

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA
PROGRAMA DE DOUTORADO EM ENGENHARIA DE TRANSPORTES

LUIZ AUGUSTO CANITO GALLEGO DE ANDRADE

Modelos de inteligência computacional para previsão de demanda desagregada em cadeias varejistas do setor de bens de consumo

São Paulo
2020

LUIZ AUGUSTO CANITO GALLEGO DE ANDRADE

Modelos de inteligência computacional para previsão de demanda
desagregada em cadeias varejistas do setor de bens de consumo

Versão Corrigida

Tese apresentada à Escola Politécnica da Universidade de
São Paulo para obtenção do título de Doutor em Ciências
Área de Concentração: Engenharia de Transportes
Orientador: Prof. Dr. Cláudio Barbieri da Cunha

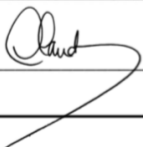
São Paulo
2020

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 08 de dezembro de 2020

Assinatura do autor: Luiz Augusto Canito Gallego de Andrade

Assinatura do orientador: 

Catálogo-na-publicação

Andrade, Luiz Augusto Canito Gallego de
Modelos de inteligência computacional para previsão de demanda desagregada em cadeias varejistas do setor de bens de consumo / L. A. C. G. Andrade – versão corr. -- São Paulo, 2019.
264 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo.
Departamento de Engenharia de Transportes.

1.CADEIA DE SUPRIMENTOS 2.VAREJO 3. ESTATÍSTICA APLICADA
4.APRENDIZADO COMPUTACIONAL I. Universidade de São Paulo.
Escola Politécnica. Departamento de Engenharia de Transportes II.t.

AGRADECIMENTOS

Agradeço à minha família pela compreensão durante todos os momentos em que não pude estar presente. Em especial ao meu avô Venâncio Monteiro Gallego (*in memoriam*) por ter me passado os valores corretos e sempre ter me apontado na direção do estudo e aprimoramento pessoal.

À minha querida esposa Mariana Peres Ciriano por seu suporte, compreensão e incentivo nas minhas decisões profissionais, e por me incentivar a perseguir meus objetivos e me apoiar nos meus erros e acertos.

Ao meu orientador Prof. Dr. Cláudio Barbieri da Cunha por seu inestimável apoio, confiança e pela orientação sempre firme e justa. Suas sugestões e críticas foram imprescindíveis para a execução dessa pesquisa.

Ao meu amigo e mentor Eng. José Guilherme Whitaker Ribeiro por compartilhar suas experiências de vida e pelo inestimável incentivo em minha carreira profissional, sem o qual muitas das minhas realizações não seriam possíveis.

E a todos que direta ou indiretamente contribuíram para a realização deste trabalho.

RESUMO

O problema de previsão de demanda desagregada no varejo de bens de consumo consiste na previsão da demanda futura de cada produto vendido em cada loja de uma empresa varejista. A efetividade com que uma empresa varejista consegue posicionar seus estoques para atendimento da demanda dos consumidores depende diretamente da sua capacidade de realizar tais previsões. Previsões acima da demanda resultam em excessos e perdas de estoque e previsões abaixo resultam em rupturas de estoque e consumidores frustrados. Apesar dos avanços dos sistemas de gestão das empresas varejistas, esse ainda é um problema em aberto, visto que a ocorrência de rupturas e perdas ainda se mostra como um dos principais problemas do setor.

Para realizar previsões de demanda é possível utilizar modelos convencionais de séries temporais. Porém, no caso do segmento varejistas alguns fatores dificultam sua aplicação: os consumidores são influenciados por datas especiais, sazonalidades e campanhas de marketing, a quantidade de modelos necessária é proporcional à multiplicação da quantidade de produtos e lojas e por último, existem problemas típicos na qualidade dos dados como por exemplo a ocorrência de rupturas e *outliers* de vendas. Por isso observa-se que na prática a demanda não é prevista de forma desagregada resultando em rupturas e excessos de estoque nas lojas das empresas varejistas de bens de consumo.

Essa pesquisa propõe diferentes caracterizações para o referido problema e uma metodologia de construção de modelos de previsão que considera o cenário das empresas varejistas: alta disponibilidade de dados, necessidade de aplicação de limpezas e correção de dados de vendas e estoques, necessidade de baixa supervisão e intervenção humana na construção dos modelos e capacidade de capturar efeitos de marketing e de datas especiais. Dentro da metodologia, são empregados modelos de aprendizado computacional para construção dos modelos de previsão. A metodologia é validada numa aplicação prática considerando duas empresas varejistas nacionais e os resultados são comparados com técnicas convencionais de previsão de séries temporais.

ABSTRACT

The disaggregated demand forecasting problem in consumer goods retail consists in forecasting of future demand for each product sold in each store of a retailer. The effectiveness with which a retailer can place its inventories to meet consumer demand depends directly on its ability to make such forecasts. Forecasting above demand results in excess inventory and potential losses, forecasting below demand results in stockouts and frustrated consumers. Despite advancements in retail companies management systems, this is still an open problem, as the occurrence of stockouts and losses is still one of the main problems in the sector.

To perform demand forecasts it is possible to use conventional time series models. However, in the case of the retail segment some factors hinder its application: consumers are influenced by special dates, seasonality and marketing campaigns, the required number of models equals to the multiplication of the quantity of products and stores and lastly, there are typical data quality problems, such as stockouts and outliers. Therefore, it is observed that in practice demand is not forecasted in a disaggregated manner resulting in stockouts and losses in the stores of consumer goods retailers.

This research proposes different characterizations for the aforementioned problem and a methodology for the construction of forecasting models considering the scenario of retail companies: high data availability, need for cleaning and correction of sales and inventory data, need for low supervision and human intervention in model building and the ability to capture marketing effects and special dates. Within the methodology, machine learning models are employed to construct the prediction models. The methodology is validated in a practical application considering two national retailers, and the results are compared with conventional time series forecasting techniques.

SUMÁRIO

1.	INTRODUÇÃO	1
1.1.	Contextualização do problema	1
1.2.	Objetivos.....	8
1.3.	Delineamento da Metodologia Proposta	8
1.4.	Estrutura da tese	10
2.	REVISÃO BIBLIGRÁFICA	12
2.1.	Perspectiva histórica.....	12
2.2.	Aplicações de modelos de inteligência computacional para previsão de demanda	15
2.3.	Aplicação de modelos de séries temporais	42
2.4.	Visão geral da revisão bibliográfica	46
3.	CARACTERIZAÇÃO DO PROBLEMA	49
3.1.	Cadeias varejistas	49
3.2.	Previsão de demanda	50
3.3.	Previsão de múltiplos de vendas	55
3.4.	Previsão do comportamento comum de demanda	56
3.5.	Comparação entre as visões do problema	57
4.	METODOLOGIA	60
4.1.	Justificativa	60
4.2.	Visão geral	62
4.3.	Tratamento e preparação dos dados	65
4.3.1.	Identificação de datas especiais	68
4.3.2.	Correção de dados de vendas observadas em períodos com rupturas de estoque	76
4.3.3.	Correção de outliers	82

4.3.4.	Reamostragem dos dados ou agregação temporal dos dados.....	85
4.3.5.	Seleção de variáveis quantitativas e redução de dimensão	88
4.3.6.	Seleção de variáveis qualitativas	94
4.3.7.	Conversão de variáveis qualitativas.....	100
4.3.8.	Inclusão de sinais de tempo	103
4.3.9.	Binarização da saída	105
4.4.	Teste preliminar	106
4.4.1.	Conjunto de treino e separação de dados.....	108
4.4.2.	Modelos para o teste preliminar	111
4.5.	Otimização de parâmetros	113
4.6.	Estimativa de desempenho	114
5.	APLICAÇÃO DA METODOLOGIA	117
5.1.	Descrição do experimento.....	118
5.2.	Conjuntos de dados.....	120
5.3.	Problema de previsão de vendas de um único produto.....	123
5.4.	Problema de previsão de múltiplos de movimentação de um único produto	134
5.5.	Problema de previsão de vendas de um grupo de produtos.....	145
5.6.	Problema de previsão de múltiplos de movimentação de um grupo de produtos	156
5.7.	Resultados comparados	167
5.8.	Sumário dos resultados	171
6.	CONCLUSÕES E RECOMENDAÇÕES	174
	REFERÊNCIAS BIBLIOGRÁFICAS.....	179
	APÊNDICE A. CONCEITOS FUNDAMENTAIS	187
A1.	Aprendizado computacional e aprendizado estatístico.....	187
A2.	Principais modelos de inteligência computacional.....	195

A.2.1. Associadores lineares e não lineares	195
A.2.2. Árvores de decisão	198
A.2.3. Máquinas de vetores de suporte.....	204
A.2.4. Redes Neurais Artificiais	215
A.2.5. Redes Neurais Fuzzy	227
A.2.6. Gradient Boosting Machines	231
A3. Modelos de séries temporais	240
A4. Legitimidade dos dados.....	247

LISTA DE FIGURAS

Figura 2.1 – Vendas agregadas do setor varejista dos Estados Unidos	20
Figura 2.2 – Topologia ilustrativa de uma rede RBF	26
Figura 2.3 – Resultados do modelo GA-RBF	28
Figura 2.4 – Série de vendas simulada para avaliação da capacidade de previsão dos elos da cadeia de suprimentos	29
Figura 2.5 – (a) Wolf’s sunspot dat, (b) Canadian lynx e (c) British pound/US dollar exchange rate data	32
Figura 2.6 – Normalização do perfil de vendas.....	39
Figura 2.7 – Exemplos do agrupamento de perfis de venda	40
Figura 2.8 – Vendas mensais de cinco categorias do varejista Foreva	43
Figura 4.1 – Ilustração da metodologia	64
Figura 4.2 – Fluxo de tratamento e preparação das amostras de dados	68
Figura 4.3 – Vendas dos produtos de uma loja	70
Figura 4.4 – Vendas médias normalizadas de um ponto de venda	71
Figura 4.5 – Vendas médias normalizadas de um ponto de movimentação separadas por data especial.....	73
Figura 4.6 – Pseudo-código do método de identificação de datas especiais	76
Figura 4.7 – Pseudo-código do método de identificação de rupturas.....	78
Figura 4.8 – Série de vendas contaminada por rupturas de estoque.....	79
Figura 4.9 – Aplicação do método de detecção de rupturas nos dados da Figura 4.8 para D igual a 90 dias	80
Figura 4.10 – Aplicação do filtro de médias móveis nos dados da Figura 4.9 para J igual a 30 dias	81
Figura 4.11 – Pseudo-código da metodologia de detecção de outliers por critério de distância estatística	85
Figura 4.12 – Comparação do efeito de reamostragem de dados	87

Figura 4.13 – Exemplo de determinação de lags relevantes pela função de autocorrelação	90
Figura 4.14 – Pseudo-código do método de seleção de lags relevantes da variável de interesse	90
Figura 4.15 – Exemplo de determinação de lags relevantes pelo coeficiente de pearson. 91	
Figura 4.16 – Pseudo-código do método de seleção de lags relevantes de variáveis de entrada	92
Figura 4.17 – Pseudo-código do método de redução de dimensão de variáveis de entrada	94
Figura 4.18 – Amostra de dados de vendas de um produto submetido a promoções.....	99
Figura 4.19 – Amostras separadas de dados de vendas de um produto submetido a promoções.....	99
Figura 4.20 – Pseudo-código do método de ajuste de escala	100
Figura 4.21 – Ilustração da técnica one-hot encoding.....	103
Figura 4.22 – Pseudo-código do método de conversão de variáveis qualitativas	102
Figura 4.23 – Pseudo-código do método de inclusão de sinais de tempo	104
Figura 4.24 – Pseudo-código do método de binarização da saída	106
Figura 4.25 – Fluxo de testes preliminares	107
Figura 4.26 – Separação entre dados de treino e teste.....	110
Figura 4.27 – Seleção de dados para validação cruzada do estudo de caso	111
Figura 5.1 – Sequência de processamento para a caracterização 1 do problema	123
Figura 5.2 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa A considerando o problema 1	126
Figura 5.3 – Frequência de melhores modelos para os dados da empresa A para o problema 1.....	127
Figura 5.4 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa B considerando o problema 1.....	127
Figura 5.5 – Frequência de melhores modelos para os dados da empresa B para o problema 1.....	128

Figura 5.6 – Análise de erros da dos modelos otimizados considerando os dados da empresa A e o problema 1	129
Figura 5.7 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa A e o problema 1.....	130
Figura 5.8 – Análise de erros da dos modelos otimizados considerando os dados da empresa B e o problema 1	130
Figura 5.9 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa B e o problema 1.....	131
Figura 5.10 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A Problema 1	132
Figura 5.11 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B Problema 1	133
Figura 5.12 – Sequencia de processamento para a caracterização 2 do problema	135
Figura 5.13 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa A considerando o problema 2	137
Figura 5.14 – Frequência de melhores modelos para os dados da empresa A para o problema 2	137
Figura 5.15 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa B considerando o problema 2.....	138
Figura 5.16 – Frequência de melhores modelos para os dados da empresa B para o problema 2	139
Figura 5.17 – Análise de erros da dos modelos otimizados considerando os dados da empresa A e o problema 2	140
Figura 5.18 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa A e o problema 2.....	141
Figura 5.19 – Análise de erros da dos modelos otimizados considerando os dados da empresa B e o problema 2	142
Figura 5.20 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa B e o problema 2.....	142
Figura 5.21 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A Problema 2	143
Figura 5.22 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B Problema 2	144

Figura 5.23 – Sequencia de processamento para a caracterização 3 do problema	146
Figura 5.24 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa A considerando o problema 3	147
Figura 5.25 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa B considerando o problema 3.....	149
Figura 5.26 – Análise de erros da dos modelos otimizados considerando os dados da empresa A e o problema 3	151
Figura 5.27 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa A e o problema 3.....	151
Figura 5.28 – Análise de erros da dos modelos otimizados considerando os dados da empresa B e o problema 3	152
Figura 5.29 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa B e o problema 3.....	153
Figura 5.30 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A Problema 3	154
Figura 5.31 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B Problema 3	155
Figura 5.32 – Sequencia de processamento para a caracterização 4 do problema	157
Figura 5.33 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa A considerando o problema 4	158
Figura 5.34 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa B considerando o problema 4.....	159
Figura 5.35 – Análise de erros da dos modelos otimizados considerando os dados da empresa A e o problema 4	161
Figura 5.36 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa A e o problema 4.....	162
Figura 5.37 – Análise de erros da dos modelos otimizados considerando os dados da empresa B e o problema 4	163
Figura 5.38 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa B e o problema 4.....	163
Figura 5.39 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A Problema 4	165

Figura 5.40 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B Problema 4	166
Figura 5.41 – Resultados comparados entre todas as abordagens de definição do problema para os dados da Empresa A	168
Figura 5.42 – Resultados comparados entre todas as abordagens de definição do problema para os dados da Empresa A	169
Figura 5.43 – Resultados comparados entre todas as abordagens de definição do problema para os dados da Empresa B.....	170
Figura 5.44 – Resultados comparados entre todas as abordagens de definição do problema para os dados da Empresa B.....	172
Figura A.1 – Pontos linearmente separáveis num espaço bidimensional.....	193
Figura A.2 – Associador Linear.....	195
Figura A.3 – Árvore de decisão	198
Figura A.4 – Pseudo-código da metodologia de construção de uma árvore de classificação	202
Figura A.5 – Representação geométrica em \mathbb{R}^2 do hiperplano ótimo de separação	207
Figura A.6 – Representação geométrica em \mathbb{R}^2 do hiperplano ótimo de regressão	212
Figura A.7 – Taxonomia de modelos de RNA.....	215
Figura A.8 – Unidade de processamento genérica (a) e rede neural artificial (b)	216
Figura A.9 – Multilayer Perceptron	218
Figura A.10 – Pseudo-código do algoritmo backpropagation na modalidade batelada ...	222
Figura A.11 – Arquiterura do modelo FNN	228
Figura A.12 – Ilustração do princípio da extensão de Zadeh	229
Figura A.13 – Pseudo-código do algoritmo de Gradient Boosting.....	237

LISTA DE TABELAS

Tabela 2.1 – Comparação de previsão dos modelos de previsão de vendas agregadas do varejo americano (período 1)	22
Tabela 2.2 – Comparação de previsão dos modelos de previsão de vendas agregadas do varejo americano (período 2)	23
Tabela 2.3 – DM test e SR test comparando a precisão dos modelos	23
Tabela 2.4 – Comparação de precisão entre modelo ARIMA, RNA e abordagem hibrida ..	33
Tabela 3.1 – Comparação entre as diferentes caracterizações do problema.....	59
Tabela 4.1 – Exemplo de aplicação do teste-t para identificação de datas especiais considerando o Natal.....	74
Tabela 4.2 – Exemplo de aplicação do teste-t para identificação de datas especiais considerando a Black Friday	75
Tabela 4.3 – Comparação do coeficiente de variação para diferentes janelas de reamostragem	87
Tabela 5.1 – Conjuntos de dados do estudo de caso	121
Tabela 5.2 – Estatísticas descritivas dos dados da empresa A.....	121
Tabela 5.3 – Estatísticas descritivas dos dados da empresa B.....	122
Tabela 5.4 – Hiperparâmetros do teste preliminar – Problema 1	125
Tabela 5.5 – Grade de busca para a otimização de parâmetros – Problema 1	128
Tabela 5.6 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A – Problema 1	132
Tabela 5.7 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B – Problema 1	133
Tabela 5.8 – Hiperparâmetros do teste preliminar – Problema 2	136
Tabela 5.9 – Grade de busca para a otimização de parâmetros – Problema 1	139
Tabela 5.10 – Taxa de hit para o problema 2 considerando os dados da empresa A.....	140
Tabela 5.11 – Taxa de hit para o problema 2 considerando os dados da empresa B.....	143

Tabela 5.12 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A – Problema 2	144
Tabela 5.13 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B – Problema 2	145
Tabela 5.14 – Estatísticas dos erros dos modelos no teste preliminar – Empresa A – Problema 3	148
Tabela 5.15 – Estatísticas dos erros dos modelos no teste preliminar – Empresa B – Problema 3	150
Tabela 5.16 – Grade de busca para a otimização de parâmetros – Problema 3 – Empresa A	150
Tabela 5.17 – Grade de busca para a otimização de parâmetros – Problema 3 – Empresa B	152
Tabela 5.18 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A – Problema 3	154
Tabela 5.19 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B – Problema 3	155
Tabela 5.20 – Estatísticas dos erros dos modelos no teste preliminar – Empresa A – Problema 4	159
Tabela 5.21 – Estatísticas dos erros dos modelos no teste preliminar – Empresa B – Problema 4	160
Tabela 5.22 – Grade de busca para a otimização de parâmetros – Problema 4 – Empresa A	161
Tabela 5.23 – Taxa de hit para o problema 4 considerando os dados da empresa A.....	162
Tabela 5.24 – Taxa de hit para o problema 4 considerando os dados da empresa B.....	164
Tabela 5.25 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A – Problema 4	165
Tabela 5.26 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B – Problema 4	166
Tabela A.1 – Taxonomia de modelos de suavização exponencial	241
Tabela A.2 – Termos de suavização para os diferentes modelos de suavização	242

1. INTRODUÇÃO

1.1. Contextualização do problema

Dentre os processos de planejamento em cadeias de suprimento a previsão de demanda se destaca como sendo o ponto de partida para decisões de planejamento de estoques, planejamento de produção, planejamento de capacidades, planejamento financeiro, dentre outros processos (HUANG, FILDES e SOOPRAMANIEN, 2019). Fildes, Ma e Kolassa (2018) apresentam uma extensiva revisão do estado da arte em relação a práticas de previsão de demanda dentro do contexto de pesquisa operacional. Os autores afirmam que é grande o interesse da comunidade científica e de profissionais que atuam em cadeias logísticas em geral com relação a práticas de previsão de demanda em *supply chain*.

A previsão da demanda realizada de forma eficaz permite a maximização da lucratividade na cadeia de suprimentos e possibilita que uma determinada empresa obtenha vantagens competitivas frente à sua concorrência posicionando seus produtos ou serviços mais próximos dos consumidores (CORSTEN e GRUEN, 2003). Uma previsão maior que a demanda real resulta em excesso de estoque na cadeia logística e eventual perda de mercadorias, ao passo que uma previsão menor resulta em falta de mercadorias, vendas perdidas e consumidores insatisfeitos.

Dentre os diversos segmentos em que a previsão de demanda é relevante, está o segmento varejista de bens de consumo. Esse setor é composto por empresas que oferecem mercadorias que são adquiridas pelos consumidores nas lojas para consumo imediato ou em curto prazo (por exemplo.: alimentos, vestuário, utensílios domésticos, materiais de construção, dentre outros). Essas empresas têm contato direto com o consumidor final em seus pontos de venda e a previsão de demanda e o dimensionamento de estoques são determinantes para que as mercadorias estejam disponíveis para consumo, sem que haja excesso de estoque prejudicial à saúde financeira da cadeia.

Corsten e Gruen (2003) afirmam que a concorrência do setor de bens de consumo em uma determinada região é proporcional ao grau de complexidade de sua economia e de sua conexão logística com outras regiões. Em economias desenvolvidas, essa concorrência é alta

e as empresas que conseguem prever a sua demanda com antecedência conseguem ser mais eficientes na movimentação de suas mercadorias dos pontos de fabricação ou fornecimento até os pontos de venda. A previsão da demanda realizada de forma eficaz permite o dimensionamento correto dos estoques em todos os elos da cadeia de suprimentos além de parâmetros e políticas otimizadas de reabastecimento.

O excesso de estoque nos pontos de venda significa altos custos de capital imobilizado e perdas financeiras pela obsolescência das mercadorias, seja expiração do prazo de validade de mercadorias perecíveis seja pela perda de janela de vendas das mercadorias (ex.: produtos promocionais de campanhas de Natal que não são vendidos, ou itens de moda de uma determinada coleção que não são vendidos). O custo da falta de estoques disponíveis para consumidores é um conceito mais abstrato pois acarreta perdas de oportunidades de venda. Os primeiros estudos sobre os impactos financeiros de faltas de estoques em pontos de venda foram realizados nas décadas de 50 e 60.

Zinn e Liu (2001) afirmam em sua pesquisa que em média 8,2% dos produtos de um supermercado nos EUA ficam sem estoque ao longo do ano. Em um estudo focado em produtos da categoria de mercearia em supermercados nos EUA, Che *et al.* (2012) afirmam que a taxa de falta de estoque média é de 7,9% o que representa uma perda de faturamento de aproximadamente 4%, uma quantia significativa dadas as pequenas margens de lucratividade das operações varejistas.

A falta de estoques no segmento varejista não é um problema novo. Peckhan (1963) investigou o comportamento de consumidores de supermercados nos Estados Unidos. De acordo com o autor, 58% dos entrevistados trocam de marca ao não encontrar um produto específico nas prateleiras de exposição do supermercado. Este tipo de resultado evidencia que a indisponibilidade de produtos nos pontos de venda pode custar mais para uma empresa do que simplesmente a perda da lucratividade da venda, ou seja, pode ocasionar a perda de consumidores para uma marca concorrente. Outros estudos detalhados dos efeitos de indisponibilidades de estoque podem ser encontrados, por exemplo, em Schary e Christopher (1979) e Schary and Becker (1978).

Corsten e Gruen (2003) realizaram uma análise de 40 estudos sobre efeitos de faltas de estoque nos Estados Unidos. Os autores dizem que seria natural crer que com o advento de sistemas de informação e novas tecnologias de processamento de dados, a disponibilidade de produtos deveria ser um problema já endereçado, no entanto o que se conclui é que as empresas varejistas ainda têm a falta de produtos para os consumidores como um dos principais desafios. O mesmo pode ser verificado num estudo das estratégias e preocupações dos principais varejistas norte-americanos realizado por Randall *et al.* (2011), que evidencia que a disponibilidade de produtos também é uma das principais preocupações dos varejistas analisados.

No caso do mercado brasileiro, Vasconcelos e Sampaio (2009) realizaram um levantamento de níveis de quebras de estoque (*stock out rates*) para uma amostra de supermercados e hipermercados do estado de São Paulo. Em seu levantamento constata-se que o indicador de *stockout* para a amostra selecionada é de 8,3% com desvio de 6,4%, valores similares à supermercados nos EUA (ZINN e LIU, 2001; CHE *et al.* 2012). Esses índices de indisponibilidade evidenciam um potencial de aumento de receita e lucratividade nas operações varejistas uma vez que a indisponibilidade de mercadorias poderia ser convertida em vendas caso os estoques estivessem corretamente dimensionados.

Se por um lado a previsão de demanda é importante em cadeias varejistas para melhorar a disponibilidade de produtos, por outro lado esse tipo de cadeia contém algumas características que tornam essa atividade complexa:

- (i) **Diversidade de produtos no portfólio e capilaridade geográfica:** os níveis de competição no setor de bens de consumo fazem com que as empresas busquem maior variedade no seu portfólio com o objetivo de diferenciação no mercado e atendimento das expectativas dos consumidores finais. Além disso, há uma constante busca na renovação do portfólio com lançamentos de novos produtos. Um portfólio amplo de produtos aliado a necessidade de posicionar os estoques em diversos locais para atendimento do consumidor traz complexidade para a previsão de demanda;

- (ii) **Alta capilarização dos pontos de venda:** com o objetivo de penetração geográfica, as empresas do setor de bens de consumo geralmente possuem estoques em muitas lojas simultaneamente. Há casos em que as empresas do setor gerenciam seus próprios estoques em pontos de venda de terceiros (processo chamado de *Vendor-Managed Inventory*);
- (iii) **Alta influência de promoções e outras ações de estímulo ao consumo:** o setor de bens de consumo utiliza práticas de estímulo de demanda tais como promoções, descontos e formação de *kits* de venda. Esse tipo de ação não coordenada com o dimensionamento de estoques e com o restante das operações logísticas causa efeitos sistêmicos de rupturas de estoque nos diversos elos da cadeia;
- (iv) **Sazonalidades e eventos:** o consumo do setor varejista possui ciclos, ou sazonalidades e eventos específicos nos quais a projeção das vendas é difícil. Por exemplo, as vendas de Natal concomitantes com lançamentos do período, fazem com que a acurácia das projeções do período seja em geral baixa.
- (v) **Dificuldade de coordenação com ações comerciais:** as ações promocionais (por exemplo: campanhas de *marketing* e descontos) muitas vezes não possuem informações estruturadas ou são executadas sem o devido planejamento logístico (HUANG, FILDES e SOOPRAMANIEN, 2019), especialmente em cadeias que não possuem processos integrados de planejamento (ex.: *Sales and Operations Planning*) cujo objetivo é justamente promover essa coordenação. Isso resulta em rupturas de estoque.

Esses fatores de complexidade resultam em constantes rupturas de estoques ou obsolescência e desperdício.

A atividade de previsão ou projeção de demanda consiste em estimar a quantidade que será vendida dado um produto ou serviço e uma região geográfica. Isso pode ser feito por meio de métodos qualitativos ou métodos quantitativos (MORETTIN e TOLOI, 2004). Métodos qualitativos envolvem o julgamento de especialistas e análises qualitativas das vendas passadas. Métodos quantitativos envolvem modelar o mecanismo gerador da demanda de

uma empresa utilizando modelos matemáticos para previsão do seu comportamento em cenários futuros. Ainda com relação aos métodos quantitativos, toma-se como premissa que a venda observada é a mensuração de um fenômeno desconhecido sobre o qual se tem interesse em modelar para fazer previsões.

Do ponto de vista matemático, modelos de previsão compreendem um tema abrangente que envolve diversas áreas da matemática, desde modelos de regressão linear simples até a utilização de processos estocásticos para modelagem de séries temporais (DE GOOIJER e HYNDMAN, 2006). Há uma grande quantidade de metodologias de previsão que podem ser utilizadas para previsão de demanda, como por exemplo, modelos de suavização, modelos da classe *Autoregressive Integrated Moving Average* (ARIMA), modelos de espaço de estados, modelos causais e modelos de aprendizado computacional. A escolha da melhor técnica para cada caso específico depende das características de cada problema, a quantidade de produtos a serem previstos, o tamanho do histórico de dados disponível e a influência de fatores promocionais são parâmetros que devem ser considerados na escolha do método.

A previsão de demanda em especial pode ser analisada como um problema de análise de séries temporais em que as vendas observadas são ordenadas no tempo. As séries temporais por sua vez podem ser modeladas como um processo estocástico em que cada venda observada é um evento decorrente de uma distribuição condicional de probabilidade e cada série é um traço do processo estocástico (MORETIN e TOLOI, 2004).

Num contexto de cadeias varejistas, é possível agrupar as séries de vendas de um determinado produto considerando elementos da cadeia como lojas, centros de distribuição, ou é possível tratar cada série isoladamente. Caso as séries sejam agrupadas, o problema de previsão recebe o nome de previsão de demanda agregada. Caso cada série seja tratada isoladamente o problema de previsão recebe o nome de previsão de demanda desagregada (FILDES *et al.*, 2018). A previsão agregada é relevante para dimensionamento de capacidade de produção, dimensionamento de centros de distribuição e planejamento de compras. Já a previsão desagregada é relevante para tomadas de decisão de abastecimento de lojas.

Segundo Huang *et al.* (2019) existem várias abordagens propostas na literatura para o problema de previsão agregada. Já no caso da previsão desagregada, as abordagens ainda são limitadas. Para o caso de cadeias varejistas, os autores afirmam que dois requisitos tornam o problema de previsão desagregada particularmente desafiador:

1. **Necessidade de automatização do processo:** empresas varejistas atualmente trabalham com uma grande variedade de produtos. Por isso, procedimentos de modelagem que requerem intervenções de especialistas são inviáveis na prática.
2. **Necessidade de tratar situações de mudança estrutural do fenômeno de demanda:** a demanda dos produtos nas lojas de cadeias varejistas é muito influenciada por descontos e ações promocionais. Isso faz com que modelos convencionais que assumem estabilidade da distribuição de probabilidade sejam pouco efetivos em épocas promocionais.

Boone *et al.* (2019) apresentam uma revisão do impacto da disponibilidade de informações na previsão de demanda na cadeia de suprimentos. Segundo os autores a popularização de sistemas de informação avançados nos pontos de venda e a digitalização dos consumidores finais são dois fatores que atualmente permitem a construção de sistemas avançados de análise de dados. Ainda segundo os autores, considerando a disponibilidade de dados, técnicas de aprendizado computacional ou inteligência artificial são mais indicadas para previsão de demanda que técnicas convencionais de análise de séries temporais.

Há uma lacuna na literatura no que se refere à aplicação de métodos de aprendizado computacional para o problema de previsão de demanda desagregada. Esse caminho pode potencialmente auxiliar na solução do problema mencionado de indisponibilidade de produtos em cadeias varejistas e pode impactar positivamente o resultado dessas empresas, tanto em lucratividade quanto na satisfação dos consumidores finais.

Modelos de aprendizado computacional são capazes de mapear as relações de entradas e saídas de forma automática sem a intervenção de um especialista. Assim, tais métodos cumprem os requisitos citados por Huang *et al.* (2019), ou seja, podem ser aplicados com

alto grau de automatização e podem ser capazes de identificar alterações estruturais no processo de geração de demanda decorrente de ações promocionais.

Em resumo, sobre o problema de previsão de demanda desagregada em cadeias varejistas do setor de bens de consumo é possível observar que:

- (i) Esse problema ainda se mostra presente nas empresas do setor de bens de consumo se manifestando na forma de frequentes rupturas e excessos de estoque nos pontos de venda, representando uma oportunidade significativa de aumento de receitas e lucratividade;
- (ii) O problema não foi tratado de forma satisfatória pelas aplicações encontradas na literatura de forma a promover uma maior automação do processo de análise de dados e escolha de modelos sem que haja a necessidade de supervisão de especialistas;
- (iii) Modelos de inteligência computacional se mostram como um caminho promissor para melhorar a previsão de demanda em cadeias varejistas (ALON, QI e SADOWSKI, 2001; ZHANG, 2003; VEIGA *et al.*, 2016);
- (iv) A disponibilidade de informações nas bases de dados das empresas e em fontes externas de informações possibilitam o uso de algoritmos de aprendizado computacional necessários para a aplicação de modelos de previsão de demanda com baixa supervisão de especialistas.

Dentro desse contexto, essa pesquisa visa preencher a lacuna encontrada pela análise da aplicação de modelos de aprendizado computacional para o problema de previsão desagregada no varejo. Entende-se que a proposição de uma metodologia capaz de melhorar o processo de previsão de demanda em tais cadeias, pode aumentar a disponibilidade de produtos e reduzir as perdas, impactando positivamente as operações das empresas varejistas e a satisfação dos consumidores finais.

1.2. Objetivos

O objetivo geral da pesquisa é analisar o problema de previsão de demanda desagregada em cadeias varejistas com especial foco na aplicação de modelos de aprendizado computacional. Mais detalhadamente a pesquisa visa:

- (i) Caracterizar o problema de previsão de demanda desagregada de cadeias varejistas do setor de bens de consumo;
- (ii) Propor uma metodologia de previsão de demanda desagregada que possa ser aplicada com alto grau de automação em cadeias com muitos pontos de venda e muitos produtos, ou seja, uma metodologia que possa ser aplicada com baixa supervisão de especialistas;
- (iii) Avaliar a aplicação de técnicas e modelos de aprendizado computacional dentro da metodologia proposta em comparação com modelos convencionais de séries temporais;
- (iv) Realizar uma aplicação prática com dados reais de cadeias varejistas para validar a metodologia proposta.

1.3. Delineamento da Metodologia Proposta

Para a resolução do problema de previsão desagregada em cadeias varejistas a metodologia de pesquisa empregada envolve inicialmente uma revisão da literatura quanto a aplicações de modelos de aprendizado computacional em problemas de previsão de demanda em geral e aplicações convencionais de modelos de séries temporais.

Em seguida são propostas caracterizações para o problema de previsão de demanda desagregada em cadeias varejistas. São propostas definições alternativas com base em diferentes premissas e considerações sobre a demanda e sobre o fenômeno gerador da demanda.

Tendo em vista as possíveis caracterizações do problema, essa pesquisa propõe uma metodologia de construção e validação de modelos de previsão baseados em modelos de

aprendizado computacional. A metodologia visa endereçar as particularidades do problema de previsão de demanda no contexto varejista, em especial a necessidade de um processo de construção de modelos com baixa supervisão a alto grau de automação.

A metodologia proposta nessa pesquisa tem como objetivo a resolução de algumas questões práticas de tratamento dos dados de séries de vendas de cadeias varejistas, bem como a proposição de um processo estruturado de para construção de modelos de previsão de demanda desagregada em cadeias varejistas. A aplicação da metodologia proposta resulta em modelos preditivos para auxiliar empresas varejistas do setor de bens de consumo a abastecerem corretamente seus pontos de venda.

De forma geral, a metodologia desenvolvida nessa pesquisa compreende os seguintes passos:

1. **Tratamento de dados:** nessa etapa da metodologia são propostos mecanismos de correção, transformação e tratamento de dados característicos de séries de vendas de cadeias varejistas de bens de consumo;
2. **Teste preliminar de modelos:** compreende a primeira etapa de um processo estruturado de modelagem de problemas de aprendizado computacional. O teste preliminar de modelos visa identificar rapidamente tipos ou classes de modelos com bom desempenho para que apenas nesses seja realizada uma atividade de busca de parâmetros;
3. **Busca de parâmetros:** essa etapa compreende a busca de parâmetros adequados para os modelos preditivos identificados no teste preliminar com o objetivo de minimizar os erros de previsão;
4. **Validação e estimativa de desempenho *out-of-sample*:** uma vez que um modelo seja selecionado e tenha seus parâmetros otimizados é necessário fornecer uma estimativa do desempenho desse modelo em termos de uma medida de erro para previsões de amostras ainda desconhecidas.

Essa metodologia proposta visa estabelecer um processo que pode ser aplicado na construção de modelos de previsão para que sejam utilizados em processos de cadeias varejistas de bens de consumo de forma abrangente.

Não foram identificadas na literatura metodologias similares direcionadas para o problema de previsão de demanda desagregada de cadeias varejistas. Os trabalhos encontrados na literatura lidam com problemas de previsão agregadas com características diferentes do problema abordado nessa pesquisa (por exemplo: RAMOS, SANTOS e REBELO, 2015; VEIGA *et al.*, 2016; GOODNESS *et al.*, 2015).

Para validar a metodologia proposta, realiza-se uma aplicação prática considerando dados de duas empresas varejistas de abrangência nacional. A aplicação da metodologia proposta permite validar os passos propostos e analisar os resultados comparados com metodologias convencionais de previsão de demanda.

Por fim é realizada uma análise crítica dos resultados. O objetivo dessa análise é identificar qual das caracterizações do problema é a mais adequada e se a metodologia proposta resulta em modelos de previsão mais precisos que metodologias convencionais de previsão de demanda.

1.4. Estrutura da tese

O capítulo 2 contempla uma revisão bibliográfica do tema da pesquisa. Primeiro é apresentada uma perspectiva cronológica da proposição de modelos de previsão de séries temporais, desde os primeiros modelos estatísticos mais simples até os modelos de Inteligência computacional. Além disso é apresentado um conjunto de exemplos de aplicações de modelos de inteligência computacional para a solução de problemas similares ao dessa pesquisa

No capítulo 3 é realizada uma caracterização formal do problema da pesquisa. São apresentadas quatro visões alternativas do problema que serão tratadas pela metodologia proposta, que é efetivamente detalhada no capítulo 4.

No capítulo 5 é apresentado uma aplicação prática da metodologia que consiste na sua aplicação em dois casos reais com o objetivo de validar os passos propostos. O capítulo 5 também contém toda a descrição da aplicação da metodologia bem como uma discussão dos resultados obtidos.

Por fim, no capítulo 6 são traçadas as conclusões e são apresentadas recomendações de continuidade da pesquisa e temas relacionados que podem ser perseguidos.

O APÊNDICE A apresenta alguns conceitos fundamentais de inteligência e aprendizado computacional além de apresentar os fundamentos de alguns modelos e algoritmos relacionados ao tema de inteligência computacional. O final da seção detalha o conceito de legitimidade dos dados para a construção de modelos, um tema pertinente para a construção de modelos efetivos.

2. REVISÃO BIBLIOGRÁFICA

De acordo com Morettin e Tolo (2004), os métodos de previsão podem ser classificados como métodos qualitativos ou métodos quantitativos. No caso de métodos qualitativos, utiliza-se preponderantemente o julgamento de indivíduos ou grupos de especialistas para avaliação de cenários de previsão. No caso de modelos quantitativos são aplicadas técnicas de modelagem matemática em conjuntos de dados para determinação de modelos de previsão.

Esta pesquisa trata de métodos quantitativos de previsão de demanda e, portanto, essa revisão da literatura possui seu foco nesse tipo de metodologia.

São tratados três tópicos:

- (i) É apresentada uma perspectiva histórica do desenvolvimento de modelos de previsão de demanda, desde os primeiros modelos de séries temporais até os atuais modelos de inteligência computacional;
- (ii) São apresentadas aplicações e revisões de modelos de inteligência computacional para problemas de previsão de séries temporais;
- (iii) São apresentadas aplicações e revisões de modelos de séries temporais para problemas similares ao dessa pesquisa.

2.1. Perspectiva histórica

Modelos de séries temporais foram as primeiras abordagens matemáticas aplicadas a problemas de previsão de demanda em meados de 1950. Tais modelos incluem os métodos de suavização e os métodos de decomposição de séries temporais.

Com o crescimento da capacidade computacional, abordagens mais sofisticadas foram desenvolvidas para o problema. Nos anos 1960s, Box e Jenkins (1990) propuseram uma metodologia para modelagem de séries temporais baseada em processos estocásticos. Essa metodologia deu origem aos modelos da classe *Autoregressive Integrated Moving Average* (ARIMA).

Chu e Zhang (2003) realizaram um estudo comparativo de modelos tradicionais lineares e técnicas avançadas de modelagem para previsão. Sobre os modelos resultantes da metodologia de Box e Jenkins, os autores afirmam que além da necessidade de assumir uma estrutura linear para o problema, essa metodologia possui a desvantagem de ser necessário assumir a forma do modelo *a priori*, sem necessariamente conhecer a estrutura de relações presentes na série temporal a ser modelada.

Modelos de espaço de estado (*State Space Models – SSM*) são uma abordagem que considera a representação de um sistema dinâmico por meio de um sistema de equações diferenciais de primeira ordem (DURBIN e KOOPMAN, 2012). Esse tipo de modelo possui aplicações em problemas de diversas naturezas, como por exemplo modelagem de fenômenos econômicos (ZENG; WU, 2013), filtragem de sinais de áudio (ALZAMENDI; SCHLOTTHAUER; TORRES, 2015) e identificação de falha de sistemas de engenharia (SUN *et al.*, 2012). As séries temporais também foram modeladas segundo essa representação resultando no surgimento de modelos estruturais. A aplicação de modelos de espaço de estados para previsão de demanda exige que seja proposta uma estrutura matricial para o mapa de estados do fenômeno modelado, exigindo também conhecimento *a priori* do fenômeno gerador de demanda. Uma revisão detalhada dos modelos de espaço de estados pode ser encontrada em Durbin e Koopman (2012).

Com o aprimoramento de técnicas computacionais de modelagem matemática, métodos de inteligência computacional começaram a ser aplicados no contexto de previsão de demanda. Historicamente, os primeiros modelos de inteligência computacional surgiram na década de 1940 com o trabalho do neurofisiologista Warren McCulloch e do matemático Walter Pitts (MCCULLOCH e PITTS, 1943). Os autores propuseram um modelo matemático teórico para representar o mecanismo de funcionamento de neurônios.

Em 1949, Donald Hebb (HEBB, 1949) propôs uma descrição qualitativa do mecanismo de aprendizado por reforço, cuja aplicação posterior em linguagem matemática se tornaria a base fundamental dos algoritmos de aprendizado das técnicas modernas de inteligência computacional. Essas duas publicações foram as bases para a proposição dos primeiros modelos de redes neurais artificiais, sendo o trabalho de Frank Rosenblatt em 1957 a

primeira proposição de uma rede neural com múltiplas camadas capaz de resolver problemas complexos de reconhecimento de padrões (ROSENBLATT, 1962).

A partir desses trabalhos iniciais, outros modelos de inteligência computacional foram propostos e aplicados em problemas de reconhecimento de padrões, incluindo redes neurais com diferentes arquiteturas e mecanismos de aprendizado, Árvores de Classificação e Regressão (*Classification and Regression Trees - CART*), Máquinas de Vetores de Suporte (*Support Vector Machines – SVM*) e modelos de lógica nebulosa (*Fuzzy Logic*).

A ideia de aplicar Redes Neurais Artificiais (RNAs) para previsão é antiga sendo a primeira aplicação um estudo conduzido por Hu (1964) no qual uma rede linear adaptativa foi utilizada para previsão climática. Contudo a falta de algoritmos práticos de aprendizado de redes com arquitetura de múltiplas camadas inviabilizou grande parte das tentativas de aplicação de RNAs, resultando em apenas estudos teóricos e aplicações limitadas (ROJAS 1996). A partir de 1986 com a proposição do algoritmo de *Backpropagation* (RUMELHART, HINTON E WILLIAMS; 1986) a aplicação de RNAs para problemas de previsão apresentou grandes desenvolvimentos.

No contexto de previsão de demanda, a aplicação de técnicas de inteligência computacional se mostra promissora uma vez que nesse tipo de modelo poucas premissas precisam ser assumidas a respeito do fenômeno gerador da demanda, bastando para sua aplicação, uma quantidade suficiente de dados (BOONE *et al.*; 2019). A lógica de modelagem das técnicas de aprendizado computacional busca definir a estrutura do modelo com base em um algoritmo de treinamento que reforça as relações entre as variáveis de entrada e saída com a apresentação de amostras. Zhang, Patuwo e Hu (1998) apresentam uma revisão de aplicações de modelos de redes neurais para problemas de previsão.

As técnicas mais recentes de inteligência computacional compreendem uma classe de modelos que recebem o nome de modelos de aprendizado profundo (*deep learning*) (GOODFELLOW, BENGIO E COURVILLE; 2016). Esses modelos podem ser encarados como modelos de redes neurais com muitas camadas interconectadas o que confere ao modelo grande capacidade de modelagem de fenômenos e grande capacidade de recombinação de entradas para correlacionar com saídas desejadas. Os modelos de *deep learning* são

utilizados em problemas de aprendizado computacional complexos, que necessitam modelos com alta capacidade de abstração de entradas, como por exemplo, reconhecimento de imagens, processamento de linguagem natural e transcrição áudio-texto.

Sob um aspecto prático, os modelos de *deep learning* são difíceis de treinar devido a um fenômeno denominado *vanishing gradient problem*. A técnica convencional de treinamento de RNAs (*backpropagation*) depende do cálculo sequencial de gradientes do modelo, porém ocorre que em redes com muitas camadas, os gradientes nas camadas mais distantes da saída do modelo são muito próximos de zero, o que inviabiliza o treinamento desses modelos em tempo computacional viável.

Com o avanço das técnicas computacionais, em especial o uso de unidades gráficas de processamento (*Graphical Processing Units*) para cálculo paralelizado de multiplicações tensoriais, o treinamento de modelos de *deep learning* vem se popularizando e tem encontrado aplicações práticas em sistemas computacionais. Esses modelos de aprendizado profundo compreendem as mais recentes aplicações de inteligência computacional num contexto mais geral que o dessa pesquisa.

2.2. Aplicações de modelos de inteligência computacional para previsão de demanda

A importância do processo de previsão de demanda na atividade empresarial cresce na medida em que a complexidade das operações e os níveis de concorrência aumentam. De acordo com Corsten e Gruen (2003), o sucesso de uma cadeia logística depende da habilidade de seus planejadores de antever eventos futuros e ajustar os parâmetros logísticos antecipadamente, mantendo assim a eficiência da cadeia. Sem a capacidade de previsão, as operações de uma cadeia de suprimentos podem apenas responder a estímulos de forma retroativa, levando a planos de produção pouco eficientes, perdas de vendas, baixo nível de serviço ao consumidor final, utilização ineficiente de recursos e consequentemente resultado financeiro reduzido.

Winklhofer, Diamantopoulos e Witt (1996) apresentaram um estudo abrangente de aplicações de previsão de demanda. Os autores apontam que os fatores que fazem com que o tema seja cada vez mais pertinente são:

- (i) Conforme a complexidade das operações aumenta, bem como a complexidade do ambiente em que as empresas estão inseridas, se torna mais difícil para os tomadores de decisão planejar a alocação de seus recursos considerando todos os fatores sem o uso de modelos matemáticos adequados;
- (ii) As organizações migraram para uma condição de tomada de decisão sistemática baseada em argumentos concretos para alocação de recursos e planejamento. Dessa forma, modelos matemáticos fornecem subsídios mais sustentáveis do que argumentos subjetivos de profissionais com experiência de negócio;
- (iii) O mercado de tecnologia avançou no desenvolvimento de ferramentas analíticas para gestores e participantes da cadeia de suprimentos, aproximando teoria e prática nos processos de previsão de demanda;

O primeiro uso de modelos de aprendizado computacional foi no sentido de automatizar o passo de identificação de ordem do modelo dentro da metodologia ARIMA. Lee e Oh (1996) realizaram uma aplicação de RNAs combinada com árvores de classificação para a solução do problema de identificação de séries temporais (*Neural Network Driven Tree Classifier – DTC*). Uma das atividades necessárias para aplicação da metodologia ARIMA é a determinação da ordem do modelo, definida pela quantidade de termos autorregressivos e de termos de médias móveis a serem considerados. Os autores propõem a utilização de técnicas de reconhecimento de padrões para automatizar essa etapa da modelagem de séries temporais, a qual tradicionalmente é realizada por um especialista de forma iterativa.

Uma série temporal observada pode ser caracterizada pela sua função de autocorrelação amostral estendida (*Extended Sample Autocorrelation Function – ESACF*). O objetivo do modelo proposto pelos autores é classificar a ESACF de uma série temporal observada de acordo com a ESACF de um modelo teórico específico por meio de uma árvore de classificação em que a função de decisão de cada nó é dada por uma RNA. Para a validação

do modelo proposto, Lee e Oh (1996) realizaram uma série de experimentos com dados simulados e dados reais de séries observadas. No experimento conduzido com dados simulados, as redes neurais de cada nó do classificador possuem 36 inputs, representando uma ESACF de ordem (5,5) e funções de ativação do tipo sigmoide bipolar em cada neurônio. Os dados simulados foram gerados por modelos ARMA de ordem conhecida acrescidos de ruído. O percentual de acerto de classificação do modelo DTC para o experimento realizado foi de 90,5%, o que significa que o modelo foi capaz de classificar corretamente a maior parte das séries apresentadas, mesmo na presença de ruído.

A abordagem proposta por Lee e Oh (1996) é uma aplicação de reconhecimento de padrões para identificação da ordem das séries temporais. O uso da metodologia proposta e do modelo DTC pode ser usado no contexto de previsão de demanda na seleção de *inputs* autorregressivos para a construção de modelos de previsão. Os autores trabalharam apenas com modelo estacionários, sendo necessária a validação da metodologia para séries não-estacionárias.

O trabalho de Lee e Oh (1996) tenta automatizar o passo da metodologia ARIMA que dificulta a sua aplicação com pouca supervisão. Os autores são bem-sucedidos na aplicação de modelos de inteligência computacional na identificação da ordem dos modelos ARIMA, porém o modelo resultante continua sendo um modelo linear e não faz uso de variáveis importantes que se relacionam as vendas em cadeias varejistas como as campanhas de marketing. Além disso, o uso dos modelos de inteligência computacional é feito de forma indireta como parte do processo convencional e não diretamente para modelagem da série temporal.

Outros trabalhos utilizam os modelos de inteligência computacional de forma direta. Zhang, *et al.* (1998) apresentaram uma revisão abrangente de modelos e aplicações de redes neurais artificiais, sendo que o uso para previsão de séries temporais é um dos principais. De acordo com os autores, a utilização de RNAs para problemas de previsão atrai interesse das mais diversas áreas do conhecimento por três motivos:

- (i) Ao contrário da abordagem tradicional de modelagem matemática na qual é necessário assumir uma forma funcional para o fenômeno em análise, as RNAs

derivam as relações entre as variáveis a partir dos dados de forma adaptativa e evolutiva, sendo necessário pouco conhecimento *a priori* do fenômeno modelado. As RNAs aprendem as relações entre as variáveis a partir de exemplos de entradas e saídas e são capazes de capturar padrões sutis e difíceis de descrever;

- (ii) As RNAs são capazes de generalizar o conhecimento mapeado na sua estrutura, ou seja, elas são capazes de prever a saída para um vetor de entradas ainda desconhecido;
- (iii) Pode ser demonstrado que as RNAs são capazes de aproximar qualquer função contínua dado um grau de precisão desejado. A abordagem estatística tradicional assume que para um conjunto de observações (entradas e saídas) de um fenômeno em análise há um modelo teórico que gera as saídas observadas a partir de entradas também observadas. Muitas vezes é difícil especificar o modelo teórico, sendo as RNAs uma alternativa que substitui essa necessidade;

As RNAs podem ser consideradas como modelos não lineares. Os modelos tradicionais de previsão, como os modelos ARIMA, assumem processos lineares de geração de uma série temporal observada.

Ainda segundo Zhang *et al.* (1998), é irrealista assumir *a priori* que um fenômeno é gerado por um processo linear, especialmente um fenômeno que envolve interações humanas como a demanda por produtos e outros fenômenos econômicos. Os autores comentam que apesar da existência de modelos estatísticos não lineares, como por exemplo o modelo *Autoregressive Conditional Heterocedastic* (ARCH), o modelo Bilinear de Granger e Anderson e o modelo *Threshold Autoregressive* (TAR), esses ainda são limitados pela necessidade de especificação de um modelo teórico de geração do fenômeno.

A pesquisa de Zhang *et al.* (1998) não só aborda os pontos que fazem com que RNAs sejam modelos promissores para aplicações de previsão de séries temporais. Os autores também discutem as dificuldades dessa metodologia em relação a definições de arquitetura (escolha de nós de entrada, intermediários e de saída, e escolha de funções de ativação), algoritmos

de treinamento, separação de amostras de treino e teste, métricas de performance e normalização de dados. Apesar de abrangente, o trabalho de Zhang *et al.* (1998) não trata de problemas específicos de aplicações práticas e não recomenda parâmetros nem decisões de arquitetura, sendo um trabalho valioso como levantamento do estado da arte, mas pouco efetivo para direcionamento de implementações práticas como o caso dessa pesquisa.

Além das RNAs existem outros modelos de aprendizado computacional que foram aplicados a problemas de previsão de séries temporais. Um exemplo é o trabalho de Müller *et al.* (2001) no qual os autores apresentam uma aplicação de SVMs de regressão para previsão de séries temporais. Os autores fazem um detalhamento da modelagem de SVMs de regressão e utilizam duas medidas de erro para modelagem, o erro ϵ -sensível e o erro de Huber (seção A.2.3 do Apêndice A). As SVMs geradas são comparadas entre si e comparadas com um modelo de RNA de base radial (*Radial Basis Function - RBF*).

As SVMs e a rede RBF são treinadas e testadas considerando dois conjuntos de dados, um gerado a partir da equação de Mackey-Glass e outro com os dados de uma competição de modelagem de séries temporais. Ambos os conjuntos de dados são acrescidos de ruídos gaussianos e uniformes para comparação da precisão dos modelos treinados nas diferentes situações. Os dados utilizados como *inputs* para os modelos são definidos exclusivamente em termos de valores passados das respectivas séries temporais.

Os resultados indicam que as SVMs obtêm melhores resultados nos conjuntos de dados gerados com o acréscimo de ruídos uniformes enquanto a rede RBF apresenta melhor resultado em conjuntos de dados com ruído gaussiano. Os autores ainda comentam que uma das vantagens das SVMs é a estabilidade dos modelos gerados e conseqüentemente das previsões geradas. Isso ocorre, pois, as SVMs são treinadas por meio da solução de um problema de otimização quadrática com solução única, enquanto a rede RBF é treinada por um algoritmo de otimização numérica de descida em gradiente, sujeito a ótimos locais dependendo do curso da otimização.

A pesquisa de Müller *et al.* (2001) contém uma comparação relevante de dois algoritmos de aprendizado computacional no contexto de previsão séries temporais. Apesar de não terem sido utilizados dados relacionados ao tema dessa pesquisa o trabalho dos autores indica

situações em que as SVMs podem apresentar bom desempenho em termos de previsão. Todavia, uma vez que os modelos construídos utilizam apenas valores passados da série como variáveis explicativas, a aplicação é distante do cenário em que essa pesquisa se insere.

Com relação a aplicação de modelos de aprendizado computacional no domínio de aplicação dessa pesquisa é possível citar o trabalho de Alon *et al.* (2001) que apresenta uma comparação de modelos de previsão tradicionais e modelos de RNA para séries de vendas agregadas do mercado de varejo dos Estados Unidos. Os autores comparam o desempenho das RNAs com o desempenho de modelos ARIMA, modelos de suavização de Winters e modelos de regressão linear.

São utilizados dados agregados de vendas do setor varejista americano em janelas de tempo mensais (Figura 2.1).

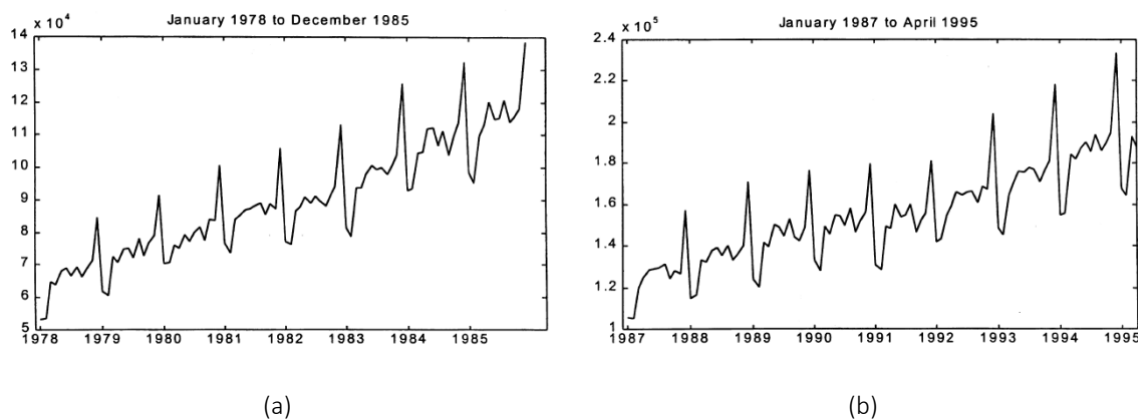


Figura 2.1 – Vendas agregadas do setor varejista dos Estados Unidos

Fonte: Alon *et al.* (2001)

A série de dados do setor varejista foi escolhida por apresentar tendência e padrões sazonais e cíclicos evidentes. São considerados dois períodos, um de 1978 a 1985 (período um – Figura 2.1a) e outro de 1986 a 1995 (período dois – Figura 2.1b). No período um, a economia americana sofreu duas recessões, alta de juros e incentivos de oferta na economia, já no período dois a economia americana passou por um período mais estável.

Os autores utilizam o erro absoluto percentual médio (MAPE) para avaliar os erros de previsão de cada modelo. Para analisar estatisticamente a diferença de precisão de dois modelos diferentes são utilizados o teste de Diebold e Mariano (*DM test*) e o teste de *Signed-Ranks* de Wilcoxon (*SR test*).

Seja $\hat{\varepsilon}_{A,t}$ o erro percentual de previsão do modelo A no instante t . A diferença de erros de previsão quadráticos de dois modelos diferentes A e B é dada pela expressão (2.1).

$$d_t = \hat{\varepsilon}_{B,t}^2 - \hat{\varepsilon}_{A,t}^2 \quad (2.1)$$

O *DM test* é baseado na estatística dada pela expressão (2.2), em que \bar{d} é o desvio percentual quadrático médio, e $\hat{f}_d(0)$ é uma estimativa da densidade espectral de d_t na frequência 0, e N é a quantidade de erros considerados para cálculo da estatística. A hipótese nula do teste representa erros iguais dos dois modelos de previsão e a distribuição da estatística dada pela expressão (2.2) é uma distribuição normal padrão.

$$DM = \frac{\bar{d}}{\sqrt{2\pi\hat{f}_d(0)N^{-1}}} \quad (2.2)$$

Já o *SR test* usa a estatística dada pela expressão (2.3), em que $I_+(d_t)$ é igual a 1 se d_t é maior que 0 e é igual a 0 caso contrário conforme a expressão (2.4), e $rank(|d_t|)$ é o rank do valor absoluto de d_t . A estatística *SR* segue assintoticamente uma distribuição normal padrão de acordo com uma escala (expressão (2.5)).

$$SR = \sum_{t=1}^N I_+(d_t) \times rank(|d_t|) \quad (2.3)$$

$$I_+(d_t) = \begin{cases} 1 & \text{se } d_t > 0 \\ 0 & \text{caso contrário} \end{cases} \quad (2.4)$$

$$\frac{SR - N(N+1)/4}{\sqrt{N(N+1)(2N+1)/24}} \approx N(0; 1) \quad (2.5)$$

Com os testes *DM test* e *SR test* é possível avaliar se um modelo é estatisticamente mais preciso que outro.

Os autores utilizam uma RNA de três camadas sendo uma camada de *inputs*, uma camada intermediária com funções logísticas de ativação dos neurônios e uma camada de *output* linear. Os *inputs* utilizados são variáveis *dummy* nos períodos sazonais e uma variável de tendência que representa o período *t* em que cada amostra se encontra. O algoritmo de inicialização dos pesos da RNA é o algoritmo proposto por Nguyen e Widrow (1990) tal que os pesos iniciais são arbitrados de acordo com a variação de cada *input* da rede. O algoritmo de treinamento utilizado é o algoritmo de Levenberg-Marquardt. Foram utilizados testes empíricos com os dados para determinação da quantidade ideal de neurônios na camada intermediária, sendo o valor ideal encontrado igual a 8.

O plano de experimento realizado pelos autores inclui previsões realizadas com múltiplos períodos no horizonte de previsão (*multistep forecast*) e previsões com apenas um período no horizonte de previsão (*one-step forecast*), sendo que no segundo caso o modelo é atualizado a cada novo dado apresentado.

Como métodos de comparação, os autores utilizaram a Suavização Exponencial de Winters, modelos ARIMA e um modelo de regressão linear múltipla que utiliza variáveis de vendas de períodos passados e variáveis *dummy* nos períodos sazonais.

A Tabela 2.1 apresenta os resultados comparados de precisão dos modelos utilizados por Alon *et al.* (2001) para os dados do período um. A Tabela 2.2 apresenta os mesmos resultados para os dados do período dois.

Tabela 2.1 – Comparação de previsão dos modelos de previsão de vendas agregadas do varejo americano (período 1 – período com oscilação econômica)

Modelo	Previsão de 1 período	(Rank)	Previsão de múltiplos períodos	(Rank)	Média	(Rank)
RNA	1.79	(1)	1.67	(1)	1.73	(1)
ARIMA	2.20	(2)	2.02	(2)	2.11	(2)
Regressão	2.44	(3)	2.66	(4)	2.55	(3)
Winters	3.11	(4)	2.27	(3)	2.69	(4)
Média	2.39		2.15		2.27	

Fonte: Alon *et al.* (2001)

Os resultados mostram que no primeiro período da série a precisão das previsões realizadas com múltiplos períodos de tempo possuem maior precisão, o que é um resultado inesperado conforme mencionado pelos autores, dado que as previsões com apenas um período no futuro usam mais informações para atualização constante dos modelos. No segundo período da série ocorre o contrário, ou seja, as previsões com um período futuro no horizonte têm maior precisão que as previsões realizadas com múltiplos períodos no horizonte. Isso sugere que em situações de instabilidade econômica, previsões com múltiplos períodos possuem maior precisão. Os autores explicam essa constatação pelo fato de que em situações de instabilidade os dados das séries temporais são mais ruidosos e a inserção de informação para atualização contínua dos modelos prejudica a precisão.

Tabela 2.2 – Comparação de previsão dos modelos de previsão de vendas agregadas do varejo americano (período 2 – período com estabilidade econômica)

Modelo	Previsão de 1 período	(Rank)	Previsão de múltiplos períodos	(Rank)	Média	(Rank)
ARIMA	1.18	(1)	1.26	(2)	1.22	(1)
RNA	1.26	(2)	1.29	(3)	1.27	(2)
Winters	2.22	(3)	1.16	(1)	1.69	(3)
Regressão	2.93	(4)	2.97	(4)	2.69	(4)
Média	1.90		1.67		1.79	

Fonte: Alon *et al.* (2001)

A Tabela 2.3 apresenta a comparação estatística da precisão dos modelos utilizados. Foram aplicados o *DM test* e o *SR test* comparando a RNA com cada um dos modelos alternativos.

Tabela 2.3 – DM test e SR test comparando a precisão dos modelos

Comparação	Período 1				Período 2			
	<i>DM test</i>	<i>p-valor</i>	<i>SR test</i>	<i>p-valor</i>	<i>DM test</i>	<i>p-valor</i>	<i>SR-test</i>	<i>p-valor</i>
Regressão vs RNA	2.4432	(0.0073)	2.0396	(0.0207)	1.8450	(0.0326)	1.8043	(0.0356)
Winters vs RNA	1.5711	(0.0581)	0.8629	(0.1941)	-1.1461	(0.8741)	-0.3922	(0.6526)
ARIMA vs RNA	1.1558	(0.1239)	1.0983	(0.1360)	-0.5972	(0.7248)	-0.8629	(0.8059)

Fonte: Alon *et al.* (2001)

Com relação ao período um, pode-se afirmar que a RNA teve melhor precisão que o modelo de regressão a um nível de 5% de significância, além de ter apresentado melhor precisão que o modelo de suavização de Winters a um nível de 10% de significância. Nesse mesmo

período de dados, a diferença de precisão do modelo de RNA e do modelo ARIMA não foi estatisticamente significativa a um nível de 10% de significância.

Com relação ao período dois, o modelo de RNA apresentou melhor precisão que o modelo de regressão a um nível de 5% de significância e não apresentou precisão estatisticamente diferente em relação aos outros modelos a um nível de 10% de significância.

Com base nos resultados do experimento e nos testes estatísticos, Alon *et al.* (2001) concluem os seguintes pontos:

- (i) Em geral, comparando diferentes tipos de previsão (*one-step* e *multiple step*) e diferentes períodos de dados, o modelo de RNA teve a melhor precisão, independente dos testes estatísticos;
- (ii) O modelo de RNA teve melhor precisão especialmente no período um de dados, que apresenta maior variância da série temporal de dados. No período dois, que apresenta maior estabilidade das condições econômicas, os modelos ARIMA e de suavização de Winters tiveram desempenho similar ao modelo de RNA;
- (iii) Em períodos de maior instabilidade previsões com múltiplos períodos no horizonte apresentam melhor precisão que previsões realizadas com atualização contínua dos modelos.

A pesquisa realizada por Alon *et al.* (2001) apresenta técnicas estatísticas válidas para comparação da precisão de diferentes modelos de previsão. Apesar dos testes estatísticos não terem encontrado diferenças significativas do modelo de RNA em comparação com os modelos utilizados como referência, os autores afirmam que a abordagem do problema de previsão com o uso de redes neurais artificiais é promissora e apresenta bons resultados. Apesar de terem sido usadas séries temporais de vendas do setor varejistas, os dados foram agregados em um nível nacional, apresentando padrões de tendência, sazonalidade e ciclo evidentes mesmo sob inspeção visual, diferentemente das séries de venda tipicamente encontradas em pontos de vendas.

Outro exemplo de aplicação de modelos de aprendizado computacional para previsão de vendas no segmento de bens de consumo pode ser encontrado em Doganis *et al.* (2006), em que os autores utilizam um modelo de inteligência computacional para previsão de vendas em uma indústria de laticínios. Nesta indústria, o tempo de vida dos produtos é curto, o que inviabiliza a manutenção de altos níveis de estoque nos diferentes elos da cadeia e por isso a previsão de vendas é uma atividade importante para a eficiência de estoques na cadeia.

Os autores comentam que os modelos tradicionalmente utilizados na indústria de alimentos são modelos da classe ARIMA e classes especiais de modelos ARIMA não lineares. A desvantagem desses modelos é que se deve conhecer a estrutura do modelo antes da aplicação do algoritmo de ajuste dos dados, o que traz a necessidade de um procedimento de tentativa e erro para determinar o melhor modelo de previsão para cada produto. Na sua pesquisa, Doganis *et al.* (2006) argumentam que técnicas de inteligência computacional permitem incorporar estruturas mais genéricas de modelos não lineares e eliminam a necessidade de procedimentos de tentativa e erro quanto a estrutura do modelo a ser ajustado.

O modelo utilizado pelos autores, denominado GA-RBF, é uma RNA com função de base radial (*Radial Basis Function* – RBF) com um algoritmo genético (GA) para seleção de *inputs*. A rede RBF é um tipo específico de rede neural com três camadas: uma camada de *inputs*, uma camada intermediária com neurônios que aplicam uma transformação não linear nos dados e uma camada linear de *output*. A transformação não linear aplicada pelos neurônios da camada intermediária ocorre de acordo com uma função de base radial, sendo essa a origem da nomenclatura desse tipo de RNA. A Figura 2.2 ilustra a topologia da rede RBF.

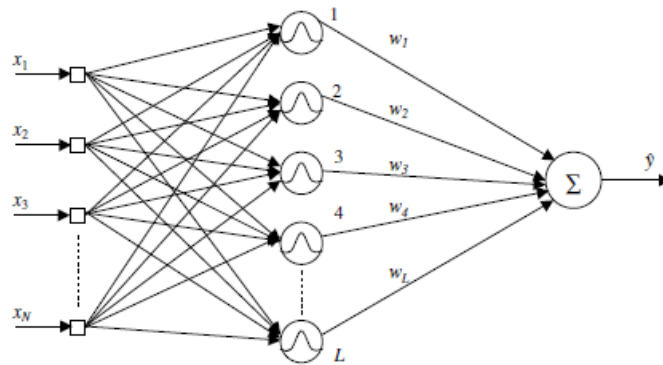


Figura 2.2 – Topologia ilustrativa de uma rede RBF

Fonte: Doganis *et al.* (2006)

Cada nó da camada intermediária da rede RBF está associado a um centro \vec{c} (centros das distribuições representadas nos nós da Figura 2.2), um vetor de dimensão igual a quantidade de *inputs* da rede RBF (N na Figura 2.2). A atividade v de um nó intermediário é equivalente a distância euclidiana entre o vetor de inputs x e o centro do nó. A função de base radial aplicada em cada nó intermediário no caso da pesquisa de Doganis *et al.* (2006) é dada pela expressão (2.6).

$$f(v) = v \log(v) \quad (2.6)$$

O algoritmo de treinamento das redes RBF pode ser formulado como um problema de minimização. Sendo L o número de nós da camada intermediária, c_j o centro do nó intermediário j , w_j o peso da conexão do nó intermediário j ao nó de output, x_i um vetor de amostras de input associado a uma saída y_i , o problema de treinamento da rede RBF é definido pelas expressões (2.7) e (2.8).

$$\min J(L, c_j, w_j) = \sum_{i=1}^K (y_i - \hat{y}_i)^2 \quad (2.7)$$

$$\hat{y}_i = \sum_{j=1}^L w_j f(\|x_i - c_j\|) \quad (2.8)$$

O problema de minimização definido por (2.7) e (2.8) é um problema de otimização não linear inteira mista, no qual a função objetivo deve ser minimizada tanto em termos da estrutura da rede (quantidade de nós intermediários), quanto em termos dos parâmetros da rede (centros e pesos das conexões). Para a solução desses problemas os autores utilizam um algoritmo denominado *Fast and Efficient Fuzzy Means Clustering Algorithm*.

Para a determinação dos inputs da rede RBF os autores utilizam um algoritmo genético em que cada indivíduo representa uma coleção possível de inputs da rede RBF. Cada indivíduo possui um vetor binário em que cada componente i representa se o input i está presente ou não na topologia da rede. Além do vetor de inputs, cada indivíduo possui um gene adicional inteiro que indica a quantidade de conjuntos *fuzzy* presentes no domínio de cada variável, sendo esse um parâmetro específico utilizado pelo algoritmo de ajuste da rede RBF.

Para validação do modelo GA-RBF os autores realizaram um experimento com dados reais de vendas diárias de um produto de uma fábrica de laticínios localizada na cidade de Atenas, Grécia. São utilizados dados de vendas diárias de um produto específico nos anos de 2001 e 2002 com o objetivo de prever as vendas diárias do mesmo produto. Para prever as vendas de um determinado dia, são consideradas 14 variáveis de input candidatas: as vendas nos últimos 6 dias do ano corrente, as vendas dos últimos 6 dias do ano passado considerando o mesmo dia no ano anterior, o percentual de alteração de vendas totais entre o ano corrente e o ano anterior e uma variável inteira que corresponde ao dia da semana do dia de uma observação. O algoritmo GA-RBF é testado em duas modalidades, uma com atualização dos pesos da rede a cada rodada de previsão na medida em que novas observações são introduzidas, e outra sem que haja ajuste dos pesos da rede a cada rodada de previsão do algoritmo. Os autores comparam o desempenho do modelo GA-RBF com outros modelos tais como modelos ARIMA e modelos de Suavização Exponencial, sendo que os resultados mostram que o modelo com melhor precisão é de fato o GA-RBF.

O resultado do experimento é apresentado na Figura 2.3. Pode-se observar que as previsões são aderentes à série diária de vendas indicando que o modelo é apropriado para prever as vendas diárias do produto. Vale notar que a série modelada é muito regular, diferente de

séries desagregadas de vendas de produtos em lojas de empresas do varejo de bens de consumo.

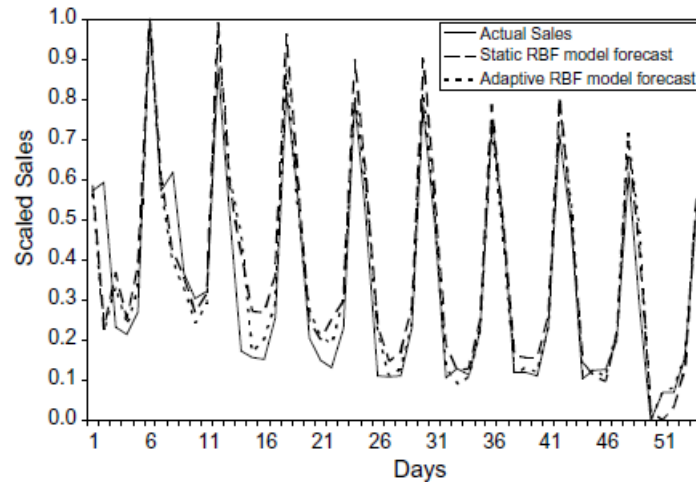


Figura 2.3 – Resultados do modelo GA-RBF

Fonte: Doganis *et al.* (2006)

Do ponto de vista do problema em questão, o algoritmo GA-RBF possui a vantagem de que seu algoritmo de treinamento ajusta não somente os pesos da estrutura da rede, mas também os inputs mais adequados e a própria estrutura em si. Essa característica de auto ajuste de estrutura, pesos e inputs é útil para configuração de sistemas de previsão não supervisionados, que devem realizar previsões de quantidades grandes de séries de vendas. No entanto, o teste realizado pelos autores foi feito em uma série temporal de vendas consolidadas, que apresenta padrões de ciclo e sazonalidade que podem ser identificados por inspeção visual da série. As séries temporais desagregadas de cadeias varejistas podem apresentar comportamentos menos estáveis o que dificulta sua previsão.

Carbonneau, Laframboise e Vahidov (2008) realizaram um estudo de aplicações de modelos de aprendizado computacional para previsão de demanda em cadeias de suprimentos. O estudo tem como foco entender como uma cadeia que utiliza esses modelos para prever a demanda reage frente a variações no sinal de demanda no elo final da cadeia. Os autores simulam sinais de demanda com variância controlada no elo final de uma cadeia fictícia e

avaliam a capacidade de previsão dos elos anteriores em preverem a demanda futura com uso de modelos de aprendizado computacional. São utilizados como modelos de previsão as RNAs e SVMs. Os autores afirmam que os resultados em termos de eficiência dos estoques são superiores em relação a metodologias convencionais de previsão de demanda.

A Figura 2.4 apresenta a série de demanda simulada. Diferente de Doganis *et al.* (2006) e Alon *et al.* (2001), a pesquisa de Carbonneau *et al.* (2008) considera uma série de vendas com padrões menos claros e não identificáveis visualmente.

A pesquisa de Carbonneau *et al.* (2008) não aborda cadeias varejistas, mas sim cadeias logísticas como um todo. Além disso o enfoque está na capacidade de elos anteriores realizarem previsões de demanda frente a distorções na demanda realizada do consumidor final. Apesar disso, o trabalho é relevante pois indica que os modelos de inteligência computacional são uma forma viável de melhorar a eficiência dos estoques na cadeia como um todo. Apesar de considerar apenas uma série de vendas, o que é diferente do cenário de previsão de demanda de um varejo, a série considerada não possui ciclos claros, nem tendência aparente, o que se aproxima das séries de vendas de cadeias varejistas.

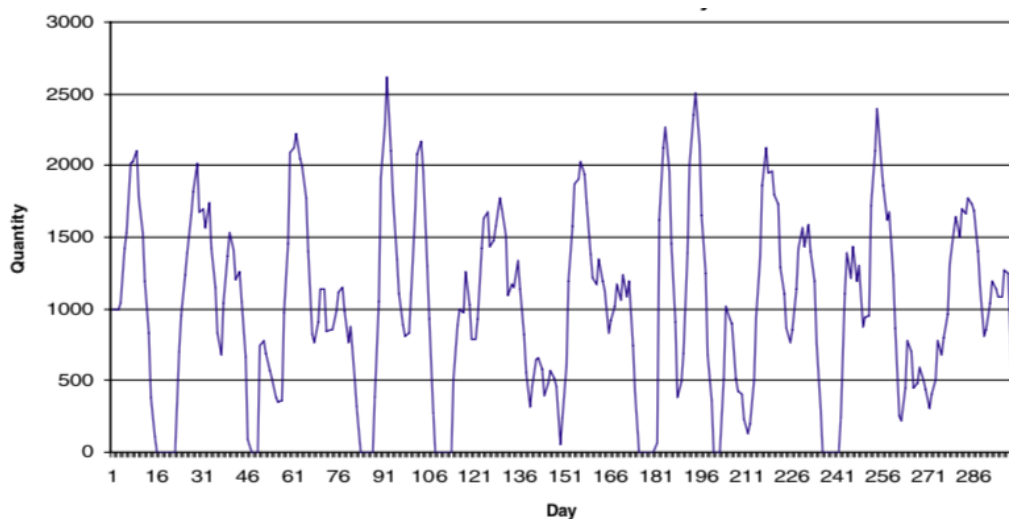


Figura 2.4 – Série de vendas simulada para avaliação da capacidade de previsão dos elos da cadeia de suprimentos

Fonte: Carbonneau *et al.* (2008)

Existem também linhas de pesquisa que investigam o uso de modelos híbridos, ou seja, a combinação de modelos convencionais (por exemplo: ARIMA) com modelos de aprendizado computacional. Nesse sentido Zhang (2003) propõe a aplicação de um modelo híbrido que combina modelos da classe ARIMA e RNAs para previsão de séries temporais em geral. O autor busca combinar a capacidade de modelagem de fenômenos lineares dos modelos ARIMA com a capacidade de modelagem de fenômenos não lineares das RNAs.

O autor considera um modelo simples em que uma série temporal y_t é composta por uma parte linear L_t e uma parte não linear N_t .

$$y_t = L_t + N_t \quad (2.9)$$

O autor utiliza um modelo ARIMA para análise da parte linear L_t , resultando nas estimativas \hat{L}_t . Os resíduos e_t resultantes dessa modelagem devem conter apenas as características não lineares do fenômeno.

$$e_t = y_t - \hat{L}_t \quad (2.10)$$

A análise de resíduos é uma parte importante da análise de modelos lineares, de modo que não deve haver estruturas de covariância presentes no ruído para que o modelo seja considerado adequado. Contudo a análise de resíduos tradicional feita pela avaliação da autocorrelação e da autocorrelação parcial dos resíduos não é capaz de evidenciar padrões não lineares nos resíduos. Assim, mesmo um modelo ARIMA, no qual os resíduos tenham sido validados pela análise tradicional, pode conter padrões não lineares não modelados o que prejudica a precisão do modelo para realização efetiva de previsões.

Assim, para a modelagem do resíduo e_t o autor utiliza uma RNA com arquitetura do tipo *feedforward* contendo uma camada de inputs, uma camada intermediária e uma camada de output. Não existem retroalimentações dentro da arquitetura da rede como no caso de redes neurais recorrentes. De acordo com Hornik, Stinchcombe e White (1990) mesmo esse tipo simples de arquitetura de RNA é muito flexível e pode aproximar qualquer função não linear dada uma quantidade suficiente de neurônios na camada intermediária. Os inputs utilizados neste caso são os termos anteriores da série de resíduos filtrada pelo modelo

ARIMA, de acordo com a expressão (2.11) em que f é uma função não linear modelada pela RNA e ε_t um termo aleatório.

$$e_t = f(e_{t-p}, e_{t-p-1}, \dots, e_{t-1}) + \varepsilon_t \quad (2.11)$$

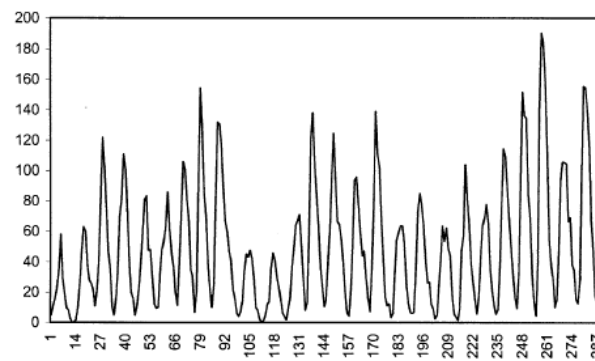
Assim, o modelo híbrido da série temporal y_t é dado pela expressão (2.12).

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \quad (2.12)$$

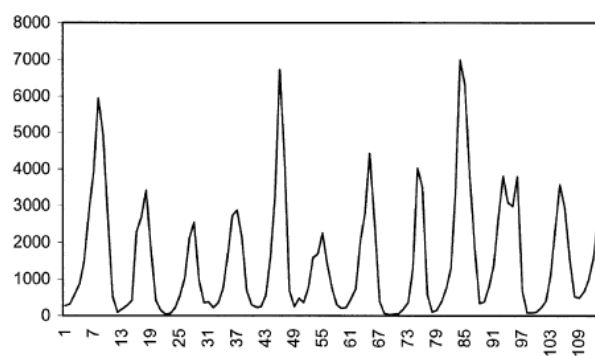
Para a aplicação da RNA proposta por Zhang (2003) é necessária a parametrização da quantidade de neurônios da camada intermediária (q) e da quantidade de resíduos que devem ser utilizadas como inputs (p). O autor realiza uma determinação empírica desses parâmetros sem o uso de nenhum resultado teórico.

Para validação da metodologia proposta, Zhang (2003) aplica o modelo proposto em três séries utilizadas frequentemente para comparação de modelos de previsão: a série *Wolf's sunspot data* (Figura 2.5a), a série *Canadian lynx* (Figura 2.5b) e a série histórica *British pound/US dollar exchange rate data* (Figura 2.5c).

Pode-se observar que as séries possuem padrões sazonais característicos, especialmente a série *Wolf's sunspot data* com um ciclo característico de aproximadamente 11 anos de acordo com estudos geofísicos. A série *Canadian lynx* representa a quantidade de lincas canadenses presos todo ano no Rio Mackenzie ao Norte do Canadá e também apresenta um padrão sazonal com período de aproximadamente 10 anos. A série *British pound/ US dollar exchange rate data* é o conjunto de dados dos três avaliados que possui maior aleatoriedade e cuja modelagem é mais difícil de acordo com o autor.



(a)



(b)



Figura 2.5 – (a) Wolf's sunspot data, (b) Canadian lynx e (c) British pound/US dollar exchange rate data

Fonte: Zhang (2003)

Os testes realizados comparam a precisão do modelo híbrido contra a aplicação de um modelo ARIMA e de uma RNA para modelagem das séries de teste. São consideradas as medidas de desvio médio absoluto e erro quadrático médio. Em todos os testes, as medidas

de precisão são calculadas considerando apenas a previsão de um período seguinte da série (*one step ahead accuracy*).

A Tabela 2.4 apresenta os resultados obtidos comparando os três modelos. Pode-se observar que o modelo híbrido possui melhor precisão em ambas as medidas de erro para todos os conjuntos de dados, exceto no caso da série *Wolf's sunspot data* na medida desvio médio absoluto. Com base nesse resultado o autor conclui que a abordagem híbrida possui melhor desempenho que os modelos ARIMA e RNA utilizados isoladamente. Não são feitos testes estatísticos para validar a melhor precisão obtida pelo modelo híbrido.

Tabela 2.4 – Comparação de precisão entre modelo ARIMA, RNA e abordagem híbrida

Série de dados	Medida de erro	Modelo		
		ARIMA	RNA	Híbrido
<i>Wolf's sunspot data</i>	MAD	11,319	10,243	10,831
	MSE	216,965	205,302	186,827
<i>Candian lynx</i>	MAD	0.112255	0.112109	0.103972
	MSE	0.020486	0.020466	0.017233
<i>British pound/US dollar exchange rate data</i>	MAD	0.005016	0.004218	0.004146
	MSE	3.68493	2.76375	2.67259

Fonte: Zhang (2003)

As séries utilizadas por Zhang (2003) possuem padrões cíclicos que podem ser observados por análise visual, diferente de séries temporais de vendas de produtos em cadeias varejistas nas quais os padrões não são tão evidentes. Além disso, as séries utilizadas possuem histórico bastante longo e possuem dados completos em todos os períodos. No caso de séries de vendas de produtos, o histórico muitas vezes não é longo, dado o ciclo de lançamentos e constantes modificações do portfólio de produtos das empresas.

Os resultados obtidos por Zhang (2003) evidenciam um aumento de precisão com a utilização de modelos híbridos para previsão de séries temporais, no entanto, uma vez que não são realizados testes estatísticos, não fica clara a real vantagem do modelo híbrido na pesquisa apresentada.

Além disso, o modelo proposto por Zhang (2003) é bem geral pois utiliza apenas os valores da série temporal, e em teoria poderia ser aplicado ao problema de previsão de demanda.

O autor também traz uma evidência de que modelos híbridos tendem a ter melhores resultados que modelos estatísticos convencionais (ARIMA) e modelos de aprendizado computacional (RNAs) aplicados isoladamente (Tabela 2.4). Contudo o autor testou apenas em séries muito simples, com padrões de tendência e sazonalidade identificáveis por inspeção visual. Além disso, dado que a parte linear do modelo proposto por Zhang (2003) é um modelo ARIMA que precisa ser parametrizado por especialistas, ele não atende o critério de automação para previsão de muitos produtos em muitas lojas como no varejo de bens de consumo.

Guo, Wong e Li (2013) aplicaram um modelo híbrido para previsão de demanda em uma cadeia varejista chinesa do setor de vestuário e moda. A organização de vendas desse setor trabalha com planejamento de demanda de uma categoria de vestuário para campanhas que ocorrem a cada estação do ano. Sendo assim, o problema tratado pelos autores é determinar as vendas de uma categoria de produtos em estação com base nas vendas iniciais da campanha (*early sales*) e em outros fatores exógenos. Para isso é proposto um modelo denominado *Multivariate Intelligent Decision-Making Model* (MID). O modelo MID contém três módulos:

- (i) um módulo de pré-processamento dos dados de entrada é utilizado para capturar os dados dos pontos de venda e organizar os dados das variáveis exógenas;
- (ii) um módulo de seleção de variáveis chamada de *Harmony Search-wrapper-based variable selection* que determina um subconjunto adequado de variáveis de entrada para previsão das vendas de uma campanha;
- (iii) e um módulo de previsão denominado *Multivariate Intelligent Forecaster* que é treinado com dados amostrais e é efetivamente utilizado para prever as vendas de uma estação.

No problema em questão, Guo *et al.* (2013) consideram as vendas iniciais da campanha e algumas variáveis exógenas como candidatas a fatores de influência nas vendas, são elas:

- (i) Vendas iniciais: os autores consideram as vendas acumuladas nos primeiros 3, 7, 10 e 14 dias;
- (ii) Preço de venda original: preço planejado do produto sem considerar descontos;
- (iii) Estilo do produto: indica o tipo de mercadoria, por exemplo, camisetas, calças ou jaquetas;
- (iv) Material do produto: indica o material principal do produto, por exemplo, algodão, couro ou poliéster;
- (v) Estratégia de promoção: os autores classificam as estratégias de promoção em muito agressiva, parcialmente agressiva e média de acordo com o orçamento de *marketing* de cada produto;
- (vi) Quantidade de lojas: indica a quantidade de lojas em que o produto estará disponível de acordo com o planejamento;
- (vii) Data de lançamento: data em que o produto é efetivamente colocado a disposição dos consumidores nas lojas físicas;
- (viii) Tempo de vida: indica por quanto tempo o produto ficará disponível para compra;
- (ix) Índice climático: indica a temperatura média prevista da campanha de um produto contada a partir da data de lançamento por todo o tempo de vida. Diferentes produtos possuem diferentes índices climáticos dependendo da data de lançamento e tempo de vida;
- (x) Índice econômico: média dos últimos seis meses de indicadores de produção e de preço do setor de vestuário.

O módulo de pré-processamento, além de organizar essas variáveis, realiza a limpeza dos dados, interpolando observações faltantes ou preenchendo janelas de dados faltantes com a média dos dados vizinhos.

O módulo de seleção de variáveis do modelo MID visa determinar a melhor combinação de variáveis de entrada para a previsão das vendas de um determinado produto. Algumas das variáveis candidatas são qualitativas e os autores usam uma conversão numérica para que sejam tratadas corretamente no módulo de seleção de variáveis. Após a seleção de variáveis de entrada realizada pelos autores, é utilizado um algoritmo chamado *Harmony Search* para explorar o espaço de busca de todas as combinações possíveis de inputs.

O módulo de previsão é um algoritmo de RNA que recebe o subconjunto de inputs da camada de seleção de variáveis e os dados de entrada para realização das previsões. Para cada previsão de vendas a ser realizada, a RNA é aplicada por um número finito de vezes e o conjunto de previsões é analisado para compor a previsão final de vendas. São removidos valores extremos e a média das previsões restantes no conjunto é tomada como a previsão de vendas para os próximos períodos. Os autores concluem que o módulo de seleção de variáveis foi vantajoso para todos testes realizados. Considerando o problema abordado nessa pesquisa, verifica-se o benefício em termos de melhoria de precisão resultante da combinação de modelos de previsão e técnicas de otimização para seleção de variáveis de entrada.

Por um lado, o trabalho de Guo *et al.* (2013) é relevante para essa pesquisa pois apresenta uma combinação de várias técnicas de inteligência computacional para solução de um problema de previsão de demanda em uma empresa varejista de itens de moda, que também são caracterizados como bens de consumo. Os autores também fazem recomendações interessantes de variáveis de entrada incluindo variáveis relacionadas a preços e campanhas de *marketing*. Além disso os autores também resolvem o problema de seleção de variáveis com um algoritmo de inteligência computacional. Por outro lado, o trabalho dos autores é muito direcionado para empresas de moda, as variáveis utilizadas são específicas desse mercado (por exemplo: estilo e material do produto) e não estão disponíveis em casos mais gerais de empresas de bens de consumo. Por fim as vendas são previstas de forma agregada, ou seja, é feita uma previsão para cada produto, mas não para cada loja da empresa, o que foge do propósito dessa pesquisa.

Existem também trabalhos que possuem foco em problemas específicos da previsão de demanda no varejo de bens de consumo. Thomassey e Fiordaliso (2006) propuseram a aplicação de um algoritmo de previsão de perfil de vendas baseado em modelos de agrupamento e árvores de decisão para solucionar o problema de previsão de vendas de lançamentos na indústria de moda. Os autores analisam as características da indústria têxtil e citam cinco fatores significativos desse mercado que tornam complexa a previsão de vendas:

- (i) Alto número de produtos diferentes a cada coleção devido a pressão de *marketing* por diversidade (15.000 itens por ano no caso tratado pelos autores);
- (ii) Baixo ciclo de vida dos produtos devido ao rápido ciclo de inovação da indústria de moda (6 a 12 semanas no caso tratado pelos autores);
- (iii) Renovação dos portfólios das empresas a cada coleção;
- (iv) Longos *leadtimes* para desenvolvimento de produtos, produção e distribuição e consequente necessidade de horizontes longos de previsão;
- (v) Grande influência de fatores externos ao mercado como por exemplo: aspectos climáticos, ocorrência de feriados e datas especiais de vendas, promoções e ambiente econômico.

O domínio de aplicação da pesquisa de Thomassey e Fiordaliso (2006) é o mesmo que o dessa pesquisa, contudo o problema não é a previsão de vendas em si, mas sim o perfil ou distribuição de vendas no tempo. Não obstante, o modelo dos autores contém aspectos interessantes para essa pesquisa. O modelo proposto pelos autores segue os seguintes passos:

1. Preparação dos dados: as vendas históricas dos itens são normalizadas para caracterizar seu perfil de vendas;
2. Agrupamento: Aplicação do algoritmo k-médias para determinar grupos de itens de acordo com seus perfis de venda normalizados. Isso produz classificações para os itens que serão utilizadas no estágio subsequente de classificação do processo;

3. Classificação: os itens agrupados, juntamente com atributos descritivos dos mesmos (preço, estilo, data de lançamento, etc), são submetidos a um processo de construção de uma árvore de classificação. O processo *k-fold cross validation* com *k* igual a 10 é aplicado para a construção do melhor classificador.
4. Previsão: novos itens, juntamente com atributos descritivos, são classificados pela árvore, resultando em perfis de venda previstos para cada item novo.

Os autores argumentam que apesar das redes neurais possuírem maior capacidade de generalização de dados não observados, as árvores de decisão são melhores para fornecer conhecimento sobre o problema modelado uma vez que resultam em modelos cuja estrutura pode ser interpretada e até traduzida em simples regras de negócio. Existem múltiplos algoritmos de treinamento de árvores de classificação. Os autores utilizam o algoritmo C4.5 (QUINLAN; 1993) por sua capacidade de previsão e possibilidade de tratamento de atributos numéricos e nominais. O apêndice A.2.2 apresenta maiores detalhes sobre o funcionamento de árvores de decisão.

O processo de preparação dos dados envolve duas etapas: normalização da quantidade total de vendas e normalização do ciclo de vida. A normalização da quantidade total de vendas é feita dividindo-se a venda de cada período pela quantidade total de vendas no período. A normalização do ciclo de vida de um item é feita por uma transformação homotética. A Figura 2.6 ilustra a normalização do perfil de vendas de um item.

O processo de agrupamento é realizado considerando as vendas de cada período do perfil normalizado dos itens como uma dimensão de uma amostra. Dessa forma, se os perfis foram normalizados em 52 semanas, cada perfil de cada item possui 52 dimensões para o algoritmo de agrupamento. Os autores utilizam a distância euclidiana como medida de dissimilaridade das amostras. O objetivo é determinar os centroides dos grupos, que representam perfis de venda médios para os itens do grupo. A Figura 2.7 apresenta exemplos de centroides e perfis de venda agrupados pelo algoritmo *k-means*. Thomassey e Fiordaliso (2006) dão o nome de “protótipo” ao perfil de vendas do centroide de cada cluster.

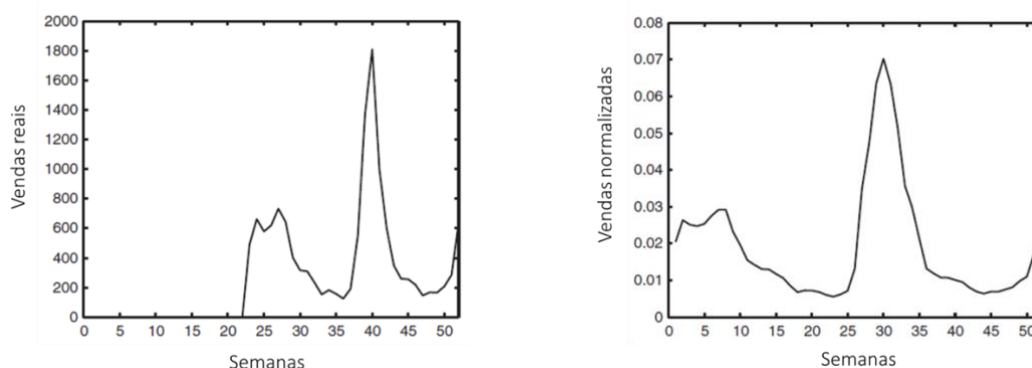


Figura 2.6 – Normalização do perfil de vendas

Fonte: Thomassey e Fiordaliso (2006)

O experimento conduzido por Thomassey e Fiordaliso (2006) para validação do modelo proposto considera dados de uma empresa francesa de desenvolvimento e distribuição de artigos de vestuário. São utilizados dados de itens de 1998 e 1999 para construção do modelo (total de 482 itens) e dados de 2000 (285 itens) para validação de protótipos previstos pela árvore de classificação resultante. A transformação homotética de normalização do perfil de vendas foi realizada considerando um perfil de 52 semanas de vendas que representa um período de um ano. Os atributos descritivos utilizados foram o preço de lançamento, semana de lançamento e tempo de vida dos itens.

O modelo proposto por Thomassey e Fiordaliso (2006) é uma abordagem interessante para a previsão de vendas de lançamentos em cenários com constante substituição de portfólio, o que ocorre com frequência em cadeias varejistas. Além disso, o modelo se mostra de fácil aplicação com baixa necessidade de preparação de dados. Diferentemente de outros trabalhos apresentados nessa revisão bibliográfica, os autores não utilizam RNAs, mas sim árvores de decisão, sendo que o principal motivador dessa escolha é a interpretabilidade de tais modelos. No entanto, a pesquisa de Thomassey e Fiordaliso (2006) resulta na previsão apenas do perfil agregado de vendas (protótipos de vendas) e não as vendas totais dos itens em diferentes pontos de venda, o que seria necessário para planejamento de abastecimento de cadeias varejistas.

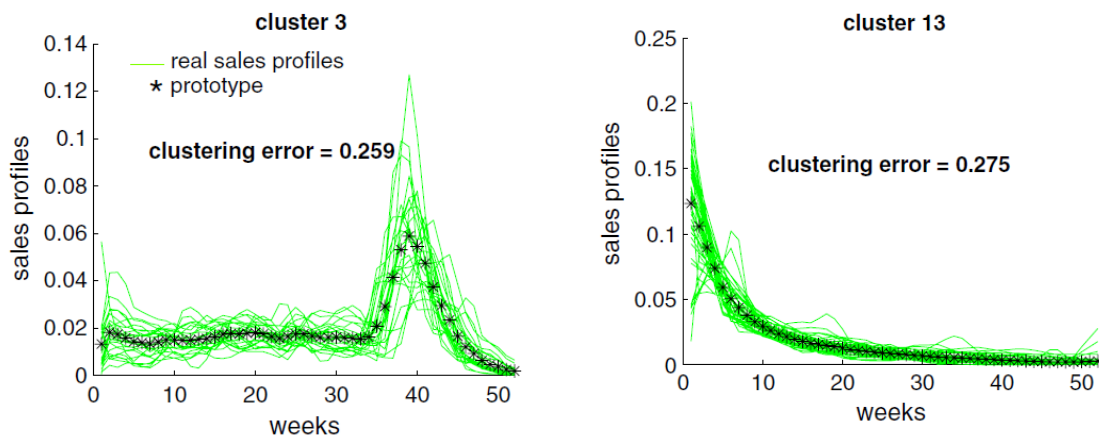


Figura 2.7 – Exemplos do agrupamento de perfis de venda

Fonte: Thomassey e Fiordaliso (2006)

A atenção dada ao problema de previsão desagregada no varejo é recente. Huang *et al.* (2019) apresentam em sua pesquisa uma metodologia de ajuste de modelos de previsão para o problema de previsão desagregada no varejo. Dentre os trabalhos encontrados na literatura esse foi o único que trata do mesmo tema desta pesquisa. Os autores afirmam que ações promocionais e de marketing fazem com que a demanda dos produtos nas lojas de uma cadeia varejista apresente uma mudança de regime. Essa afirmação equivale a dizer que o fenômeno gerador de demanda fora de épocas promocionais é um e dentro de épocas promocionais é outro. Por isso, metodologias estatísticas convencionais que modelam a série de vendas de forma única precisam ser ajustadas para reduzir os erros de previsão.

Os autores propõem um método de previsão de demanda em três etapas:

1. Na primeira etapa aplica-se um modelo chamado *Least Absolute Shrinkage and Selection Operator* (LASSO) para identificação das variáveis mais relacionadas com a demanda de um produto. Nesta etapa os autores buscam identificar a influência das variáveis de preços, atributos classificadores (por exemplo o peso) e tipo de exposição dos produtos de uma categoria, nas vendas de um produto focal;
2. Com base no resultado da primeira etapa, os autores utilizam novamente o modelo LASSO para montar um modelo de previsão utilizando os valores autorregressivos

das variáveis identificadas. Adicionalmente, os autores introduzem variáveis que representam os ciclos do mês e do ano;

3. Como último passo, são aplicados dois métodos de correção das previsões denominados Correção de Intercepto (*Intercept Correction* – IC) e Combinação de Janelas de Estimação (*Estimation Window Combining* – EWC) para contemplar possíveis mudanças de regime na série de vendas.

Os autores realizam um teste em uma base de dados com informações de vendas e campanhas promocionais coletadas por 202 semanas para 1831 produtos e 28 lojas e comparam os resultados com um modelo de suavização exponencial. Os resultados apontam que o método proposto é mais preciso que o método de comparação e funciona adequadamente mesmo sem a presença das variáveis de atributos classificadores e tipo de exposição.

O trabalho de Huang *et al.* (2019) é relevante para essa pesquisa pois trata do mesmo tema. Os resultados apontam que métodos de aprendizado computacional como o LASSO podem trazer ganhos de precisão na previsão de demanda. No entanto, a metodologia proposta ainda possui algumas escolhas humanas, como por exemplo a quantidade de *lags* autorregressivos a ser utilizada na segunda parte da metodologia proposta, e utiliza apenas um método de aprendizado computacional (LASSO) que é apenas capaz de capturar relações lineares entre as variáveis de entrada e a demanda. Um dos propósitos dessa pesquisa é que a metodologia seja ainda mais independente de escolhas humanas e seja capaz de avaliar uma variedade maior de métodos de aprendizado computacional.

Trabalhos mais recentes apontam que as condições atuais do segmento varejista são particularmente favoráveis para o uso de modelos de aprendizado computacional. Boone *et al.* (2019) realizaram uma pesquisa sobre as mudanças e oportunidades na prática de previsão de demanda no varejo, considerando o recente aumento de disponibilidade de dados. Os autores argumentam que nos últimos 10 anos houve uma popularização de sistemas de informação e ferramentas de coleta de dados em cadeias varejistas. Com isso há oportunidade de aprimoramento na teoria e prática de previsão de demanda e análise do comportamento do consumidor em geral. Os autores afirmam que a prática de mercado

mais usual é o uso de métodos qualitativos em que especialistas fazem julgamentos sobre a demanda futura e modificam previsões quantitativas feitas por modelos matemáticos. Boone *et al.* (2019) dizem que isso resulta num decréscimo de precisão uma vez que os especialistas tentem a introduzir um viés nas suas estimativas. Por fim, os autores dizem que modelos de aprendizado computacional são um caminho promissor para uso de toda a informações disponível e que isso deve vir acompanhado de ações de capacitação das pessoas que conduzem processos de previsão de demanda.

2.3. Aplicação de modelos de séries temporais

A utilização de modelos de suavização e modelos de séries temporais da classe ARIMA tem sido a metodologia convencionalmente utilizada para previsão de séries temporais em geral (MORETTIN e TOLOI, 2004). Mais recentemente, modelos de espaços de estado, por serem capazes de generalizar a classe de modelos ARIMA, têm sido aplicados na literatura em estudos de casos de previsão de séries temporais (FILDES *et al.*, 2008).

Ramos *et al.* (2015) comparam a performance dos modelos ARIMA e dos modelos de espaços de estado para previsão de consumo de produtos no varejo. Os autores afirmam que a modelagem de séries de vendas do varejo ainda é uma questão em aberto no campo de previsão de demanda. Os autores realizam o estudo levando em consideração os dados de uma empresa varejista portuguesa do ramo de calçados chamada Foreva, que possui 70 lojas por todo o território português. Foram analisados dados de 2007 a 2012 de vendas agregadas mensalmente de 5 categorias (Figura 2.8). Os autores realizaram o ajuste de modelos ETS e ARIMA para cada série temporal. Vale ressaltar que as séries de vendas possuem padrões regulares identificáveis visualmente, ou seja, são muito distintas das séries que devem ser consideradas no problema de previsão desagregada em cadeias varejistas.

No seu estudo de caso, Ramos *et al.* (2015) utilizaram uma parte dos dados para ajuste do modelo e outra parte para testes de acurácia. Os autores apresentam diferentes medidas de acurácia e concluem que os modelos ARIMA e ETS produzem resultados equivalentes não sendo possível eleger uma classe de modelo com maior precisão.

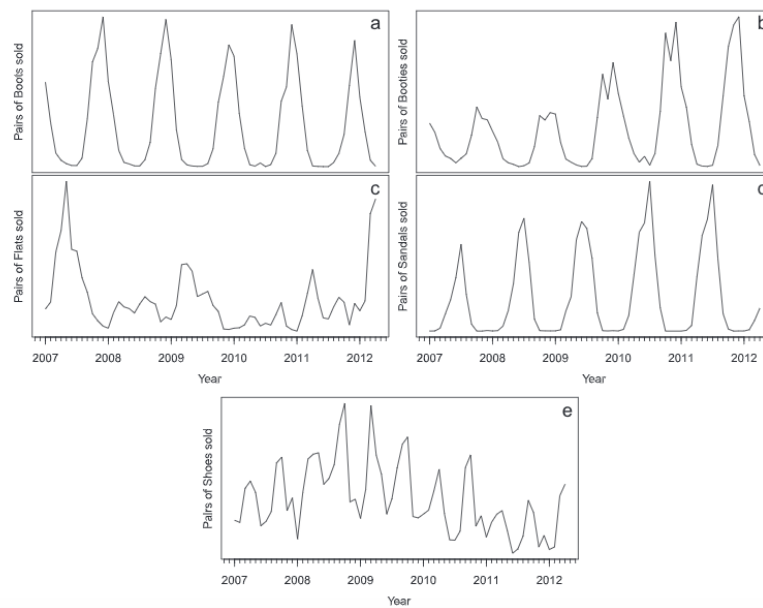


Figura 2.8 – Vendas mensais de cinco categorias do varejista Foreva

Fonte: Ramos *et al.* (2015)

O estudo é relevante na medida em que determina uma equivalência entre as classes de modelos ETS e ARIMA e é aplicado no mercado de vestuário que pode ser classificado como um tipo de varejo de bens de consumo. Contudo, os autores tratam de poucas séries com padrões muito regulares, e utilizam metodologias que necessitam de conhecimento de especialistas para serem aplicadas, de modo que tal metodologia não poderia ser aplicada de forma viável para o problema de previsão desagregadas no varejo de bens de consumo.

Veiga *et al.* (2016) apresentam um estudo comparando modelos convencionais de séries temporais com modelos não lineares para o problema de previsão de demanda de três grupos de alimentos perecíveis numa cadeia logística. Em seu estudo, os autores comparam modelos convencionais, nominalmente modelos ARIMA e a suavização exponencial de Holt Winters (HW), com modelos não lineares, nominalmente redes neurais de ondaletas (*Wavelet Neural Networks* – WNN) e Takagi-Sugeno Fuzzy System (modelo TS). A razão para escolha dos modelos ARIMA e HW é devido ao fato de modelos ARIMA cobrirem boa parte do espectro de modelos de suavização e o modelo HW ser consideravelmente utilizado na prática.

Os autores utilizam dados de vendas uma empresa de laticínios brasileira que opera no território nacional de 2005 a 2013. Os dados considerados são referentes a três categorias de produtos (iogurte, leite fermentado e sobremesas). As séries de venda são agrupadas para todas as lojas atendidas pela empresa, para todos os produtos de uma categoria e em janelas mensais. Os autores afirmam que as séries agregadas exibem padrões de tendência e sazonalidade que podem ser capturados pelos modelos propostos.

Além de utilizarem métricas convencionais de medição de acurácia, os autores utilizam uma métrica de *fill rate* que avalia o quanto da demanda foi atendida em cada período de previsão. A utilização dessa métrica ajuda a avaliar o impacto do modelo nos resultados de uma operação varejista. Os autores concluem que o modelo WNN tem melhor desempenho considerando todas as métricas de avaliação.

O estudo apresentado por Veiga *et al.* (2016) é relevante para a pesquisa pois aponta que modelos de inteligência computacional podem ser promissores para previsão de vendas no varejo em comparação com técnicas convencionais de séries temporais. No entanto, são consideradas séries num nível de agregação muito alto, o que não pode ser utilizado no dia a dia de operações de abastecimento de cadeias varejistas que precisam de previsões de mais curto prazo e em níveis de granularidade maiores. Além disso, uma vez que as séries são agrupadas por categoria de produtos, a alteração de produtos dentro da categoria não afeta o histórico de vendas, sendo possível trabalhar com uma disponibilidade muito maior de histórico de vendas. Isso não se verifica quando se utilizam séries desagregadas de produtos, que possuem um histórico limitado pelas datas de lançamento dos produtos.

Goodness *et al.* (2015) realizaram uma aplicação de modelos de séries temporais para previsão de vendas agregadas no setor de varejo da África do Sul. Os autores comentam que o desafio de previsão de demanda é ainda mais complexo em economias emergentes que estão sujeitas a alterações estruturais com maior frequência que mercados e economias mais estabelecidas. Os autores consideram dados de consumo no varejo da economia sul-africana de 1970 a 2012, um período longo em que tanto o país quanto o restante do mundo passaram por mudanças estruturais e geopolíticas significativas. Os autores buscam

identificar as estratégias de previsão que permitem ser efetivas tanto em épocas de crise financeira e recessão, quanto em período de estabilidade e períodos de crescimento.

Na pesquisa de Goodness *et al.* (2015) são considerados 23 modelos diferentes, divididos em 2 grupos: um grupo de modelos com variáveis sazonais auxiliares (*dummy seasonal variables*) e outro grupo com estruturas de sazonalidade implícitas. Os modelos variam de modelos da classe ARIMA, modelos da classe ETS e até mesmo modelos baseados em redes neurais *feed forward*. Além dos 23 modelos, os autores também consideram conjuntos de modelos combinados (*ensemble*) de acordo com estratégias diferentes. As metodologias de combinação são: combinação pela média aritmética simples dos modelos, combinação pela média ponderada em que os pesos são dados pelo inverso dos erros dos modelos na amostra de treino e combinação pelos componentes principais das previsões dos modelos. Os autores argumentam que combinações de modelos produzem metodologias de previsão com melhor acurácia e menor variância.

Os autores concluem que os melhores resultados são obtidos pela combinação de modelos utilizando a estratégia de ponderação pelo inverso dos erros do modelo. O argumento que explica esse resultado é que diferentes modelos capturam diferentes características da série temporal e a metodologia de ponderação permite dar maior importância para os modelos que capturam as características mais proeminentes.

O trabalho de Goodness *et al.* (2015) apesar de tratar de um problema de previsão de vendas no segmento varejista, o faz num contexto macroeconômico, diferente do contexto microeconômico em que a presente pesquisa se insere. Além disso, Goodness *et al.* (2015) tratam de apenas uma única série temporal e os autores realizam uma análise bastante detalhada da série, o que é oposto ao problema abordado nessa pesquisa que considera um problema de muitas séries e a necessidade de construir modelos com pouca intervenção humana dada uma quantidade grande de séries de vendas. Não obstante, a vantagem de combinar modelos para estruturar metodologias de previsão é algo também verificado nas aplicações de modelos de inteligência computacional.

2.4. Visão geral da revisão bibliográfica

Do ponto de vista de aplicação de modelos de aprendizado computacional, a revisão bibliográfica inicia com a primeira tentativa de automação do procedimento de modelagem ARIMA realizada por Leh e Oh (1996). Os autores utilizam métodos de aprendizado computacional visando eliminar a necessidade de um especialista para modelagem de séries temporais. Ainda que bem-sucedidos, os modelos resultantes são lineares, não capturando eventuais efeitos não lineares nas séries e não fazem uso direto dos modelos de aprendizado computacional.

Na sequência são considerados os trabalhos de Zhang (1998), Muller *et al.* (2001), Alon *et al.* (2001), Doganis *et al.* (2006) e Carbonneau *et al.* (2008). Essas pesquisas aplicam modelos de inteligência computacional diretamente no problema de modelagem de séries temporais de demanda. Os trabalhos analisados são bem-sucedidos na aplicação, porém tratam de quantidades pequenas de séries de vendas com padrões de sazonalidade de tendência identificáveis por inspeção visual. Além disso as abordagens possuem alta dependência de intervenção de especialistas quanto a aplicação dos modelos.

As pesquisas de Zhang (2003), Guo *et al.* (2013) e Thomassey e Fiordalisso (2006) constituem um passo adiante na investigação do problema pois consideram abordagens híbridas, combinando modelos convencionais de séries temporais com modelos de aprendizado computacional. Duas das aplicações são realizadas no segmento de vestuário que pode ser categorizado como um varejo de bens de consumo. A questão principal desses trabalhos que diverge do tema desta pesquisa é que as séries são tratadas de forma agregada. Não é dada atenção ao problema de previsão desagregada que de fato subsidia as decisões de abastecimento de pontos de venda.

Numa pesquisa mais recente Huang *et al.* (2019) abordam o problema de previsão desagregada em cadeias varejistas e propõem um modelo linear de inteligência computacional. Os resultados são comparados com um modelo muito simples de suavização exponencial simples. Não obstante, os resultados indicam que o modelo proposto fornece previsões com menos erros.

Com relação aos trabalhos de previsão de demanda com modelos convencionais os trabalhos levantados na revisão bibliográfica tratam de pequenas quantidades de séries, algumas com padrões identificáveis por inspeção visual e tratam as séries de forma agregada. Os trabalhos relacionados tratam do problema de previsão sob uma ótica agregada e não desagregada como é o caso dessa pesquisa.

Em resumo, as aplicações de previsão de demanda encontradas na literatura possuem as seguintes características:

- (i) Em geral o enfoque é na previsão de demanda em nível agregado, ou seja, previsões de demanda de fábricas e centros de distribuição, ou mesmo de mercados regionais inteiros. Não foram encontrados muitos trabalhos referentes ao problema de previsão desagregada de demanda no varejo (RAMOS *et al.*; 2015, VEIGA *et al.*; 2016, GOODNESS *et al.*; 2015);
- (ii) As séries de vendas encontradas no varejo possuem descontinuidade de informações devido a lançamentos e descontinuidade de comercialização de produtos em períodos específicos, possuem informações de demanda prejudicadas por rupturas de estoques, além de possuírem picos e vales de demanda ocasionados por eventos locais dos pontos de venda não mapeados por variáveis específicas (por exemplo: promoções e descontos). As aplicações de inteligência computacional encontradas na literatura lidam com problemas agregados de previsão de demanda que tratam de séries com comportamento característico de tendência e sazonalidade. Portanto, as aplicações encontradas na literatura não lidam com dados de demanda relativos ao problema de previsão de demanda desagregada (GUO *et al.*, 2013; ZHANG, 2003; Doganis *et al.*, 2006 e THOMASSEY e FIORDALISO, 2006);
- (iii) Os modelos de inteligência computacional aplicados na literatura levantada exigem alta atuação humana na sua construção, ou seja, não há um enfoque específico na automação do tratamento dos dados e construção dos modelos para que sejam aplicados numa grande quantidade de pontos de venda e

produtos com baixa intervenção humana (GUO *et al.*, 2013; Doganis *et al.*, 2006 e THOMASSEY e FIORDALISO, 2006).

Sendo assim, conclui-se que as aplicações encontradas na literatura deixam uma lacuna no caso de aplicações práticas de previsão de demanda em cadeias varejistas. A revisão também indica que modelos de aprendizado computacional são um caminho promissor para solucionar esse problema. Essa pesquisa busca preencher essa lacuna pela proposição de uma metodologia de previsão de demanda que contém algoritmos de inteligência computacional para utilização massificada no planejamento de demanda de cadeias varejistas.

3. CARACTERIZAÇÃO DO PROBLEMA

Este capítulo visa caracterizar o problema de previsão de demanda desagregada em cadeias varejistas. A seção 3.1 apresenta uma definição para cadeias de varejistas. Em seguida, nas seções 3.2, 3.3 e 3.4, são apresentadas quatro definições alternativas para o problema de previsão de demanda no contexto da pesquisa. Por fim, na seção 3.5 é apresentada uma discussão sobre as diferentes visões do problema em questão.

3.1. Cadeias varejistas

Empresas varejistas são aquelas que conduzem negócios diretamente com os consumidores finais, seja por meios físicos (ex.: lojas) seja por meios digitais (ex.: e-commerce). Além disso, uma cadeia varejista também se caracteriza pela venda de seus produtos em pequenas quantidades para consumo e não para revenda (RANDALL *et al.*, 2011).

De acordo com Adivar, Hüseyinoğlu e Christopher (2019), considerando a digitalização do consumidor final que ocorreu nos últimos anos, as operações varejistas devem ter dois objetivos:

- Um no consumidor final, com objetivo de melhorar a experiência de compra, aumentar a efetividade de canais de atendimento e suporte, direcionar as ações promocionais e adequar a variedade de produtos;
- Outro na cadeia logística, com objetivo de otimizar a disponibilidade de estoque nos pontos de consumo, reduzir os tempos de entrega e reduzir a ocorrência de *stockouts*.

Adivar *et al.* (2019) afirmam que as operações varejistas evoluíram nos seguintes passos:

- Nos anos 1970 as operações varejistas eram caracterizadas por cadeias de suprimentos de dois elo, em que fornecedores realizavam entregas direcionadas às lojas que atendiam aos consumidores finais;

- Os anos 1980 foram marcados pela centralização das operações varejistas em cadeias de três elos, com centros de distribuição que consolidam entregas de fornecedores e enviam as mercadorias para as lojas;
- Durante os anos 1990, a implementação de conceitos de cooperação na cadeia de suprimentos e *global sourcing* as cadeias logísticas varejistas se tornaram mais complexas englobando três ou mais elos, envolvendo integração de fornecedores, e mais pontos de consolidação e fracionamento de cargas.
- Os anos 2000 foram caracterizados pela expansão do canal digital e as cadeias varejistas iniciaram uma competição no *e-commerce* que elimina as barreiras físicas entre o consumo e a oferta. Nesse sentido, a exigência dos consumidores por tempos de entrega menores tornou ainda mais importante a necessidade de estoques mais bem posicionais.
- As décadas seguintes, envolvendo tempos atuais foram marcadas pelo crescimento dos canais digitais, como por exemplo o canal *mobile*, e pela criação do conceito de *omni-channel* que aumenta ainda mais a concorrência entre os varejistas.

Randall *et al.* (2011) realizam uma pesquisa sobre as estratégias e principais preocupações de empresas varejistas. Os autores afirmam que as decisões estratégicas de definição de malha logística (localização de instalações e políticas de níveis de serviço) que tradicionalmente eram tomadas pelos fabricantes de bens de consumo são hoje mais influenciadas pelos varejos uma vez que esses detêm o canal de relacionamento com o consumidor final. Os autores ainda afirmam que a disponibilidade de produtos é uma das principais preocupações dos varejistas, decorrente de desafios de atendimento e previsão de demanda.

3.2. Previsão de demanda

Seja \mathcal{U} uma companhia varejista que comercializa k produtos e possui i pontos de venda. O conjunto de i pontos de venda contempla não somente o canal de lojas físicas, mas também canais virtuais de vendas (*e-commerce*), sendo cada canal modelado com pontos de venda. As vendas são realizadas para o consumidor final.

Para cada par i, k é desejado prever qual será a demanda futura para uma quantidade h de períodos de tempo.

São conhecidas as vendas históricas para esses k produtos nos i pontos de venda. Cada produto possui uma data de lançamento diferente, portanto a quantidade de observações das séries de venda é variável. Com essa definição são excluídos lançamentos de produtos novos.

O conceito de produto deve ser interpretado em cada caso de aplicação. Deve-se considerar o produto como a unidade mínima de planejamento de demanda da companhia \mathcal{U} . Por exemplo, no segmento supermercadista, o produto pode ser considerado como a mercadoria definida pelo *Stock Keeping Unit* (SKU).

Uma vez que o problema está inserido no contexto de comercialização de produtos ao consumidor final, há um conhecimento *a priori* do comportamento de demanda em períodos específicos. Exemplos desse tipo de situação são datas comemorativas em que as vendas são conhecidamente mais elevadas (e.g.: vendas de chocolates próximo ao feriado de Páscoa), ou mesmo estações do ano em que o patamar de vendas é conhecidamente mais baixo (ex.: vendas de ar condicionado no inverno). Assim, são conhecidos os períodos t em que a demanda pelos produtos k da companhia \mathcal{U} é influenciada por eventos sazonais. Para cada evento sazonal r é definido um sinal s_r , para cada evento conhecido *a priori*.

Seja k o índice que representa os produtos do conjunto K dos produtos da companhia \mathcal{U} . Seja i um índice que representa um ponto de venda do conjunto I de pontos de venda de \mathcal{U} . Seja t um índice que representa o tempo T agrupado em períodos iguais e ordenados de forma crescente.

$$i \in I = \{0, 1, 2, 3, \dots, n\} \quad (3.1)$$

$$k \in K = \{0, 1, 2, 3, \dots, p\} \quad (3.2)$$

$$t \in T \quad (3.3)$$

Em cada período de tempo passado t são conhecidas as seguintes grandezas:

1. Estoque disponível em número de unidades (E_{ik}^t);
2. Vendas realizadas em número de unidades (V_{ik}^t);
3. Preço de venda médio praticado em valor monetário (P_{ik}^t)

Para a caracterização do problema assume-se que as vendas de um ponto de venda i_1 não possuem relação com as vendas de outro ponto de venda $i_2 \neq i_1$.

Existe a possibilidade de indisponibilidade de estoque do produto k no ponto de venda i no período t . Denomina-se de ruptura, ou *stockout*, o evento de indisponibilidade de estoque do produto k . O evento de ruptura é caracterizado por uma variável binária conforme a expressão (3.4).

$$rupt_{ik}^t = \begin{cases} 1 & \text{se } E_{ik}^t = 0 \text{ e } V_{ik}^t = 0 \\ 0 & \text{caso contrário} \end{cases} \quad (3.4)$$

Considerando a ocorrência de rupturas, há a necessidade de especificar o conceito de demanda e diferenciá-la do conceito de vendas observadas. A demanda por um produto k em um ponto de venda i no período t é dada por D_{ik}^t , e representa toda a venda potencial do produto em (i, k, t) . Em outras palavras, o termo demanda é equivalente às vendas irrestritas, ou seja, as vendas caso o produto demandado esteja sempre disponível nos pontos de venda. Essa definição é necessária pois idealmente os pontos de venda i deveriam ter capacidade de vender toda a demanda pelos produtos k . A capacidade física de estocagem das lojas não é considerada na formulação do problema.

Caso não ocorra ruptura em (i, k, t) , ou seja $rupt_{ik}^t = 0$, é possível afirmar que as vendas observadas em (i, k, t) são iguais a demanda em (i, k, t) .

Caso ocorra a ruptura de um produto k em um ponto de venda i durante um conjunto g de instantes de tempo consecutivos $\{t - g, t - g + 1, \dots, t\}$, isso implica que as vendas observadas entre $(t - g)$ e t serão iguais a 0, dada a indisponibilidade de produtos. No

entanto, caso o produto estivesse disponível entre $(t - g)$ e t possivelmente seriam observadas vendas do produto.

Dessa forma, valem as relações das expressões (3.5) e (3.6).

$$\text{se } rupt_{ik}^t = 0 \rightarrow D_{ik}^t = V_{ik}^t \quad (3.5)$$

$$\text{se } rupt_{ik}^t = 1 \rightarrow V_{ik}^t = 0 \text{ e } D_{ik}^t \geq V_{ik}^t \quad (3.6)$$

Seja R um processo capaz de transformar um conjunto de vendas observadas de um produto em um conjunto de demandas com base também nas informações de disponibilidade de estoques do produto (expressão (3.7)).

$$R(V_{ik}^t, E_{ik}^t) \rightarrow (D_{ik}^t) \quad \forall (i, k, t) \quad (3.7)$$

Considerando essas definições, deseja-se obter um modelo, ou um conjunto de modelos genericamente denominado \mathbb{M} , que seja(m) capaz(es) de prever a demanda pelo produto k , no ponto de venda i para h períodos futuros.

$$\mathbb{M}(V_{ik}^t, E_{ik}^t, P_{ik}^t) \rightarrow (\hat{D}_{ik}^{t+1}, \hat{D}_{ik}^{t+2}, \dots, \hat{D}_{ik}^{t+h}) \quad \forall (i, k, t) \quad (3.8)$$

Além das informações de estoque, vendas, preço e sinais de eventos da companhia \mathcal{U} , existem outras variáveis que podem influenciar D_{ik}^t . Existem fatores endógenos do próprio comportamento de demanda e existem fatores exógenos.

Os fatores endógenos podem ser representados pela existência de correlação entre o comportamento de demanda D_{ik}^t com comportamentos passados observados em períodos anteriores de tempo (*lags*). Define-se L_k como o conjunto de lags relevantes do comportamento de demanda do produto k .

Exemplos de fatores exógenos são (i) ações de marketing expostas nos pontos de venda, ou em meios de divulgação como televisão, internet e rádio e (ii) variáveis climáticas como temperatura e pluviosidade. Essas variáveis recebem o nome de fatores de influência no contexto do problema abordado nessa pesquisa. Podem existir fatores de influência

específicos F_{ik}^t (que influenciam cada (i, k, t)) ou mesmo gerais para um ponto de venda em um instante F_i^t (que influenciam cada (i, t)).

Assim, a expressão (3.8) pode ser revista, incluindo as variáveis de sinais de eventos e fatores de influência, resultando na expressão (3.9).

$$\mathbb{M}(V_{ik}^t, E_{ik}^t, P_{ik}^t, S_r, F_{ik}^t, F_i^t) \rightarrow (\widehat{D}_{ik}^{t+1}, \widehat{D}_{ik}^{t+2}, \dots, \widehat{D}_{ik}^{t+h}) \quad \forall(i, k, t) \quad (3.9)$$

A eficácia do modelo \mathbb{M} deve ser mensurada de acordo com sua precisão ou analogamente de acordo com seu erro de previsão. Duas medidas adequadas para mensuração de erros de previsão são o erro quadrático médio (MSE) e o erro percentual absoluto médio (MAPE). Para o problema em questão, define-se um horizonte de erros médios E_α^{avg} que é equivalente à média de uma medida de erro α ao longo de um conjunto h de períodos futuros de tempo, e um horizonte de desvios de erro E_α^{std} que é equivalente ao desvio padrão de uma medida de erro α ao longo do mesmo conjunto h de períodos de tempo. Assim, E_α^{avg} pode ser representado por um vetor de medidas de erro $e_\alpha^{avg,h}$ em que cada elemento do vetor representa uma média de medida de erro em um período de tempo futuro h . O horizonte de desvios de erro E_α^{std} pode ser representado de forma análoga.

$$E_{RMSE}^{avg} = (e_{MSE}^{avg,1} \quad e_{MSE}^{avg,2} \quad \dots \quad e_{MSE}^{avg,h}) \quad (3.10)$$

$$E_{RMSE}^{std} = (e_{MSE}^{std,1} \quad e_{MSE}^{std,2} \quad \dots \quad e_{MSE}^{std,h}) \quad (3.11)$$

$$E_{MAPE}^{avg} = (e_{MSE}^{avg,1} \quad e_{MSE}^{avg,2} \quad \dots \quad e_{MSE}^{avg,h}) \quad (3.12)$$

$$E_{MAPE}^{std} = (e_{MSE}^{std,1} \quad e_{MSE}^{std,2} \quad \dots \quad e_{MSE}^{std,h}) \quad (3.13)$$

Considerando as definições de (3.9), (3.10), (3.11), (3.12) e (3.13), o problema pode ser caracterizado como encontrar \mathbb{M} que minimiza (3.10) ou (3.12) dependendo da medida escolhida de mensuração de erros.

Esse problema pode ser classificado como um problema de regressão no contexto de aprendizado computacional, no sentido em que se deseja realizar previsões de variáveis

contínuas e a mensuração do erro avalia uma distância entre a previsão e um valor real numérico observado.

3.3. Previsão de múltiplos de vendas

Na caracterização do problema apresentada na seção 3.2, a variável de demanda foi definida como toda a venda potencial de um produto em um ponto de vendas em um período t . Alternativamente, pode-se modificar a variável de demanda para que essa represente uma quantidade discreta de vendas.

Nas cadeias de abastecimento as mercadorias são movimentadas desde seus locais de produção ou fornecimento para os pontos de venda em embalagens fechadas que contém múltiplos dos produtos. Assim, pode-se definir a demanda potencial de (i, k, t) em termos do múltiplo do produto k na cadeia.

Seja Q_k uma quantidade definida do produto k que é movimentada na cadeia. Pode-se definir a variável B_{ikn}^t como uma variável binária igual a 1 se a demanda potencial do produto k no ponto de venda i no período t é maior que n vezes o múltiplo Q_k e menor que $n + 1$ vezes o mesmo múltiplo.

$$B_{ikn}^t = \begin{cases} 1 & \text{se } n \cdot Q_k \leq D_{ik}^t \leq (n + 1) \cdot Q_k \\ 0 & \text{caso contrário} \end{cases} \quad (3.14)$$

Nesse caso, o problema reside em prever a demanda potencial de múltiplos de um produto k em um ponto de vendas i num período t . Considerando a variável B_{ikn}^t , a expressão (3.14) pode ser revista resultando na expressão (3.15).

$$\mathbb{M}(V_{ik}^t, E_{ik}^t, P_{ik}^t, S_r, F_{ik}^t, F_i^t) \rightarrow (\hat{B}_{ikn}^{t+1}, \hat{B}_{ikn}^{t+1}, \dots, \hat{B}_{ikn}^{t+h}) \quad \forall (i, k, t) \quad (3.15)$$

Diferente da definição do problema da seção 3.2, a definição dada nessa seção posiciona o problema como um problema de classificação no contexto de aprendizado computacional no sentido em que se deseja realizar previsões de variáveis discretas que representam a classificação ou a janela de demanda de um produto em um ponto de venda em um determinado período.

3.4. Previsão do comportamento comum de demanda

As definições do problema descritas nas seções 3.2 e 3.3 consideram a modelagem do comportamento de demanda de um produto k em uma loja i exclusivamente como grandeza de interesse, podendo ele ser influenciado por características intrínsecas do comportamento de venda do produto (fatores endógenos) ou mesmo por fatores externos como as vendas de outros produtos, feriados, entre outros (fatores exógenos).

Alternativamente pode-se definir o problema em termos de um comportamento comum de demanda, supondo que o mecanismo de geração de demanda de todo um conjunto de produtos nas lojas é o mesmo. Nesse caso, não só as vendas de um produto específico seriam as variáveis endógenas, mas também as vendas de todos os produtos de um conjunto. A saída do que se deseja prever passa então a ser a venda de um produto genérico k em uma loja genérica i . As amostras históricas de vendas dos produtos de um ponto de venda podem ser colocadas como exemplos ou amostras do mesmo fenômeno.

A formulação apresentada na seção 3.2 deve sofrer algumas alterações considerando essa visão do problema. Primeiro, para que as vendas dos diferentes produtos possam ser tratadas como ocorrências do mesmo fenômeno, os valores observados das vendas V_{ik}^t devem ser normalizados para serem representados dentro de uma mesma ordem de grandeza. Um processo de normalização T_k deve ser construído para cada produto do conjunto para que o histórico de vendas V_{ik}^t possa ser representado de forma normalizada por v_{ik}^t .

Sobre as vendas normalizadas de um produto em um ponto de venda, se aplicam os mesmos comentários e premissas acerca de disponibilidade de estoques e rupturas. As vendas normalizadas podem ser utilizadas para construir histórico de demandas por meio do processo R (expressão (3.7)).

Com base no histórico de demanda normalizado de cada produto em cada ponto de venda devem ser construídas amostras ou exemplos do comportamento de demanda. Uma amostra pode ser construída com D_{ik}^t , seus lags relevantes e fatores exógenos. Cabe

mencionar que a construção de um exemplo deve seguir as exigências de legitimidade dos dados conforme descrito no apêndice A4.

Seja S_w uma amostra de demanda dentro de um conjunto W de todos os exemplos disponíveis de produtos e lojas. A expressão (3.16) exemplifica a construção de todos os exemplos possíveis da amostra.

$$\{D_{ik}^w | w \in L_k\}_{i,k,t} \cup \{P_{ik}^w | w \in L_k\}_{i,k,t} \cup \{S_r, F_{ik}^t, F_i^t\}_{i,k,t,r} \rightarrow S_w \quad (3.16)$$

Assim, a expressão (3.16) pode ser modificada para expressar o problema de modelagem do comportamento conjunto de vendas resultando na expressão (3.17).

$$\mathbb{M}\left(\{V_{ik}^t, E_{ik}^t, P_{ik}^t, S_r, F_{ik}^t, F_i^t\}_{i,k,t,r}\right) \rightarrow \mathbb{M}(S_w) \rightarrow (\hat{D}_{ik}^{t+1}, \hat{D}_{ik}^{t+2}, \dots, \hat{D}_{ik}^{t+h}) \quad \forall(i, k, t) \quad (3.17)$$

Assim como a representação do problema da seção 3.3 em termos de valores binários que representam múltiplos de vendas, o mesmo pode ser aplicado na modelagem do comportamento comum de demanda.

$$\mathbb{M}\left(\{V_{ik}^t, E_{ik}^t, P_{ik}^t, S_r, F_{ik}^t, F_i^t\}_{i,k,t,r}\right) \rightarrow \mathbb{M}(S_w) \rightarrow (\hat{B}_{ikn}^{t+1}, \hat{B}_{ikn}^{t+1}, \dots, \hat{B}_{ikn}^{t+h}) \quad \forall(i, k, t) \quad (3.18)$$

3.5. Comparação entre as visões do problema

As diferentes visões do problema apresentadas nas seções 3.2, 3.3 e 3.4 trazem consigo vantagens e desvantagens.

A caracterização do problema como um simples problema de previsão de demanda possui a vantagem de ser uma modelagem mais objetiva do problema, em que apenas uma grandeza de interesse é analisada. São envolvidas menos transformações de dados em relação as definições alternativas. Além disso a medida de erro pode ser calculada de forma mais direta, comparando-se a previsão realizada pelo modelo com valores observados de venda. Essa abordagem possui duas desvantagens: (i) a variabilidade da grandeza de interesse é maior em comparação a abordagens do problema que consideram múltiplos de movimentação e (ii) a quantidade de dados disponíveis para treinamento dos modelos é

menor em relação abordagens que consideram a venda de múltiplos produtos em múltiplos pontos de venda como parte do mesmo fenômeno.

A segunda caracterização do problema trata a demanda em intervalos discretos, representados por variáveis binárias. Dado que em geral os múltiplos de movimentação de mercadorias são da mesma ordem de grandeza das vendas observadas, essa visão do problema reduz significativamente a variabilidade da grandeza de interesse, potencialmente facilitando a captura de padrões por modelos de aprendizado computacional. No entanto, ao considerar a demanda em intervalos discretos, perde-se a resolução das previsões que podem ser apenas feitas em termos dos múltiplos de movimentação. Dependendo do contexto de aplicação, essa perda de resolução pode não fazer diferença, como por exemplo no planejamento de produção ou para planejamento de reabastecimento de pontos de venda, mas existem casos em que é desejado o valor exato da demanda para planejamento, como por exemplo no planejamento orçamentário e projeção de resultados de pontos de venda.

A terceira caracterização do problema trata as vendas de um grupo de produtos como realizações de um mesmo mecanismo gerador para o qual se deseja extrair o comportamento padrão. A vantagem dessa abordagem é que uma vez que vendas de produtos diferentes são transformadas em exemplos de um mesmo fenômeno, há uma disponibilidade significativamente maior de amostras para treinamento dos modelos de aprendizado computacional, fato que pode ser determinante para aplicação desses modelos.

A quarta abordagem compreende a consideração das vendas de um conjunto de produtos em um conjunto de pontos de venda como realizações de um mesmo fenômeno e trata a demanda em intervalos discretos. Caso essa seja uma premissa válida, tem-se uma definição com maior disponibilidade de dados e redução da variabilidade das previsões. As desvantagens dessa abordagem são que ela requer o maior número de transformações e manipulações dos dados e possui uma medida de erro indireta, assim como a segunda representação do problema.

A Tabela 3.1 resume os pontos fortes e fracos de cada caracterização do problema.

Tabela 3.1 – Comparação entre as diferentes caracterizações do problema

#	Caraterização	Vantagens	Desvantagens
1	Previsão de demanda de um único produto em uma loja	Modelagem direta do problema Medida de erro direta	Alta variabilidade da grandeza de interesse Baixa disponibilidade de dados
2	Previsão de múltiplos de movimentação de um único produto em uma loja	Baixa variabilidade da grandeza de interesse	Baixa disponibilidade de dados Medida de erro indireta
3	Previsão de demanda de um grupo de produtos e lojas	Alta disponibilidade de dados Medida de erro direta	Alta variabilidade da grandeza de interesse
4	Previsão de múltiplos de movimentação de um grupo de produtos e lojas	Alta disponibilidade de dados Baixa variabilidade da grandeza de interesse	Medida indireta de erro

Fonte: Próprio autor

4. METODOLOGIA

Esse capítulo descreve a metodologia proposta. A primeira seção apresenta uma justificativa para a escolha de modelos de aprendizado computacional para abordar o problema de previsão desagregada em cadeias varejistas. A segunda seção apresenta uma visão geral da metodologia e de seus passos intermediários. As subseções seguintes detalham cada passo da metodologia, sendo apresentados exemplos de aplicação.

4.1. Justificativa

Os modelos de inteligência computacional representam um conjunto de técnicas utilizadas para extração e reconhecimento de padrões, a partir de um conjunto de dados. Alguns exemplos de modelos dessa categoria são as redes neurais artificiais, as árvores de classificação e regressão e as máquinas de vetores de suporte (para maiores detalhes dos conceitos básicos vide APÊNDICE A). De acordo com Alpaydin (2010), em geral esse tipo de modelo possui premissas menos restritivas para a sua aplicação em comparação com métodos estatísticos convencionais, como por exemplo, suposições a respeito das distribuições das variáveis e estrutura funcional do modelo, porém para compensar esse fato necessitam de maior quantidade de informações para ajuste dos parâmetros do modelo em comparação com modelos convencionais.

No caso específico do setor varejista, os modelos de aprendizado computacional possuem algumas características que os tornam aderentes as características do problema de previsão desagregada de demanda:

- (i) É necessário pouco ou nenhum conhecimento *a priori* das séries de venda: em métodos estatísticos por exemplo, antes da aplicação de algum modelo de previsão é necessário caracterizar a distribuição das variáveis do modelo ou a estacionariedade da série de vendas e estabilidade da variância da mesma, isso faz com que seja necessária uma análise prévia para utilizar o modelo, tornando complexa a automação da seleção e ajuste de parâmetros. Um exemplo dessa necessidade de análise prévia é a própria aplicação da metodologia de Box e

Jenkins para construção de modelos ARIMA que requer o julgamento do analista para determinação da ordem dos modelos (DE GOOIJER e HYNDMAN, 2006);

- (ii) Pode-se automatizar a escolha da estrutura funcional do modelo: diferente de modelos estatísticos tradicionais, não é necessário assumir uma estrutura funcional *a priori*, ou seja, o próprio algoritmo de aprendizado dos métodos de inteligência computacional mapeia as relações entre as variáveis explicativas e a variável de interesse;
- (iii) Modelos de inteligência computacional são naturalmente auto-adaptativos: a análise de uma série de vendas de um produto, pode estar relacionada a dois padrões de demanda distintos ou mesmo um padrão que está em transição. A utilização de um modelo estatístico pode ser comprometida ao tentar utilizar uma estrutura funcional para modelar dois padrões distintos de demanda (HUANG *et al.*, 2019). No caso de modelos de inteligência computacional isso não é um problema pois esses modelos são capazes de representar mais de uma estrutura funcional e podem ser continuamente ajustados frente a novas observações e alterações no comportamento da demanda;
- (iv) A inserção e descarte de variáveis pode ser automatizada no algoritmo de ajuste do modelo: tanto em modelos de inteligência computacional quanto em modelos estatísticos, dado um conjunto de variáveis disponíveis, busca-se encontrar o conjunto mínimo de variáveis que sejam explicativas em relação à variável de interesse. A introdução de variáveis que não são relacionadas ao fenômeno pode prejudicar a aplicação de métodos estatísticos, aumentando artificialmente os coeficientes de ajuste. No caso de modelos de inteligência computacional, existem técnicas de redução de dimensões que auxiliam na remoção de variáveis que não são relacionadas com o fenômeno gerador de demanda, possibilitando uma maior automação do processo de escolha de variáveis e ajuste do modelo;
- (v) Um mesmo algoritmo de ajuste pode ser utilizado para muitos pontos de venda e produtos: uma vez que modelos de inteligência computacional possuem a

característica de aprender o comportamento de demanda a partir dos dados, pode-se aplicar esses modelos em uma quantidade muito grande de séries de vendas sem necessidade de supervisão humana.

Pelos fatores mencionados, os modelos de inteligência computacional possuem vantagens para aplicação em processos de previsão de demanda em cadeias varejistas. Contudo, existe a desvantagem de que esses modelos necessitam de maiores quantidades de dados para a utilização dos algoritmos de aprendizado e para o ajuste de parâmetros.

Com o avanço da informatização dos processos logísticos, a disponibilidade de dados sobre vendas e estoques se tornou abundante nas cadeias varejistas (BOONE, 2019). Paralelamente, há uma crescente disponibilidade de dados que podem ser utilizados como variáveis exógenas, externas ao processo da cadeia de suprimentos, mas que podem se relacionar com a demanda, como por exemplo bases de dados de grandezas climáticas e indicadores econômicos setoriais.

Em decorrência do aumento da disponibilidade de dados, técnicas de inteligência computacional estão sendo aplicadas para modelagem da demanda de cadeias logísticas em diversos cenários.

A aplicação de modelos de aprendizado computacional depende não somente da quantidade de dados disponíveis, mas também da qualidade dos mesmos (WANG *et al.*, 2016). Assim, também é necessário se preocupar com a limpeza e correção de dados previamente à aplicação dos modelos de aprendizado computacional. Isso justifica a necessidade de técnicas de limpeza e correção de dados propostas nesta pesquisa como forma de garantir a qualidade dos dados previamente à aplicação dos modelos de aprendizado computacional

4.2. Visão geral

Para a solução do problema caracterizado no capítulo 3, de previsão da demanda desagregada de produtos em lojas de cadeias varejistas considerando instantes futuros de tempo, é proposta uma metodologia (ilustrada pela Figura 4.1) que contempla:

1. **Técnicas de limpeza e preparação de dados:** a metodologia inclui técnicas para limpeza e correção das variáveis de entrada de vendas e estoques definidas na caracterização do problema (capítulo 3). A metodologia contempla também algumas transformações necessárias para que essas variáveis possam ser utilizadas nas aplicações de modelos de aprendizado computacional. Além disso, algumas das variáveis exógenas (relativas as variáveis F_{ik}^t da caracterização do problema no capítulo 3) podem estar não estar numa forma numérica (por exemplo a categoria de um produto pode ser uma descrição textual); por isso a metodologia contempla alguns passos para a transformação dessas variáveis;
2. **Construção e teste de modelos de aprendizado computacional:** a metodologia contempla a aplicação de diferentes tipos de modelos de aprendizado computacional para identificação daquele que apresenta menores erros. São considerados quatro tipos de modelos: redes neurais artificiais, árvores de decisão, máquinas de vetores de suporte e máquinas de descida em gradiente. Esses quatro tipos foram escolhidos pois possuem paradigmas distintos de modelagem: as redes neurais artificiais são baseadas numa teoria própria que as caracteriza como aproximadores universais de funções treinados por algoritmos de descida em gradiente (ALPAYDIN, 2010), as árvores de decisão são modelos não paramétricos cujo algoritmo de treinamento visa dividir o espaço das variáveis de entrada maximizando a separação de amostras de classes distintas (HASTIE *et al.*, 2008), as máquinas de vetores de suporte se baseiam na teoria do aprendizado estatístico e são treinadas por um algoritmo de otimização linear (ALPAYDIN, 2010), e por último as máquinas de descida em gradiente são baseadas numa teoria de *ensembles* que consideram conjuntos de modelos que se especializam em partes da amostra de entrada (NATEKIN E KNOLL, 2013). Além disso, um teste com todos os tipos possíveis de modelos de aprendizado computacional foge do escopo dessa pesquisa. Os modelos são testados com parâmetros recomendados pela literatura, sem que haja uma parametrização específica dos mesmos. O objetivo desse passo da metodologia é identificar quais modelos devem ser explorados numa etapa seguinte de parametrização.

3. **Otimização de modelo:** a metodologia contempla uma etapa de escolha de parâmetros ótimos para o tipo de modelo que alcança melhores resultados. É proposta uma metodologia de teste e escolha de parâmetros baseada numa busca de varredura em um conjunto de parâmetros (*grid search*). Um detalhamento do conjunto de parâmetros de cada tipo de modelo considerado nesta etapa pode ser encontrado no APÊNDICE A;

4. **Testes de validação para estimativa de desempenho:** por último, a metodologia contempla uma etapa de teste para comparação do modelo com uma técnica convencional de previsão. Busca-se avaliar os erros em dados não utilizados na construção do modelo para estimar o erro de previsão e comparar o desempenho com um método convencional. O modelo selecionado na segunda etapa é treinado novamente com os dados de treinamento e considerando os parâmetros selecionados na terceira etapa. Em seguida é ajustado um modelo de previsão convencional da classe ARIMA e os erros de ambos os modelos são comparados.

Vale ressaltar que o capítulo 3 menciona quatro diferentes formulações para o problema. As formulações são parecidas, divergindo apenas na forma de modelagem da demanda (se numérica ou categórica) e na consideração conjunta ou não de series de vendas. A metodologia proposta pode ser aplicada para cada uma das quatro definições.

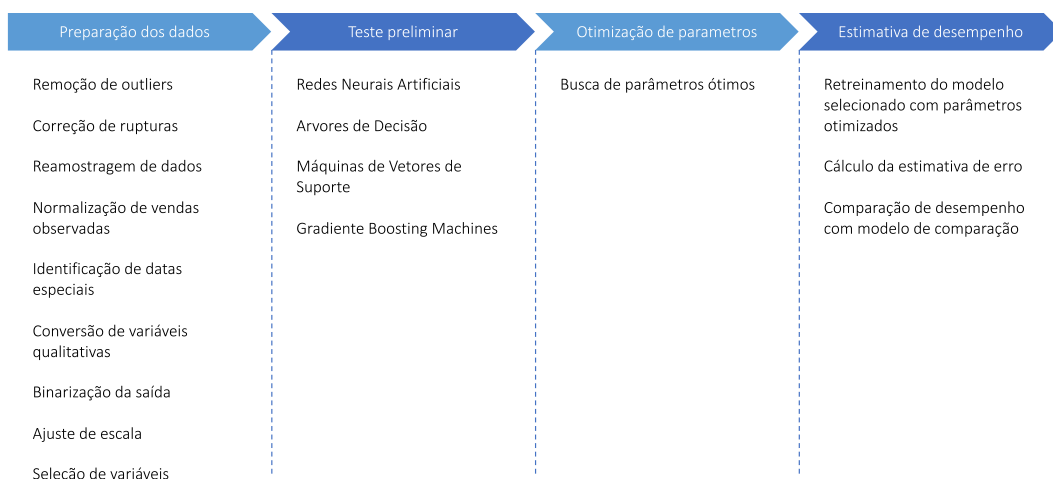


Figura 4.1 – Ilustração da metodologia

Fonte: Próprio autor

4.3. Tratamento e preparação dos dados

Na primeira etapa da metodologia são realizadas as atividades de limpeza e tratamento dos dados de vendas, estoques e variáveis exógenas. Essa etapa de pré-processamento é importante para preparar os dados para utilização nos modelos de previsão.

São considerados os seguintes processos de limpeza, preparação e transformação de dados:

1. **Identificação de datas especiais** (subseção 4.3.1): no setor de bens de consumo existem datas em que o comportamento de demanda é conhecidamente diferente do comportamento normal, como por exemplo em datas comemorativas e épocas de campanhas de *marketing*. Faz parte do pré-processamento a identificação dessas datas para que variáveis que as representam sejam corretamente consideradas, permitindo que os modelos de inteligência computacional sejam capazes de prever o efeito dessas datas. É importante que as vendas nessas datas especiais não sejam consideradas como *outliers* e removidas das amostras de dados, pois nesse caso a séries de vendas corrigidas já não teriam as informações relevantes das datas comemorativas e campanhas;
2. **Correção de dados de vendas por conta de rupturas** (subseção 4.3.2): os dados de uma amostra de vendas representam a demanda de um produto em um ponto de venda apenas se em todo o período observado há estoque disponível para a realização das vendas. Caso contrário, ou seja, dada a ocorrência de rupturas, os dados observados de vendas não representam a demanda potencial integralmente. Pela definição do problema (capítulo 3) é preciso modelar a demanda potencial e não somente as vendas observadas, e por isso se faz necessária a correção dessas observações contaminadas por rupturas por meio de uma aproximação estatística. Além disso, pode ocorrer que a informação de estoque nos sistemas de informação das empresas não corresponde ao estoque físico realmente disponível nos pontos de venda, ou seja, a informação de estoque indica que há disponibilidade, mas na realidade não há. Esse evento recebe o nome de estoque virtual (UÇKUN, KARAESMEN e SAVAS, 2008), que é equivalente a um evento de ruptura e precisa ser identificado e corrigido. A metodologia propõe um método empírico de identificação do estoque virtual. Vale ressaltar que os dados de

estoque são utilizados apenas nesse processo de tratamento, sendo descartados na sequência;

3. **Limpeza de *outliers*** (subseção 4.3.3): séries de vendas podem conter valores muito distantes da média. Essas observações podem ser resultado de eventos não repetitivos ou erros de mensuração (JOHNSON e WICHERN, 2008). Tais observações recebem o nome de pontos extremos, ou *outliers*. Esses pontos, se mantidos na amostra de dados, podem comprometer o treinamento dos modelos de previsão (ALPAYDIN, 2010), acarretando um viés de estimação indesejado. Esses dados precisam ser identificados, removidos ou mesmo corrigidos. Vale ressaltar que vendas elevadas observadas em datas comemorativas, ou campanhas do varejo (por exemplo: Natal, Páscoa, campanhas de mídia de categorias de produtos ou campanhas de *trade-marketing* locais nos pontos de venda) não devem ser removidas nem corrigidas; caso contrário não podem ser capturadas pelos modelos de aprendizado computacional; apenas eventos não planejados e não repetitivos devem ser considerados como *outliers*. Como exemplo de evento não repetitivo que caracterizaria um *outlier*, podemos citar casos em que um cliente específico vai até um ponto de venda e realiza uma compra em massa para alguma finalidade empresarial;
4. **Reamostragem ou agregação dos dados** (subseção 4.3.4): os dados de vendas são em geral definidos em janelas diárias. No entanto, há casos em que no contexto de negócio de uma empresa, basta que a previsão de demanda seja definida em janelas semanais, como é o caso do problema de reabastecimento de pontos de venda. A agregação de dados em janelas maiores de tempo em geral tem um efeito benéfico para modelos de reconhecimento de padrões, pois tal agregação tende a diminuir a aleatoriedade dos dados observados. Esse efeito de redução da aleatoriedade pode ser encarado como uma redução da razão entre o ruído e o sinal presente em uma amostra de dados;
5. **Seleção de variáveis quantitativas e redução de dimensão** (seção 4.3.5): esta é a etapa em que são realizadas análises preliminares da relevância de uma variável quantitativa para explicação do comportamento de demanda de um produto em uma loja. A utilização de variáveis irrelevantes no processo de aprendizado de um modelo de

inteligência computacional é prejudicial, pois aumenta a dimensão do problema pelo aumento da quantidade de parâmetros que precisam ser capturados pelo algoritmo de treinamento do modelo. Deseja-se então utilizar a menor quantidade possível de variáveis explicativas desde que não haja perda da capacidade de explicação. Esse processo envolve a seleção de *lags* relevantes das variáveis de vendas e preços e a redução de dimensão das mesmas. Vale ressaltar que o processo de redução de dimensão já ajusta a escala das variáveis selecionadas, não sendo necessária uma normalização de dados posterior, usualmente requerida por modelos de aprendizado computacional

6. **Seleção e conversão de variáveis qualitativas** (subseções 4.3.6 e 4.3.7): alguns dos fatores externos das observações de venda podem estar definidos em termos de variáveis qualitativas. Exemplos dessas variáveis são: a categoria de um ponto de venda, uma classificação do tipo de venda como bonificada ou não, ou mesmo a categoria de um produto sendo vendido. Para a sua utilização nos algoritmos de treinamento é necessário primeiramente analisar sua importância e em seguida realizar conversões que as tornem aptas a serem processadas pelos modelos corretamente;
7. **Inclusão de sinais de tempo** (seção 4.3.8): considerando que as vendas no varejo seguem ciclos, como por exemplo os ciclos anuais e mensais, essa etapa visa enriquecer a amostra de dados com sinais binários referentes aos diferentes ciclos de vendas do varejo. Além de variáveis binárias que representam os ciclos de tempo (por exemplo: meses do ano) são também inseridas as variáveis binárias que representam as datas especiais identificadas no primeiro passo do tratamento de dados;
8. **“Binarização” da saída** (subseção 4.3.9): no caso das caracterizações do problema em termos de vendas de múltiplos (seção 3.3) é necessário transformar as variáveis de interesse (demanda) em variáveis binárias que indicam quantos múltiplos são vendidos em uma determinada observação;

As subseções 4.3.1 a 4.3.9 a seguir detalham cada um desses processos acima e a Figura 4.2 ilustra a sequência de passos para preparação dos dados. Ao final do processo as variáveis

são recompostas numa amostra única de dados. Todo o processo encontra-se representado na Figura 4.2 .

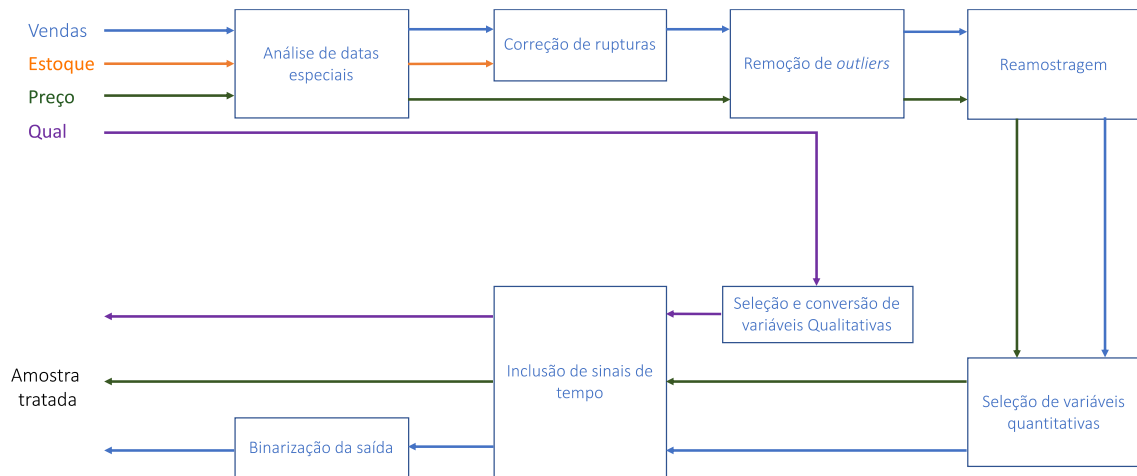


Figura 4.2 – Fluxo de tratamento e preparação das amostras de dados

Fonte: Próprio autor

4.3.1. Identificação de datas especiais

As cadeias varejistas atendem os consumidores finais e é comum a ocorrência de períodos de venda em que há um conhecimento *a priori* do padrão de vendas, como por exemplo durante os feriados de final de ano. Por outro lado, na maioria das vezes esse conhecimento *a priori* não possui validação metodológica e é definido de forma empírica a partir do conhecimento dos gestores e profissionais das respectivas cadeias logísticas. Parte dessas datas especiais possuem abrangência nacional como é o caso dos feriados nacionais, porém existem particularidades regionais (por exemplo, festas folclóricas regionais, feriados municipais, dentre outros).

Em geral o conhecimento dessas datas especiais não está sistematizado nem presente nas amostras de dados e pode ser capturado por meio de entrevistas com os profissionais e gestores de uma cadeia logística. Parte desse conhecimento *a priori* são as datas especiais em si, mas também a antecedência com que elas fazem efeitos nas vendas aos consumidores finais.

Por isso, parte da metodologia proposta para preparação de dados é dedicada a levantar as datas especiais conhecidas e a antecedência com que afetam a demanda por meio de entrevistas com gestores de uma cadeia varejista e validar o efeito dessas datas na demanda através de um processo de cálculo estatístico. Somente dessa forma é possível comprovar o comportamento diferenciado dessas datas especiais para que sejam corretamente incluídas nos modelos de previsão. Para os exemplos apresentados nessa subseção, foram realizadas entrevistas com gestores de um varejo específico para levantamento de quais seriam as datas especiais e com que antecedência elas fazem efeito sobre as vendas. A metodologia de análise de datas especiais visa validar estatisticamente o conhecimento *a priori* dos gestores e profissionais envolvidos na operação da cadeia varejista.

De forma geral, tais datas especiais tem um efeito em múltiplos produtos de uma cadeia varejista; é o caso, por exemplo, do feriado de Natal que impulsiona as vendas do setor varejista como um todo. Dessa forma, para a finalidade específica de validação de datas especiais a metodologia propõe uma análise conjunta das vendas observadas dos produtos de uma cadeia, de forma que as vendas de diferentes produtos são, na verdade observações de um mesmo fenômeno de interesse. Vale ressaltar que o problema de previsão desagregada busca determinar uma previsão de vendas para cada produto em cada loja de uma cadeia varejista, mas nesta etapa da metodologia as séries de vendas são analisadas conjuntamente a fim de apenas determinar se uma data especial conhecida *a priori* faz ou não efeito sobre as vendas da cadeia como um todo.

A Figura 4.3 apresenta uma amostra de vendas observadas não normalizadas de 52 produtos de uma mesma loja de uma cadeia varejista agrupadas em janelas semanais. Cada série é representada por uma cor e corresponde a um produto diferente vendido na loja. Percebe-se que se analisadas numa escala absoluta, as séries são difíceis de serem comparadas devido aos seguintes fatores: as séries de produtos com vendas baixas ficam mascaradas pelos produtos com vendas altas e não é possível identificar picos claros nas datas sazonais; existem produtos com amplitudes de vendas bastante diferentes e o período de início das vendas dos produtos não é o mesmo, o que é comum no segmento de bens de consumo. Por isso, para analisar o efeito das datas sazonais nas vendas o primeiro passo da análise de datas especiais é normalizar a amostra de dados.

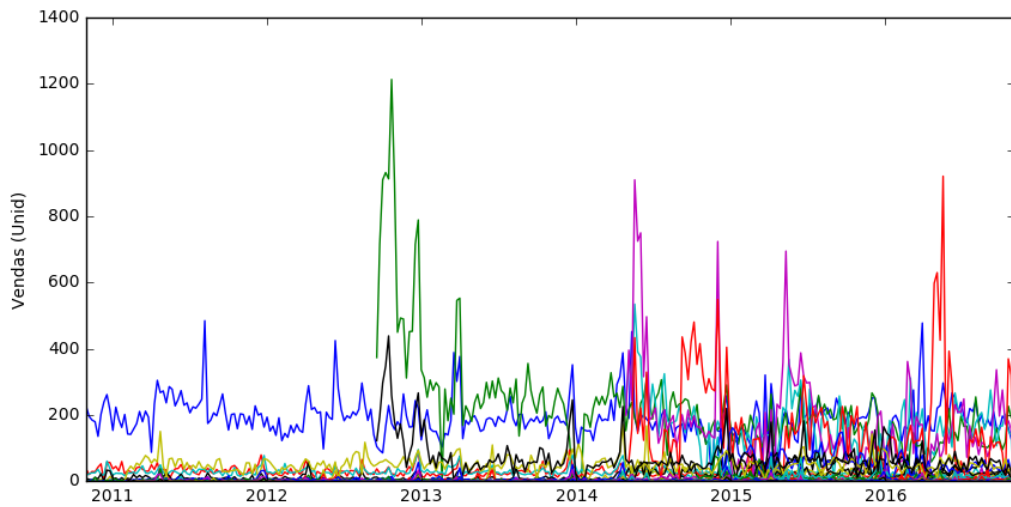


Figura 4.3 – Vendas dos produtos de uma loja

Fonte: Próprio autor

A normalização das vendas de um produto em uma loja é um processo simples no qual a venda observada é subtraída da venda mínima observada e dividida pela amplitude de vendas observadas. Seguindo a notação da seção 3 seja V_{ik}^t a venda observada do produto k na loja i no período t , V_{ik}^{max} a venda máxima observada do produto k na loja i e V_{ik}^{min} a venda mínima observada do produto k na loja i , a venda normalizada do produto k na loja i , v_{ik}^t , é dada pela expressão (4.1).

$$v_{ik}^t = \frac{V_{ik}^t - V_{ik}^{min}}{V_{ik}^{max} - V_{ik}^{min}} \quad (4.1)$$

As datas especiais em geral fazem efeito sobre as vendas num período no entorno de um feriado ou de uma data comemorativa. É o caso por exemplo das vendas de Natal, que impulsionam as vendas do varejo antes da data específica do Natal, ou mesmo o dia das mães que impulsiona as vendas semanas antes do feriado quando as pessoas realizam a compra de presentes e bens de consumo. Por isso, além de normalizar as vendas dos produtos nas lojas essa metodologia propõe como segundo passo agregar as vendas em janelas semanais.

Seja v_{ik}^t o valor da venda normalizada do produto k na loja i , sendo a normalização realizada tendo em vista o histórico de vendas do próprio produto k em i . A venda média entre produtos e lojas é dado por \bar{v}^t . A Figura 4.4 apresenta os valores de \bar{v}^t para os mesmos dados da Figura 4.3.

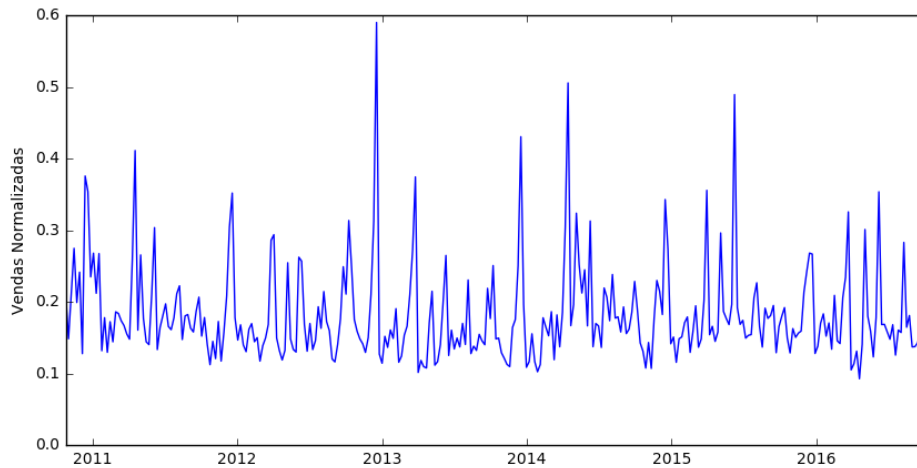


Figura 4.4 – Vendas médias normalizadas de um ponto de venda

Fonte: Próprio autor

Após a normalização e agregação semanal, o passo seguinte da metodologia para identificação de datas especiais é separar as vendas dos produtos em dois conjuntos para cada data especial: um conjunto N de vendas normais e outro conjunto complementar E de vendas em datas especiais. Seja T o conjunto de períodos de vendas observadas normalizadas e seja t_e um período para o qual há um conhecimento *a priori* que indica que se trata de uma data especial. A data especial e por sua vez faz parte do conjunto E de todas as datas especiais conhecidas *a priori*. As vendas normais ou usuais são dadas pelas vendas normalizadas em $T - \{t_e\}_{e \in E}$. O conjunto de períodos com vendas usuais é denominado por N .

Conforme mencionado, a influência de uma data especial não se dá apenas no período t_e podendo ser iniciada alguns períodos anteriores. Seja d a janela de antecedência, tal que a influência da data especial t_e se inicia em t_{e-d} . Assim, como segundo passo de separação as

vendas nesses períodos antecedentes também são incluídas no conjunto E de vendas especiais.

Dessa forma, o conjunto de períodos de vendas normais N é dado pela expressão (4.2) e o conjunto E é o conjunto complementar.

$$N = T - \{t_{e-d}, t_{e-d+1}, \dots, t_e\}_{e \in E} \quad (4.2)$$

Para os dados de vendas da Figura 4.3 um conhecimento *a priori* do comportamento de vendas dos produtos indica que as datas especiais que influenciam as vendas são os feriados de Natal, Páscoa, Dia das Crianças, Dia das Mães, Dia dos Pais, Dia dos Namorados e o dia de promoções relativo a *Black Friday* (época de descontos do setor varejista, tradicional nos EUA e instituído no Brasil em 2010) sendo que a influência desses feriados se inicia com duas semanas de antecedência.

A Figura 4.5 apresenta os dados de vendas médias normais separadas dos dados de vendas médias em datas especiais considerando uma janela de antecedência de dois períodos para os dados da Figura 4.3 (as lacunas entre a serie azul de vendas normais e as séries de datas especiais são geradas por estarem sendo representadas em conjuntos de dados diferentes). Uma inspeção visual indica que para algumas datas, o comportamento de vendas é bastante diferente como é o caso dos feriados de Natal e Páscoa. Já para outros feriados como o Dia dos Pais, tal diferença não é aparente.

Após a separação das vendas nos conjuntos N e E , a metodologia propõe a realização de um teste estatístico da diferença entre duas médias: a média de vendas normalizadas usuais e a média de vendas em uma data especial e . Seja \bar{v}_n o valor médio das vendas normalizadas do conjunto N e seja \bar{v}_e o valor médio das vendas na data especial e pertencentes ao conjunto E . As hipóteses nula e alternativa do teste quanto aos valores de \bar{v}_e e \bar{v}_n são dadas pela expressão (4.3) em que a hipótese nula representa o caso de não existência de diferença significativa no comportamento de vendas na data especial e , e a hipótese alternativa representa o caso em que a diferença é significativa.

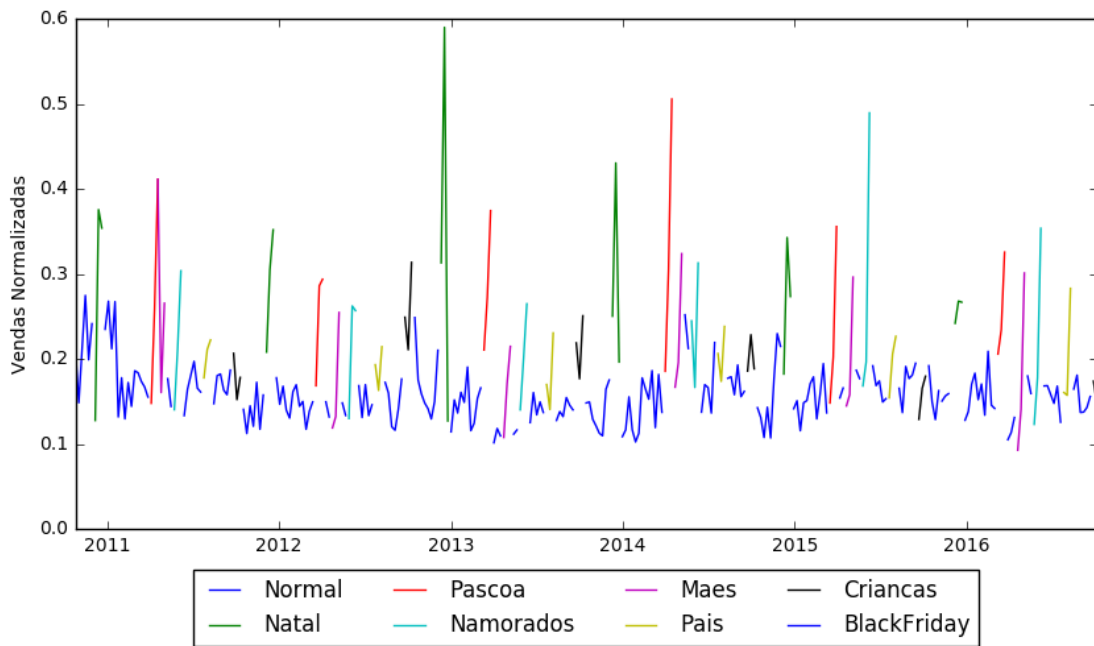


Figura 4.5 – Vendas médias normalizadas de um ponto de movimentação separadas por data especial

Fonte: Próprio autor

$$\begin{aligned}
 H_0: \bar{v}_e - \bar{v}_n &= 0 \\
 H_1: \bar{v}_e - \bar{v}_n &\neq 0
 \end{aligned}
 \tag{4.3}$$

As hipóteses representadas pela expressão (4.3) podem ser testadas por meio de um teste t de amostras não balanceadas e com variâncias diferentes (JOHNSON e WICHERN, 2008). Para a realização do teste, assume-se que as vendas usuais e as vendas na data especial possuem distribuições gaussianas e que as amostras de vendas dos dois grupos de dados foram coletadas independentemente.

A estatística de teste t é dada pela expressão (4.4), em que n_e é o número de vendas normalizadas referente a data especial e , s_e representa o desvio padrão das vendas normalizadas referentes a e , n_n representa o número de vendas normalizadas descontando as vendas em datas especiais e s_n representa o desvio padrão das vendas normalizadas

usuais. A estatística do teste t segue uma distribuição de probabilidade do tipo *t-Student* com $n_e + n_n - 2$ graus de liberdade.

$$t = \frac{\bar{v}_e - \bar{v}_n}{\sqrt{\frac{(n_e - 1)s_e^2 + (n_n - 1)s_n^2}{n_e + n_n - 2} \cdot \left(\frac{1}{n_e} + \frac{1}{n_n}\right)}} \quad (4.4)$$

A aplicação do teste estatístico proposto consiste em:

1. Calcular o valor de t dado pela expressão (4.4);
2. Dado um nível de significância α , calcular os valores críticos do teste e a região crítica dada pela distribuição *t-Student* com os graus de liberdade apropriados;
3. Rejeitar a hipótese nula caso o valor calculado de t esteja fora da região crítica do teste.

Considerando os dados da Figura 4.3 e o feriado de Natal, a Tabela 4.1 apresenta um resumo da aplicação do teste proposto para um nível de significância de 5%. Neste caso, pode-se rejeitar a hipótese nula a um nível de significância de 5% e é possível afirmar que o comportamento das vendas de Natal é diferente do comportamento usual.

Tabela 4.1 – Exemplo de aplicação do teste-t para identificação de datas especiais considerando o Natal

Vendas normalizadas	Média	Variância	Observações
Usual	0.1573	0.1623	5247
Especial (Natal)	0.2798	0.2649	441
t =	14.33		Resultado
α =	5%		
graus de liberdade =	5686		
	Prob (t < $\alpha/2$)	Prob (t > 1- $\alpha/2$)	Rejeita H0
Região crítica	-1.96	1.96	

Fonte: Próprio autor

A Tabela 4.2 apresenta um resumo da aplicação da metodologia proposta para um nível de significância de 5% considerando os mesmos dados da Figura 4.3 e a data especial de *Black Friday*. Neste caso, não é possível rejeitar a hipótese nula a um nível de significância de 5%, o que significa que não é possível afirmar que o comportamento das vendas durante o período de *Black Friday* é diferente do comportamento usual.

Tabela 4.2 – Exemplo de aplicação do teste-t para identificação de datas especiais considerando a *Black Friday*

Vendas normalizadas	Média	Variância	Observações
Usual	0.1573	0.1623	5247
Especial	0.1557	0.1489	507
t =	-0.21		Resultado Não Rejeita H0
alpha =	5%		
graus de liberdade =	5752		
	Prob (t < alpha/2)	Prob (t > 1-alpha/2)	
Região crítica	-1.96	1.96	

Fonte: Próprio autor

A Figura 4.6 apresenta o pseudo-código da metodologia proposta para identificação de datas especiais significativas. Cabe ressaltar que é necessário conhecimento *a priori* para determinação de quais datas devem ser incluídas como argumentos do método e a janela de antecedência de influência da respectiva data especial.

A identificação de datas especiais dentro do contexto do problema abordado na pesquisa permite identificar períodos em que vendas diferentes do comportamento usual são esperadas e portanto, não devem ser consideradas como *outliers* (vide seção 4.3.3). Além disso, com a identificação das datas especiais é possível adicionar variáveis explicativas nos modelos de previsão de demanda para que os padrões de vendas potenciais sejam capturados (vide seção 4.3.8).

```

IdentificacaoDatasEspeciais(VendasObservadas, datas_especiais, janela, alpha)
  VendasNorm ← []
  DatasEspeciais ← []
  Para cada produto k e para cada loja i em VendasObservadas faça
    VendasNorm ← AjusteEscala(VendasObservadas[k,i])
  VendasUsuais ← VendasNorm
  VendasEspeciais ← []
  Para cada data especial te em datas_especiais faça
    VendasEspeciais[te] ← VendasNorm[te],...,VendasNorm[te - janela]
    VendasUsuais ← VendasUsuais - VendasEspeciais[te]
  Calcula media_usual e variância_usual
  Para cada data especial te em datas_especiais faça
    Calcula media_especiale e variância_especiale
    Calcula estatística t //expressão (4.4)
    Aplica o teste de hipótese com significância alpha
    Caso rejeita a hipótese nula faça
      DatasEspeciais ← DatasEspeciais + te
  retorna DatasEspeciais

```

Figura 4.6 – Pseudo-código do método de identificação de datas especiais

Fonte: Próprio autor

4.3.2. Correção de dados de vendas observadas em períodos com rupturas de estoque

A correção de dados de vendas por conta de rupturas é um tipo de tratamento de dados que faz sentido apenas no contexto de cadeias de abastecimento no qual as informações de estoque e vendas estão presentes (UÇKUN *et al.*, 2007). Conforme detalhado no capítulo 3, a informação presente nos históricos de venda não representa necessariamente a demanda potencial de um produto em um ponto de venda. Isso decorre de eventos de ruptura. Por isso, é necessário identificar as rupturas e corrigir os dados de vendas nos períodos em que foram identificadas as rupturas.

Inicialmente, a identificação de rupturas é uma tarefa que decorre da aplicação direta da expressão (3.4). No entanto, de acordo com Uçkun *et al.*, (2007), as informações de estoque podem não ser corretas em relação ao estoque fisicamente disponível nas lojas sendo que essa divergência pode ocorrer pelos seguintes fatores: erros nas anotações de transações de movimentação de mercadorias, erros na alocação dos estoques dentro dos armazéns ou perdas não registradas. Nesse caso diz-se que há ocorrência de estoque virtual e a aplicação da expressão (3.4) não é suficiente para identificação das rupturas. A verificação de estoques

virtuais é um pouco mais complexa que a simples verificação de valores de estoques zerados e pode ser feita apenas de forma aproximada, uma vez que a sua ocorrência poderia apenas ser confirmada pelo confronto da informação de posições de fechamento de estoque com o estoque físico disponível.

Para identificação de estoques virtuais, essa pesquisa propõe o uso adaptado da técnica proposta por Karabati, Tan e Öztürk (2009) para identificação de falta de estoque com base na informação de vendas observadas. Trata-se de um método empírico, mas que apresenta bons resultados em análises de substituição de produtos em lojas físicas.

Para cada período t em que as informações de vendas e estoques estão definidas calculam-se os seguintes indicadores: venda média diária dos últimos D dias e intervalo médio de dias sem venda nos últimos D dias, sendo D um parâmetro do método que representa a quantidade de dias que caracteriza um comportamento normal de demanda. Trata-se de um parâmetro empírico que deve ser ajustado para cada empresa varejista, considerando quanto tempo é necessário para caracterizar uma distribuição de vendas estável para o produto nos pontos de venda. Uma distribuição de probabilidade é considerada estável se a combinação linear de duas variáveis aleatórias que seguem a distribuição resulta numa outra variável aleatória com a mesma distribuição deslocada quanto à média e ao desvio padrão (MORETIN E TOLOI, 2004). No caso em questão, isso equivale a ter uma distribuição de vendas com média e desvio padrão invariantes no tempo.

Com base nesses indicadores a ocorrência de ruptura por estoque virtual num período t é caracterizada pelas seguintes condições:

- (i) Estoque disponível em t diferente de 0;
- (ii) Venda média diária dos últimos D dias maior que 0;
- (iii) Dias desde a última venda até t (intermitência entre vendas) maior que o intervalo médio de dias sem venda nos últimos D dias.

O pseudo-código para identificação de rupturas (reais e virtuais) é apresentado na Figura 4.7 e recebe como entrada o histórico de vendas e posições de estoque e o parâmetro D . O

procedimento é capaz de identificar rupturas pela aplicação direta da expressão (3.4) e pela aplicação do método empírico de identificação de estoques virtuais descrito nessa seção.

```

DeteccaoRupturas(SerieVendasEstoque, D)
    dias_rupturas_estoque_zero ← {}
    dias_rupturas_estoque_virtual ← {}
    registro ← []
    Para cada dia de registro em SerieVendasEstoque faça
        registro[dia]=[estoque, venda]
        Calcula vendas médias nos últimos D dias = s
        Calcula intervalo médio de dias sem venda nos últimos D dias = r
        Calcula quantidade de dias desde a ultima venda = t
        registro[dia]=[estoque, venda, s, r, t]
    Para cada dia de registro em SerieVendasEstoque faça
        Se registro[dia].venda = 0 e registro[dia].estoque=0 faça
            dias_rupturas_estoque_zero ← dia
        Se registro[dia].venda = 0 e
            registro[dia].estoque>0 e
            registro[dia].t> registro[dia].r faça
                dias_rupturas_estoque_virtual ← dia
    Retorna dias_rupturas_estoque_zero + dias_rupturas_estoque_virtual
    
```

Figura 4.7 – Pseudo-código do método de identificação de rupturas

Fonte: Próprio autor

O parâmetro D do método é empírico e deve ser analisado para cada empresa varejista. Uma forma de determinar D é realizar um teste estatístico de comparação entre distribuições para diferentes amostras de vendas de comprimento D . Caso a hipótese nula do teste de que as distribuições são iguais não seja rejeitada, é possível afirmar que o comprimento D caracteriza um comportamento estável de vendas.

A Figura 4.8 apresenta uma série de vendas observadas de um produto em uma loja varejista, assim como a posição de estoques ao longo do tempo. Os valores de vendas estão representados no eixo vertical esquerdo e os valores de estoque no eixo vertical direito. Pode-se observar que próximo aos meses de dezembro de 2014 e junho de 2015 as vendas observadas são nulas, possivelmente devido a indisponibilidade de estoque no período, ou seja, por conta de ruptura de estoque.

Além disso existem alguns períodos em que o estoque se mantém constante e diferente de zero e as vendas são nulas, indicando possíveis períodos de estoques virtuais, como por exemplo no entorno do mês de setembro de 2014.

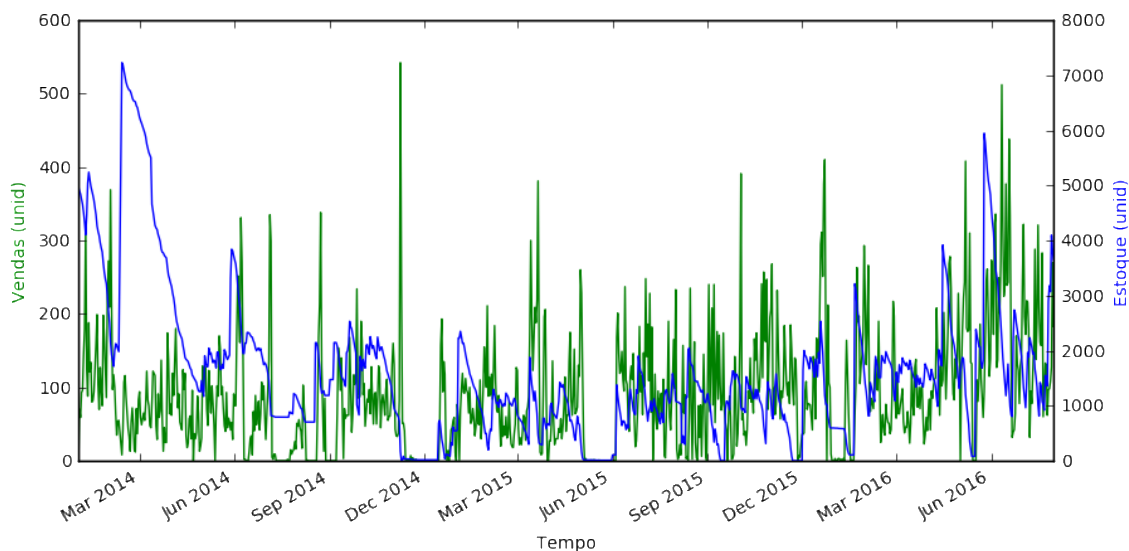


Figura 4.8 – Série de vendas contaminada por rupturas de estoque

Fonte: Próprio autor

A Figura 4.9 contém uma representação ilustrativa da aplicação do procedimento de detecção de rupturas para os dados da Figura 4.8 considerando o parâmetro D igual a 90 dias (representando os últimos 90 dias de vendas observadas de um produto para caracterização da intermitência entre vendas).

Os dias identificados receberam um valor fictício de 100, apenas para que fossem representados na mesma escala que as vendas observadas, por isso a legenda (100 * Ruptura). Os dias identificados com rupturas por estoque zerado e vendas zeradas (de acordo com a expressão (3.4)) estão representados pelas linhas vermelhas, já os dias identificados pelo método empírico de identificação de estoques virtuais estão representados pelas linhas azuis. Pode-se observar que há uma grande ocorrência de estoques virtuais e apenas uma ocorrência de estoque zerado, ou seja, o método proposto nessa pesquisa foi eficaz na identificação de ambos os tipos de rupturas, reais e virtuais.

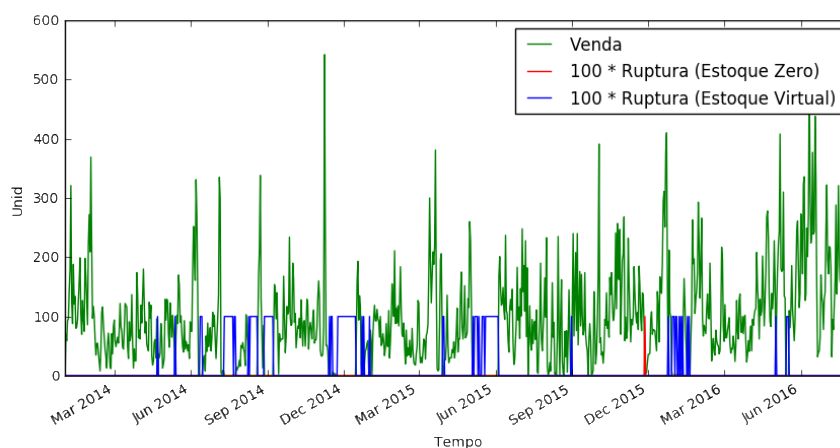


Figura 4.9 – Aplicação do método de detecção de rupturas nos dados da Figura 4.8 para D igual a 90 dias

Fonte: Próprio autor

Os dias com as linhas azuis e vermelhas no gráfico da Figura 4.9 são os dias para os quais foi identificada ruptura. Após a identificação das rupturas, os valores das vendas observadas dos períodos identificados devem ser corrigidos por estimativas da demanda potencial diária.

Existem diversas técnicas de preenchimento e correção de dados para séries temporais, como por exemplo: filtros de médias móveis, filtros de Kalman, interpolação linear e substituição pela média ou mediana (ANDIOJAYA e DEMIRHAN, 2019). A escolha do método de correção depende da quantidade de dimensões da série, do percentual de dados a serem corrigidos e da origem do que causou a necessidade de correção. No caso das séries de vendas, a dimensão da série é igual a um. Quanto ao percentual de dados a serem corrigidos, assume-se que é pequeno em relação ao tamanho da série, abaixo de 5% (conforme evidenciado no exemplo da Figura 4.9). Também é assumido que os dados a serem corrigidos ocorrem aleatoriamente (condição equivalente a *Missing at Random* – MAR). Nesse caso é possível utilizar um filtro de médias móveis que corrige a série analisando apenas a distribuição da vizinhança do ponto a ser corrigido. Caso a quantidade de pontos a serem corrigidos seja maior ou a necessidade de correção não seja aleatória, sugere-se utilizar métodos mais sofisticados de correção como indicado em Andiojaya e Demirhan (2019).

O filtro de médias móveis funciona como uma suavização de um valor num dado período t pela média dos últimos J períodos, sendo que J define o tamanho da janela do filtro. Seguindo a notação da seção 3, o valor calculado pelo filtro de médias móveis em um período t é descrito pela expressão (4.5). O tamanho da janela J deve ser equivalente ao maior período de ruptura da série ou um valor no entorno desse.

$$\hat{D}_{ik}^t = \frac{\sum_{i=0}^J V_{ik}^{t-i-1}}{J} \quad (4.5)$$

A Figura 4.10 contém o resultado da aplicação do filtro de médias móveis para os dados da Figura 4.9 com J igual a 30, equivalente ao dobro da maior duração, em dias, de ruptura de estoque identificada na série. Nos dias identificados com rupturas ou com estoques virtuais, os valores de vendas observadas devem ser substituídos pelos valores de vendas ajustadas (representados pela linha vermelha). Isso consiste numa aproximação de qual seria a demanda potencial caso não houvesse ruptura nos dias identificados.

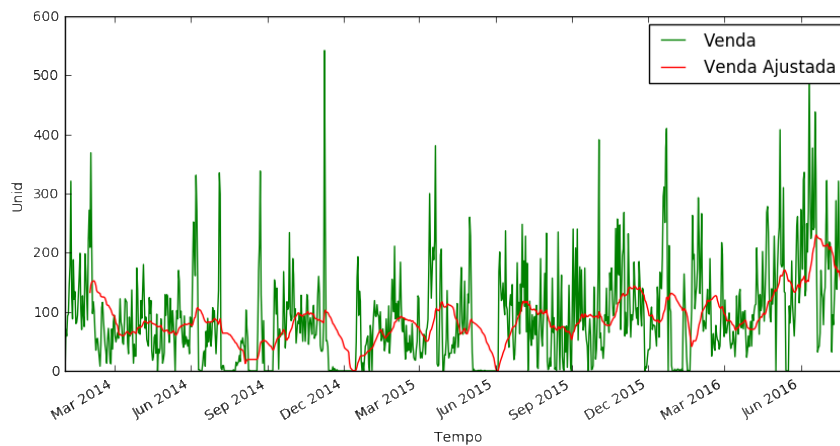


Figura 4.10 – Aplicação do filtro de médias móveis nos dados da Figura 4.9 para J igual a 30 dias

Fonte: Próprio autor

Períodos prolongados de rupturas fazem com que o valor da estimativa da demanda potencial seja próximo de zero. Isso decorre da aplicação do filtro de médias móveis: uma vez que o filtro utiliza os valores passados da observação para calcular a média e valores nulos observados diminuem a média, o valor estimado pelo filtro tende a zero quando as

observações passadas são nulas. É o que pode ser observado próximo aos meses de dezembro de 2014 e junho de 2015. Por conta disso sugere-se a utilização de um valor do parâmetro J maior que o maior período consecutivo de ruptura encontrado em um conjunto de séries de vendas e estoques de um determinado caso em estudo.

4.3.3. Correção de outliers

Detecção e correção de *outliers* é uma atividade que visa a identificação de observações que diferem excessivamente do comportamento esperado (CHEN e LIU, 1993). Em algumas áreas do conhecimento a detecção dessa observação é uma finalidade em si, como é o caso de identificação de fraudes bancárias e diagnósticos médicos por inteligência computacional. Os primeiros estudos sobre esse tipo de técnica datam do século 19 (EDGEWORTH; 1887). No caso dessa pesquisa deseja-se identificar observações *outliers* para que sejam corrigidas nas amostras de treinamento dos modelos de inteligência computacional. Intuitivamente um *outlier* pode ser definido como sendo uma observação que desvia do comportamento do restante da amostra, tal que há suspeita que a observação foi gerada por um mecanismo diferente do fenômeno observado (SINGH e UPADHYAYA, 2012).

A ocorrência de *outliers* pode ser explicada pela própria aleatoriedade dos dados, mas em geral essas observações indicam erros de medição ou que o fenômeno observado possui uma função de distribuição de probabilidade com cauda pesada e com altos coeficientes de assimetria. Nesse segundo caso, é necessário aplicar alguma transformação nos dados ou utilizar métodos de inteligência computacional que não assumem distribuições normais para os dados de entrada.

Existem diversas técnicas para detecção de *outliers*, algumas baseadas em testes estatísticos, outras baseadas em critérios de distância e outras ainda baseadas em critérios de densidade de observações no espaço de variáveis da amostra. Singh e Upadhyaya (2012) apresentam uma análise da literatura acerca do tema e uma visão abrangente das técnicas disponíveis para tratamento de *outliers*. Essa pesquisa propõe o uso uma técnica para detecção de *outliers* baseada em um critério estatístico.

As observações em que o comportamento de vendas é sabidamente diferente do usual são chamadas de *outliers a priori* (vide seção 4.3.1). O critério para determinação de um *outlier a priori* é a repetitividade do evento que causou a observação elevada de vendas, se o evento for repetitivo (por exemplo: Natal e Páscoa) ou planejado (por exemplo: campanhas de marketing e promoções) a observação correspondente é um *outlier a priori*. Antes da aplicação da técnica de detecção de *outliers*, devem ser separados os *outliers a priori*. Em outras palavras, a detecção de *outliers* é aplicada apenas nas observações de vendas fora de épocas comemorativas e campanhas promocionais que são repetitivas ou planejadas. Isso é feito pois é desejável que os *outliers a priori* não sejam identificados nem corrigidos e que sejam mantidos na amostra para que os modelos de aprendizado computacional sejam capazes capturar e prever o comportamento de vendas nesses eventos repetitivos ou planejados.

A técnica para detecção de *outliers* é descrita por Johnson e Wichern (2008) e compreende duas etapas para cada observação:

- (i) Uma análise unidimensional da distância da observação em relação ao centro da distribuição de cada dimensão;
- (ii) Uma análise multidimensional da distância generalizada em relação ao centroide da distribuição das observações da amostra, também chamada de distância de Mahalanobis.

Para a aplicação dessa técnica assume-se que as observações da amostra possuem um processo gerador com distribuição normal multivariada.

Seja \vec{x}_j uma observação de uma amostra de n dados p -dimensionais, x_{jk} o valor da observação j na dimensão k , \bar{x}_k o valor médio da amostra em relação a dimensão k e s_{kk} a variância da amostra em relação a dimensão k . O primeiro passo do procedimento de detecção de *outliers* consiste em calcular as distâncias relativas entre cada dimensão da observação j em relação aos valores médios da amostra (JOHNSON e WICHERN, 2008). Essa distância é dada pela expressão (4.6).

$$z_{jk} = \frac{(x_{jk} - \bar{x}_k)}{\sqrt{S_{kk}}} \quad (4.6)$$

Caso os valores de z_{jk} para uma observação sejam muito altos, a observação j pode ser considerada como um *outlier* e pode ser marcada como uma observação *outlier* unidimensional. Johnson e Wichern (2008) recomendam que um valor limite de 3,5 seja utilizado como referência para identificação de *outliers* em relação a uma dimensão k .

Em seguida, devem ser calculados os valores das distâncias de Mahalanobis de cada observação que são as distâncias entre cada observação e o centroide de uma distribuição amostral multidimensional. Essa distância é dada pela expressão (4.7) em que \bar{x} representa o contróide da amostra e S representa a matriz de covariância da amostra.

$$d_j = (\vec{x}_j - \bar{x})^T S^{-1} (\vec{x}_j - \bar{x}) \quad (4.7)$$

Caso o valor de d_j seja alto, a observação j pode ser marcada como uma observação *outlier* multidimensional. Sabe-se que d_j segue uma distribuição χ_p^2 (chi-quadrado com p graus de liberdade). Assim, a determinação do valor limite de d_j depende da definição de um percentual adequado da distribuição e depende da dimensão da amostra. Johnson e Wichern (2008) recomendam um valor de 5% como percentual adequado para determinação do valor limite de d_j . A quantidade de graus de liberdade depende da quantidade de observações da amostra.

Cabe ressaltar que esse método de detecção de *outliers* pode apenas ser aplicado para analisar as dimensões de variáveis reais de uma amostra de dados (JOHNSON e WICHER, 2008). A Figura 4.11 apresenta o pseudo-código dessa metodologia proposta de detecção de *outliers*.

Uma vez identificados, essa metodologia propõe que os valores de venda de observações *outlier* sejam substituídos por valores suavizados por um filtro de médias móveis assim como no caso da seção 4.3.2, em que observações de vendas com rupturas de estoque são substituídas pelo mesmo método.

DeteccaoOutliersEstatístico(X , $\Theta_{unidimensional}$, $\Theta_{multidimensional}$)

```

Outliers = {} //Conjunto de outliers
 $X_r \leftarrow \{x_j \mid j \text{ não é outlier a priori}\}$ 
Para cada dimensão  $k$  de  $X_r$  faça
    Calcula  $x_k$  e  $s_{kk}$ 
    Para cada observação  $j$  de  $X_r$  faça
        Calcula  $z_{jk}$ 
        Se  $z_{jk} > \Theta_{unidimensional}$  faça
            Outliers  $\leftarrow j$  //outlier unidimensional
    Para cada observação  $j$  de  $X_r$  faça
        Calcula  $d_j$ 
        Se  $d_j > \Theta_{multidimensional}$  faça
            Outliers  $\leftarrow j$  //outlier multidimensional
 $X = \{X\} - \text{Outliers}$ 
retorna  $X$ 
    
```

Figura 4.11 – Pseudo-código da metodologia de detecção de *outliers* por critério de distância estatística

Fonte: Johnson e Wichern (2008)

4.3.4. Reamostragem dos dados ou agregação temporal dos dados

A reamostragem dos dados, também chamada de agregação temporal, consiste no agrupamento de valores de vendas observadas em janelas de tempo diferentes da frequência original da amostra de dados. Caso a agregação seja feita de uma frequência alta para uma frequência menor (por exemplo a agregação de dados diários de vendas em dados mensais), essa operação tem um efeito de suavização dos dados da série e redução da variância (VEREDAS e SILVESTRINI, 2008). A agregação pode ser feita para: reduzir a variância da série, ajustar os dados para o propósito do modelo resultante ou anular problemas de dados faltantes. A janela de agregação para reamostragem dos dados depende do contexto de negócio em que a previsão de demanda será utilizada.

No caso das cadeias varejistas, as informações sobre estoques e vendas de cadeias varejistas estão em geral disponíveis em janelas diárias (BOONE *et al.*, 2019). Contudo, dados diários podem ter a variância muito alta, especialmente para produtos com vendas baixas e baixo giro de estoque.

De acordo com Kourentzes, Rostami-Tabar e Barrow (2017), a janela de agregação dos dados deve se adequar ao propósito da tomada de decisão do modelo que será treinado com a

amostra. Para decisões estratégicas de longo prazo como investimentos em capacidade na cadeia de suprimentos, janelas mensais são adequadas, já para decisões de estoques, compras e abastecimento, janelas semanais são mais adequadas. Os autores afirmam que a agregação em janelas temporais maiores, tem o efeito de suavização da série, redução de ruído e variância e simplifica a geração de modelos de previsão de demanda.

No caso de dados de séries de vendas, a reamostragem de dados é um processo trivial do ponto de vista computacional, na medida em que consiste apenas na agregação através da soma das vendas diárias observadas em uma janela que representa o período de reamostragem e uma agregação através da média para os preços praticados dentro da janela de reamostragem.

Um exemplo de aplicação da reamostragem pode ser encontrado na Figura 4.12 que apresenta um exemplo considerando dados de vendas observadas de um produto de alto valor agregado em uma loja varejista. A parte superior contém as vendas diárias do produto e a parte inferior contém uma comparação entre as vendas diárias e as vendas agregadas em janelas semanais e mensais. A venda diária observada do produto contém um padrão bastante intermitente com vendas variando entre zero e 10. Visualmente pode-se perceber uma redução da volatilidade da série de vendas observadas com a agregação semanal e uma redução ainda maior com a agregação mensal.

A Tabela 4.3 apresenta uma análise do coeficiente de variação para os dados de vendas observadas da Figura 4.12. Pode ser observado que, conforme há um aumento da janela de reamostragem, da janela diária para a semanal e da semanal para a mensal, há também uma redução do coeficiente de variação da série de vendas observadas, o que por sua vez caracteriza uma redução na volatilidade.

O efeito de redução da volatilidade pode ser explicado considerando a venda observada em um dia t como uma variável aleatória com média μ e variância σ . Apenas para fins ilustrativos, pode-se considerar que as vendas em um dia t não se relacionam com as vendas dos dias anteriores e assim a venda agregada em uma janela de reamostragem J pode ser encarada como uma soma de variáveis aleatórias independentes e identicamente distribuídas. Por consequência, a soma das vendas observadas na janela cresce na

proporção $J\mu$, enquanto o desvio padrão cresce na proporção $\sqrt{J}\sigma$. Sendo o coeficiente de variação a razão entre a média e o desvio, observa-se um decréscimo nesse coeficiente proporcional ao aumento da janela de reamostragem.

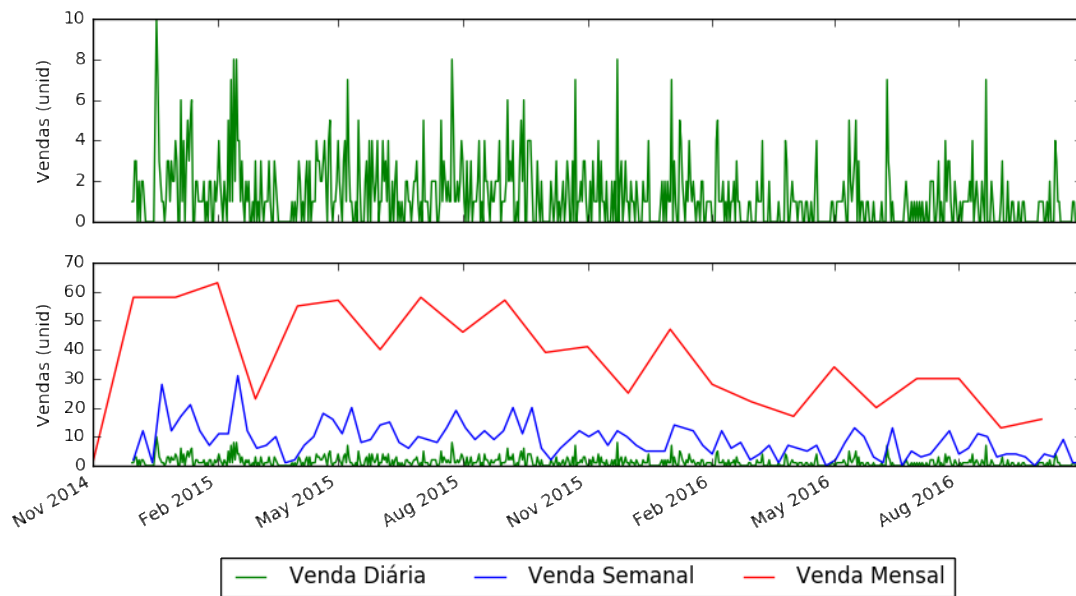


Figura 4.12 – Comparação do efeito de reamostragem de dados

Fonte: Próprio autor

Tabela 4.3 – Comparação do coeficiente de variação para diferentes janelas de reamostragem

	média	desvio padrão	coeficiente de variação
diária	1.26	1.57	1.24
semanal	8.69	5.79	0.67
mensal	36.58	17.55	0.48

Fonte: Próprio autor

Uma vez que variâncias elevadas na série dificultam o aprendizado de modelos de inteligência computacional (ALPAYDIN, 2010), considerando também que os modelos de previsão serão utilizados para tomadas de decisão de compras e estoques nas lojas e por

último, assumindo que uma cadeia varejista em que a metodologia esteja sendo aplicada opere com ciclos de reabastecimento menores que um mês, uma agregação semanal é adequada para a metodologia proposta.

4.3.5. Seleção de variáveis quantitativas e redução de dimensão

A seleção de variáveis é um conjunto de processos que visa identificar as melhores variáveis para um problema de aprendizado computacional. De acordo com Guyon e Elisseeff (2003) a seleção de variáveis tem como objetivos: (i) melhorar o desempenho de modelos de previsão pelo uso das variáveis com capacidade de explicação de um fenômeno, (ii) eliminar variáveis irrelevantes de uma amostra e com isso otimizar o treinamento dos modelos de previsão e (iii) dar melhor entendimento do processo gerador do fenômeno em análise.

No contexto dessa pesquisa a seleção de variáveis abrange três passos:

- (i) Seleção de *lags* relevantes de vendas observadas passadas;
- (ii) Análise de *lags* relevantes de outras variáveis;
- (iii) Redução de dimensão;

A seleção de *lags* relevantes de vendas observadas passadas consiste em determinar quais valores passados da demanda devem ser utilizados para modelar a demanda potencial futura dentro do horizonte de previsão do problema.

Para isso é possível a utilizar da função de autocorrelação (MORETTIN e TOLOI, 2004). A autocorrelação de uma variável aleatória X com relação a um *lag* específico l (ρ_l), considerando uma amostra de n observações, pode ser estimada de acordo com a expressão (4.8) em que σ_X^2 é a variância real da variável aleatória e S_X^2 é um estimador da variância. O intervalo de confiança para a estimativa da autocorrelação é dado por z_α/\sqrt{n} em que z_α representa a função inversa de uma distribuição normal de probabilidade com significância α .

$$\rho_l(X) = \frac{cov(X_t, X_{t+l})}{\sigma_X^2} \approx \frac{1}{(n-l)s_X^2} \sum_{t=1}^{n-l} (x_t - \bar{x})(x_{t+l} - \bar{x}) \quad (4.8)$$

Com base na estimativa da autocorrelação das vendas observadas para diferentes *lags* é possível determinar os *lags* que possuem correlação diferente de zero com base no intervalo de confiança dado por z_α/\sqrt{n} .

A Figura 4.13 ilustra a função de autocorrelação até o *lag* 50 para as vendas observadas de um produto em uma loja varejista, agrupadas em janelas semanais. Também são apresentados os intervalos de confiança construídos com 95% de significância para as diferentes estimativas das autocorrelações de diferentes *lags* (parte sombreada da Figura 4.13).

Os *lags* que possuem autocorrelações fora dos intervalos de confiança são aqueles para os quais é possível afirmar, com pelo menos 95% de confiança, que a autocorrelação é diferente de zero. Para o exemplo ilustrado para a Figura 4.13, é possível afirmar que os *lags* 1, 2 e 21 possuem autocorrelação diferente de zero com 95% de confiança (além do *lag* 0 que é a própria observação sem defasagem). Assim, na construção dos modelos de previsão os valores passados de vendas observadas que devem ser considerados são relativos aos *lags* 1, 2 e 21.

A Figura 4.14 ilustra o pseudo-código do método proposto para determinação de *lags* relevantes da variável de interesse. Na Figura 4.14, y é um vetor de vendas observadas, *max_lags* é um valor máximo a partir do qual os *lags* são automaticamente julgados não relevantes e α representa o nível de significância utilizado para determinar os intervalos de confiança das autocorrelações de diferentes *lags*. Conforme recomendado por Morettin e Tolo (2004), o valor utilizado nessa pesquisa para a constante α é de 95%. Não foram encontradas recomendações sobre a constante *max_lags*; por isso para essa pesquisa foi tomado como premissa que após 104 semanas (o que corresponde a aproximadamente dois anos), os *lags* não são mais analisados pelo método.

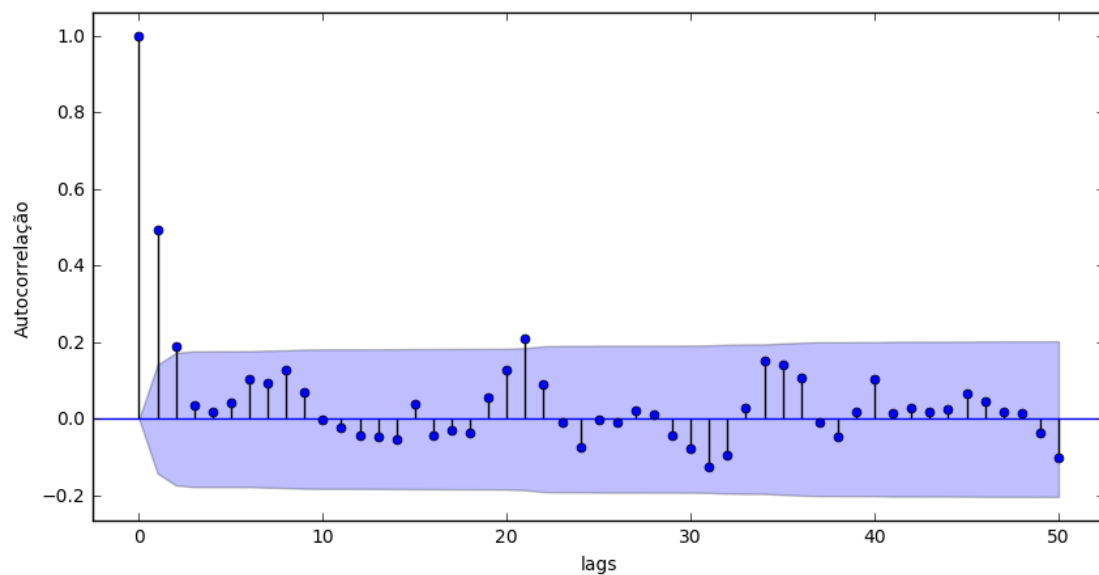


Figura 4.13 – Exemplo de determinação de *lags* relevantes pela função de autocorrelação

Fonte: Próprio autor

```

SelecaoLagsTarget(y, max_lags, alpha)
  lags_selecionados = {}
  autocorrelacao_target = autocorrelacao(y)
  intervalos_confianca_target = intervalos_confianca(y, alpha)
  lag ← 1
  enquanto lag < max_lag faça
    Se |autocorrelacao_target[lag] | > intervalo_confianca[lag] faça
      lags_selecionados = lags_selecionados + {lag}
      lag ← lag + 1
  retorna lags_selecionados
    
```

Figura 4.14 – Pseudo-código do método de seleção de *lags* relevantes da variável de interesse

Fonte: Próprio autor

O segundo passo da seleção de variáveis quantitativas é a identificação de *lags* relevantes de outras variáveis em relação à variável de interesse, como por exemplo: preços praticados, valores investidos em propaganda e tamanho dos estoques de exposição (planogramas). Um procedimento similar pode ser utilizado para analisar a correlação de outras variáveis

quantitativas além com as vendas observadas (o procedimento não pode ser aplicado em variáveis qualitativas). Ao invés da autocorrelação, nesse caso é a correlação da variável de interesse com os *lags* de outra variável que deve ser analisada, sendo essa correlação equivalente ao coeficiente de Pearson (JOHNSON e WICHERN, 2008). Uma vez calculado o coeficiente de correlação entre uma variável de interesse e o *lag* de outra variável presente na amostra, é possível testar a hipótese de não correlação considerando os *p-valores* das estimativas.

A Figura 4.15 ilustra um exemplo que considera os dados de vendas observadas e os dados de preços praticados de um produto em um supermercado (os dados foram agregados semanalmente). O gráfico superior da Figura 4.15 apresenta os valores do coeficiente de Pearson entre a variável de vendas observadas numa determinada semana e o preço médio praticado na semana correspondente ao *lag* do eixo horizontal. O gráfico inferior apresenta os *p-valores* das estimativas para teste de não correlação (linha verde) e um limite equivalente a um *p-valor* de 5% (linha vermelha). Os dados indicam que há correlação significativa entre as vendas observadas e os preços praticados nas semanas correspondentes aos *lags* 0, 11, 18 e 24 (*lags* em que o *p-valor* correspondente fica abaixo do limite de 5%) o que significa que os preços praticados com essas defasagens de tempo são variáveis explicativas da demanda que se deseja prever.

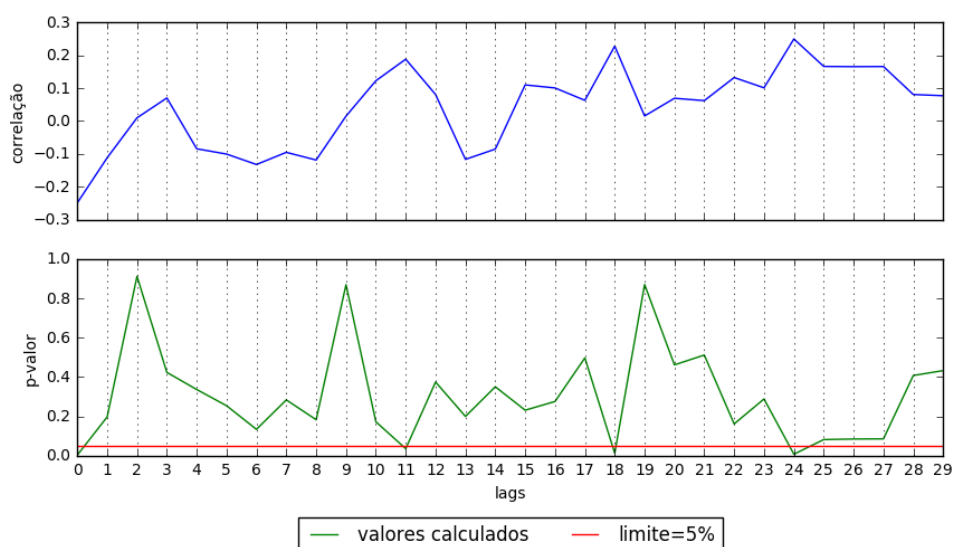


Figura 4.15 – Exemplo de determinação de lags relevantes pelo coeficiente de pearson

Fonte: Próprio autor

O exemplo da Figura 4.14 indica que as vendas observadas nos *lags* 1, 2 e 21 são significativas para um modelo de previsão de demanda, e o exemplo da Figura 4.15 indica que os preços praticados nos *lags* 0, 11, 18 e 24 são significativos para o mesmo modelo de previsão.

A Figura 4.16 contém o pseudo-código da metodologia de identificação de *lags* relevantes de variáveis de entrada em relação a uma variável de interesse em que y é um vetor de observações das vendas, X é uma matriz de variáveis de entrada, max_lags é o valor máximo de *lags* analisadas para cada dimensão da matriz de entrada, e p_valor_limite é o valor mínimo para que um *lag* de uma determinada variável seja julgado relevante para explicar o comportamento da variável de interesse.

```

SelecaoLagsEntrada( $y$ ,  $X$ ,  $max\_lags$ ,  $p\_valor\_limite$ )
   $lags\_selecionados = \{\}$ 
  Para cada dimensão  $d$  de  $X$  faça
    Caso  $X[d]$  represente uma variável quantitativa faça
       $lag \leftarrow 1$ 
      enquanto  $lag < max\_lag$  faça
         $correlação\_target\_d\_lag = correlação(y, X[d], lag)$ 
         $p\_valor = p\_valor\_correlacao(correlação\_target\_d\_lag, y, X[d], lag)$ 
        Se  $p\_valor < p\_valor\_limite$  faça
           $lags\_selecionados[d] \leftarrow lags\_selecionados[d] + \{lag\}$ 
         $lag \leftarrow lag + 1$ 
      Caso contrário faça
         $lags\_selecionados[d] \leftarrow \{\}$ 
  retorna  $lags\_selecionados$ 

```

Figura 4.16 – Pseudo-código do método de seleção de *lags* relevantes de variáveis de entrada

Fonte: Próprio autor

A seleção de *lags* das vendas e *lags* das demais variáveis quantitativas pode resultar na seleção de muitas variáveis explicativas para um modelo de previsão. Segundo Alpaydın (2010), para casos em que a quantidade de dimensões da matriz de variáveis é da mesma ordem de grandeza ou maior que a quantidade de observações, é recomendável aplicar um procedimento de redução de dimensão das variáveis explicativas. Quanto menor for a

dimensão do espaço de entrada, menor a quantidade de parâmetros que devem ser encontrados pelos algoritmos de aprendizado dos modelos de inteligência computacional e, conseqüentemente, melhor tende a ser o desempenho dos modelos em termos de tempo de treinamento e até mesmo em termos de capacidade de generalização (VAPNIK, 1998).

A metodologia proposta nessa pesquisa sugere o uso da Análise de Componentes Principais (*Principal Components Analysis* – PCA) para a redução da dimensão do espaço de entradas. De acordo com Johnson e Wichern (2008) o método PCA consiste em analisar e simplificar a estrutura de variância e covariância de um sistema com p dimensões.

Seja X um conjunto de dados p -dimensional e seja Σ a respectiva matrix de covariância com pares de autovetores e autovalores representados por $(\vec{e}_1, \lambda_1), \dots, (\vec{e}_p, \lambda_p)$. Pode-se demonstrar que a i -ésima componente principal de X é dada pela expressão (4.9). Isso significa que uma amostra de dados X pode ser convertida numa amostra baseada em suas componentes principais pela determinação de seus autovetores e autovalores.

$$Y_i = \vec{e}_i^T X \quad (4.9)$$

Cabe mencionar que a tradução de X em suas componentes principais produz uma matriz de dados com exatamente p dimensões. A redução de dimensão ocorre com a eliminação das componentes principais menos significativas.

Pode-se demonstrar que a variância total da amostra é equivalente a soma dos autovalores da matriz de covariância da amostra (JOHNSON e WICHERN, 2008). Assim, a fração da variância explicada pela i -ésima componente principal F_i é dada pela expressão (4.10).

$$F_i = \frac{\lambda_i}{\sum_{j=0}^p \lambda_j} \quad (4.10)$$

Dado um valor limite de explicação de variância é possível selecionar um número reduzido de componentes principais para redução da dimensão da amostra. Essa metodologia propõe um limite de 80% de explicação para retenção dos componentes principais, ou seja, apenas os primeiros componentes principais que representam 80% da variância são mantidos na

amostra de dados, os demais são eliminados. Um detalhamento maior do método PCA pode ser encontrado em Johnson e Wichern (2008).

A Figura 4.17 contém o pseudo-código do método de redução de dimensões proposto na metodologia em que X é uma matriz de variáveis de entradas reais e F é a fração de explicação da variância que deve ser mantida após a redução de dimensões. Cabe ressaltar que a redução de dimensão é realizada apenas para a matriz de variáveis de entrada selecionada pelas metodologias de determinação de *lags* relevantes (Figura 4.14 e Figura 4.16).

```
ReducaoDimensaoEntrada( $X, F, lags\_relevantes$ )  
   $X\_transform, autovalores\_X = PCA(X)$   
   $X\_transform, autovalores\_X = ordenar\_por\_autovalor(X\_transform, autovalores\_X)$   
   $explicação\_acumulada = 0$   
   $d = 0$   
   $dimensões\_selecionadas = \{\}$   
  enquanto  $explicação\_acumulada < F$  faça  
     $fracao\_explicacao\_d = autovalores\_X[d] / soma(autovalores\_X)$   
     $explicação\_acumulada \leftarrow explicação\_acumulada + fracao\_explicacao\_d$   
     $dimensões\_selecionadas \leftarrow dimensões\_selecionadas + \{d\}$   
  retorna  $X\_transform[dimensões\_selecionadas]$ 
```

Figura 4.17 – Pseudo-código do método de redução de dimensão de variáveis de entrada

Fonte: Próprio autor

Os métodos de seleção de variáveis por autocorrelação e por correlação podem apenas ser aplicados em variáveis numéricas.

4.3.6. Seleção de variáveis qualitativas

Não foram encontradas recomendações na literatura sobre como identificar as variáveis qualitativas para explicar comportamentos de vendas. Por isso, essa pesquisa propõe modelar esse problema como um problema de identificação de tratamentos. Assumindo que os estados, ou valores, que uma variável qualitativa pode assumir são como tratamentos a que uma amostra de vendas está submetida, podem ser utilizados procedimentos estatísticos para validar se as propriedades da amostra de vendas são estatisticamente diferentes sob a influência de múltiplos tratamentos.

Para a avaliação da significância de variáveis qualitativas, a metodologia proposta nessa pesquisa sugere o uso de uma análise de variância multivariada (*multivariate analysis of variance* – MANOVA) com múltiplas classes não hierárquicas (*non hierarchical multiple-way MANOVA*) (JOHNSON e WICHERN, 2008). Esta subseção apresenta a análise de variância multivariada com dois fatores (*two-way MANOVA*), sendo que a definição da análise com múltiplos fatores pode ser estendida a partir dos conceitos apresentados. A definição exposta omite a notação vetorial para manter a concisão das expressões.

A análise de variância parte da premissa que um fenômeno de interesse é analisado sob diferentes condições experimentais. Uma condição experimental pode ser caracterizada por uma condição de fatores que assumem diferentes níveis. No caso da *two-way MANOVA* assume-se que existem dois fatores experimentais que atuam de forma conjunta no fenômeno de interesse, nominalmente fator um e fator dois. Assume-se que o fator um pode assumir g diferentes níveis e o fator dois pode assumir b diferentes níveis.

O modelo de probabilidade em que a *two-way MANOVA* se baseia assume que uma observação aleatória p -dimensional de índice r , observada nos níveis l e k dos fatores um e dois respectivamente, é definida pela expressão (4.11) em que μ é o valor médio global esperado do fenômeno, τ_l é o efeito do fator um, β_k é o efeito do fator dois, γ_{lk} é a interação cruzada dos fatores um e dois e e_{lkr} é um fator de erro com distribuição $N_p(\mathbf{0}, \Sigma)$ (distribuição normal p -dimensional com média $\mathbf{0}$ e variância Σ). Todos os componentes da expressão (4.11) são vetores p -dimensionais.

$$X_{lkr} = \mu + \tau_l + \beta_k + \gamma_{lk} + e_{lkr} \quad (4.11)$$

Uma observação do fenômeno pode ser escrita conforme a expressão (4.12) em que todas as parcelas são também p -dimensionais.

$$x_{lkr} = \bar{x} + (\bar{x}_l - \bar{x}) + (\bar{x}_k - \bar{x}) + (\bar{x}_{lk} - \bar{x}_l - \bar{x}_k + \bar{x}) + (x_{lkr} - \bar{x}_{lk}) \quad (4.12)$$

Com base na representação (4.12), é possível escrever as somas de produtos quadráticos (*sums of square products* – SSP) conforme a expressão (4.13). O lado esquerdo da expressão (4.13) representa a soma de quadrados total, ou corrigida (SSP_{corr}), a primeira parcela do

lado direito representa a soma de quadrados relativa ao fator um (SSP_{fac1}), a segunda parcela representa a soma de quadrados do fator dois (SSP_{fac2}), a terceira parcela representa a soma de quadrados da interação entre os fatores (SSP_{int}) e a última parcela representa a soma de quadrados dos resíduos (SSP_{res}). Vale ressaltar que as somas de produtos quadráticos são matrizes de dimensão $(p \times p)$.

$$\begin{aligned}
 & \sum_{l=1}^g \sum_{k=1}^b \sum_{r=1}^n (x_{lkr} - \bar{x}) (x_{lkr} - \bar{x})' \\
 &= \sum_{l=1}^g bn(\bar{x}_l - \bar{x}) (\bar{x}_l - \bar{x})' + \sum_{k=1}^b gn(\bar{x}_k - \bar{x}) (\bar{x}_k - \bar{x})' \\
 &+ \sum_{l=1}^g \sum_{k=1}^b n(\bar{x}_{lk} - \bar{x}_l - \bar{x}_k + \bar{x}) (\bar{x}_{lk} - \bar{x}_l - \bar{x}_k + \bar{x})' \\
 &+ \sum_{l=1}^g \sum_{k=1}^b \sum_{r=1}^n (x_{lkr} - \bar{x}_{lk}) (x_{lkr} - \bar{x}_{lk})'
 \end{aligned} \tag{4.13}$$

Com base nos valores das somas de produtos quadráticos é possível determinar se os efeitos individuais e cruzados dos fatores são significativos. A expressão (4.14) apresenta o teste de hipótese para verificar se há interação significativa entre os dois fatores. A hipótese nula indica que todas as interações são nulas, e a hipótese alternativa indica que pelo menos uma interação é diferente de zero.

$$\begin{aligned}
 H_0: \gamma_{11} = \gamma_{12} = \dots = \gamma_{gb} = 0 \\
 H_1: \exists \gamma_{ij} \neq 0
 \end{aligned} \tag{4.14}$$

Para realizar o teste de hipótese da expressão (4.14) é necessário calcular a estatística Λ^* chamada de Lambda de Wilks (JOHNSON e WICHER, 2008).

$$\Lambda^* = \frac{|SSP_{res}|}{|SSP_{int} + SSP_{res}|} \tag{4.15}$$

Johnson e Wichern (2008) demonstram que para amostras grandes a estatística segue uma distribuição $\chi^2_{(g-1)(b-1)p}$. Dessa forma é possível testar a hipótese nula de (4.14).

Rejeita-se H_0 com um nível de significância α caso seja verificada a condição da expressão (4.16).

$$-\left[gb(n-1) - \frac{p+1-(g-1)(b-1)}{2}\right] \ln \Lambda^* > \chi_{(g-1)(b-1)p}^2(\alpha) \quad (4.16)$$

A expressão (4.17) indica o teste de hipótese para verificar se há efeito significativo do fator um.

$$\begin{aligned} H_0: \tau_1 = \tau_2 = \dots = \tau_g = 0 \\ H_1: \exists \tau_i \neq 0 \end{aligned} \quad (4.17)$$

A estatística Λ^* para o fator um e o critério para rejeição da hipótese nula do teste (4.17) são apresentados nas expressões (4.18) e (4.19) respectivamente.

$$\Lambda^* = \frac{|SSP_{res}|}{|SSP_{fac1} + SSP_{res}|} \quad (4.18)$$

$$-\left[gb(n-1) - \frac{p+1-(g-1)}{2}\right] \ln \Lambda^* > \chi_{(g-1)p}^2(\alpha) \quad (4.19)$$

De maneira similar o teste para o fator dois pode ser desenvolvido de acordo com a expressão (4.20).

$$\begin{aligned} H_0: \beta_1 = \beta_2 = \dots = \beta_b = 0 \\ H_1: \exists \beta_i \neq 0 \end{aligned} \quad (4.20)$$

A estatística Λ^* para o fator dois e o critério para rejeição da hipótese nula do teste (4.20) são apresentados nas expressões (4.21) e (4.22) respectivamente.

$$\Lambda^* = \frac{|SSP_{res}|}{|SSP_{fac2} + SSP_{res}|} \quad (4.21)$$

$$-\left[gb(n-1) - \frac{p+1-(b-1)}{2}\right] \ln \Lambda^* > \chi_{(b-1)p}^2(\alpha) \quad (4.22)$$

Um maior detalhamento da metodologia MANOVA pode ser encontrado em Johnson e Wichern (2008).

No contexto da pesquisa, a MANOVA pode ser utilizada para testar a relevância de variáveis qualitativas nas vendas de um produto numa loja. Nesse sentido, assume-se que cada variável representa um fator na metodologia, e cada classificação possível da variável representa um nível do fator. Assim, a metodologia é aplicada para verificar se há diferença significativa nas vendas observadas de acordo com os níveis de cada fator. Os fatores para os quais forem rejeitadas as hipóteses nulas de impactos nulos, podem ser caracterizadas como variáveis significativas para os modelos de previsão.

A Figura 4.18 apresenta a distribuição de vendas de um produto influenciado por promoções ao longo de seu histórico de vendas em uma loja de um varejo alimentar. Para avaliar se uma variável que indica a existência de promoções é significativa ou não, podemos aplicar a MANOVA proposta na metodologia dessa pesquisa. Nesse caso, o fenômeno de interesse é caracterizado pelas vendas em si; o único fator atuante é a presença de promoções, que possui dois níveis, com e sem promoção.

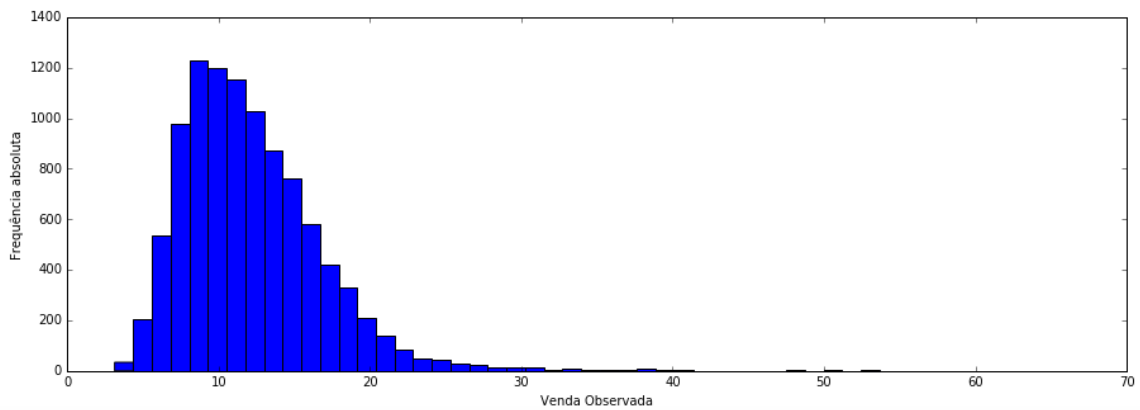


Figura 4.18 – Amostra de dados de vendas de um produto submetido a promoções

Fonte: Próprio autor

A Figura 4.19 apresenta os mesmos dados da Figura 4.18, porém com histogramas distintos para vendas observadas em períodos promocionais e fora de períodos promocionais. Pode-se observar uma diferença entre os centros das distribuições. A aplicação da MANOVA visa identificar se essa diferença entre as distribuições é estatisticamente significativa para que a variável qualitativa de promoções seja incluída como variável explicativa das vendas.

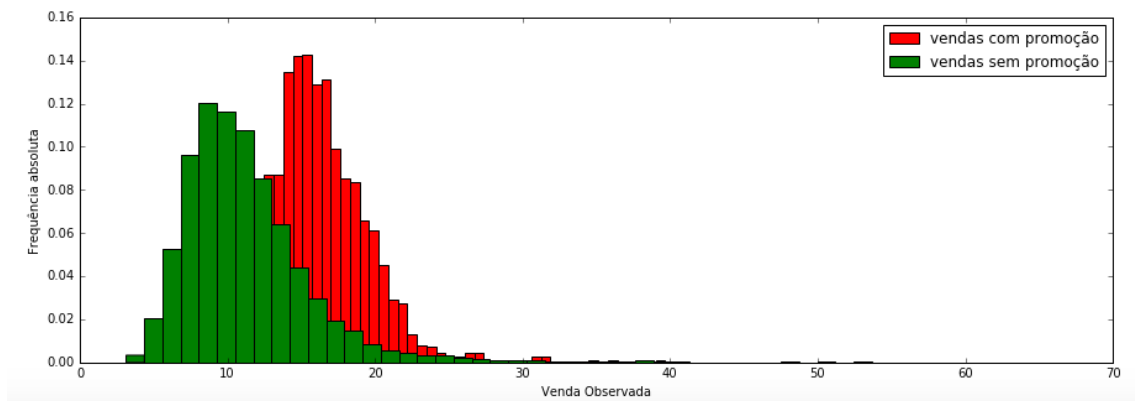


Figura 4.19 – Amostras separadas de dados de vendas de um produto submetido a promoções

Fonte: Próprio autor

A aplicação da metodologia MANOVA aos dados da Figura 4.19 resulta em um valor da estatística de teste igual a 1923,44 e um p-valor igual a 10^{-8} . Com isso é possível rejeitar H_0 e afirmar que a variável qualitativa que indica a existência de promoção é significativa.

A Figura 4.20 apresenta o pseudo-código da metodologia proposta para seleção de variáveis qualitativas em que X é uma matriz de variáveis de entradas qualitativas e y é a variável de vendas observadas.

4.3.7. Conversão de variáveis qualitativas

Muitos dos fatores que podem explicar a demanda potencial de um produto podem ser definidos por variáveis qualitativas. Exemplos desse tipo de variável são classificações do produto como um lançamento, informações sobre a categoria do produto, informações sobre a loja ou mesmo informações de campanhas de marketing, como por exemplo se o produto estava presente em uma campanha de grande divulgação ou não. Os valores que essas variáveis podem assumir recebem o nome de classes.

```

SelecaoVariaveisQualitativas( $X, y$ )
    fatores_significativos = {}
    fatores = {}
    niveis = {}
    Para cada dimensão  $d$  de  $X$  faça
        fatores ← fatores + { $d$ }
        niveis[ $d$ ] = diferentes classificações de  $X$ [ $d$ ]
        fatores, lambdas_wilks = MANOVA( $X, y, fatores, niveis$ )
    Para cada dimensão  $d$  de  $X$  faça
        Caso teste_hipotese(lambda_wilks[ $d$ ]) rejeita  $H_0$  faça
            fatores_significativos ← fatores_significativos + {fatores[ $d$ ]}
    retorna fatores_significativos
    
```

Figura 4.20 – Pseudo-código do método de ajuste de escala

Fonte: Próprio autor

Essas variáveis podem ser classificadas em ordinais ou categóricas (ALPAYDIN, 2010):

1. Uma variável qualitativa ordinal recebe esse nome pois há uma ordem implícita nas classes dessa variável, ainda que os valores das classes não representem valores numéricos. Assumindo que uma variável qualitativa pode assumir valores “A”, “B” e “C”, para que essa variável seja ordinal, deve haver uma ordem implícita, por

exemplo “A” > “B” > “C”. Um exemplo de uma variável qualitativa ordinal seria a classificação de um ponto de venda como pequeno, médio ou grande;

2. Uma variável qualitativa se diz categórica se não há nenhum tipo de ordenação nas suas classes. Nesse caso as classes representam algum tipo de categorização do objeto a que a variável diz respeito. Um exemplo de uma variável categórica é a classificação de um produto como parte da linha contínua de produtos de uma loja ou como algum lançamento especial.

Essas variáveis qualitativas não podem ser diretamente introduzidas nos modelos matemáticos. Por isso devem ser aplicadas técnicas de codificação dessas variáveis para aquelas que forem relevantes para explicação das vendas observadas (vide seção 4.3.6).

Para as variáveis qualitativas ordinais basta codificar os valores qualitativos da variável em números naturais de forma a respeitar a ordenação implícita nos valores. Por exemplo, os valores “pequeno”, “médio” e “grande” que descrevem um ponto de venda e podem ser substituídos pelos valores um, dois e três. Esse tipo de codificação recebe o nome de *Label Encoding* ou etiquetamento.

Para as variáveis qualitativas categóricas propõe-se a utilização de uma técnica chamada *one-hot encoding* (ALPAYDIN, 2010). Essa técnica converte cada variável categórica em um conjunto de variáveis binárias, uma para cada classe da variável. Para uma determinada observação, apenas a variável binária que representa a classe da observação é igual a um e as demais são iguais a zero. A vantagem dessa técnica para transformar variáveis qualitativas categóricas é que os vetores resultantes não possuem uma ordenação implícita, assim como os valores das categorias originais.

A Figura 4.21 apresenta o pseudo-código do método para conversão de variáveis qualitativas proposto nessa pesquisa.

ConversaoVariaveisCategoricas(X)

Para cada dimensão d de X **faça**

Se d é uma variável qualitativa **faça**

Se d é uma variável ordinal **faça**

Para cada observação o de X **faça** // atribui o valor correspondente

$X[o,d] \leftarrow d_{corr}$ // atribui um valor correspondente a classe

Se d é uma variável categórica **faça**

Cria as variáveis $[d1_bin, \dots, dn_bin]$ para cada classe de d

Para cada observação o de X **faça** // atribui o valor correspondente

$X[o,d] \leftarrow [d1_bin=0, \dots, d_corr=1, \dots, dn_bin]$

retorna X

Figura 4.21 – Pseudo-código do método de conversão de variáveis qualitativas

Fonte: Próprio autor

A Figura 4.22 ilustra um exemplo de conversão de duas variáveis categóricas pela técnica de *one-hot encoding*. São consideradas duas variáveis de exemplo, uma que identifica a região do país na qual uma loja está localizada e outra que identifica se a loja está localizada dentro de um *shopping center* ou não. A variável que identifica a região possui cinco classes e a variável que indica se a loja está dentro de um *shopping center* possui duas classes. Nesse caso após a aplicação da codificação *one-hot*, são criadas sete variáveis binárias diferentes, cinco mutuamente excludentes para a primeira variável e duas mutuamente excludentes para a segunda variável. Uma observação de vendas de uma loja na região sul dentro de um *shopping center* seria convertida num vetor binário igual a $(1,0,0,0,0,1,0)$ pela codificação exemplificada na Figura 4.22 .

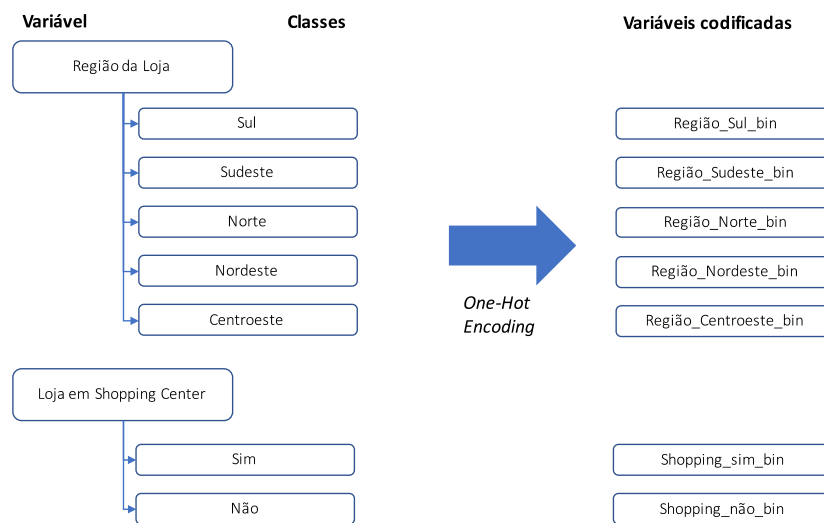


Figura 4.22 – Ilustração da técnica *one-hot encoding*

Fonte: Próprio autor

Uma ressalva deve ser feita caso a quantidade de classes de uma variável qualitativa ordinal ou cardinal seja muito grande. Nesses casos, a aplicação direta do etiquetamento ou mesmo da codificação *one-hot* pode aumentar muito o número de dimensões da amostra de observações. Nesses casos, conforme mencionado por Alpaydin (2010) é necessário realizar uma análise da frequência de ocorrência de cada classe de uma variável e agrupar aquelas com representatividade menor que 5% em classes agregadas e em seguida aplicar a técnica de codificação apropriada.

4.3.8. Inclusão de sinais de tempo

O comportamento de consumo no varejo possui ciclos que refletem os ciclos do cotidiano dos consumidores. Para que modelos de aprendizado computacional sejam capazes de capturar a variação da demanda nesses ciclos é necessário introduzir contadores ou sinais de tempo que representem esses ciclos. Essas variáveis permitem com que os modelos de aprendizado computacional mapeiem o comportamento específico de cada período do ciclo.

Conforme Ehrenthal, Honhon e Van Woensel (2014), o comportamento de vendas no varejo possui um ciclo anual e um ciclo intramensal associado ao pagamento de salários. Assim, os ciclos relevantes propostos pela metodologia são:

- Mensal: ciclo dos 12 meses que compõem o ano;
- Intramensal: ciclo de 4 a 5 semanas que compõem o mês;
- Semanal: ciclo de 52 semanas que compõem o ano.

Considerando esses três ciclos, é possível criar variáveis binárias indicando para uma dada observação em que posição do ciclo ela se encontra. A Figura 4.23 ilustra o pseudo-código do método de inclusão de sinais de tempo, sendo que **mês** é uma função que retorna o número do mês de um ano dada uma data, **semana_mes** é uma função que retorna o número da semana dentro do mês dada uma data e **semana_ano** é uma função que retorna o número da semana dentro de um ano dada uma data.

```
SinaisTempo(X)  
  Para cada observação d de X faça  
    data ← d.data  
    Para i de 1 ate 12 faça  
      Se mes(d) = i faça  
        X[d, mes[i]] = 1  
      Caso contrário faça  
        X[d, mes[i]] = 0  
    Para i de 1 ate 5 faça  
      Se semana_mes(d) = i faça  
        X[d, semana_mes[i]] = 1  
      Caso contrário faça  
        X[d, semana_mes[i]] = 0  
    Para i de 1 ate 52 faça  
      Se semana_ano(d) = i faça  
        X[d, semana_ano[i]] = 1  
      Caso contrário faça  
        X[d, semana_ano[i]] = 0  
  retorna X
```

Figura 4.23 – Pseudo-código do método de inclusão de sinais de tempo

4.3.9. Binarização da saída

No caso da caracterização do problema em termos da demanda definida em múltiplos de movimentação (vide seção 3.5), há a necessidade de converter as vendas de uma amostra de dados em termos dos múltiplos de movimentação.

Seguindo a notação da seção 3, a demanda potencial D_{ik}^t de um produto k em uma loja i no período t deve ser expressa em termos de seu múltiplo de movimentação Q_k . Isso é obtido pela utilização das variáveis binárias B_{ikn}^t . Essa conversão pode ser feita através da expressão (4.23) em que $\lceil x \rceil$ representa o operador teto de um número real x .

$$B_{ikn}^t = \begin{cases} 1 & \text{se } \left\lceil \frac{D_{ik}^t}{Q_k} \right\rceil = n \\ 0 & \text{caso contrário} \end{cases} \quad (4.23)$$

Deve-se atentar a questão de quantas variáveis B_{ikn}^t devem ser criadas, o que depende do número máximo que pode ser assumido pelo teto do quociente entre a demanda potencial e o múltiplo de movimentação. Esse número máximo recebe o nome de n_{ik}^{max} e pode ser determinado pela própria série histórica de demanda potencial (expressão (4.24)).

$$n_{ik}^{max} = \max_{t=\{1,2,\dots,T\}} \left(\left\lceil \frac{D_{ik}^t}{Q_k} \right\rceil \right) \quad (4.24)$$

A binarização da saída é um processo simples de conversão das vendas observadas e tem como entrada apenas a série histórica de demanda potencial e o múltiplo de movimentação. O pseudo-código encontra-se representado na Figura 4.24, em que y representa a série de vendas que será convertida nas variáveis binárias.

BinarizaçãoSaída(y, Q)

$y_{max} \leftarrow \max(y)$

$n_{max} \leftarrow \text{teto}(y_{max}/Q)$

Cria as variáveis $B_n, n=\{1,2,\dots,n_{max}\}$

Para cada observação y_j em y **faça**

$n_j \leftarrow \text{teto}(y_j/I) \quad y_j \leftarrow [B_1=0, B_2=0, \dots, B_{n_j}=1, \dots, B_{n_{max}}=0]$

retorna y

Figura 4.24 – Pseudo-código do método de binarização da saída

Fonte: Próprio autor

4.4. Teste preliminar

Após o tratamento de dados pela aplicação dos procedimentos descritos na seção 4.3, a segunda etapa da metodologia proposta compreende a execução de testes preliminares de modelos de previsão de demanda com o objetivo de identificar quais modelos são mais promissores. A Figura 4.25 ilustra o fluxo dos testes preliminares.

A execução de testes preliminares exige a definição de um conjunto de dados, uma metodologia de separação entre dados de treino e teste, uma lista de modelos a serem testados e uma métrica de desempenho para a comparação dos modelos.

No caso da metodologia proposta, o conjunto de dados depende da caracterização do problema considerado (seção 3). No caso da caracterização dos problemas um e dois, cada conjunto de dados consiste na série de vendas observadas e variáveis exógenas de um produto apenas. No caso da caracterização dos problemas três e quatro, o conjunto de dados abrange as séries de vendas observadas de um conjunto de produtos em um conjunto de lojas.

A respeito do conjunto de dados, de acordo com a caracterização do problema apresentada no capítulo 3 a variável de interesse de cada amostra depende do horizonte de previsão h definido para o problema, ou seja, deve-se definir quantos períodos futuros devem ser previstos pelos modelos para a construção das amostras.

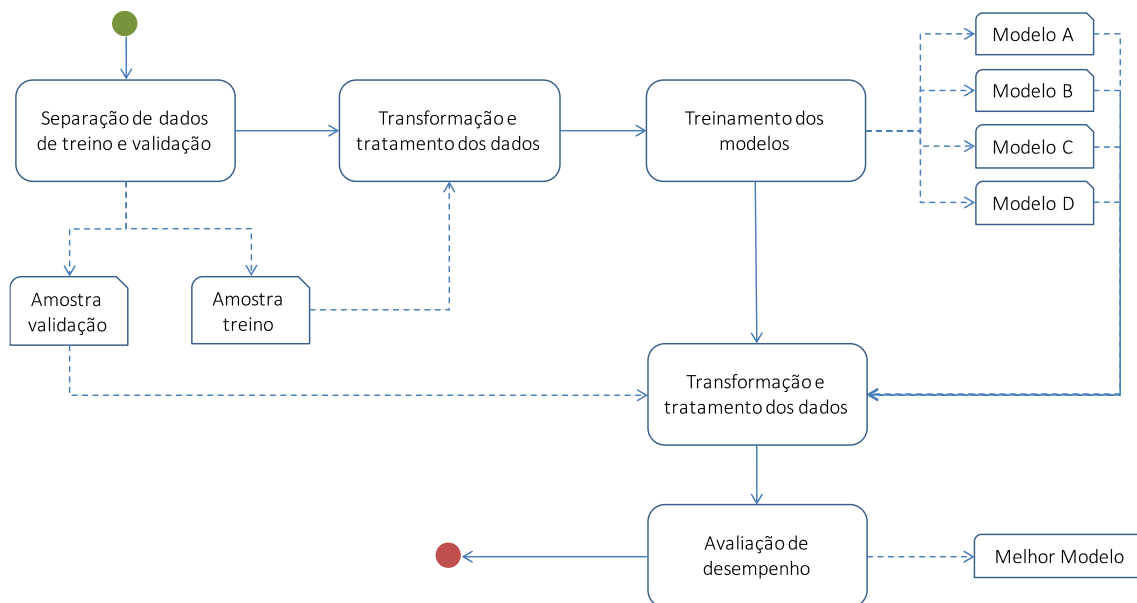


Figura 4.25 – Fluxo de testes preliminares

Fonte: Próprio autor

Sobre a separação de dados entre treinamento e teste, uma vez que o problema abordado na pesquisa se refere a um problema de previsão temporal, faz sentido que o método de separação de dados seja definido conforme um critério temporal. As recomendações encontradas na literatura sobre a proporção de dados a serem utilizados para treinamento e para teste são aplicáveis apenas a problemas específicos, porém Alpaydin (2010) afirma que uma prática comum é separar 20% dos dados para teste e o restante para treinamento. O método de separação de dados proposto para os testes preliminares consiste na separação de 20% dos dados mais recentes das séries de vendas observadas para teste, sendo os 80% restantes utilizados para treinamento dos modelos. Todas as transformações e técnicas de tratamento de dados propostas na seção 4.3 são utilizadas na amostra de treino.

A seção 4.4.1 detalha o procedimento de separação dos dados. Não foi empregada nenhuma técnica de validação cruzada, como por exemplo os métodos *k-fold* e *leave-one-out*, pois essas metodologias envolvem a separação de porções aleatórias da amostra de dados, o que por sua vez representa uma quebra da estrutura temporal dos dados. É necessário

explicitamente separar dados passados e dados futuros para realmente avaliar a capacidade de previsão dos modelos no contexto temporal.

A lista de modelos propostos para esse teste são: rede neural *Multi-Layer Perceptron* (MLP), árvore de decisão, máquina de vetor de suporte (*Support Vector Machine*) e *Gradient Boosting Machine*. A especificação de cada um desses modelos se encontra detalhada no APÊNDICE A. Não seria possível testar com uma lista exaustiva de modelos; sendo assim, a escolha desses quatro modelos se justifica pois cada um deles representa um paradigma diferente de modelos de aprendizado computacional.

Ao término do treinamento de um determinado modelo a amostra de teste é submetida às mesmas transformações da amostra de treino e os *inputs* são utilizados no modelo para que sejam obtidas previsões. As previsões são comparadas com os valores reais da amostra de teste para cálculo da métrica de avaliação.

Quanto a medida de erro utilizada para comparar os modelos, tendo em vista que o objetivo primordial da metodologia é gerar modelos de previsão de vendas, propõe-se o uso de uma medida de erro para avaliação de grandezas numéricas que compara as vendas previstas e as vendas observadas na amostra de teste. No caso, recomenda-se o uso do erro absoluto percentual médio (*Mean Absolute Percentage Error*) por duas razões: é uma medida de erro relativa e, portanto, não sofre influência da amplitude das vendas, como seria o caso com o erro quadrático médio, e é uma medida de erro facilmente comunicável com gestores de cadeias varejistas e profissionais envolvidos em previsão de vendas (BOONE *et al.*, 2019).

Cabe ressaltar que essa medida de erro para avaliação do desempenho de cada tipo de modelo é diferente das funções de perda que são utilizadas no treinamento dos modelos. As funções de perda devem ser definidas e aplicadas individualmente no treinamento de cada modelo. O APÊNDICE A apresenta detalhes sobre o treinamento de cada um dos modelos considerados no teste preliminar.

4.4.1. Conjunto de treino e separação de dados

Para determinação dos conjuntos de dados de treinamento e teste são necessárias duas definições: (i) qual a caracterização do problema de acordo com as definições do capítulo 3

e (ii) qual o horizonte de previsão, ou seja, quantos períodos futuros se deseja prever. Com essas duas definições pode-se construir a amostra de dados para treinamento e teste de um modelo de previsão.

O conjunto de dados de treinamento cumpre a finalidade de ser utilizado para a parametrização dos modelos de previsão e o conjunto de teste é utilizado para avaliar o desempenho de cada modelo segundo uma métrica pré-estabelecida. Uma vez que o problema de previsão de demanda tem uma estrutura temporal em que se deseja utilizar dados do passado para a previsão de valores futuros, faz sentido que a separação entre treino e teste tenha um componente temporal.

A Figura 4.26 ilustra um exemplo de separação de dados entre treino e teste considerando apenas as vendas observadas de um produto. Nesse caso consideram-se 80% dos dados como treino e os 20% restantes como teste.

Além disso, experimentos com modelos de aprendizado computacional envolvem a realização de testes com diferentes combinações de conjuntos de dados, o que se traduz em selecionar diferentes combinações de dados de treino e teste para considerar os efeitos dos dados de treinamento selecionados nos resultados dos modelos. Existem muitas formas de realizar essa separação, como, por exemplo, as metodologias de validação cruzada *k-fold* e *leave-one-out* (APLAYDIN, 2010). No entanto, essas metodologias convencionais de validação cruzada se baseiam na seleção aleatória de observações da amostra de dados e isso pode gerar amostras de treino e teste em que dados de períodos futuros sejam utilizados para previsão de períodos passados, o que não condiz com a definição do problema em que apenas dados de períodos passados podem ser utilizados para previsão de períodos futuros.

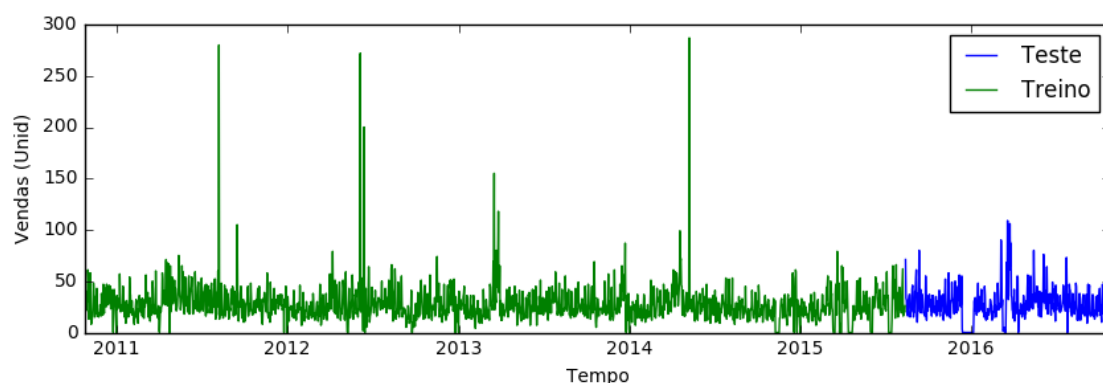


Figura 4.26 – Separação entre dados de treino e teste

Fonte: Próprio autor

No caso do problema considerado nesta pesquisa, por se tratar de um problema com estrutura temporal, foi adotada uma metodologia que consiste em selecionar janelas sequenciais para criar diferentes combinações de amostras de treino e teste. Para realizar essa separação para um conjunto de dados, basta dividir a amostra em conjuntos sequenciais com número de amostras iguais e sem seguida percorrer os grupos sequencialmente tomando um grupo como amostra de teste e os demais anteriores como amostra de treino. Esse procedimento é ilustrado na Figura 4.27. Dado um conjunto de dados e um horizonte de previsão, essa metodologia de separação gera sub amostras sequenciais de treino e teste permitindo várias execuções dos testes dos modelos.

Cabe ressaltar que caso dos problemas dois e quatro do capítulo 3 que consideram conjuntos de produtos e conjuntos de lojas, basta aplicar a mesma metodologia de separação de dados para cada um dos produtos e lojas e combinar as amostras num só conjunto de dados previamente ao treinamento ou teste dos modelos.

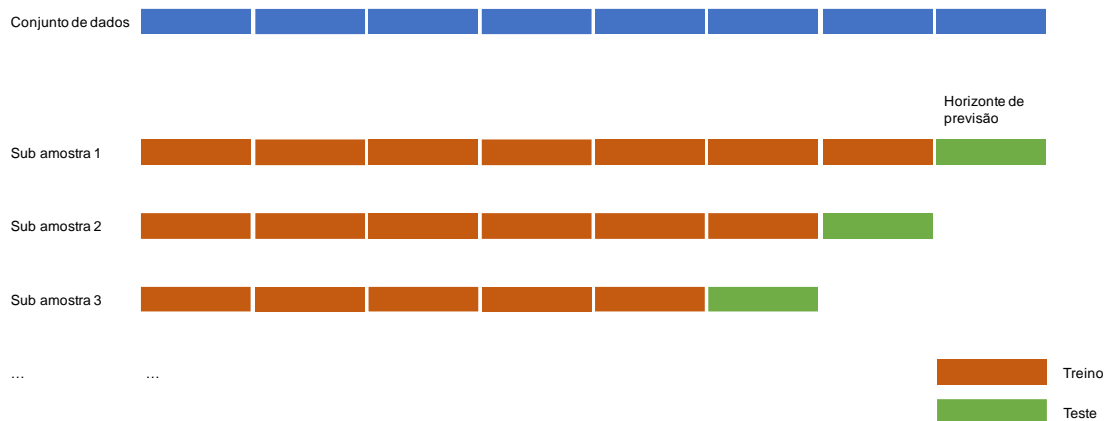


Figura 4.27 – Construção de amostras de treino e teste

Fonte: Próprio autor

4.4.2. Modelos para o teste preliminar

Esta subseção detalha os modelos utilizados no teste preliminar. A escolha de parâmetros e as estratégias de treinamento são propostas com base em recomendações encontradas na literatura. Nenhuma otimização de parâmetros é realizada nos modelos propostos, uma vez que o teste preliminar visa apenas identificar qual o melhor tipo de modelo para um determinado conjunto de dados. A otimização do modelo mais promissor é realizada em uma etapa subsequente da metodologia proposta.

Os modelos selecionados para o teste preliminar são rede neural MLP, árvore de decisão, máquina de vetor de suporte (*Support Vector Machine*) e *Gradient Boosting Machine*. Cada um desses modelos representa um paradigma diferente de modelo de aprendizado computacional.

Algumas recomendações para a implementação da rede MLP para o teste preliminar são:

- Recomenda-se que no teste preliminar seja utilizada apenas uma camada intermediária com uma quantidade de neurônios igual à quantidade de *inputs* da rede. Para as caracterizações dos problemas um e três a rede possui apenas uma saída e para as caracterizações dos problemas dois e quatro a rede possui uma saída para cada valor possível da demanda expressa em múltiplos de movimentação;

- Para as caracterizações dos problemas um e três, recomendam-se funções de ativação sigmóides nos neurônios intermediários e linear no neurônio de saída. Para as caracterizações do problema dois e quatro são recomendadas funções sigmóides nos neurônios intermediários de funções *softmax* nos neurônios de saída;
- Para treinamento da rede no teste preliminar é recomendada uma taxa de aprendizado de 0.1 e um coeficiente de momento de 0.3. Seguindo a recomendação de Rojas (1996), o treinamento das redes deve ser realizado pelo algoritmo *RMSPProp*;
- O número de épocas depende da disponibilidade de dados e das características do problema; no entanto, como valor de referência recomenda-se um número máximo de 2000 épocas, sendo que a cada época toda a amostra de treino é apresentada simultaneamente para o modelo, configurando uma estratégia de treino por batelada;
- As funções de perda que devem ser utilizadas para treinamento do modelo nos testes preliminares são o erro quadrático médio para as caracterizações dos problemas um e três e a entropia cruzada de classificação para as caracterizações dos problemas dois e quatro.

Algumas recomendações para a implementação da árvore de decisão para o teste preliminar são:

- O critério de impureza utilizado nos nós depende da caracterização do problema. No caso das caracterizações dos problemas um e três, por serem problemas de regressão, o critério de impureza recomendado é dado pelo erro quadrático médio. Já no caso das caracterizações dos problemas dois e quatro, uma vez que os mesmos podem ser encarados como problemas de classificação, recomenda-se que a medida de impureza seja o coeficiente de Gini (BREIMAN *et al.* 1984);
- O algoritmo de treinamento recomendado é o algoritmo CART. Assim, a cada iteração o algoritmo busca dentre todos *inputs*, qual produz o *split* que minimiza a medida de impureza;

- Recomenda-se como parâmetro de quantidade mínima de amostras por folha da árvore um valor de 5% da quantidade de amostras disponíveis para treinamento.

Algumas recomendações para a implementação da SVM para o teste preliminar são:

- Recomenda-se o uso de um *kernel* polinomial de baixo grau. Isso permite a modelagem de tendências mais suaves das séries de vendas observadas (SMOLA e SCHÖLKOPF; 2002);
- Além disso, recomenda-se que todas as entradas do modelo tenham seus valores ajustados para dentro da escala de 0,1 até 0,9 (SMOLA e SCHÖLKOPF; 2002);

Algumas recomendações para a implementação da GBM para o teste preliminar são:

- Para as caracterizações um e três do problema a função de perda recomendada é o erro de *huber* com fator 0,2. Para as caracterizações do problema dois e quatro a função de erro indicada é a entropia cruzada que mede o erro de classificação;
- Recomenda-se o uso de uma taxa de aprendizado pequena, entre 0,1 e 0,2. Além disso recomenda-se o uso de uma quantidade grande de estimadores sequenciais para compensar a pequena taxa de aprendizado. Um valor adequado são 250 estimadores;
- Para evitar sobre-treinamento com o uso de muitos estimadores sequenciais no teste preliminar recomenda-se utilizar a técnica de subamostragem com fator de 60% (NATEKIN E KNOLL, 2013);

4.5. Otimização de parâmetros

Uma vez que o melhor tipo de modelo seja identificado, a metodologia proposta nesta pesquisa propõe a busca de parâmetros ótimos para melhoria de desempenho de previsão. Nesta pesquisa utiliza-se uma busca por varredura no espaço de parâmetros do modelo (*grid search*) para definição do conjunto de parâmetros mais apropriado.

Podem ser aplicadas técnicas diferentes de busca de parâmetros como por exemplo uma varredura randomizada (*random search*) (ALPAYDIN, 2010). Porém, considerando que o espaço de parâmetros dos modelos é pequeno, abordagens mais sofisticadas de busca tendem a trazer ganhos pouco significativos e sem grande influência nos parâmetros selecionados pelo *grid search*.

O espaço de parâmetros depende por sua vez do tipo de modelo selecionado como mais adequado:

- No caso da MLP, o espaço de parâmetros envolve a topologia da rede (quantidade de camadas intermediárias e quantidade de neurônios em cada camada, as taxas de aprendizado e de momentum. Não são considerados outros parâmetros como: diferentes funções de ativação, diferentes algoritmos de treinamento e aplicação de estratégias de regularização de modelos;
- No caso da árvore de decisão, o espaço de parâmetros envolve a quantidade mínima de amostras por folha da árvore e o critério de decisão de *split*;
- No caso da máquina de vetor de suporte, o espaço de parâmetros envolve o tipo de função *kernel*, a constante de penalidade dos erros, e o fator de tolerância do erro;
- No caso da *gradient boosting machine*, o espaço de parâmetros envolve a quantidade de iterações que equivale ao número de estimadores sequenciais, a taxa de subamostragem, a taxa de aprendizado e a profundidade máxima dos estimadores.

4.6. Estimativa de desempenho

Após a identificação do melhor conjunto de parâmetros, a metodologia recomenda uma etapa final para diagnóstico do modelo resultante. Esse diagnóstico compreende: um novo treinamento do modelo no conjunto de dados, uma nova estimativa do erro do modelo considerando dados de teste e comparação de desempenho com relação a um modelo de referência.

O novo treinamento segue os mesmos princípios citados na seção 4.4.1 de separação de dados, ou seja, devem ser construídas amostras sequenciais e com separações entre treino e teste para validação do modelo.

No caso do problema de previsão de demanda em cadeias varejistas, a medida de acurácia que melhor comunica a qualidade do modelo para os envolvidos na gestão da cadeia de suprimentos é o erro absoluto percentual médio (*Mean Absolute Percentage Error - MAPE*). Essa é a medida de erro utilizada nessa etapa da metodologia proposta.

O modelo de comparação recomendado nesta pesquisa é a classe de modelos ARIMA. Esse tipo de modelo é tido como referência pelos profissionais envolvidos em previsão de demanda e é flexível suficiente para incluir conjuntos de dados com tendências e sazonalidades (FILDES *et al.*, 2008). Cabe ressaltar que os modelos ARIMA são, em alguns casos, equivalentes a modelos de suavização exponencial (TIAO, 2015), sendo que dessa forma suprim a necessidade de comparação com esse tipo de modelo.

Para a aplicação de modelos da classe ARIMA é necessário determinar a ordem dos modelos, ou seja, a quantidade p de termos autoregressivos, a quantidade q de termos de médias móveis e a quantidade d de diferenciações (maiores detalhes são apresentados no APÊNDICE A). Sugere-se o uso de um método de força bruta para determinação desses parâmetros sendo que o espaço de busca para os parâmetros p e q varia de 0 a 5 individualmente, e o parâmetro d varia de 0 a 2 (esses valores para p , q e d são os mesmos utilizados por Zhang (2003) para identificação automática de modelos ARIMA). Seleciona-se o modelo com menor erro em uma amostra de teste o qual passa a ser a referência de desempenho para comparação com o modelo de aprendizado computacional treinado com o conjunto otimizado de parâmetros.

Em seguida são comparados os erros produzidos pelo modelo com parâmetros otimizados e os erros produzidos pelo modelo ARIMA selecionado pelo método de força bruta. De acordo com Boone *et al.* (2019), a qualidade de um modelo de previsão depende não apenas da média de erros, como também da variância dos mesmos. De acordo com Moretin e Toloï (2004), modelos de previsão adequados possuem três características:

- Baixos erros médios de previsão em comparação com a média observada da série;
- Baixa variância de erros em comparação com a variância observada da série, e;
- Baixo viés de estimação, representada por erros centrados em zero.

Por isso, essa metodologia propõe que os erros dos modelos sejam comparados em termos de erros médios, variância de erros e viés de erros. Por isso, os modelos de aprendizado computacional ajustados com parâmetros otimizados são considerados melhores que os modelos ARIMA ajustados pelo método de força bruta apenas se os erros resultantes do primeiro tiverem uma média de erros e uma variância de erros menor e houver pouco viés de estimação.

5. APLICAÇÃO DA METODOLOGIA

Para validação da metodologia apresentada no capítulo 4 foi realizado uma aplicação prática considerando dados reais de duas empresas do setor varejista. Esse capítulo contempla a apresentação dos dados, a descrição da aplicação da metodologia, bem como a análise e interpretação dos resultados.

A seção 5.1 apresenta uma descrição da sequência experimental utilizada nesta aplicação prática. A seção 5.2 apresenta detalhamento dos dados considerados. São apresentadas algumas estatísticas descritivas da amostra de dados de cada empresa. As seções 5.3 a 5.6 apresentam os resultados da aplicação da metodologia para cada uma das quatro caracterizações do problema:

- **Previsão de demanda de um único produto:** a série de vendas de cada produto em cada loja é considerada isoladamente para a aplicação da metodologia e o problema consiste em prever a demanda dos próximos períodos (seção 3.2). Nas figuras e tabelas dessa seção essa representação do problema recebe o nome de “problema 1” ou “caracterização 1”;
- **Previsão de múltiplos de movimentação de um único produto:** a série de vendas de cada produto em cada loja é considerada isoladamente para a aplicação da metodologia e o problema consiste em prever a demanda dos próximos períodos expressa em intervalos discretos, chamados múltiplos de venda (seção 3.3). Nas figuras e tabelas desta seção essa representação do problema recebe o nome de “problema 2” ou “caracterização 2”;
- **Previsão de demanda de um grupo de produtos:** as séries de vendas dos produtos nas lojas são consideradas em conjunto para aplicação da metodologia e o problema consiste em prever a demanda dos próximos períodos (seção 3.4). Nas figuras e tabelas dessa seção essa representação do problema recebe o nome de “problema 3” ou “caracterização 3”;
- **Previsão de múltiplos de movimentação de um grupo de produtos:** as séries de vendas dos produtos nas lojas são consideradas conjuntamente para aplicação da

metodologia, e o problema consiste em prever a demanda dos próximos períodos expressa em intervalos discretos, chamados múltiplos de venda (seção 3.4). Nas figuras e tabelas desta seção essa representação do problema recebe o nome de “problema 4” ou “caracterização 4”;

A comparação dos resultados, considerando cada caracterização do problema, é apresentada na seção 5.7. Por fim, na seção 5.8 é feito um resumo dos resultados com as principais conclusões decorrentes desta aplicação prática.

5.1. Descrição do experimento

Cada uma das seções 5.3 a 5.6, em que são determinadas as previsões, compreende a aplicação dos passos da metodologia proposta no capítulo 4. Os passos são:

- Tratamento dos dados: a primeira etapa da metodologia compreende a aplicação sequencial dos métodos de tratamento descritos nas subseções 4.3.1 a 4.3.9 para preparação das amostras de dados. Os métodos de tratamento de dados foram implementados em sequências de cálculo representadas por grafos direcionais;
- Treinamento dos modelos de aprendizado computacional: a segunda etapa se inicia com a separação da amostra de dados em conjuntos de treinamento e teste (subseção 4.4.1). Para cada amostra de dados são gerados quatro pares de conjuntos de treino e teste. Com as amostras de treino e teste são executados o treinamento e avaliação dos quatro modelos sugeridos com a finalidade de identificar os modelos com menores erros de previsão (subseção 4.4.2). Os modelos foram implementados dentro dos grafos direcionais em conjunto com os métodos de tratamento de dados. Os modelos são treinados 30 vezes para cada sub-amostra de dados, mitigando efeitos de sorteios de números aleatórios presentes nos algoritmos de treinamento de cada modelo. Para esta etapa da metodologia foram utilizadas recomendações da literatura quanto a escolha dos hiperparâmetros de cada modelo. Para cada execução da sequência de cálculo para cada sub-amostra são geradas quatro previsões de vendas semanais. Isso é obtido pela aplicação recursiva da sequência de cálculo. Isso permite determinar a precisão do modelo conforme o horizonte de

planejamento se estende. Os erros de cada modelo são comparados para determinação da classe de modelo com menores erros de previsão. Para a primeira e segunda caracterizações do problema, que consideram cada conjunto de dados isoladamente, é selecionado um modelo para cada conjunto de dados. Para as caracterizações do problema três e quatro, que consideram os conjuntos de dados de forma coletiva, é selecionado apenas um modelo;

- Otimização dos parâmetros do modelo com menores erros: uma vez identificado o modelo com menores erros, a terceira etapa da metodologia é a varredura do espaço de parâmetros do modelo com menores erros para escolha do melhor conjunto de parâmetros (seção 4.5). No caso da caracterização do problema um e dois, que consideram cada conjunto de dados isoladamente, o melhor modelo para cada conjunto de dados é otimizado individualmente. Já no caso das caracterizações três e quatro, apenas o modelo selecionado passa pela etapa de otimização de parâmetros. Para cada modelo, todas as combinações de uma grade de parâmetros são testadas e comparadas. A combinação que produz menores erros médios de previsão é escolhida como conjunto ótimo de parâmetros para o respectivo modelo;
- Construção de um modelo de referência e comparação de resultados: por fim, a metodologia prevê o ajuste de um modelo da classe ARIMA para comparação de erros com os erros do modelo de aprendizado computacional com parâmetros otimizados resultante da etapa anterior (seção 4.6). Para todas as caracterizações do problema é ajustado um modelo ARIMA para cada conjunto de dados e os resultados do modelo otimizado para um determinado conjunto de dados são comparados com os resultados do respectivo modelo ARIMA. Os resultados são comparados em termos de erros médios, variância dos erros e viés de estimação.

Em consonância com a metodologia proposta no capítulo 4, os erros de previsão utilizados para análise de resultados ao longo das etapas da aplicação foram calculados segundo a métrica de erro absoluto percentual médio (MAPE). Essa métrica foi escolhida para representação dos erros nessa aplicação prática pelos seguintes motivos:

- Permite a comparação de erros de produtos com vendas de amplitudes diferentes uma vez que o erro calculado é relativo;
- Como o erro é considerado de forma absoluta, erros positivos e erros negativos não se anulam;
- É a métrica comumente utilizada em aplicações de previsão de demanda (BOONE *et al.*, 2019).

5.2. Conjuntos de dados

Foram coletadas informações de séries de vendas, preços e posições de estoque de duas empresas. Por razões de confidencialidade dos dados essas empresas são referidas nessa pesquisa com nomes fictícios. As empresas e suas principais características são:

1. Empresa A: rede de hipermercados com atuação nacional, sendo uma das três principais redes de supermercados do Brasil; possui aproximadamente 500 lojas. Além das informações de vendas diárias e estoques foram coletadas as informações de preços, promoções e ocorrência de feriados especiais (levantados por meio de entrevistas com os gestores da empresa);
2. Empresa B: empresa do segmento alimentar, produz e comercializa chocolates e doces. Atua em território nacional por meio de uma rede de lojas próprias e franquias (aproximadamente 1.000 pontos de venda). No caso da empresa B apenas as informações de vendas e estoques diárias foram disponibilizadas.

Para as duas empresas foram selecionadas amostras de produtos vendidos em lojas, ou seja, cada conjunto de dados corresponde a uma série de vendas e variáveis associadas de um produto em uma loja. A Tabela 5.1 apresenta os detalhes dessas amostras.

A Tabela 5.2 apresenta algumas estatísticas descritivas das séries de vendas diárias das amostras de dados da empresa A. Cada conjunto de dados representa as vendas de um determinado produto em uma das lojas da empresa A. Observa-se que na amostra de dados estão presentes produtos de alto giro nas lojas (ex.: A_09 e A_10), com altos valores de vendas médias diárias, assim como produtos de baixo giro com baixas vendas médias diárias

(ex.: A_27 e A_30). Também estão presentes produtos com alta volatilidade de vendas e baixa volatilidade, sendo a volatilidade representada pelo coeficiente de variação da amostra. Em geral, as séries de vendas contemplam três anos de registros diários de vendas. Entende-se que essa amostra envolve todos os tipos de produtos em termos de características de vendas e, portanto, é representativa do desafio de realizar previsões de demanda desagregada no cenário de negócios da empresa A.

Tabela 5.1 – Conjuntos de dados do estudo de caso

EMPRESA	CONJUNTOS DE DADOS	DESCRIÇÃO
Empresa A	40	Dados de produtos das categorias Casa, Açougue e Mercearia em três lojas de grande porte.
Empresa B	50	Dados de vendas de produtos de consumo rápido (bombons e barras de chocolate) e de presentes (caixas de chocolate)

Fonte: Próprio autor

Tabela 5.2 – Estatísticas descritivas dos dados da empresa A

Dataset	vendas					Dataset	vendas				
	max	min	média	desv_pad	cont		max	min	média	desv_pad	cont
A_01	498	0	49,54	68,11	937	A_21	118	0	27,47	16,77	938
A_02	3883	0	143,66	383,92	937	A_22	6456	0	653,54	608,67	938
A_03	338	0	71,06	64,62	937	A_23	5889	0	507,11	511,93	937
A_04	496	0	73,40	69,52	405	A_24	660	0	96,88	81,53	937
A_05	542	0	90,53	80,68	937	A_25	1777	0	410,61	227,67	938
A_06	551	0	99,71	84,26	937	A_26	6574	0	341,64	483,25	938
A_07	481	0	87,22	58,35	937	A_27	13	0	1,73	2,02	937
A_08	219	0	55,15	33,78	937	A_28	27	0	5,27	4,86	937
A_09	3896	0	621,45	416,02	937	A_29	29	0	3,69	5,11	815
A_10	4225	0	412,71	372,85	937	A_30	38	0	2,48	4,03	668
A_11	809	0	94,18	90,90	937	A_31	38	0	4,07	5,88	314
A_12	1035	0	319,84	158,23	937	A_32	127	0	33,44	14,27	935
A_13	7736	0	378,95	579,05	937	A_33	142	0	22,28	21,18	935
A_14	364	0	22,30	33,95	938	A_34	1338	0	56,53	60,38	935
A_15	5042	0	74,64	284,13	938	A_35	117	0	28,76	19,89	935
A_16	175	0	34,96	31,40	937	A_36	141	0	36,92	20,08	935
A_17	354	0	40,95	40,78	405	A_37	227	0	21,87	17,60	935
A_18	175	0	29,10	27,17	938	A_38	289	0	20,11	13,39	935
A_19	333	0	41,83	31,82	938	A_39	133	0	17,08	15,29	935
A_20	452	0	76,93	54,93	938	A_40	212	0	28,69	28,87	935

Fonte: Próprio autor

As mesmas estatísticas descritivas com relação aos dados da empresa B são apresentadas na Tabela 5.3. Assim como no caso da empresa A os conjuntos de dados selecionados da empresa B contemplam produtos com características distintas, representando de forma adequada o problema de previsão de demanda da empresa. Uma questão específica dos dados da empresa B é que os conjuntos de dados têm quantidades de observações bastante diferentes entre si, sendo que o produto com maior histórico de vendas possui 2191 observações e o produto com menor histórico possui 210 observações. Isso se dá, pois, a empresa B continuamente lança produtos para incrementar o portfólio oferecido ao consumidor final.

Tabela 5.3 – Estatísticas descritivas dos dados da empresa B

	vendas						vendas				
Dataset	max	min	média	desv_pad	cont	Dataset	max	min	média	desv_pad	cont
B_01	287	0	27,42	17,51	2191	B_26	11	0	0,34	0,80	701
B_02	388	0	32,25	29,24	1503	B_27	4	0	0,18	0,48	635
B_03	32	0	3,87	3,64	2191	B_28	2	0	0,09	0,31	689
B_04	31	0	4,14	3,44	2191	B_29	2	0	0,05	0,25	687
B_05	15	0	1,65	2,15	210	B_30	7	0	0,06	0,36	692
B_06	54	0	0,92	2,44	2190	B_31	24	0	1,52	2,69	751
B_07	25	0	1,27	1,97	2191	B_32	78	0	2,18	4,89	751
B_08	14	0	0,60	1,01	2188	B_33	25	0	0,57	1,56	751
B_09	13	0	0,43	0,89	2188	B_34	14	0	0,48	1,23	751
B_10	26	0	0,56	1,41	1739	B_35	4	0	0,09	0,37	688
B_11	122	0	24,32	21,73	916	B_36	1	0	0,01	0,11	751
B_12	178	0	25,73	31,81	917	B_37	61	0	1,04	4,15	751
B_13	98	0	3,57	5,79	899	B_38	6	0	0,17	0,57	303
B_14	65	0	3,66	4,81	916	B_39	7	0	0,11	0,44	751
B_15	25	0	1,06	1,87	688	B_40	8	0	0,12	0,50	751
B_16	4	0	0,31	0,64	911	B_41	67	0	7,36	6,14	2097
B_17	243	0	24,02	32,00	911	B_42	109	0	10,47	11,67	1506
B_18	30	0	1,71	2,37	318	B_43	26	0	0,84	1,81	2097
B_19	11	0	1,05	1,48	911	B_44	11	0	0,55	1,11	2094
B_20	14	0	0,48	1,03	913	B_45	6	0	0,63	0,95	213
B_21	25	0	7,15	5,38	698	B_46	18	0	0,32	1,02	2089
B_22	47	0	8,87	7,79	698	B_47	15	0	0,45	1,13	2095
B_23	10	0	1,26	1,57	697	B_48	10	0	0,18	0,64	2079
B_24	5	0	0,60	0,86	701	B_49	8	0	0,11	0,48	2094
B_25	2	0	0,22	0,44	212	B_50	7	0	0,12	0,41	1730

Fonte: Próprio autor

5.3. Problema de previsão de demanda de um único produto

Essa seção descreve a aplicação da metodologia para os dados das empresas A e B considerando o problema de previsão de vendas de cada produto isoladamente.

A primeira fase da metodologia consiste na preparação de dados, o que compreende a aplicação das técnicas de pré-processamento. A segunda fase da metodologia é a realização de testes preliminares com os quatro tipos de modelos propostos na subseção 4.4.2 para encontrar o tipo de modelo com melhor desempenho para cada conjunto de dados. Essas duas fases foram implementadas em conjunto numa sequência de processamento (Figura 5.1). No caso do problema de previsão individualizada de produtos em cada loja, os processos de pré-processamento e modelos de aprendizado computacional foram aplicados isoladamente em cada conjunto de dados das empresas A e B.

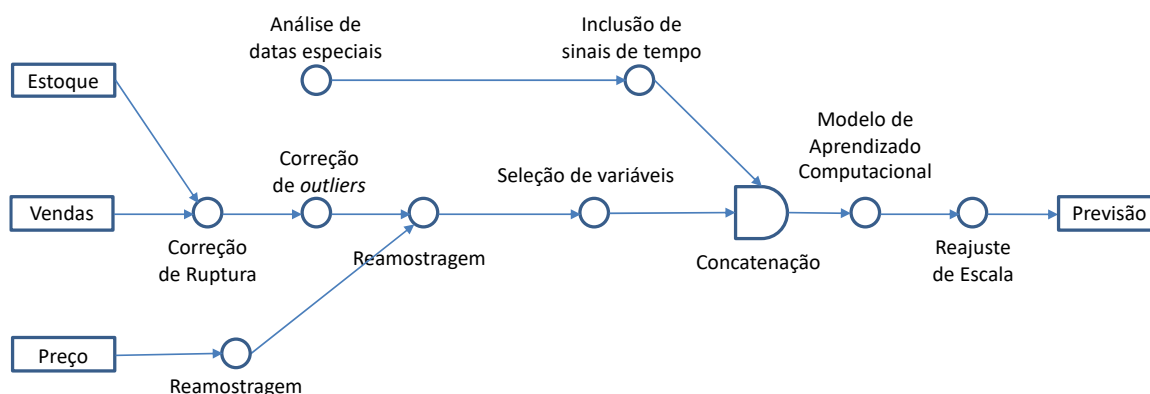


Figura 5.1 – Sequência de processamento para a caracterização 1 do problema

Fonte: Próprio autor

A sequência de cálculo da Figura 5.1 se inicia com a aplicação da heurística de identificação e correção de rupturas considerando as entradas de vendas observadas e os estoques. Em seguida as vendas corrigidas são submetidas a uma correção de *outliers*. Em seguida é feita uma reamostragem dos dados de vendas e de preço em janelas semanais. Essa reamostragem atende as necessidades de planejamento do varejo, uma vez que não são necessárias previsões diárias de vendas e, por outro lado, previsões mensais têm pouco valor para decisões operacionais de abastecimento.

As vendas e preços reamostrados são utilizadas no processo de seleção de variáveis que cumpre a finalidade de identificar os *lags* que devem ser utilizadas como atributos explicativos da venda atual.

Paralelamente ao processamento das entradas de vendas observada, estoques e preços, procede-se a criação dos sinais de feriados, a criação dos sinais binários de tempo. Os feriados utilizados são: dia dos pais, dia das mães, Páscoa, Natal e dia das crianças (essas datas especiais foram identificadas de acordo com a metodologia descrita em 4.3.1).

Todas as variáveis são compactadas numa matriz de dados (nó de concatenação) e em seguida são processadas por um modelo de aprendizado computacional. Os modelos de aprendizado computacional são treinados para prever um horizonte futuro de uma semana. A sequência de processamento é especializada para prever apenas uma semana futura e previsões de horizontes maiores são feitas com a aplicação recursiva da sequência de cálculo.

A saída do nó de modelo de aprendizado são previsões de vendas semanais dentro da escala entre zero e um. Por isso em seguida é realizado o procedimento inverso de ajuste de escala, resultando em previsões de vendas com a amplitude da amostra de dados original.

A Tabela 5.4 mostra os hiperparâmetros utilizados para a primeira fase da metodologia.

A Figura 5.2 apresenta os resultados da primeira e segunda fase da metodologia, conforme apresentado nas seções 4.3 e 4.4. para os dados da empresa A. Cada conjunto de barras representa os erros em determinados horizontes de previsão, ou seja, o primeiro conjunto de barras contém os erros de um período futuro, o segundo conjunto os erros considerando o segundo período futuro de previsão e assim sucessivamente. As diferentes barras de um conjunto representam os erros para diferentes modelos de aprendizado computacional.

Pode-se observar que o *Gradient Boosting Machine* (GBM) é o modelo que apresenta melhores resultados, tanto em termos da média quanto da dispersão dos erros. A rede neural MLP apresenta resultados ruins inclusive com muita variância nos erros.

Tabela 5.4 – Hiperparâmetros do teste preliminar – Problema 1

NOME DO HIPERPARÂMETRO	SIGNIFICADO	VALOR
GBM		
número de estimadores	quantidade de estimadores a serem empilhados na GBM	500
função de perda	função de perda a ser considerada no problema de minimização de erros	huber
coeficiente da função de perda	coeficiente de parametrização do erro de huber	0,2
sub amostra	percentual da amostra de treino a ser apresentado a cada estimador da GBM	0,6
taxa de aprendizado	taxa de multiplicação sequencial de cada estimador da GBM	0,1
máxima profundidade dos estimadores	máxima profundidade das árvores que são empilhadas na GBM	3
SVM		
kernel	tipo de kernel para recombinação dos dados de entrada	polinomial de grau 3
coeficiente de penalidade		1
tolerância do erro	percentual de erro a partir do qual se penaliza um erro de previsão	0,1
máximo de iterações	quantidade máxima de iterações do procedimento de otimização da SVM	10000
DECISION TREE		
profundidade máxima da árvore	profundidade máxima que um caminho da árvore de regressão pode atingir	nenhuma
critério de split	critério de seleção do split de cada nó	melhor split (não é utilizado nenhum procedimento estocástico)
mínimo de amostras por split	quantidade mínima de amostras que podem caracterizar um split da árvore	2
máximo de features	quantidade de atributos a serem considerados para determinação do split de um nó	não há
RNA		
Camadas intermediárias	Quantidade de camadas de neurônios entre a camada de input e a camada de output	2
Quantidade de neurônios intermediários	Quantidade de neurônios em cada camada intermediária	64
Funções de ativação intermediárias	Funções de ativação da camada intermediária	Retificadora linear
Função de ativação na camada de saída	Funções de ativação na camada de output	Retificadora linear
Épocas	Quantidade de vezes que o modelo é submetido aos dados de treinamento	5000
Algoritmo de otimização	Algoritmo utilizado para atualização dos pesos da RNA	<i>Stochastic Gradient Descent</i>

Fonte: Próprio autor

A Figura 5.3 apresenta a contagem de conjuntos de dados para os quais cada modelo utilizado produziu os menores erros para a empresa A. Pode-se observar que para aproximadamente 60% dos conjuntos de dados utilizados a GBM foi o modelo que resultou no menor erro de previsão. Já o modelo *Support Vector Machine* (SVM) resultou nos

menores erros de previsão para aproximadamente 30% dos conjuntos de dados e a árvore de decisão em 10% dos casos. Em nenhum caso a rede MLP produziu os menores erros de previsão.

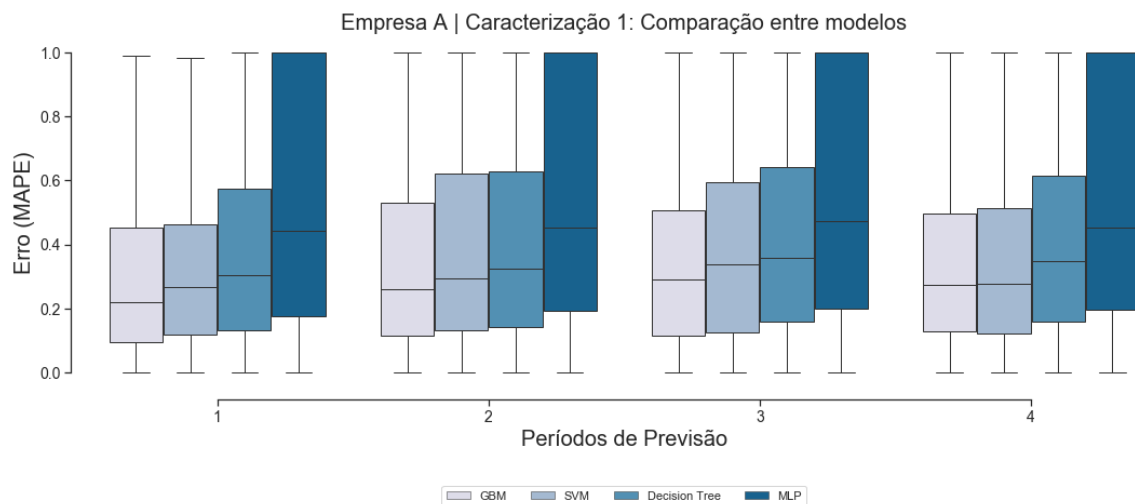


Figura 5.2 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa A considerando o problema 1

Fonte: Próprio autor

Para o problema de previsão de demanda de produtos únicos, os resultados indicam que o modelo GBM possui uma capacidade melhor de representar o problema e generalizar seus resultados para previsões da amostra de teste. De acordo com Alpaydin (2010) o melhor modelo para um determinado problema é aquele que possui a melhor capacidade de generalização, ou seja, de realizar previsões precisas para amostras não utilizadas em seu treinamento. Considerando a Teoria do Aprendizado Estatístico (VAPNIK, 1998), isso é equivalente ao modelo com melhor equilíbrio entre o risco estrutural da sua classe, e a capacidade de previsão de amostras de teste, equivalente ao risco empírico. Os resultados indicam que no caso dos dados da empresa A o modelo com essa característica é a GBM. Maiores detalhes sobre a Teoria do Aprendizado Estatístico podem ser encontrados no Apêndice A (seção A1).

A Figura 5.4 apresenta os resultados da primeira e segunda fase da metodologia para os dados da empresa B. Em comparação aos erros do da empresa A, observa-se uma maior variância dos erros, mesmo no caso dos melhores modelos para cada horizonte de previsão. Assim como no caso da Figura 5.2, a GBM é o modelo que apresenta melhores resultados

de erros médios. Analisando os resultados dos outros modelos, observa-se uma grande variância, o que é indesejado do ponto de vista de planejamento de demanda.

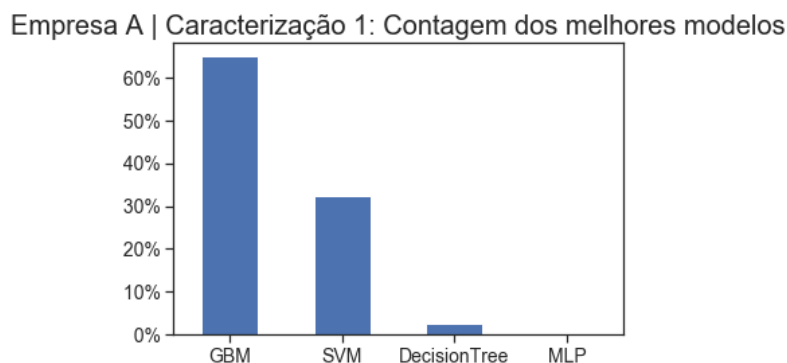


Figura 5.3 – Frequência de melhores modelos para os dados da empresa A para o problema 1

Fonte: Próprio autor

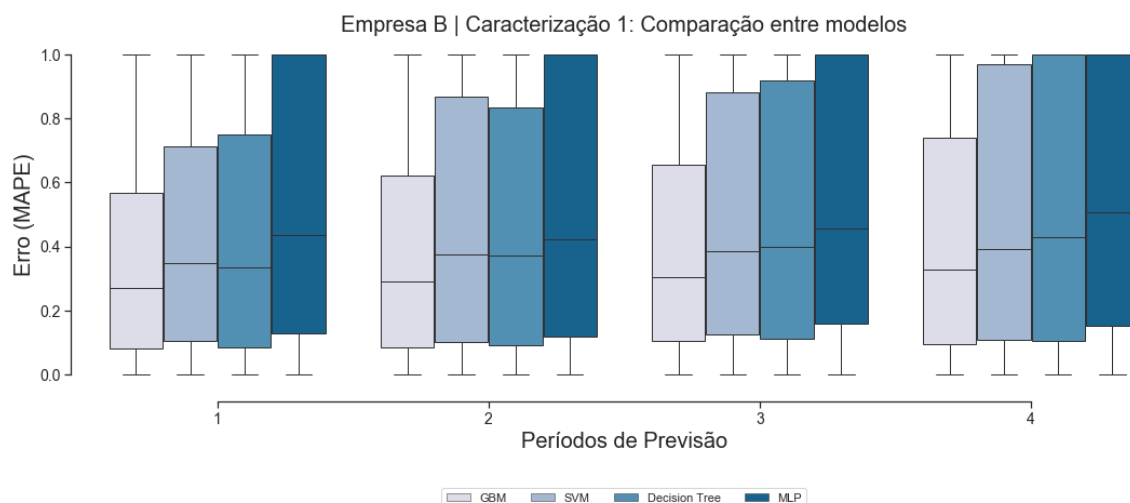


Figura 5.4 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa B considerando o problema 1

Fonte: Próprio autor

A Figura 5.5 apresenta a contagem de melhores modelos considerando os diferentes conjuntos de dados da empresa B. Nesse caso, mais de 80% dos conjuntos de dados tiveram a GBM como o modelo com menores erros de previsão. Assim como no caso dos dados da empresa A esse resultado indica que a classe de modelo GBM é a que possui maior capacidade de generalização para os dados da empresa B.

Empresa B | Caracterização 1: Contagem dos melhores modelos

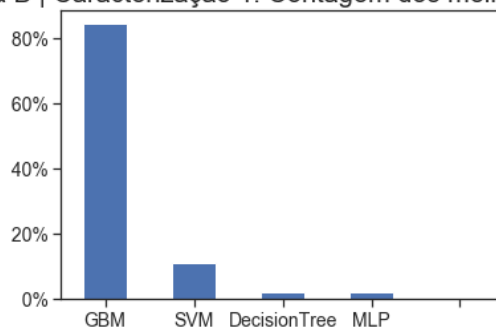


Figura 5.5 – Frequência de melhores modelos para os dados da empresa B para o problema 1

Fonte: Próprio autor

A terceira etapa da metodologia (seção 4.5) consiste na busca de parâmetros ótimos para a melhor classe de modelo associada a um conjunto de dados. A busca de parâmetros ótimos é realizada para o modelo selecionado para cada conjunto de dados.

A busca de parâmetros foi feita com uma busca por varredura de um conjunto de parâmetros dentro do espaço de parâmetros de cada modelo. A Tabela 5.5 apresenta o espaço de parâmetros para cada modelo e os valores considerados na varredura. Como a MLP não foi o melhor modelo para nenhum dos conjuntos de dados, não foram realizadas buscas com esse tipo de modelo e consequentemente não existem parâmetros na grade de busca.

Tabela 5.5 – Grade de busca para a otimização de parâmetros – Problema 1

Modelo	Parâmetro	Valores
Gradient Boosting Machine	número de estimadores	100/500
Gradient Boosting Machine	subamostragem	0,2/0,4/0,6
Gradient Boosting Machine	taxa de aprendizado	0,025/0,1/0,5
Gradient Boosting Machine	profundidade máxima dos estimadores	03/05/07
Support Vector Machine	kernel	rbf/sigmoide/linear/polinomial
Support Vector Machine	penalidade	0,1/1/10
Support Vector Machine	tolerância do erro	0,1/0,2/0,3
Support Vector Machine	grau do kernel polinomial	1/2/3
Árvore de Decisão	mínimo de amostras por folha	02/05/10
Árvore de Decisão	critério de escolha da variável de split	melhor/aleatório

Fonte: Próprio autor

A Figura 5.6 apresenta os resultados obtidos com a otimização dos parâmetros para os conjuntos de dados da empresa A. Cada conjunto de barras apresenta a distribuição do erro percentual absoluto para diferentes horizontes de previsão. Observam-se erros médio entre 20% e 30%. Na pesquisa de Huang *et al.* (2019), uma das poucas encontradas que endereçam o problema de previsão desagregada no varejo, os melhores erros reportados são da ordem de 40%. Isso indica que os erros resultantes da metodologia proposta são bons para o cenário de empresas do varejo.

A Figura 5.7 apresenta a distribuição dos erros percentuais dos modelos otimizados para os dados da empresa A. Observa-se que o centro da distribuição é levemente deslocado para a direita indicando um viés positivo de previsão.

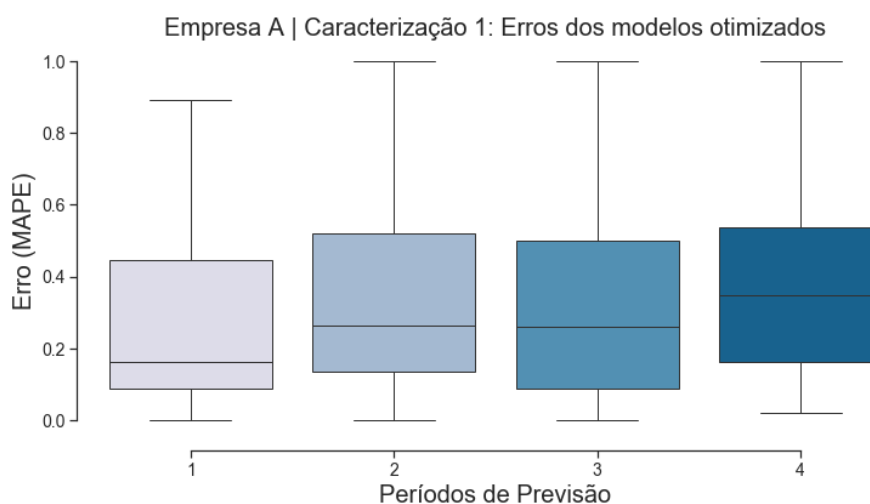


Figura 5.6 – Análise de erros da dos modelos otimizados considerando os dados da empresa A e o problema 1

Fonte: Próprio autor

A Figura 5.8 apresenta os resultados obtidos com a otimização dos parâmetros para os conjuntos de dados da empresa B. Os valores de erros médios são da ordem de 20%. Assim como no caso da empresa A, e tomando como referência os resultados de Huang *et al.* (2019), é possível afirmar que os resultados da metodologia proposta são adequados para previsão desagregada no varejo.

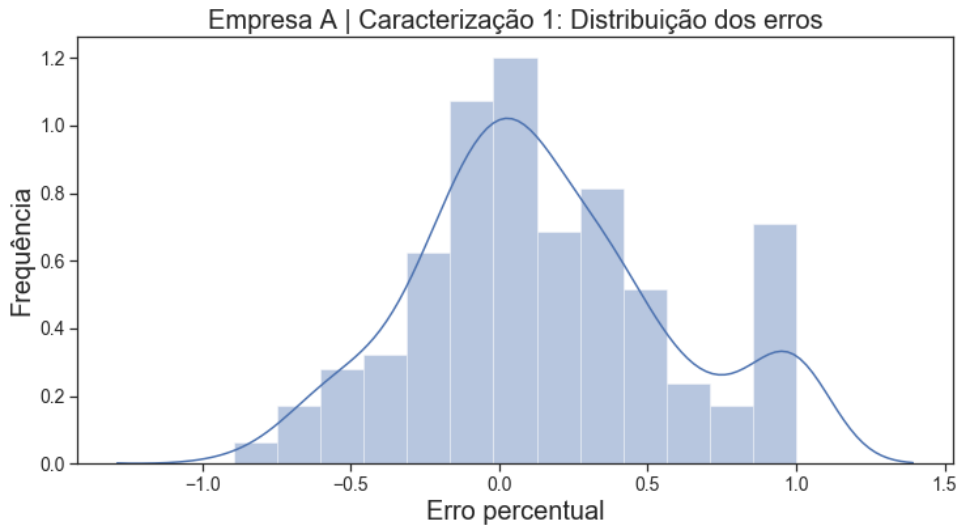


Figura 5.7 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa A e o problema 1

Fonte: Próprio autor

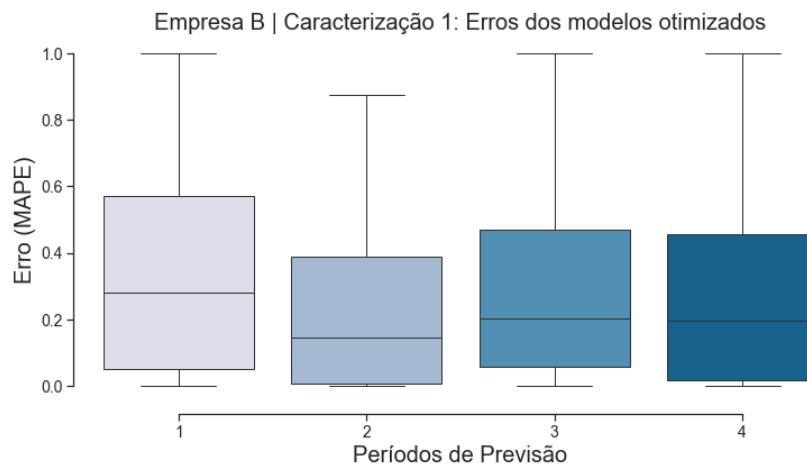


Figura 5.8 – Análise de erros da dos modelos otimizados considerando os dados da empresa B e o problema 1

Fonte: Próprio autor

A Figura 5.9 apresenta distribuição de erros percentuais para os modelos otimizados no caso dos dados da empresa B. A distribuição apresenta o centro próximo de zero e baixa dispersão, indicando pouco viés de previsão e uma menor variância do erro em comparação com o caso da empresa A. Essa baixa variabilidade também é uma característica desejada para os modelos de previsão (MORETIN e TOLOI, 2004).

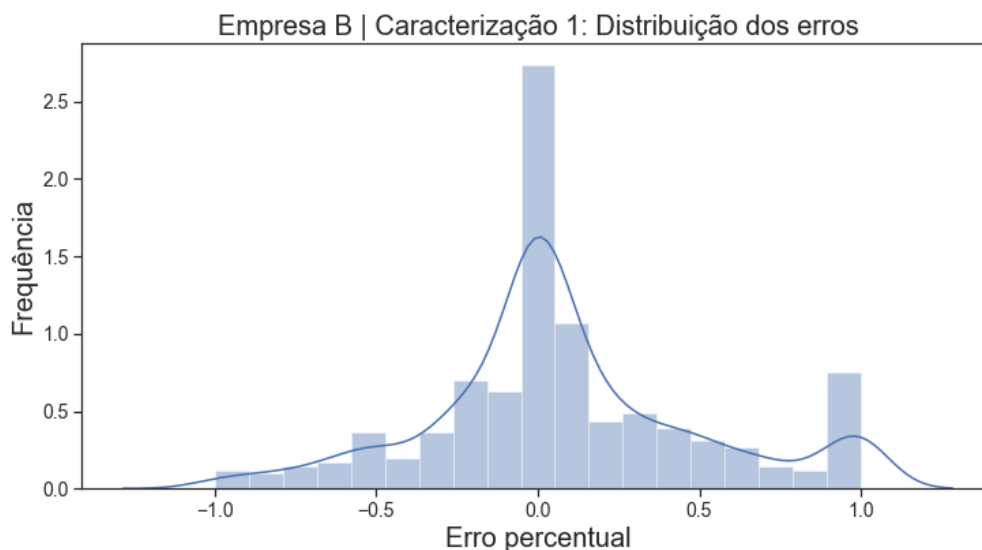


Figura 5.9 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa B e o problema 1

Fonte: Próprio autor

O último passo da metodologia (seção 4.6) consiste na comparação dos resultados de cada modelo otimizado com os com resultados de modelos ARIMA. O resultado de cada modelo de aprendizado computacional otimizado para cada conjunto de dados é comparado com o resultado do modelo ARIMA ajustado para o mesmo conjunto de dados.

A Figura 5.10 apresenta os resultados dos modelos com parâmetros otimizados resultantes da terceira etapa da metodologia aplicada aos dados da empresa A comparados com modelos ARIMA ajustados para cada conjunto de dados pelo método de força bruta (vide seção 4.6). Observa-se que os erros produzidos pelos modelos de aprendizado computacional possuem as médias e variâncias dos erros menores em todos os períodos.

A Tabela 5.6 contém os valores médios e os desvios padrão dos erros produzidos pelo modelo de aprendizado computacional e pelos modelos ARIMA para os dados da empresa A. Os dados são equivalentes aos dados da Figura 5.10.

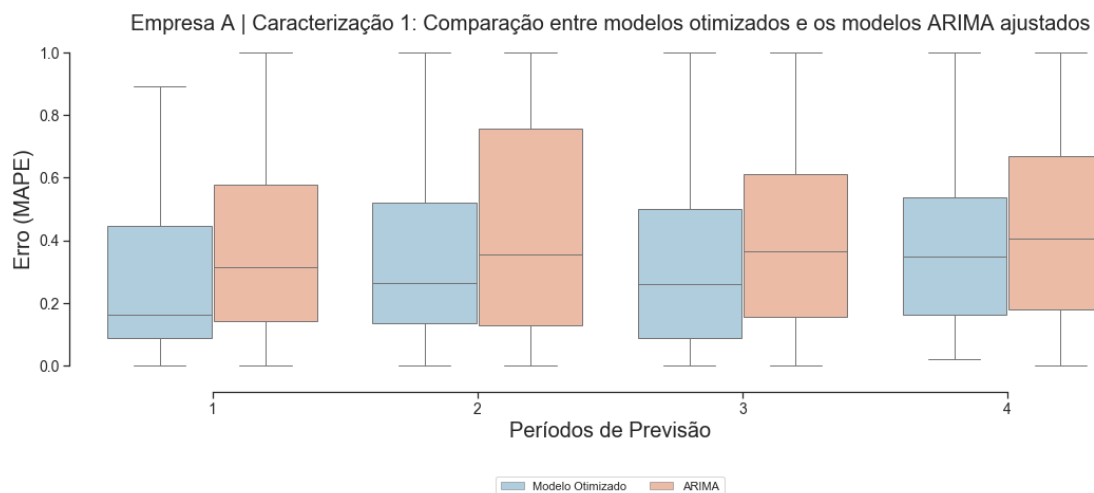


Figura 5.10 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A Problema 1

Fonte: Próprio autor

Tabela 5.6 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A – Problema 1

Horizonte	Melhor Modelo		ARIMA		Diferença	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
1	30,80%	30,61%	39,61%	31,11%	8,81%	0,50%
2	35,66%	29,23%	44,38%	34,82%	8,72%	5,58%
3	34,81%	30,70%	42,46%	32,16%	7,65%	1,45%
4	37,65%	26,84%	44,99%	32,67%	7,34%	5,83%

Fonte: Próprio autor

Pode-se observar que no caso do problema de previsão de demanda de produtos únicos para os dados da empresa A a metodologia proposta produziu modelos com menores erros médios de previsão e menores desvios em relação aos modelos ARIMA. A média de redução de erros dentre os horizontes de previsão considerados foi de 8,13%.

A Figura 5.11 apresenta os resultados dos melhores modelos com parâmetros otimizados para os dados da empresa B comparados com modelos ARIMA ajustados para cada conjunto de dados. Exceto para o primeiro período do horizonte do futuro, os modelos de aprendizado computacional possuem distribuições de erros com menores médias e

menores desvios. No caso do primeiro período do horizonte os resultados são muito próximos.

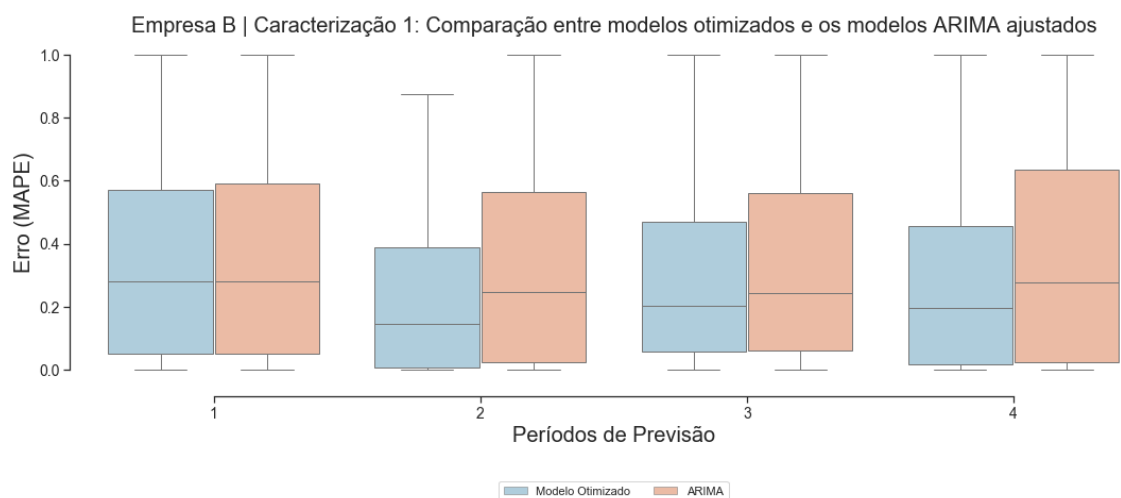


Figura 5.11 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B Problema 1

Fonte: Próprio autor

A Tabela 5.7 contém os valores médios e os desvios padrão dos erros produzidos pelos modelos de aprendizado computacional e pelos modelos ARIMA para os dados da empresa B. Os resultados da tabela indicam que a metodologia proposta foi efetiva na construção de modelos de previsão para os dados da empresa B. A redução de erros médios dentre os horizontes de previsão considerados foi de 5,69%.

Tabela 5.7 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B – Problema 1

Horizonte	Melhor Modelo		ARIMA		Diferença	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
1	33,76%	31,85%	36,29%	33,41%	2,53%	1,56%
2	26,66%	31,38%	34,94%	34,14%	8,28%	2,76%
3	30,06%	30,47%	35,28%	34,06%	5,22%	3,58%
4	29,89%	31,69%	36,62%	35,08%	6,74%	3,39%

Fonte: Próprio autor

Em ambos os casos, tanto para a empresa A quanto para a empresa B, considerando a definição do problema como um problema de previsão de demanda de cada produto em cada loja, os modelos de aprendizado computacional resultantes da metodologia possuem erros menores e com menores dispersões que os modelos de referência ARIMA.

Considerando os resultados obtidos, é possível concluir que a metodologia proposta foi capaz de produzir modelos de previsão com menores erros e com menor dispersão de erros em todos os casos. A redução de erros médios para a empresa A foi de 8,13% e para a empresa B foi de 5,69%. Ambas as reduções de erros médios estão acompanhadas de reduções de variância de erros. No contexto de decisões de abastecimento de produtos em lojas, essa melhoria de precisão e redução de variância de previsão resultam num aumento direto de nível de serviço ao consumidor final e numa redução de custos de estoque e de *stockout* segundo Corsten e Gruen (2003).

Além disso, uma vez que a metodologia foi aplicada de forma automatizada, é possível afirmar que ela poderia ser utilizada num conjunto muito maior de produtos e lojas, atendendo à uma das condições do problema de previsão desagregada no varejo, que é a necessidade de tratar muitas séries de vendas simultaneamente, sem intervenção de especialistas.

5.4. Problema de previsão de múltiplos de movimentação de um único produto

Essa seção descreve a aplicação da metodologia proposta para os dados das empresas A e B considerando o problema de previsão de múltiplos de movimentação de cada produto.

Assim como na seção 5.3, a primeira e a segunda fase da metodologia foram implementadas em uma sequência de cálculo (Figura 5.12) que aplica as técnicas de pré-processamento e realiza a execução de modelos de aprendizado computacional.

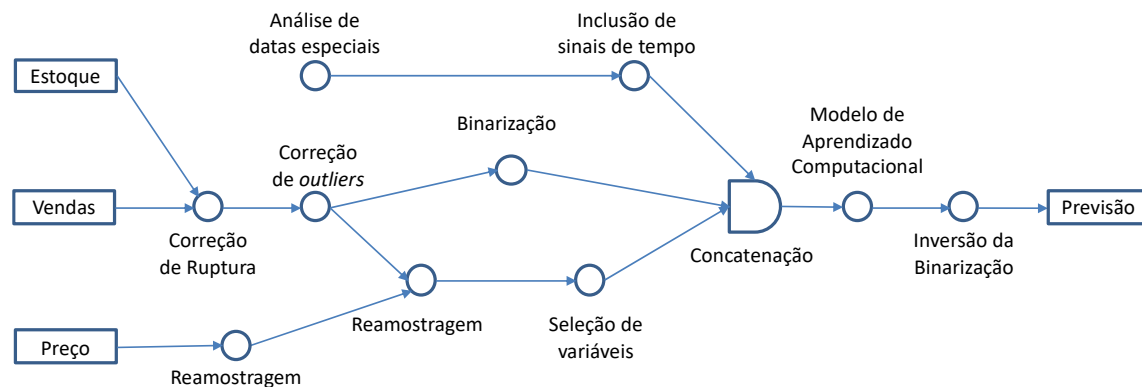


Figura 5.12 – Sequência de processamento para a caracterização 2 do problema

Fonte: Próprio autor

Essa sequência de cálculo é similar à sequência da seção 5.3 exceto pelo fato de que ao invés de se aplicar o ajuste de escala nas vendas, é aplicada a técnica de binarização (subseção 4.3.8). Isso significa que o problema de previsão das vendas se transforma num problema de classificação no qual o modelo deve aprender qual o *bin*, ou janela, de vendas mais provável dos próximos períodos.

Um dos parâmetros da técnica de binarização é a quantidade de *bins*, ou seja, a quantidade de janelas em que os valores de entrada são agrupados. Não foram encontradas na literatura recomendações sobre a quantidade de *bins* a serem utilizadas. Conseqüentemente, para a escolha da quantidade de *bins* a serem consideradas nessa aplicação prática, foi realizada uma análise da amplitude de vendas dos produtos das empresas A e B em relação aos seus respectivos múltiplos de movimentação. Nessa análise foi possível perceber que a quantidade máxima de múltiplos vendida no histórico de vendas dos produtos foi de 10 múltiplos, por isso foi escolhida uma quantidade de *bins* igual a 10 e os valores observados foram binarizados de acordo com esses *bins*.

Uma vez que a saída do modelo de aprendizado computacional é um vetor binário que indica qual o intervalo provável de vendas do próximo período, deve-se aplicar uma operação inversa e transformar esse vetor em uma variável real que representa as vendas previstas num domínio real. Tanto a operação de binarização quanto o inverso da mesma são aplicados sobre as vendas reamostradas semanalmente.

A Tabela 5.8 mostra os hiperparâmetros utilizados para a primeira fase da metodologia.

Tabela 5.8 – Hiperparâmetros do teste preliminar – Problema 2

NOME DO HIPERPARÂMETRO	SIGNIFICADO	VALOR
GBM		
número de estimadores	quantidade de estimadores a serem empilhados na GBM	500
função de perda	função de perda a ser considerada no problema de minimização de erros	Entropia cruzada multicategorias
sub amostra	percentual da amostra de treino a ser apresentado a cada estimador da GBM	0.6
taxa de aprendizado	taxa de multiplicação sequencial de cada estimador da GBM	0.1
máxima profundidade dos estimadores	máxima profundidade das árvores que são empilhadas na GBM	3
SVM		
kernel	tipo de kernel para recombinação dos dados de entrada	Polinomial de grau 3
coeficiente de penalidade		1000
máximo de iterações	quantidade máxima de iterações do procedimento de otimização da SVM	10000
DECISION TREE		
profundidade máxima da árvore	profundidade máxima que um caminho da árvore de regressão pode atingir	Não há
critério de split	critério de seleção do split de cada nó	Melhor
mínimo de amostras por split	quantidade mínima de amostras que podem caracterizar um split da árvore	Não há
máximo de features	quantidade de atributos a serem considerados para determinação do split de um nó	Não há
RNA		
Camadas intermediárias	Quantidade de camadas de neurônios entre a camada de input e a camada de output	1
Quantidade de neurônios intermediários	Quantidade de neurônios em cada camada intermediária	100
Funções de ativação intermediárias	Funções de ativação da camada intermediária	Sigmóides
Função de ativação na camada de saída	Funções de ativação na camada de output	Softmax
Épocas	Quantidade de vezes que o modelo é submetido aos dados de treinamento	1000
Algoritmo de otimização	Algoritmo utilizado para atualização dos pesos da RNA	<i>Stochastic Gradient Descent</i>

Fonte: Próprio autor

A Figura 5.13 apresenta os resultados da primeira e segunda fase da metodologia para os dados da empresa A. Diferentemente da aplicação da metodologia considerando a primeira caracterização do problema em que a GBM foi o modelo que apresentou melhores resultados no teste preliminar em todos os horizontes, no caso da segunda caracterização não há um modelo dominante para todos os horizontes de previsão.

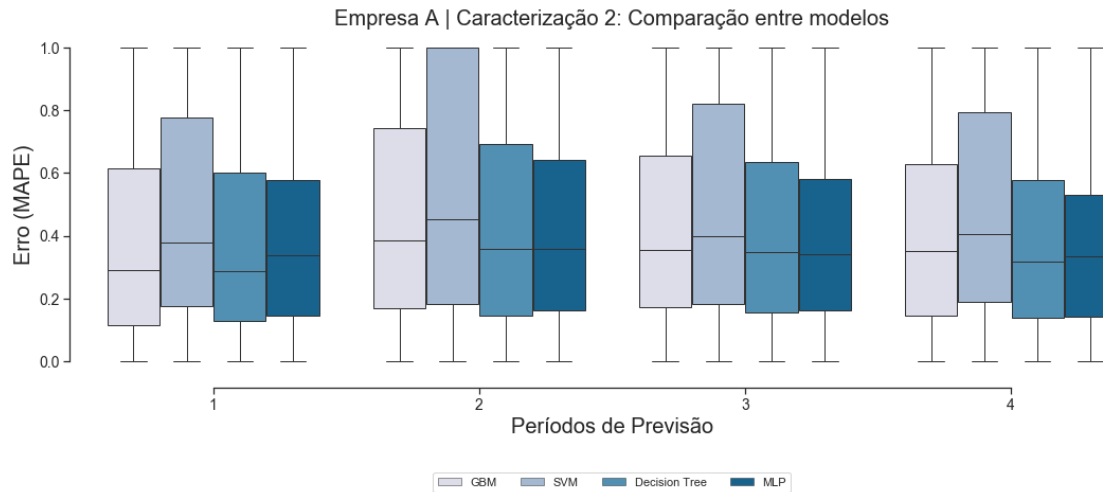


Figura 5.13 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa A considerando o problema 2

Fonte: Próprio autor

A Figura 5.14 apresenta a frequência com que cada tipo de modelo foi o melhor para os conjuntos de dados da empresa A. Observa-se que a RNA apresentou melhores resultados em cerca de 60% dos conjuntos de dados. Esse é um resultado interessante visto que a MLP não apresentou bons resultados para os dados da empresa A no caso da caracterização do problema de previsão de vendas de produtos únicos, ou seja, sem consideração de múltiplos de movimentação para caracterizar a demanda (Figura 5.3).

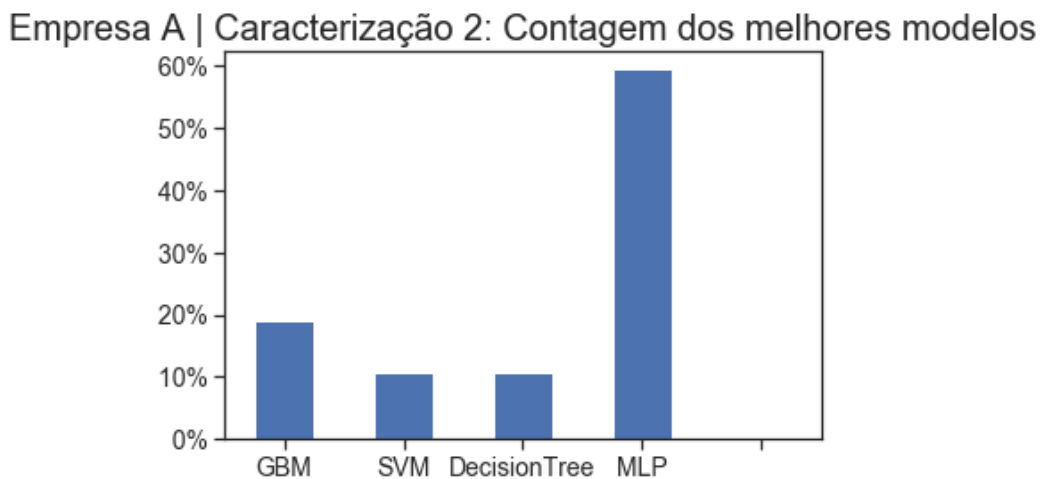


Figura 5.14 – Frequência de melhores modelos para os dados da empresa A para o problema 2

Fonte: Próprio autor

A Figura 5.15 apresenta os resultados da primeira e segunda fase da metodologia para os dados da empresa B. Observa-se uma alta dispersão do erro para os diferentes modelos e horizontes de previsão o que é indesejado.

Nota-se que a ao considerar os múltiplos de movimentação para agrupar as vendas, a metodologia proposta produziu resultados inferiores à definição do problema de previsão de produto únicos. Os erros médios também são maiores que os erros da primeira definição do problema (Figura 5.4) apresentados na seção 5.3. Neste caso, a única diferença em relação à seção 5.3 é a consideração dos múltiplos de movimentação, por isso, pode-se atribuir a isso essa piora de desempenho tanto em erros médios quanto na dispersão dos erros, indicando que a demanda representada por meio de variáveis discretas não é adequada.

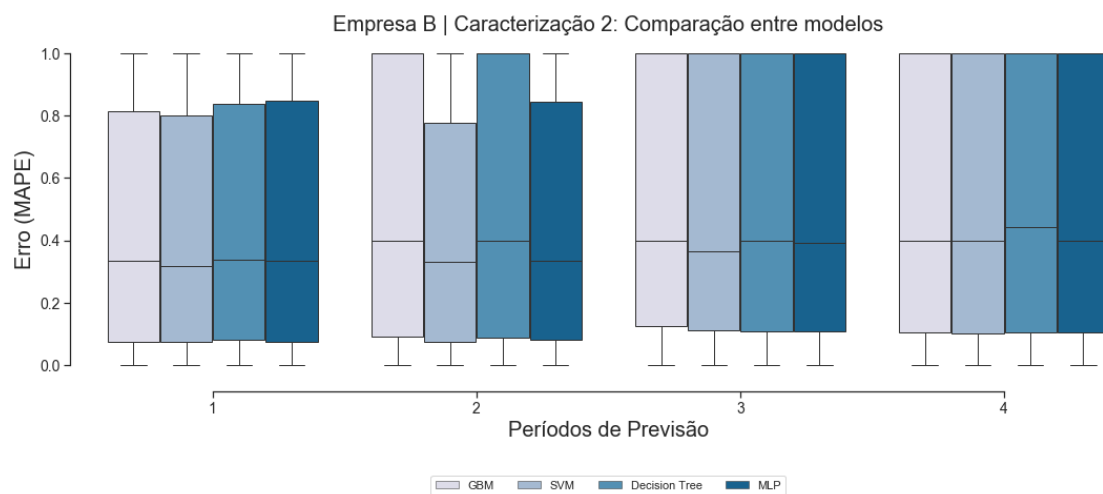


Figura 5.15 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa B considerando o problema 2

Fonte: Próprio autor

A Figura 5.16 apresenta a contagem da frequência com que cada modelo foi o melhor para os conjuntos de dados da empresa B. Não existe um modelo dominante nesse caso. Também é interessante notar que a MLP foi o melhor modelo para alguns conjuntos de dados, ao contrário do que pode ser observado quando se considera a caracterização do problema como previsão de produtos únicos em lojas específicas (Figura 5.5).

Assim como no caso da seção 5.3, a busca de parâmetros deve ser realizada para cada conjunto de dados individualmente. A Tabela 5.9 apresenta o espaço de parâmetros para cada modelo e os valores da grade de busca.

Empresa B | Caracterização 2: Contagem dos melhores modelos

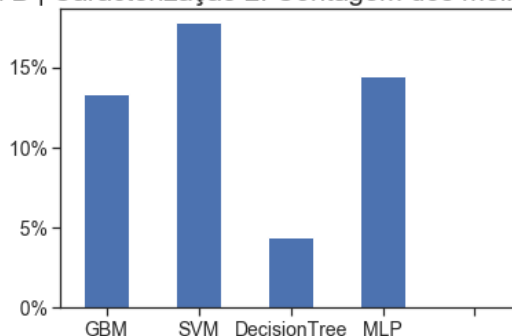


Figura 5.16 – Frequência de melhores modelos para os dados da empresa B para o problema 2

Fonte: Próprio autor

Tabela 5.9 – Grade de busca para a otimização de parâmetros – Problema 1

Modelo	Parâmetro	Valores
Gradient Boosting Machine	número de estimadores	100/500
Gradient Boosting Machine	subamostragem	0,2/0,4/0,6
Gradient Boosting Machine	taxa de aprendizado	0,025/0,1/0,5
Gradient Boosting Machine	profundidade máxima dos estimadores	03/05/07
Support Vector Machine	kernel	rbf/sigmoide/linear/polinomial
Support Vector Machine	penalidade	0,1/1/10
Support Vector Machine	grau do kernel polinomial	1/2/3
Árvore de Decisão	mínimo de amostras por folha	02/05/10
Árvore de Decisão	critério de escolha da variável de split	melhor/aleatório
MLP	número de camadas intermediárias	01/02/03
MLP	número de neurônios por camada intermediária	8/16/32/64
MLP	função de ativação dos neurônios das camadas intermediárias	sigmoide/relu/linear

Fonte: Próprio autor

A Figura 5.17 apresenta os resultados obtidos com a otimização dos parâmetros para os conjuntos de dados da empresa A. Observam-se erros médio entre 25 e 40% indicando

resultados inferiores aos obtidos considerando o problema de previsão de demanda de produtos únicos em lojas específicas que considera a demanda como uma variável numérica contínua, sem o agrupamento em *bins* (seção 5.3). Isso é uma evidência de que a binarização não foi vantajosa para a construção dos modelos, ou seja, a conversão do problema de previsão de um problema de regressão, para um problema de classificação sob a hipótese de que a menor variabilidade de resultados produziria menores erros de previsão, não se mostrou verdadeira.

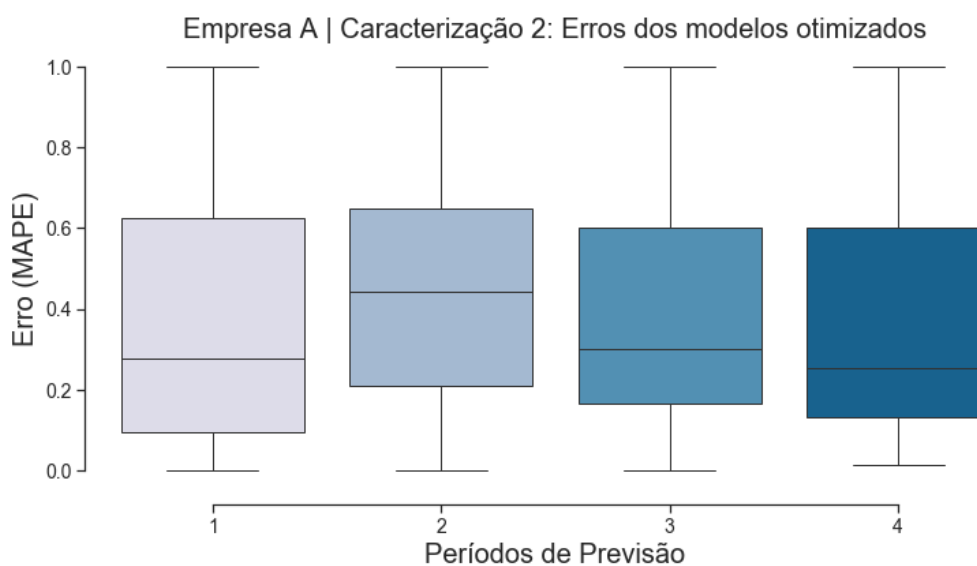


Figura 5.17 – Análise de erros da dos modelos otimizados considerando os dados da empresa A e o problema 2

Fonte: Próprio autor

Para investigar as causas do aumento de erros foi feita uma análise do percentual de vezes que os modelos treinados foram capazes de acertar o *bin* das vendas observadas. Esse percentual foi chamado de *hit* e é apresentado na Tabela 5.10.

Tabela 5.10 – Taxa de hit para o problema 2 considerando os dados da empresa A

Horizonte	Hit
1	24.36%
2	34.62%
3	24.36%
4	34.62%

Fonte: Próprio autor

Podem ser observadas taxas de *hit* baixas, entre 25 e 35%, indicando que o problema não é bem caracterizado como um problema de classificação, ou seja, o espaço de saída do problema não possui uma separação que pode ser mapeada pelos modelos de aprendizado computacional.

A Figura 5.18 apresenta a distribuição dos erros percentuais dos modelos otimizados para os dados da empresa A considerando a segunda caracterização do problema. É possível observar que a distribuição de erros possui caudas mais pesadas em relação à primeira caracterização do problema (seção 5.3), sendo essa mais uma evidência de que o agrupamento da demanda em múltiplos de movimentação foi vantajoso.

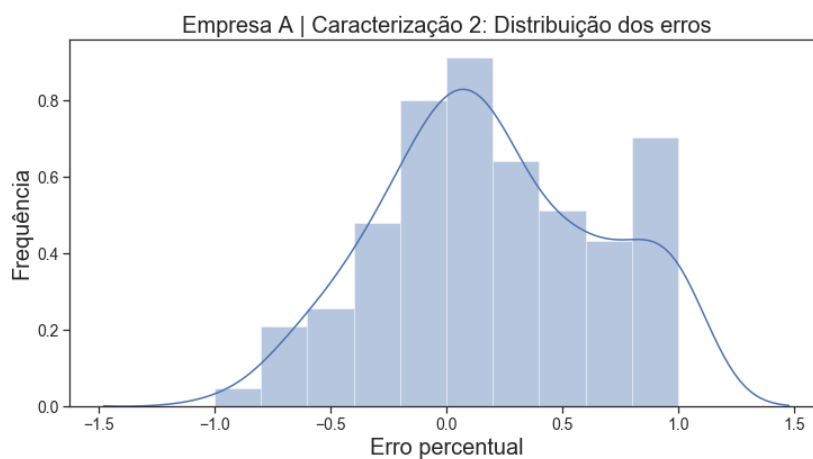


Figura 5.18 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa A e o problema 2

Fonte: Próprio autor

A Figura 5.19 apresenta os resultados obtidos após a otimização dos parâmetros dos melhores modelos para cada conjunto de dados da empresa B. Observam-se erros médios entre 30 e 40% o que é aceitável considerando os resultados de Huang *et al.* (2019), porém observam-se também altas dispersões dos erros de previsão o que confere baixa confiabilidade para as previsões feitas pelos modelos.

A Figura 5.20 apresenta a distribuição do erro percentual para os dados da empresa B e pode ser identificado que existe uma concentração de erros em torno da origem, porém existe uma concentração de erros superiores a 100% e inferiores a -100%. Isso significa que as previsões dos modelos gerados não são confiáveis para uso prático. Essa constatação é

outra evidência que reforça que a caracterização do problema considerando os múltiplos de movimentação não traz benefícios para a construção dos modelos de previsão.

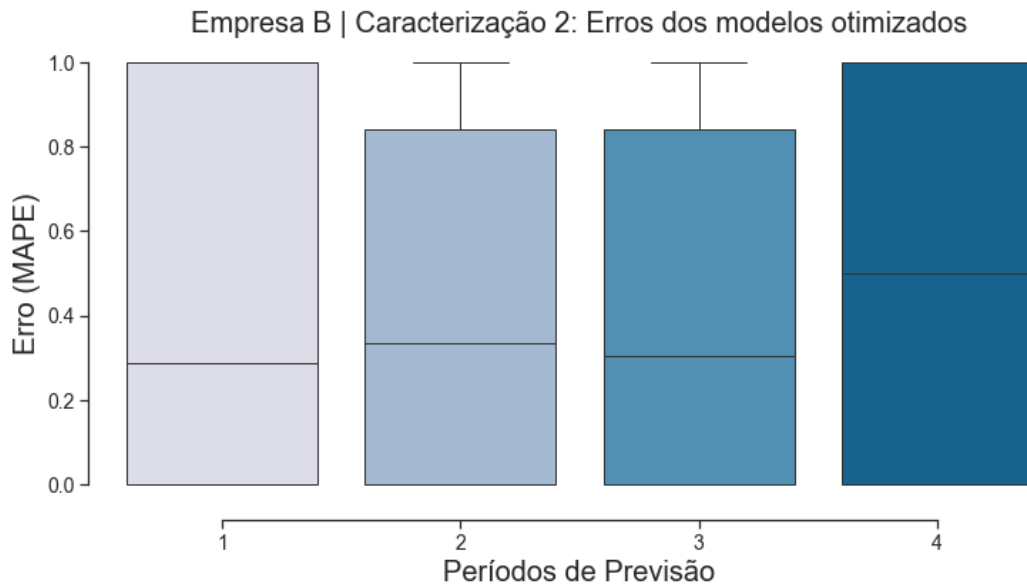


Figura 5.19 – Análise de erros da dos modelos otimizados considerando os dados da empresa B e o problema 2

Fonte: Próprio autor

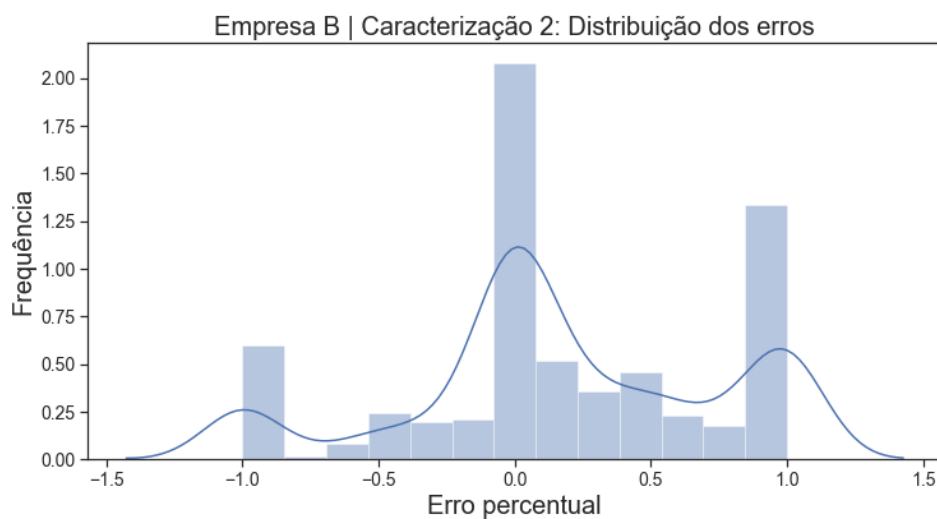


Figura 5.20 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa B e o problema 2

Fonte: Próprio autor

A Tabela 5.11 mostra o *hit* dos modelos considerando os dados da empresa B. Assim como no caso dos dados da empresa A (Tabela 5.10), podem ser observadas taxas de *hit* baixas, entre 40 e 50%. As taxas de *hit* são maiores que no caso da empresa A, mas ainda são baixas.

Tabela 5.11 – Taxa de hit para o problema 2 considerando os dados da empresa B

Horizonte	Hit
1	51.00%
2	46.00%
3	50.00%
4	39.00%

Fonte: Próprio autor

A última etapa de aplicação da metodologia é a comparação dos resultados dos modelos otimizados com modelos produzidos pela metodologia ARIMA. A Figura 5.21 apresenta uma comparação entre os modelos otimizados para os dados da empresa A e os modelos ARIMA correspondentes.

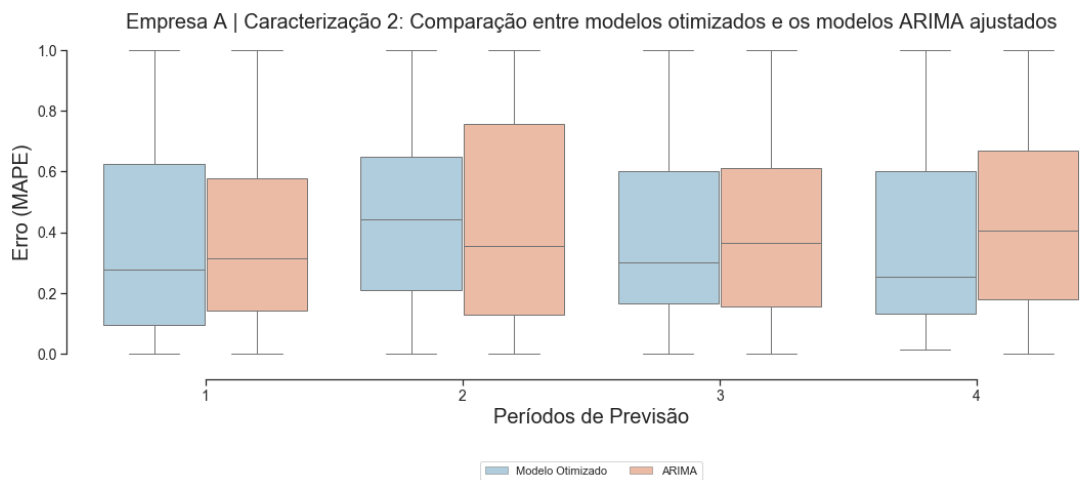


Figura 5.21 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A Problema 2

Fonte: Próprio autor

No caso do problema de previsão para cada par de produtos e lojas considerando múltiplos de movimentação para os dados da empresa A os modelos produzidos pela metodologia não apresentam resultados superiores aos modelos da classe ARIMA. Isso pode ser evidenciado pelos erros médios que são superiores para alguns horizontes e pela dispersão dos erros que também é superior em alguns casos.

A Tabela 5.12 contém os valores médios e os desvios padrão dos erros produzidos pelos modelo de aprendizado com parâmetros otimizados e pelos modelos ARIMA para os dados da empresa A para a segunda caracterização do problema. Os dados apresentados na Tabela 5.12 são os mesmo da Figura 5.21.

Tabela 5.12 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A – Problema 2

Horizonte	Melhor Modelo		ARIMA		Diferença	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
1	37,76%	31,83%	39,61%	31,11%	1,85%	(0,73%)
2	46,36%	30,05%	44,38%	34,82%	(1,98%)	4,77%
3	41,21%	31,23%	42,46%	32,16%	1,25%	0,93%
4	37,52%	31,21%	44,99%	32,67%	7,47%	1,46%

Fonte: Próprio autor

A Figura 5.22 apresenta os resultados dos modelos otimizados para os dados da empresa B comparados com modelos ARIMA. Observa-se que os resultados dos modelos otimizados foram piores que os resultados dos modelos ARIMA, tanto em termos de erros médios quanto em termos de dispersão dos erros também para a empresa B.

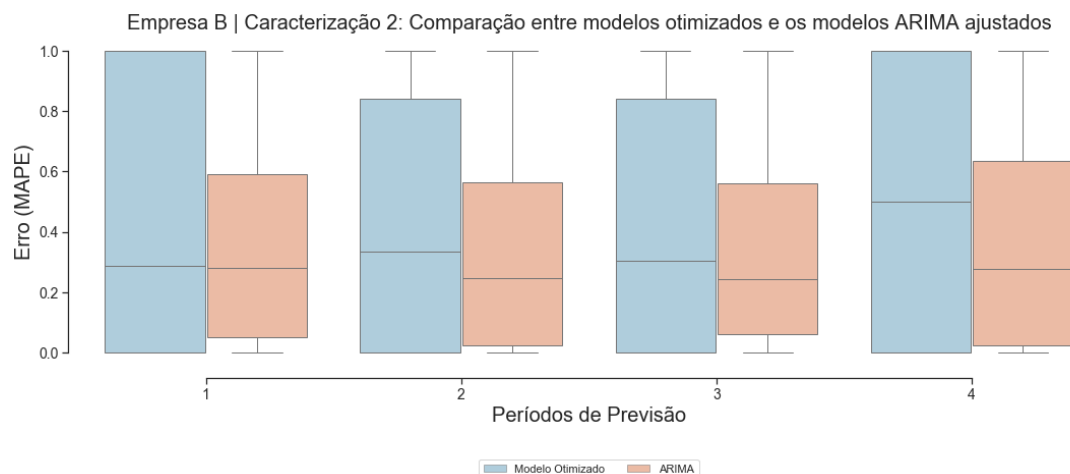


Figura 5.22 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B Problema 2

Fonte: Próprio autor

A Tabela 5.13 contém os valores médios e os desvios padrão dos erros decorrentes dos modelos de aprendizado computacional e dos modelos ARIMA para os dados da empresa B.

Tabela 5.13 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B – Problema 2

Horizonte	Melhor Modelo		ARIMA		Diferença	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
1	43,58%	41,07%	36,29%	33,41%	(7,29%)	(7,66%)
2	41,67%	39,17%	34,94%	34,14%	(6,72%)	(5,03%)
3	40,85%	39,07%	35,28%	34,06%	(5,56%)	(5,01%)
4	49,65%	44,41%	36,62%	35,08%	(13,02%)	(9,33%)

Fonte: Próprio autor

Os resultados apresentados nesta seção indicam que a metodologia não foi capaz de produzir modelos melhores que os modelos de comparação da classe ARIMA. De fato, os modelos resultantes possuem resultados inferiores em termos de erros médios e em termos de dispersão de erros. Uma análise mais detalhada mostra que as taxas de *hit* dos modelos, ou seja, a frequência com que os modelos acertam a janela do múltiplo de movimentação, são pequenas. Dado que foram considerados os inputs disponíveis de vendas, preços, estoques e datas especiais e considerando que os resultados foram inferiores ao caso que não considera a demanda em janelas, pode-se concluir que o espaço de saída, ou seja, a demanda observada não se distribui em classes com separação evidente e por isso a caracterização do problema como um problema de classificação não é vantajosa para construção de modelos de previsão de demanda desagregada no varejo.

5.5. Problema de previsão de demanda de um grupo de produtos e lojas

Essa seção descreve a aplicação da metodologia proposta para os dados da empresa A e da empresa B considerando o problema de previsão de vendas de um conjunto de produtos e lojas. Nessa caracterização, assume-se que as séries de vendas observadas dos produtos nas lojas são casos observados de um mesmo fenômeno e o problema é a previsão do valor real da demanda nos períodos futuros. Por isso as técnicas de limpeza e preparação de dados são aplicadas em cada conjunto de dados individualmente e em seguida, para cada empresa, esses dados são agrupados numa única amostra para treinamento dos modelos de

aprendizado computacional. A principal finalidade do agrupamento é aumentar o tamanho da amostra de dados utilizada para treinar um modelo de aprendizado computacional individualmente.

A primeira e a segunda fase da metodologia foram implementadas em uma sequência de cálculo (Figura 5.23) em que cada sequência representada por uma cor é relativa ao processamento da série de vendas de um produto em uma loja.

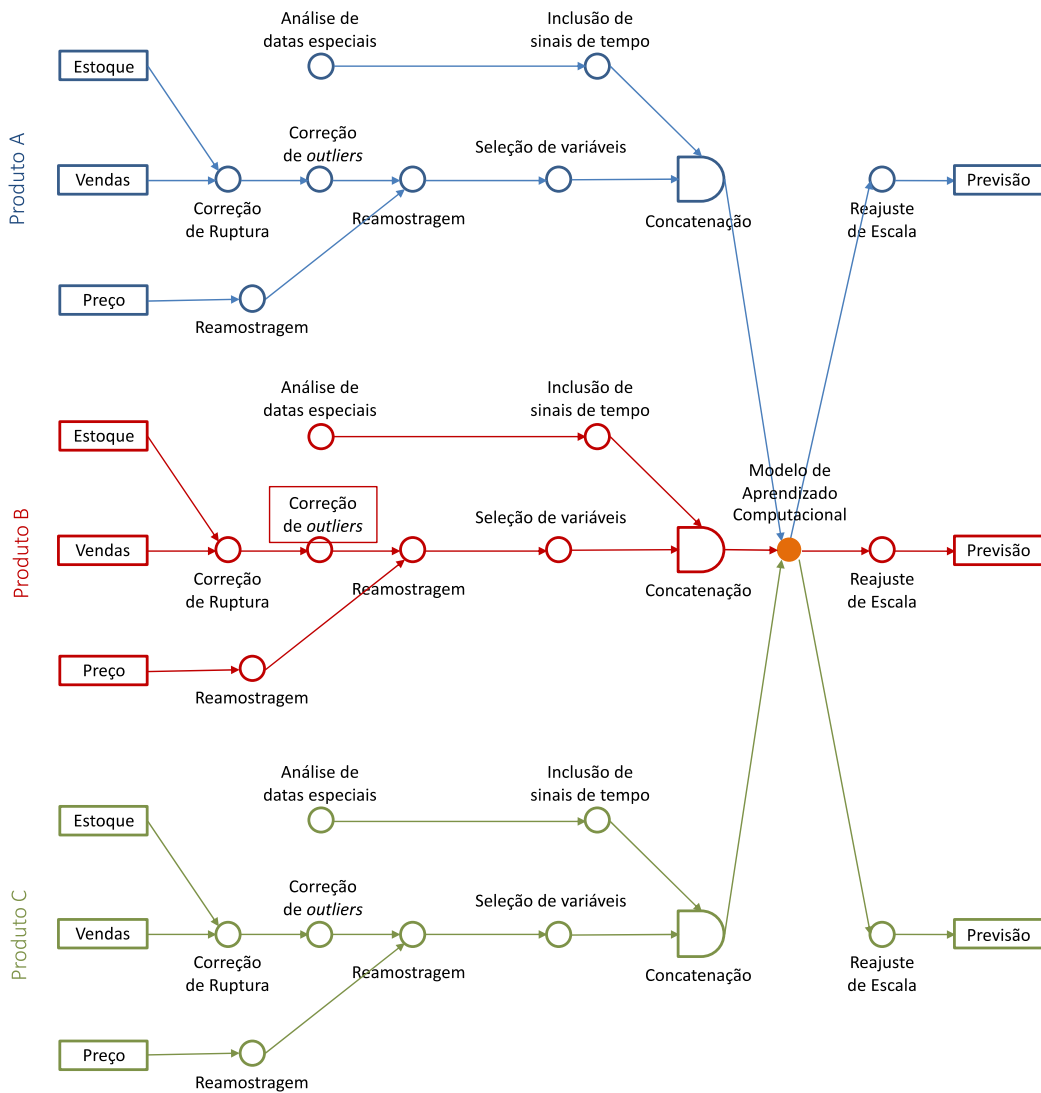


Figura 5.23 – Sequência de processamento para a caracterização 3 do problema

Fonte: Próprio autor

A sequência de cálculo é análoga à sequência da seção 5.3 que considera a previsão do valor real das vendas em períodos futuros, exceto pelo fato de que os conjuntos de dados de cada empresa são agrupados em uma única amostra para cada empresa após passarem pelos métodos de limpeza e tratamento de dados. Essa amostra agrupada é utilizada no treinamento de um único modelo de aprendizado computacional. Depois do treinamento do modelo as previsões são geradas e ajustadas para as escalas dos respectivos produtos.

De forma idêntica à seção 5.3, a escolha dos hiperparâmetros dos modelos para o teste inicial foi baseada em recomendações da literatura e resultados de testes preliminares. Foram utilizados os mesmos hiperparâmetros que na seção 5.3 (Tabela 5.4).

A Figura 5.24 apresenta os resultados da primeira e segunda fase da metodologia para os dados da empresa A. Pode-se observar que não existe um modelo dominante em todos os períodos tanto em termos de erro médio quanto em termos de dispersão do erro. Um fator interessante é que a rede neural MLP apresentou resultados muito superiores aos da seção 5.3 (Figura 5.2). Acredita-se que o fato de amostra de dados agrupada de todos os produtos conter uma quantidade maior de observações permite maior efetividade no treinamento da MLP, ou seja, permite que a MLP consiga mapear as relações entre as variáveis de entrada com a existência de um maior número de exemplos.

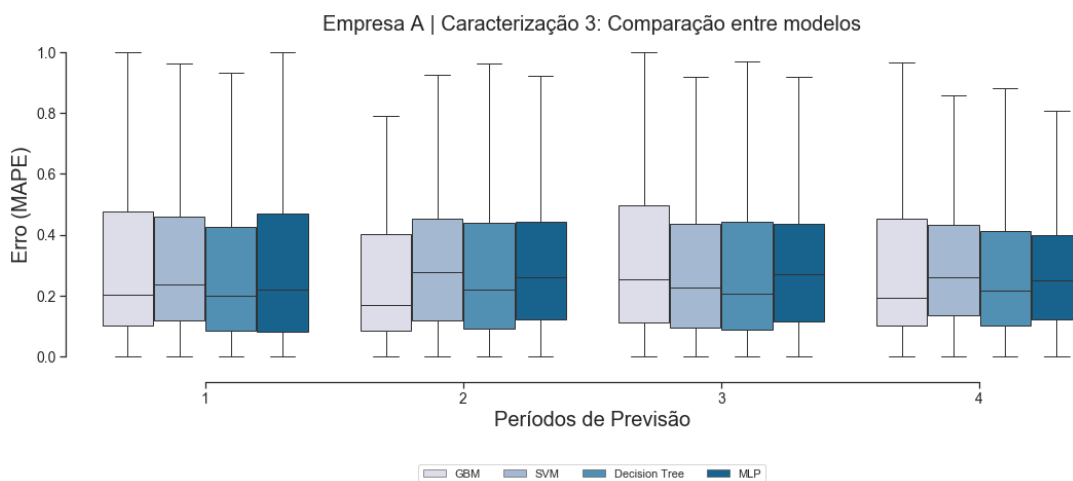


Figura 5.24 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa A considerando o problema 3

Fonte: Próprio autor

Diferentemente da caracterização do problema que considera as séries de venda isoladamente, para esta caracterização do problema que considera todo o conjunto de séries para cada empresa, é necessário escolher apenas um modelo para passar pela etapa de otimização de parâmetros.

Para a escolha do melhor modelo foi considerado o coeficiente de variação do erro médio acumulado em relação ao desvio padrão acumulado no horizonte de previsão. O erro médio acumulado é a soma dos erros médios ao longo do horizonte de planejamento. O desvio padrão acumulado é a soma do desvio padrão ao longo do horizonte de planejamento. O coeficiente de variação por sua vez é o quociente entre as duas grandezas. A Tabela 5.14 apresenta os erros médios, os desvios padrão e os coeficientes de variação no horizonte de previsão e acumulados. O critério do coeficiente de variação acumulado foi escolhido por considerar de forma combinada o erro e o desvio do erro ao longo do horizonte de previsão. Tanto o erro como o desvio ao longo do horizonte de previsão são igualmente importantes no planejamento de demanda.

Tabela 5.14 – Estatísticas dos erros dos modelos no teste preliminar – Empresa A – Problema 3

	Erro Médio				
	1	2	3	4	Acum.
gbm	29,07%	30,45%	31,18%	30,81%	121,52%
mlp	30,43%	32,24%	32,80%	31,92%	127,40%
svm	31,35%	26,84%	33,89%	32,07%	124,15%
tree	32,35%	35,17%	30,40%	32,94%	130,85%
	Desvio Padrão Do Erro				
	1	2	3	4	Acum.
gbm	26,82%	27,50%	29,21%	28,20%	111,72%
mlp	27,40%	27,18%	26,73%	27,08%	108,39%
svm	28,39%	25,62%	28,26%	29,97%	112,25%
tree	27,69%	29,42%	26,90%	26,96%	110,97%
	Coeficiente de Variação				
	1	2	3	4	Acum.
gbm	1,08	1,11	1,07	1,09	1,09
mlp	1,11	1,19	1,23	1,18	1,18
svm	1,10	1,05	1,20	1,07	1,11
tree	1,17	1,20	1,13	1,22	1,18

Fonte: Próprio autor

Pode-se notar que segundo o critério do coeficiente de variação acumulado a *Gradient Boosting Machine* (GBM) apresenta o melhor resultado com um coeficiente de 1,09, e por isso é considerada como o melhor modelo para os dados da empresa A para esta caracterização do problema.

A Figura 5.25 apresenta os resultados da primeira e segunda fase da metodologia para os dados da empresa B. Assim como no caso da empresa A, não existe um modelo dominante em todos os períodos do horizonte de previsão.

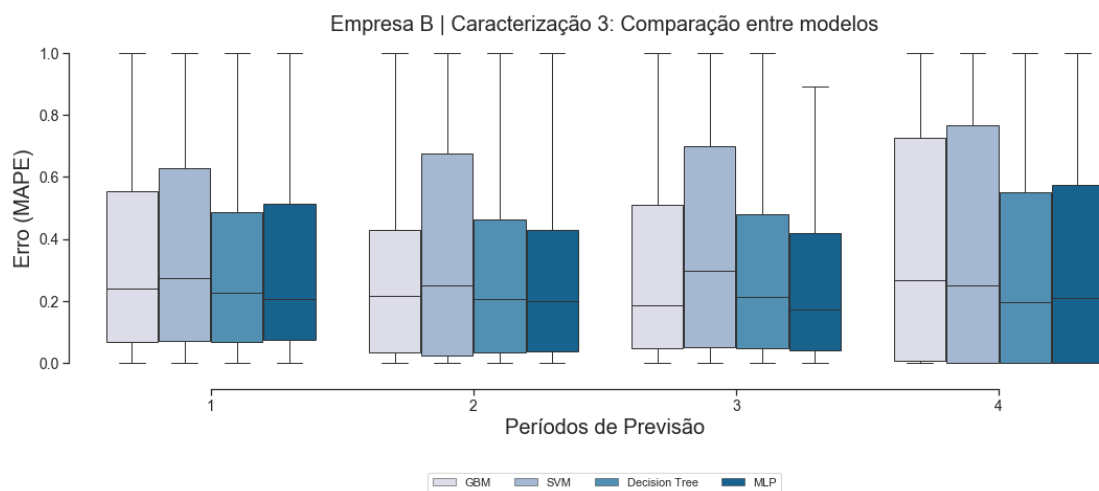


Figura 5.25 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa B considerando o problema 3

Fonte: Próprio autor

A Tabela 5.15 contém os resultados dos erros médios, desvios do erro e coeficientes de variação no horizonte e acumulados. Considerando o critério do coeficiente de variação a rede neural MLP é o melhor modelo com um coeficiente de 0,97 sendo o melhor modelo para os dados da empresa B considerando a caracterização do problema como a previsão de vendas de um grupo de produtos.

A terceira fase da metodologia compreende a busca de parâmetros para o melhor modelo do teste preliminar. No caso da empresa A, pelo critério do coeficiente de variação, o modelo que apresentou melhores resultados foi a GBM. A Tabela 5.16 apresenta o espaço de parâmetros da GBM e os valores da grade de busca.

Tabela 5.15 – Estatísticas dos erros dos modelos no teste preliminar – Empresa B – Problema 3

	Erro Médio				
	1	2	3	4	Acum.
gbm	30,81%	28,86%	29,93%	32,90%	122,51%
mlp	32,76%	28,63%	27,96%	33,92%	123,27%
svm	34,00%	29,74%	30,56%	38,98%	133,27%
tree	38,50%	37,62%	39,43%	38,23%	153,79%
	Desvio Padrão Do Erro				
	1	2	3	4	Acum.
gbm	28,89%	28,73%	28,85%	35,21%	121,68%
mlp	31,83%	29,67%	29,22%	36,41%	127,13%
svm	31,48%	29,73%	31,42%	37,83%	130,47%
tree	35,29%	36,57%	35,36%	39,42%	146,64%
	Coeficiente de Variação				
	1	2	3	4	Acum.
gbm	1,07	1,00	1,04	0,93	1,01
mlp	1,03	0,97	0,96	0,93	0,97
svm	1,08	1,00	0,97	1,03	1,02
tree	1,09	1,03	1,12	0,97	1,05

Fonte: Próprio autor

Tabela 5.16 – Grade de busca para a otimização de parâmetros – Problema 3 – Empresa A

Modelo	Parâmetro	Valores
Gradient Boosting Machine	número de estimadores	100/500/1000/2000
Gradient Boosting Machine	subamostragem	0,2/0,4/0,6
Gradient Boosting Machine	taxa de aprendizado	0,025/0,1/0,5
Gradient Boosting Machine	profundidade máxima dos estimadores	03/05/07

Fonte: Próprio autor

A Figura 5.26 apresenta os resultados obtidos com a otimização dos parâmetros para os dados da empresa A e a Figura 5.27 apresenta a distribuição dos erros percentuais dos modelos otimizados para os dados da empresa A. Os erros são próximos de 20% indicando bons resultados quando comparados com os resultados obtidos por Huang *et al.* (2019).

Além disso a dispersão dos erros também é baixa em relação aos valores médios, indicando que as previsões decorrentes do modelo otimizado são confiáveis para uso prático.

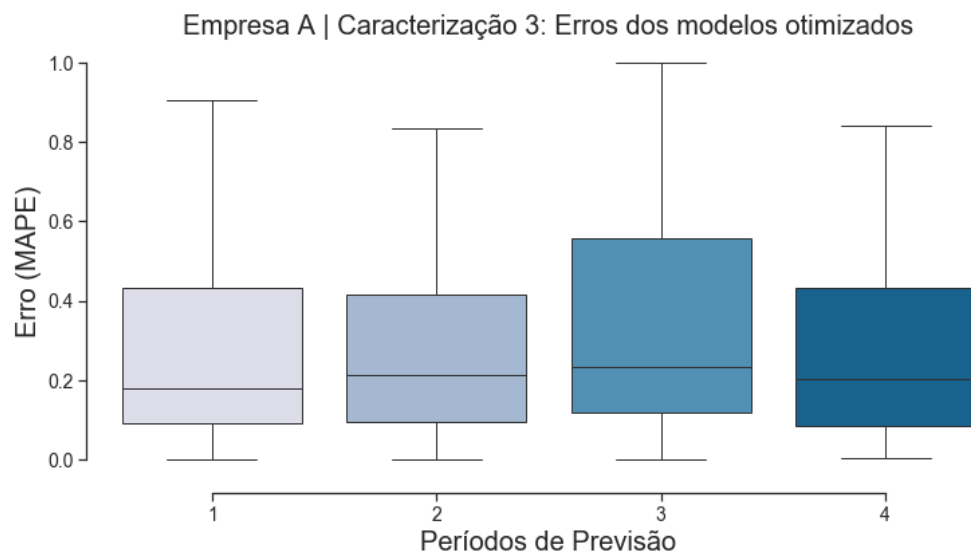


Figura 5.26 – Análise de erros da dos modelos otimizados considerando os dados da empresa A e o problema 3

Fonte: Próprio autor

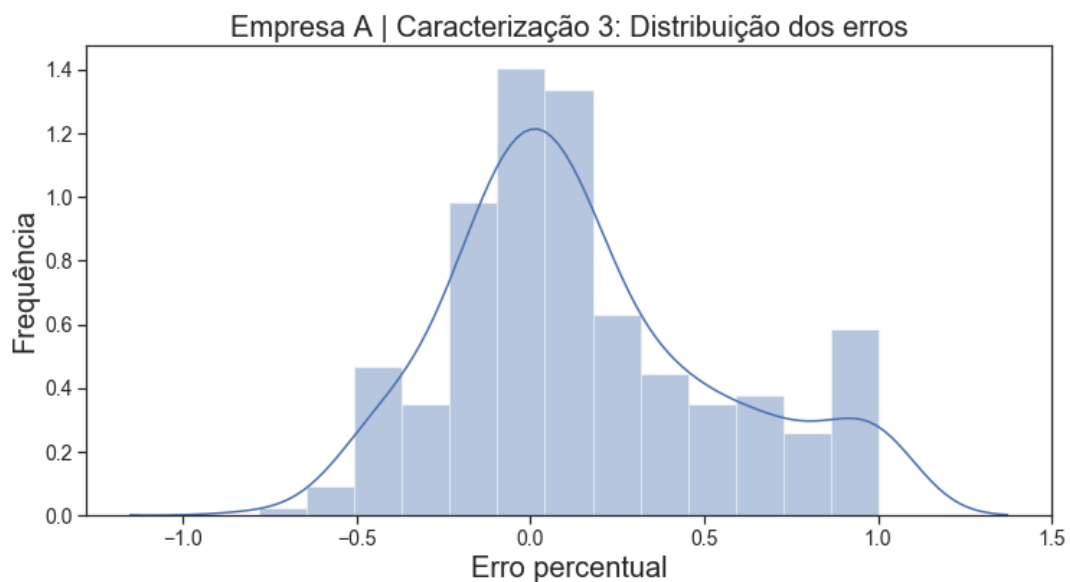


Figura 5.27 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa A e o problema 3

Fonte: Próprio autor

No caso da empresa B, o melhor modelo foi a rede neural MLP. A Tabela 5.17 apresenta o espaço de parâmetros da MLP e os valores da grade de busca.

Tabela 5.17 – Grade de busca para a otimização de parâmetros – Problema 3 – Empresa B

Modelo	Parâmetro	Valores
MLP	Ativação das camadas intermediárias	Linear/ReLU/Sigmoide
MLP	Ativação da camada de saída	Linear/ReLU/Sigmoide
MLP	Neurônios na camada intermediária	8/16/32/64
MLP	Quantidade de camadas intermediárias	1/2/3
MLP	Taxa de aprendizado	0,001/0,01/0,1

Fonte: Próprio autor

A Figura 5.28 apresenta os resultados obtidos com a otimização dos parâmetros para os conjuntos de dados da empresa B. Exceto pelo último período do horizonte de planejamento, o modelo otimizado produziu erros médios inferiores a 20% indicando bons resultados do acordo com os resultados obtidos por Huang *et al.* (2019).

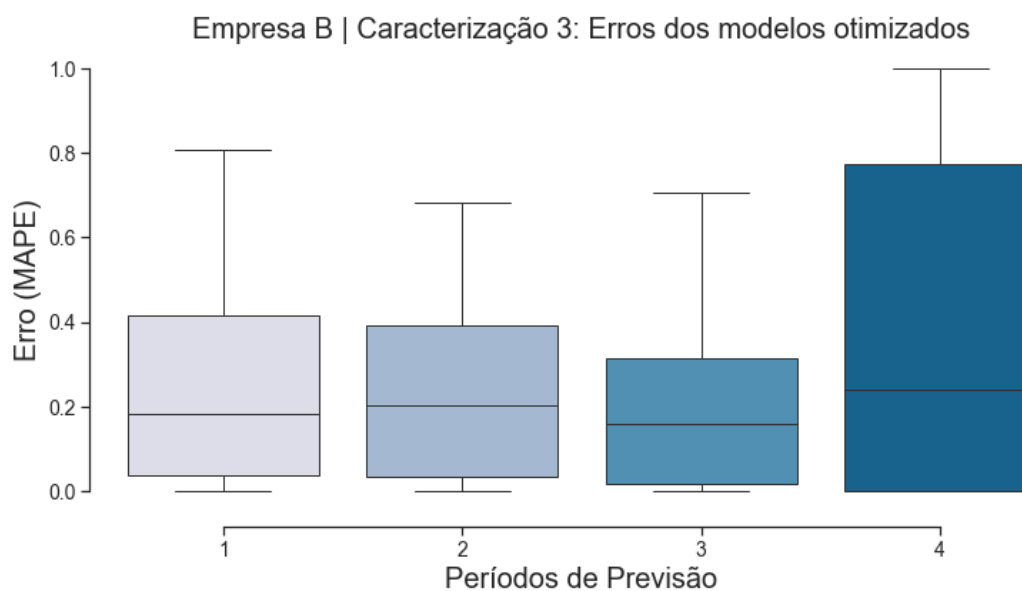


Figura 5.28 – Análise de erros da dos modelos otimizados considerando os dados da empresa B e o problema 3

Fonte: Próprio autor

A Figura 5.29 apresenta a distribuição dos erros percentuais dos modelos otimizados para os dados da empresa B.

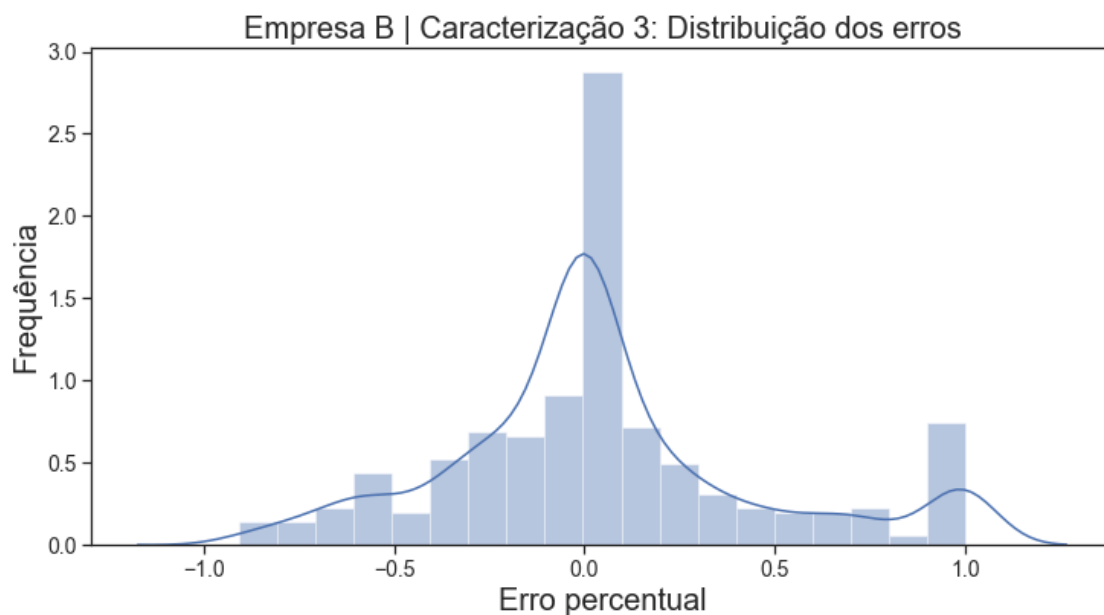


Figura 5.29 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa B e o problema 3

Fonte: Próprio autor

A distribuição de erros dos modelos otimizados no caso de ambas as empresas é centrada na origem, o que indica baixo viés de estimação. Além disso, no caso da empresa B a distribuição de erros possui caudas muito atenuadas, ou seja, há um pico de concentração na origem, o que indica um bom desempenho do modelo de previsão (erros próximos de zero e baixa variância de erros).

A última etapa é a comparação dos resultados dos modelos otimizados com modelos produzidos pela metodologia ARIMA ajustada pelo método de força bruta. A Figura 5.30 apresenta os erros do modelo com parâmetros otimizados para os dados da empresa A em comparação com os erros dos modelos ARIMA ajustados com o método de força bruta. Pode-se observar que em todos os períodos do horizonte de previsão o modelo otimizado produziu melhores resultados tanto em termos de erros médios quanto em termos de dispersão dos erros (os erros médios são menores e a dispersão de erros é igualmente menor).

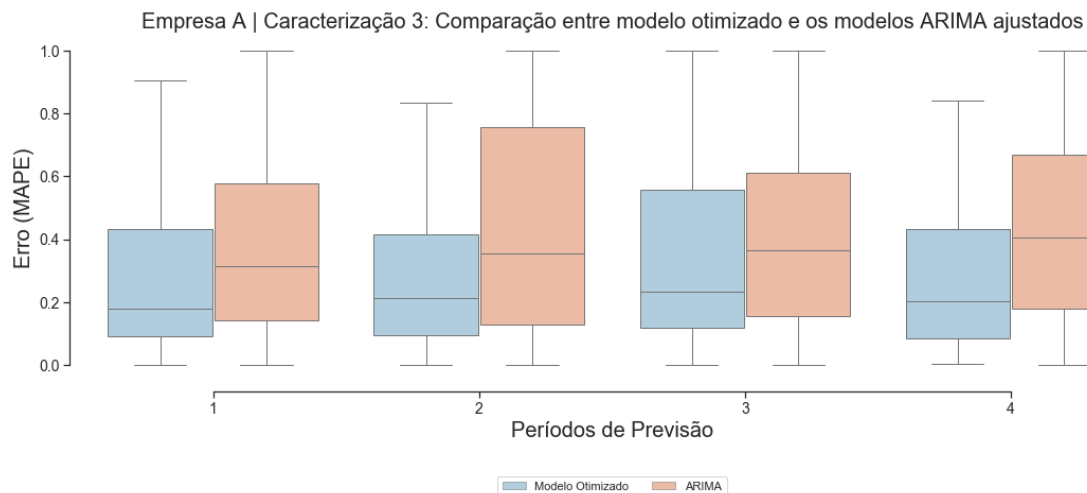


Figura 5.30 – Resultados comparados entre o modelo otimizado e modelos ARIMA – Empresa A Problema 3

Fonte: Próprio autor

A Tabela 5.18 apresenta os erros comparados do modelo com parâmetros otimizados e os erros dos modelos ARIMA para a empresa A. As diferenças dos erros médios do modelo otimizado e dos modelos ARIMA são significativas indicando que a metodologia produziu modelos com capacidade de previsão superior para os dados da empresa A neste caso da terceira definição do problema. A redução de erros média dentro do horizonte de previsão é de 11,88%.

Tabela 5.18 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A – Problema 3

Horizonte	Melhor Modelo		ARIMA		Diferença	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
1	28,40%	27,27%	39,61%	31,11%	11,21%	3,83%
2	29,46%	26,35%	44,38%	34,82%	14,92%	8,47%
3	34,29%	29,75%	42,46%	32,16%	8,18%	2,40%
4	31,78%	30,94%	44,99%	32,67%	13,21%	1,73%

Fonte: Próprio autor

A Figura 5.31 apresenta os erros do modelo com parâmetros otimizados para os dados da empresa B. Exceto pelo último período do horizonte de previsão, o modelo otimizado produziu resultados melhores em termos de erros médios e de dispersão de erros. Os resultados detalhados da comparação entre o modelo otimizado e os modelos ARIMA

encontram-se na Tabela 5.19 que comprova a efetividade da metodologia no caso desta caracterização do problema para os dados da empresa B. A redução de erros média dentro do horizonte de previsão é de 7,22%.

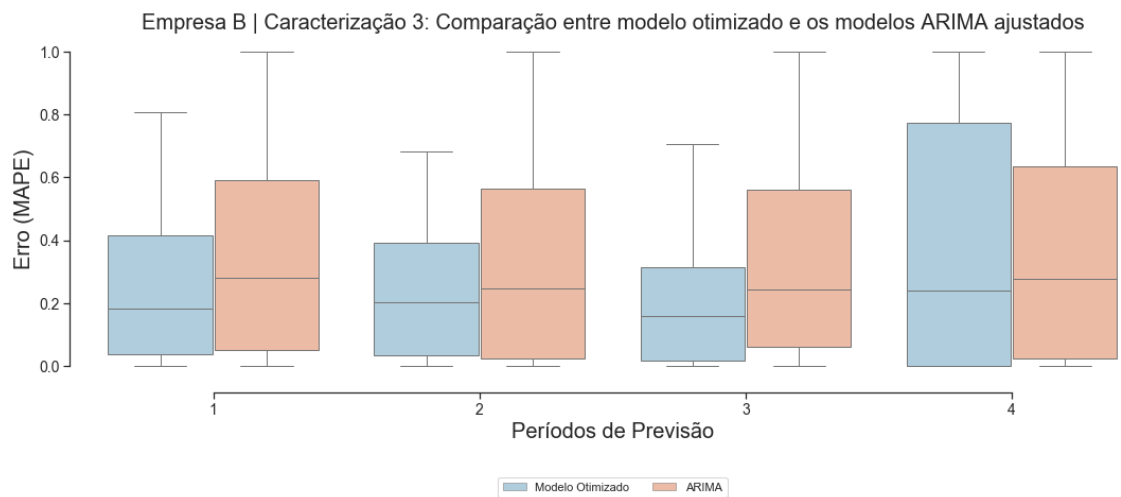


Figura 5.31 – Resultados comparados entre o modelo otimizado e modelos ARIMA – Empresa B Problema 3

Fonte: Próprio autor

Tabela 5.19 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B – Problema 3

Horizonte	Melhor Modelo		ARIMA		Diferença	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
1	26,29%	26,58%	36,29%	33,41%	10,00%	6,83%
2	24,69%	23,63%	34,94%	34,14%	10,25%	10,50%
3	23,38%	26,85%	35,28%	34,06%	11,91%	7,21%
4	39,92%	40,29%	36,62%	35,08%	(3,30%)	(5,20%)

Fonte: Próprio autor

Em síntese, em comparação com a caracterização do problema que considera as séries isoladamente, os resultados decorrentes desta aplicação prática indicam que essa caracterização que considera as séries conjuntamente produziu melhores resultados em termos de erros médios e dispersão de erros. Isso permite afirmar que considerar que todas as séries de venda observadas fazem parte de um mesmo fenômeno a ser modelado pelos modelos de aprendizado computacional é uma premissa válida que aumenta a quantidade

de exemplos na amostra de treinamento dos modelos de aprendizado computacional e resulta em erros menores e com menor variância.

5.6. Problema de previsão de múltiplos de movimentação de um grupo de produtos e lojas

Essa seção descreve a aplicação da metodologia para os dados das empresas A e B considerando o problema de previsão de múltiplos de movimentação de um conjunto de produtos e lojas. Assim como na seção 5.5, os conjuntos de dados de cada empresa foram combinados num único conjunto de dados para treinamento dos modelos. No entanto, neste caso há um agrupamento dos dados de vários produtos e lojas para o treinamento dos modelos de aprendizado computacional.

A Figura 5.32 apresenta a sequência de cálculo que implementa a primeira e a segunda etapa da metodologia. A sequência é análoga à sequência da segunda caracterização do problema que considera múltiplos de movimentação para caracterizar a demanda a ser prevista (seção 5.4) com a diferença que nesse caso há um agrupamento dos dados de vários produtos e lojas para o treinamento dos modelos de aprendizado computacional.

De forma idêntica à seção 5.4, a escolha dos hiperparâmetros dos modelos para o teste inicial foi baseada em recomendações da literatura e resultados de testes preliminares. Foram utilizados os mesmos hiperparâmetros da Tabela 5.8.

A Figura 5.33 apresenta os resultados da primeira e segunda fase da metodologia para os dados da empresa A. Assim como na seção 5.4, não existe um modelo dominante em todos os períodos tanto em termos de erro médio quanto em termos de dispersão do erro. Também de forma similar à seção 5.4, a rede neural MLP apresentou resultados muito superiores ao da seção 5.3 (Figura 5.2). Para os casos em que o problema foi caracterizado considerando a demanda representada pelas janelas dos múltiplos de movimentação, a MLP apresentou melhores resultados em comparação aos casos em que o problema foi caracterizado considerando o valor numérico da demanda. Isso indica que a MLP é um modelo com melhor desempenho caso o problema seja definido como um problema de classificação e não um problema de regressão.

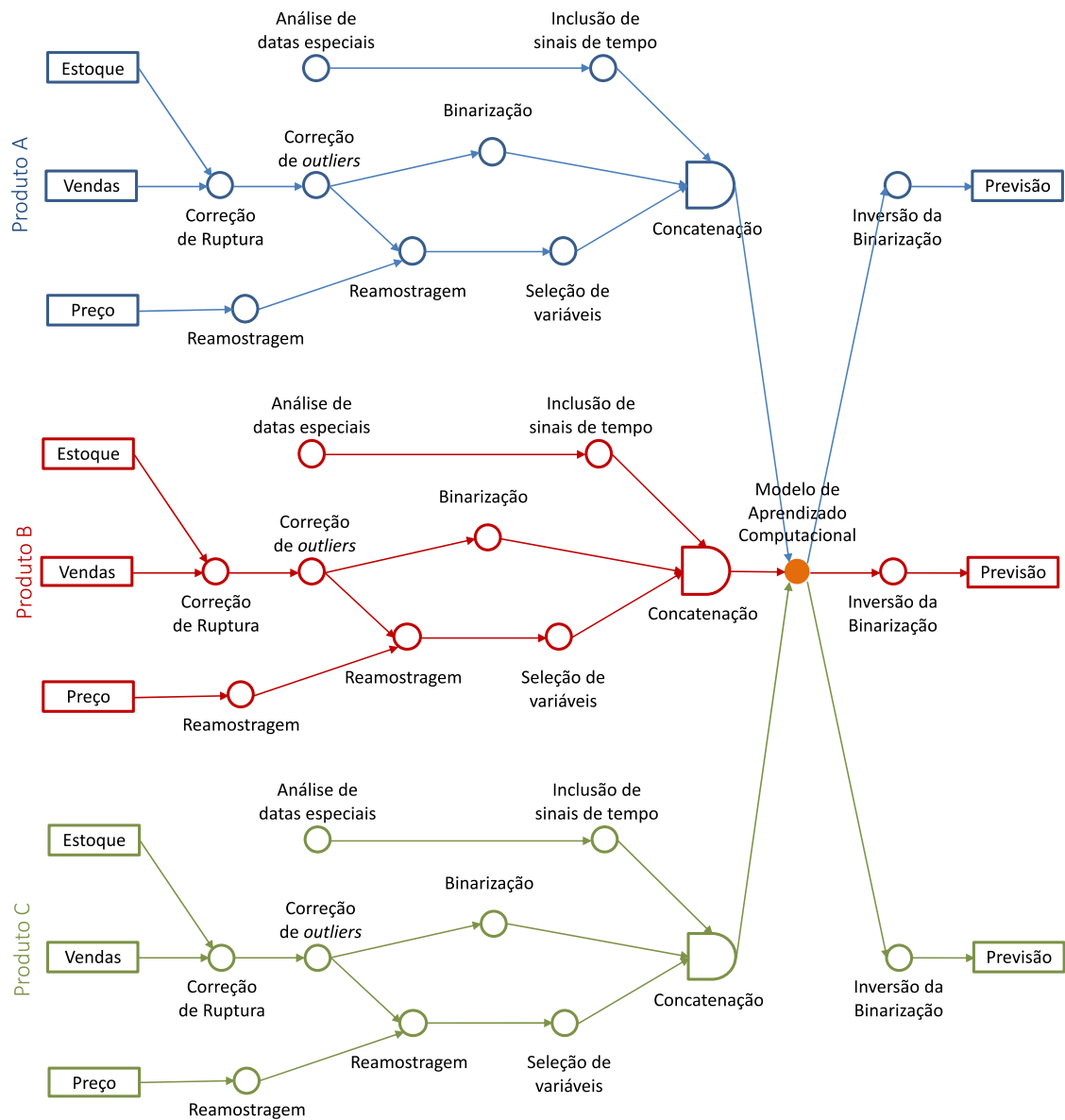


Figura 5.32 – Sequência de processamento para a caracterização 4 do problema

Fonte: Próprio autor

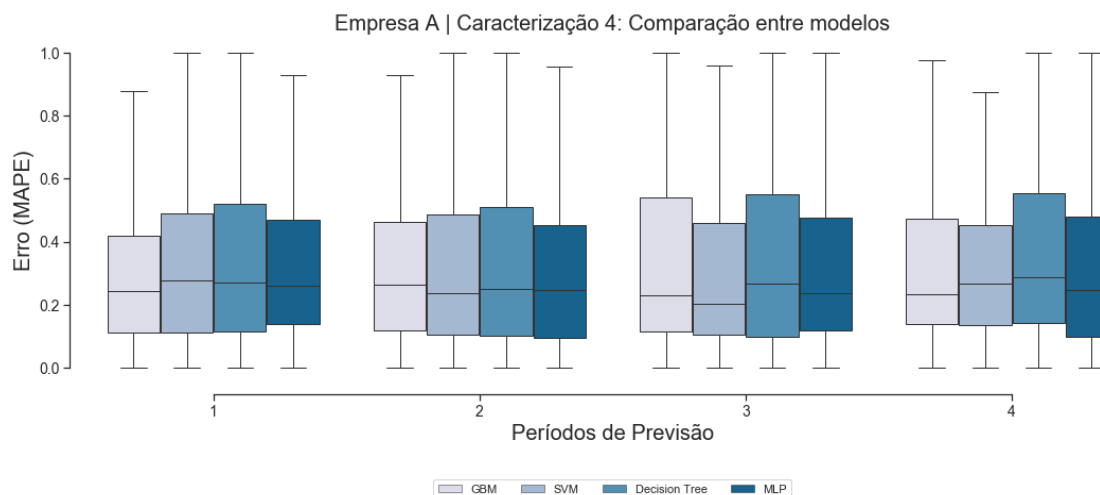


Figura 5.33 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa A considerando o problema 4

Fonte: Próprio autor

De forma idêntica à seção 5.5, para a escolha do melhor modelo foi considerado o coeficiente de variação do erro médio acumulado em relação ao desvio padrão acumulado no horizonte de previsão. A Tabela 5.20 apresenta os erros médios, os desvios padrão e os coeficientes de variação no horizonte de previsão e acumulados.

Pode-se observar que a MLP é o modelo que apresenta o menor coeficiente de variação acumulado (1,14) para os dados da empresa A considerando a definição do problema com as séries conjuntas e a demanda definida em janelas. Por isso pode ser considerado o melhor modelo para este caso.

A Figura 5.34 apresenta os resultados da primeira e segunda fase da metodologia para os dados da empresa B. Assim como no caso da empresa A, não existe um modelo dominante. Além disso, nota-se que há uma dispersão muito grande de erros em todos os modelos, independentemente dos erros médios. Isso indica um baixo desempenho considerando esta definição do problema e dados da empresa B, independente do tipo de modelo de aprendizado computacional.

Tabela 5.20 – Estatísticas dos erros dos modelos no teste preliminar – Empresa A – Problema 4

	Erro Médio				
	1	2	3	4	Acum.
gbm	35,86%	34,95%	35,70%	38,48%	144,98%
mlp	35,13%	33,14%	33,40%	33,15%	134,82%
svm	33,37%	33,08%	34,43%	35,38%	136,26%
tree	35,59%	34,94%	31,94%	32,81%	135,28%
	Desvio Padrão Do Erro				
	1	2	3	4	Acum.
gbm	29,99%	30,50%	31,06%	31,15%	122,70%
mlp	29,38%	30,00%	29,10%	29,60%	118,07%
svm	30,23%	27,28%	30,09%	30,28%	117,88%
tree	29,89%	30,88%	28,76%	26,27%	115,80%
	Coeficiente de Variação				
	1	2	3	4	Acum.
gbm	1,20	1,15	1,15	1,24	1,18
mlp	1,20	1,10	1,15	1,12	1,14
svm	1,10	1,21	1,14	1,17	1,16
tree	1,19	1,13	1,11	1,25	1,17

Fonte: Próprio autor

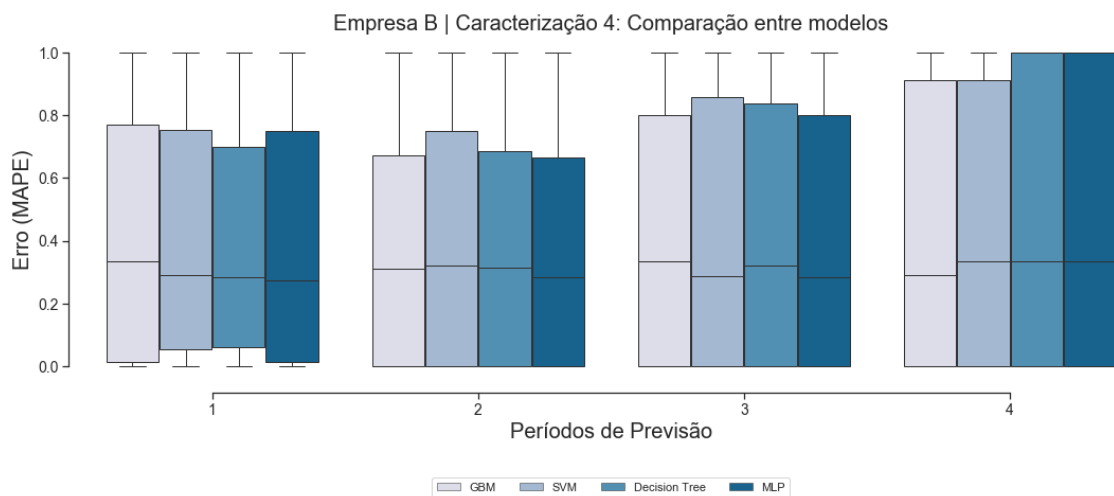


Figura 5.34 – Análise de erros da primeira e segunda fase da metodologia para os dados da empresa B considerando o problema 4

Fonte: Próprio autor

A Tabela 5.21 contém os resultados dos erros médios, desvios do erro e coeficientes de variação no horizonte e acumulados para os dados da empresa B. Considerando o critério do coeficiente de variação a MLP é o melhor modelo com um coeficiente de 1,05 sendo o melhor modelo para os dados da empresa B.

Tabela 5.21 – Estatísticas dos erros dos modelos no teste preliminar – Empresa B – Problema 4

	Erro Médio				
	1	2	3	4	Acum.
gbm	40,51%	40,08%	41,44%	42,55%	164,58%
mlp	40,87%	38,52%	39,48%	42,64%	161,52%
svm	42,48%	39,89%	41,53%	41,39%	165,29%
tree	41,63%	40,49%	41,53%	43,14%	166,78%
	Desvio Padrão Do Erro				
	1	2	3	4	Acum.
gbm	36,89%	37,23%	39,15%	40,88%	154,16%
mlp	37,93%	36,85%	38,56%	40,39%	153,73%
svm	38,11%	37,39%	38,44%	40,62%	154,55%
tree	37,53%	37,98%	39,89%	40,49%	155,90%
	Coeficiente de Variação				
	1	2	3	4	Acum.
gbm	1,10	1,08	1,06	1,04	1,07
mlp	1,08	1,05	1,02	1,06	1,05
svm	1,11	1,07	1,08	1,02	1,07
tree	1,11	1,07	1,04	1,07	1,07

Fonte: Próprio autor

A terceira fase da metodologia compreende a otimização dos parâmetros para os melhores modelos de ambas as empresas. Assim como no caso do problema que considera a previsão de vendas de um conjunto de produtos (seção 5.5), para esta definição do problema quatro apenas um modelo é escolhido para ser otimizado. No caso da empresa A o modelo que apresentou melhores resultados foi a rede neural MLP. A Tabela 5.22 apresenta o espaço de parâmetros da MLP e os valores da grade de busca. Uma vez que o problema de previsão dos múltiplos de movimentação é problema de classificação, a função *softmax* é a única adequada para o problema pois realiza a previsão da probabilidade de a venda ocorrer dentro de cada janela de múltiplo de movimentação. Por isso não faz sentido variar a função de ativação da camada de saída.

Tabela 5.22 – Grade de busca para a otimização de parâmetros – Problema 4 – Empresa A

Modelo	Parâmetro	Valores
MLP	Taxa de aprendizado	0,1/0,01/0,001
MLP	Neurônios na camada intermediária	8/16/32/64
MLP	Quantidade de camadas intermediárias	1/2/3
MLP	Ativação das camadas intermediárias	Linear/ReLU/Sigmoide

Fonte: Próprio autor

A Figura 5.35 apresenta os resultados obtidos com a otimização dos parâmetros para os conjuntos de dados da empresa A e a Figura 5.36 apresenta a distribuição dos erros percentuais dos modelos otimizados para os dados da empresa A. Os erros médios são próximos de 20% indicando bons resultados de acordo com os resultados de Huang *et al.* (2019). Além disso as dispersões de erro são pequenas, o que indica que o modelo é confiável para realizar as previsões para os quatro períodos do horizonte de previsão.

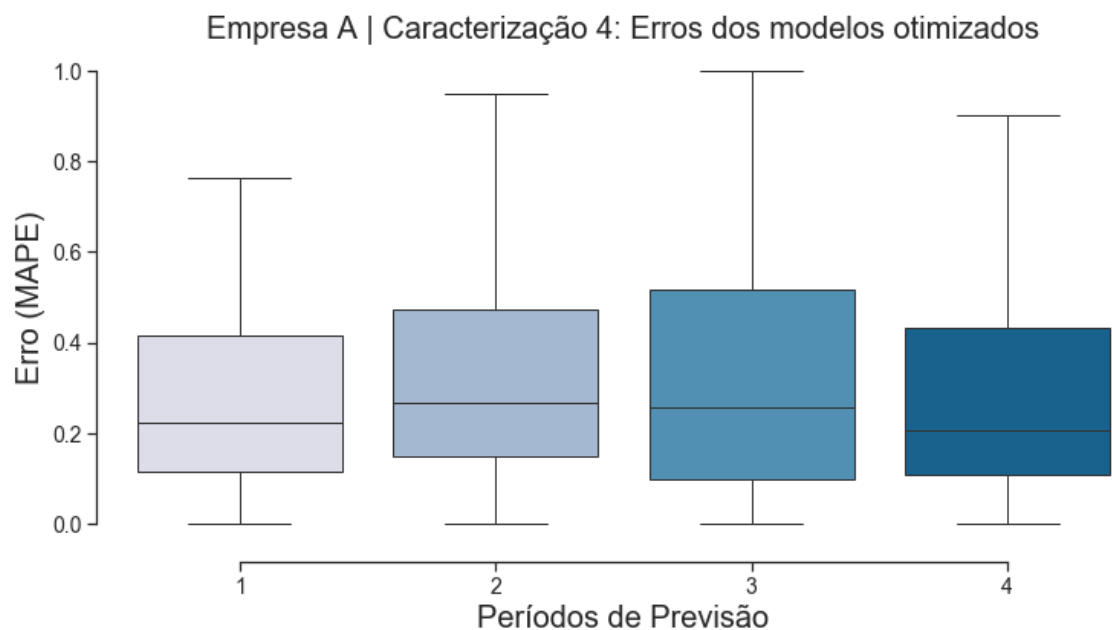


Figura 5.35 – Análise de erros da dos modelos otimizados considerando os dados da empresa A e o problema 4

Fonte: Próprio autor

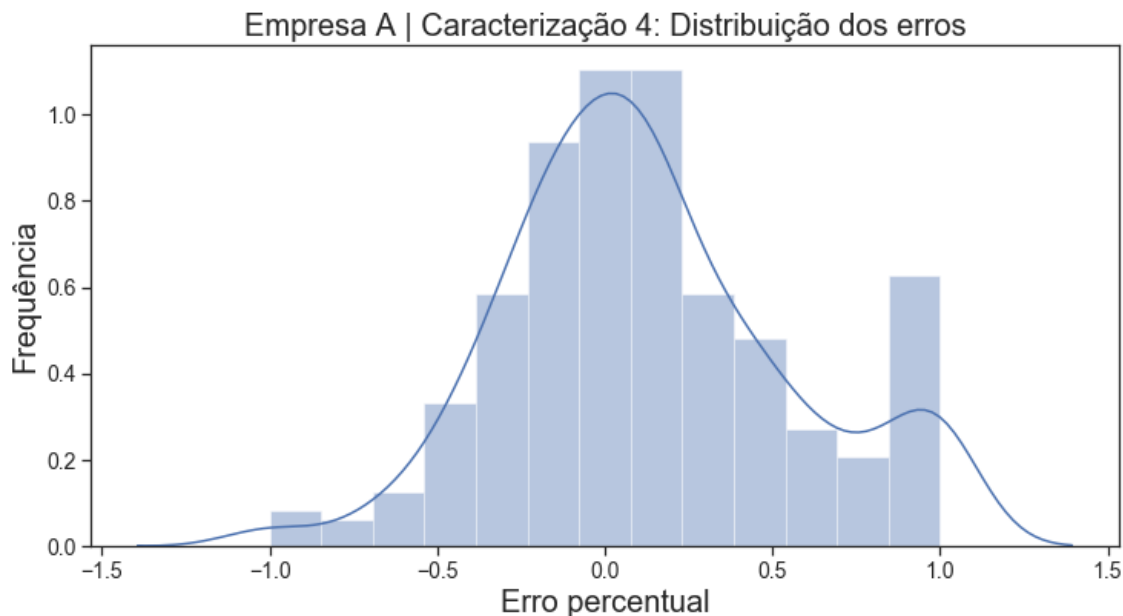


Figura 5.36 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa A e o problema 4

Fonte: Próprio autor

Assim como no caso do problema que considera as séries isoladamente e a previsão em múltiplos de movimentação (seção 5.4), pode-se medir com que precisão o modelo foi capaz de determinar a janela de vendas do produto (*hit* do modelo). A Tabela 5.23 mostra o *hit* do modelo considerando os dados da empresa A. Podem ser observadas taxas de *hit* baixas, entre 28 e 42%, porém maiores em comparação ao problema dois (Tabela 5.10). Isso é mais um indicativo de que a consideração das séries agrupadas traz um benefício para a construção dos modelos de previsão de demanda.

Tabela 5.23 – Taxa de hit para o problema 4 considerando os dados da empresa A

Horizonte	<i>Hit</i>
1	28.20%
2	42.31%
3	38.46%
4	32.05%

Fonte: Próprio autor

No caso da empresa B, o melhor modelo também foi a rede neural MLP. A grade de parâmetros considerada na otimização é a mesma que da empresa A (Tabela 5.22).

A Figura 5.37 apresenta os resultados obtidos com a otimização dos parâmetros para os dados da empresa B e a Figura 5.38 apresenta a distribuição dos erros percentuais dos modelos otimizados para os dados da empresa B.

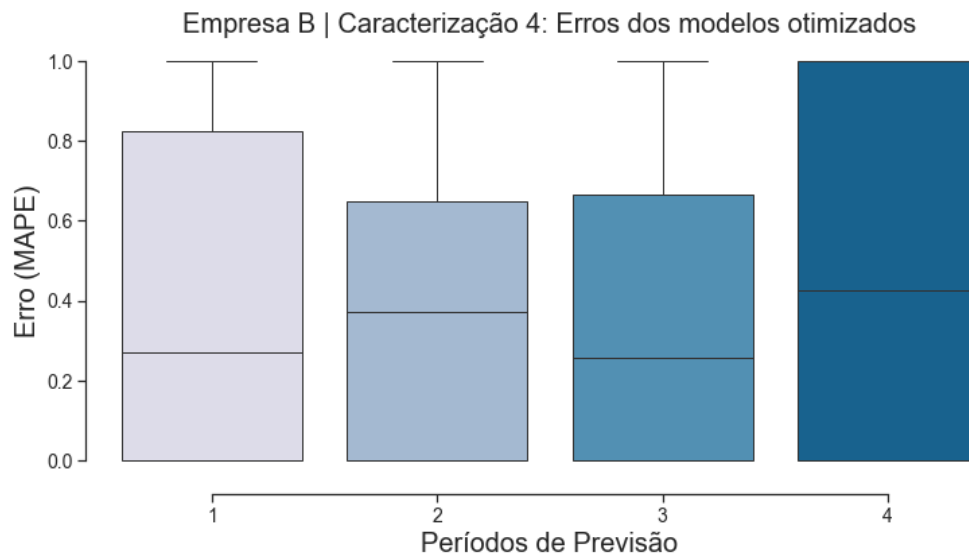


Figura 5.37 – Análise de erros da dos modelos otimizados considerando os dados da empresa B e o problema 4

Fonte: Próprio autor

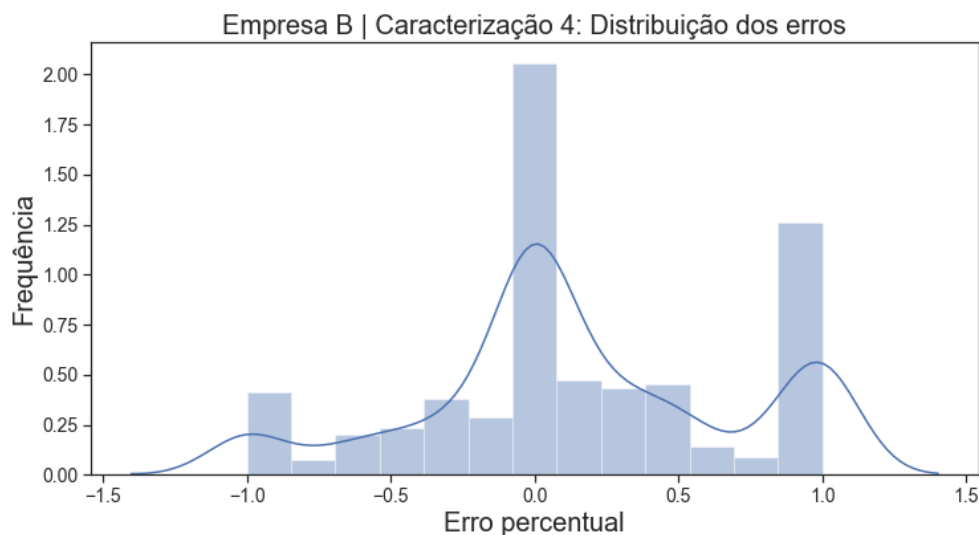


Figura 5.38 – Análise da distribuição dos erros percentuais da dos modelos otimizados considerando os dados da empresa B e o problema 4

Fonte: Próprio autor

A distribuição de erros para o caso das duas empresas é centrada na origem, indicando baixo viés de estimação. No caso da empresa B, as caudas da distribuição de erros têm acúmulos nas extremidades, indicando um desempenho ruim na tarefa de previsão, ou seja, alta variância dos erros e pouca confiabilidade nas previsões.

A Tabela 5.24 apresenta a taxa de *hit* do modelo para os dados da empresa B. Observam-se taxas de *hit* maiores que no caso da empresa A, mas ainda baixas. As baixas taxas de hits indicam que o espaço da variável de saída, no caso a janela de venda de um produto, não se distribui em classes com separação evidente. Isso indica que o problema não é bem representado como um problema de classificação.

Tabela 5.24 – Taxa de hit para o problema 4 considerando os dados da empresa B

Horizonte	Hit
1	43.33%
2	38.89%
3	46.67%
4	37,78%

Fonte: Próprio autor

A última etapa da metodologia é a comparação com modelos ARIMA. A Figura 5.39 apresenta os erros do modelo com parâmetros otimizados para os dados da empresa A em comparação com os erros dos modelos ARIMA ajustados com o método de força bruta. Observa-se que em todos os períodos do horizonte de previsão o modelo otimizado produziu melhores resultados em termos de erros médios e em termos de dispersão dos erros.

A Tabela 5.25 apresenta os erros comparados do modelo com parâmetros otimizados e os erros dos modelos ARIMA para a empresa A. As diferenças dos erros médios do modelo otimizado e dos modelos ARIMA são significativas indicando que a metodologia produziu modelos com capacidade de previsão superior no caso da definição do problema de previsão como a previsão de múltiplos de movimentação para um conjunto de produtos e lojas para os dados da empresa A.

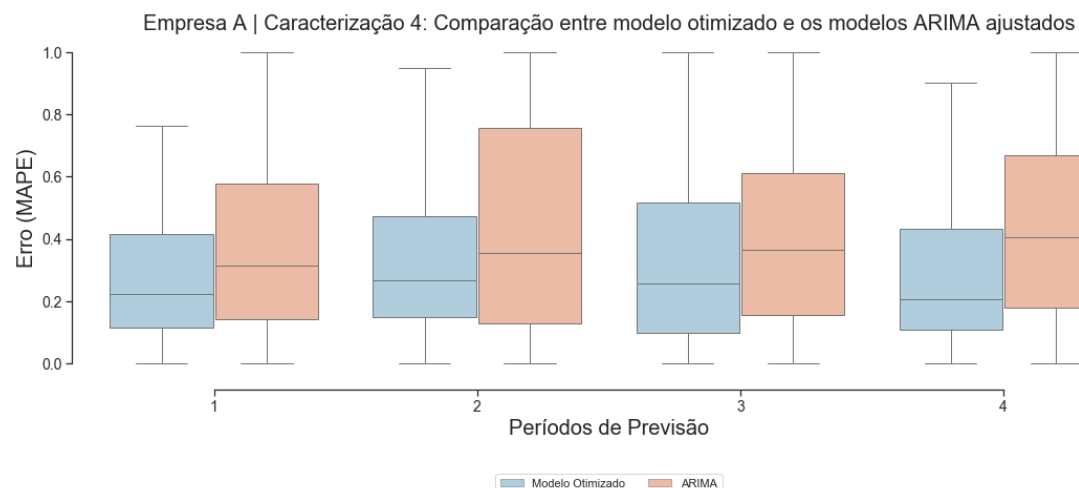


Figura 5.39 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A Problema 4

Fonte: Próprio autor

Tabela 5.25 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa A – Problema 4

Horizonte	Melhor Modelo		ARIMA		Diferença	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
1	32,01%	28,37%	39,61%	31,11%	7,60%	2,73%
2	35,87%	27,90%	44,38%	34,82%	8,51%	6,92%
3	34,30%	30,31%	42,46%	32,16%	8,17%	1,84%
4	33,56%	31,58%	44,99%	32,67%	11,43%	1,09%

Fonte: Próprio autor

A Figura 5.40 apresenta os erros do modelo com parâmetros otimizados para os dados da empresa B. Em todos os períodos do horizonte de planejamento o modelo otimizado apresentou resultados de erros médios piores que o modelo ARIMA. O modelo otimizado também produziu dispersões maiores de erros.

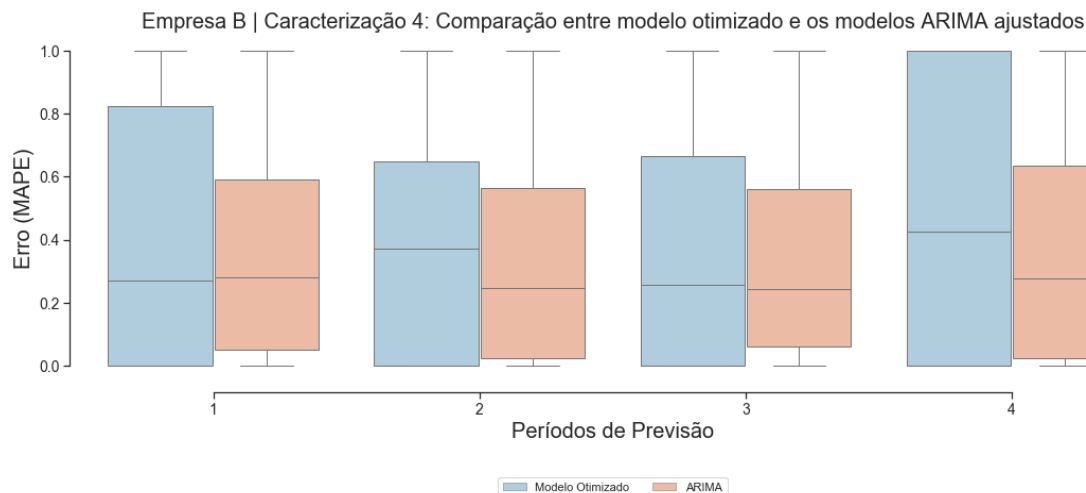


Figura 5.40 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B Problema 4

Fonte: Próprio autor

A Tabela 5.26 apresenta os erros comparados do modelo com parâmetros otimizados e os erros dos modelos ARIMA para a empresa B. Em todos os casos os resultados do modelo otimizado foram substancialmente inferiores tanto em termos de erros médios quanto em termos de dispersão de erros.

Tabela 5.26 – Resultados comparados entre os modelos otimizados e modelos ARIMA – Empresa B – Problema 4

Horizonte	Melhor Modelo		ARIMA		Diferença	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
1	40,11%	38,46%	36,29%	33,41%	(3,82%)	(5,05%)
2	40,88%	37,14%	34,94%	34,14%	(5,93%)	(3,01%)
3	37,68%	38,23%	35,28%	34,06%	(2,40%)	(4,18%)
4	46,77%	43,63%	36,62%	35,08%	(10,15%)	(8,55%)

Fonte: Próprio autor

Os resultados desta caracterização do problema com as séries analisadas conjuntamente e considerando os múltiplos de movimentação não produziu melhores resultados que os modelos ARIMA em geral. No caso da empresa B os modelos produziram erros médios maiores e erros com maior variância. Em comparação com os resultados da seção 5.4 que consideram as séries isoladamente e a demanda também representada em janelas de

múltiplos de movimentação, a caracterização dessa seção produziu erros médios menores e com menor variância. Isso é uma evidência que a representação do problema em que todas as séries são exemplos de um mesmo fenômeno é válida e resulta em erros menores e menores dispersões.

5.7. Resultados comparados

Esta seção apresenta uma síntese dos resultados comparados de todas as abordagens para os dados das empresas A e B, descritos nas seções 5.3 a 5.6.

A Figura 5.41 apresenta os resultados dos modelos otimizados para as quatro definições do problema (capítulo 3) e os resultados dos modelos ARIMA ajustados por força bruta considerando os dados da empresa A. O primeiro conjunto de *box plots*, representado pela legenda “Caracterização 1”, contém os resultados relativos à caracterização do problema que considera as séries isoladamente e a demanda como uma variável numérica. O segundo conjunto de *box plots*, representado pela legenda “Caracterização 2”, contém os resultados relativos à caracterização do problema que considera as séries isoladamente e a demanda como variáveis discretas representando múltiplos de movimentação. O terceiro conjunto de *box plots*, representado pela legenda “Caracterização 3”, contém os resultados relativos à caracterização do problema que considera as séries conjuntamente e a demanda como uma variável numérica. O quarto conjunto de *box plots*, representado pela legenda “Caracterização 4”, contém os resultados relativos à caracterização do problema que considera as séries conjuntamente e a demanda como variáveis discretas representando múltiplos de movimentação. O quinto conjunto de *box plots*, representado pela legenda “ARIMA”, contém os resultados relativos aos modelos ARIMA ajustados pelo método de força bruta. Idealmente, um modelo superior aos modelos ARIMA deveria produzir um conjunto de *box plots* com erros médios inferiores em todos os períodos do horizonte de previsão e com dispersões menores. Pode-se observar que, exceto pela segunda caracterização (“Caracterização 2”) que não produziu erros médios inferiores no segundo período do horizonte e não produziu uma dispersão de erros menor no primeiro período do horizonte, as demais caracterizações do problema cumpriram esse requisito.

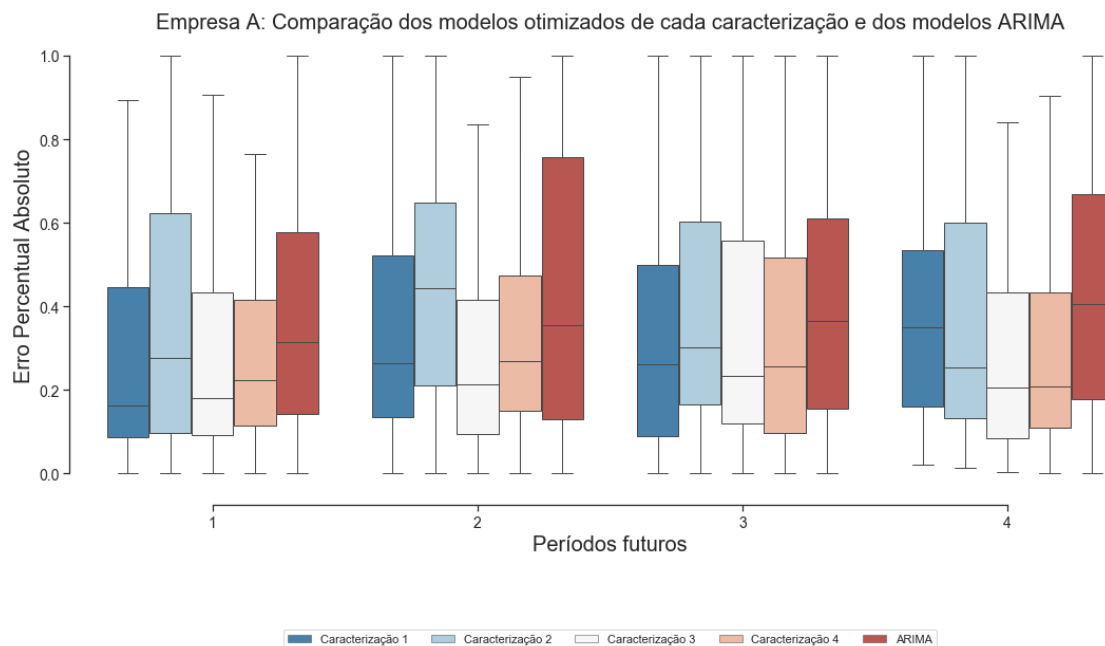


Figura 5.41 – Resultados comparados para os dados da Empresa A

Fonte: Próprio autor

Sobre a Figura 5.41 também é possível observar que as caracterizações do problema que consideram as séries de forma conjunta (“Caracterização 3” e “Caracterização 4”) produziram erros com menores dispersões em relação às caracterizações do problema que as consideram de forma isolada. Isso indica que a consideração das séries como parte de um mesmo fenômeno a ser modelado aumenta a quantidade de informações disponíveis para o treinamento dos modelos de aprendizado computacional, resulta em erros com menores dispersões e consequentemente resulta em modelos de previsão mais confiáveis.

A Figura 5.41 não permite identificar qual foi a caracterização do problema que resultou nos menores erros de previsão. A Figura 5.42 apresenta uma visão alternativa dos resultados de erros médios da Figura 5.41 em que cada eixo representa um dos quatro períodos do horizonte de previsão e cada forma representa os erros médios do modelo otimizado para cada caracterização do problema.

Empresa A: Comparação das médias de erros para cada horizonte para cada definição do problema

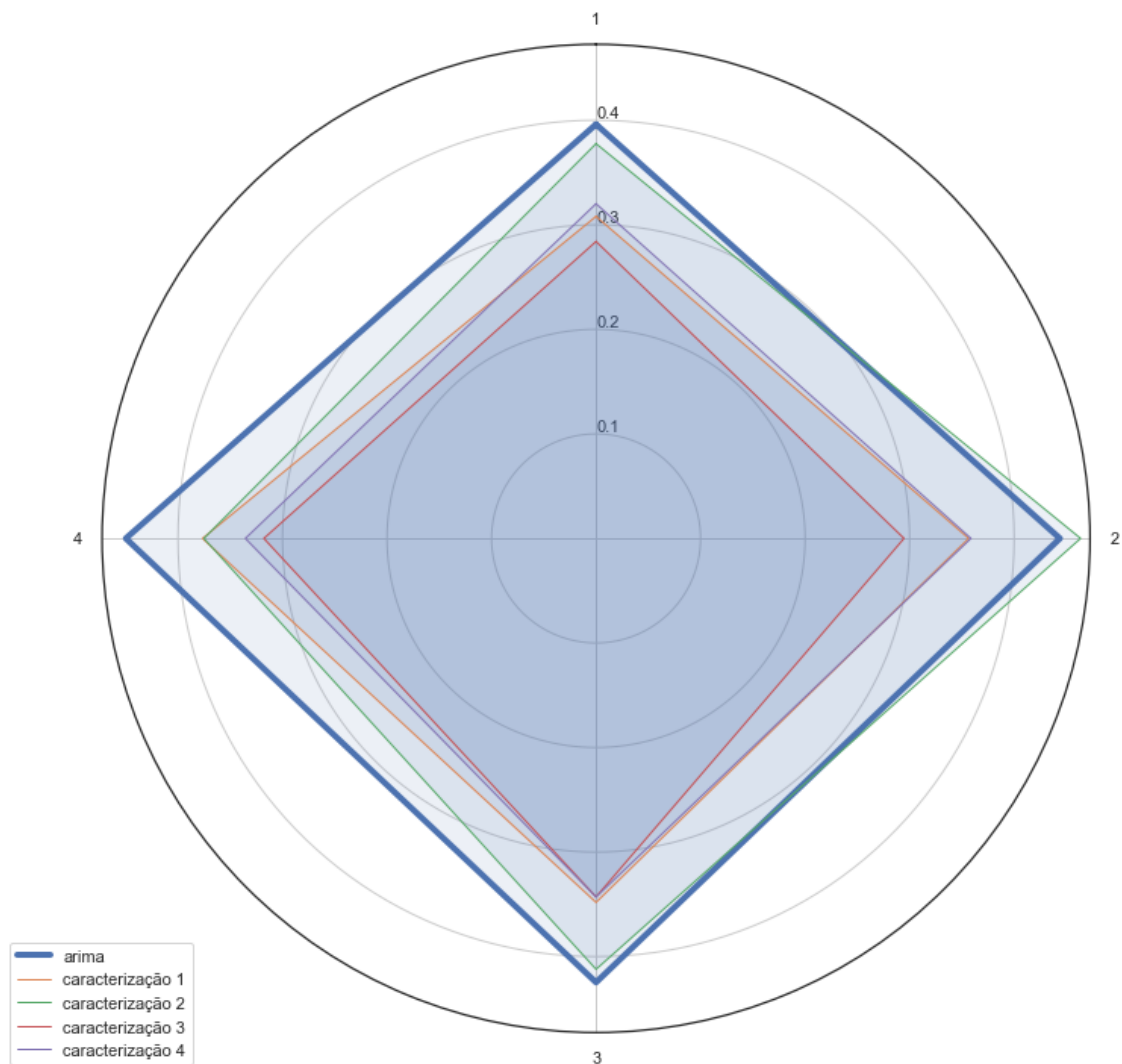


Figura 5.42 – Resultados comparados entre todas as abordagens de definição do problema para os dados da Empresa A

Fonte: Próprio autor

Exceto no caso da segunda abordagem do problema que considera os produtos isoladamente e a representação da demanda em múltiplos (forma com a legenda “Caracterização 2” na Figura 5.42), todos os modelos otimizados apresentaram resultados superiores ao ARIMA em todos os períodos do horizonte de planejamento. Dentre as definições do problema a que consistentemente apresentou menores erros e menor variância foi a abordagem do problema que considera a previsão das vendas de um conjunto de produtos em lojas (forma com linha vermelha com legenda “Caracterização 3”).

Dessa forma, podemos dizer que para os dados da empresa A a definição do problema como a modelagem do comportamento comum de demanda, contemplando um modelo único para todas as séries de vendas da amostra é a mais adequada para o propósito de construir modelos de previsão de demanda desagregada.

A Figura 5.43, análoga à Figura 5.41, apresenta os resultados comparados dos modelos otimizados para as quatro definições do problema em comparação com os resultados dos modelos ARIMA ajustados por força bruta considerando os dados da empresa B. Neste caso, a segunda e a quarta caracterizações do problema (legendas “Caracterização 2 e Caracterização 4” respectivamente) produziram erros médios maiores que com maiores dispersões que os modelos ARIMA e todos os períodos do horizonte. A representação da demanda em valores discretos dos múltiplos de movimentação não foi benéfica para a construção dos modelos de previsão.

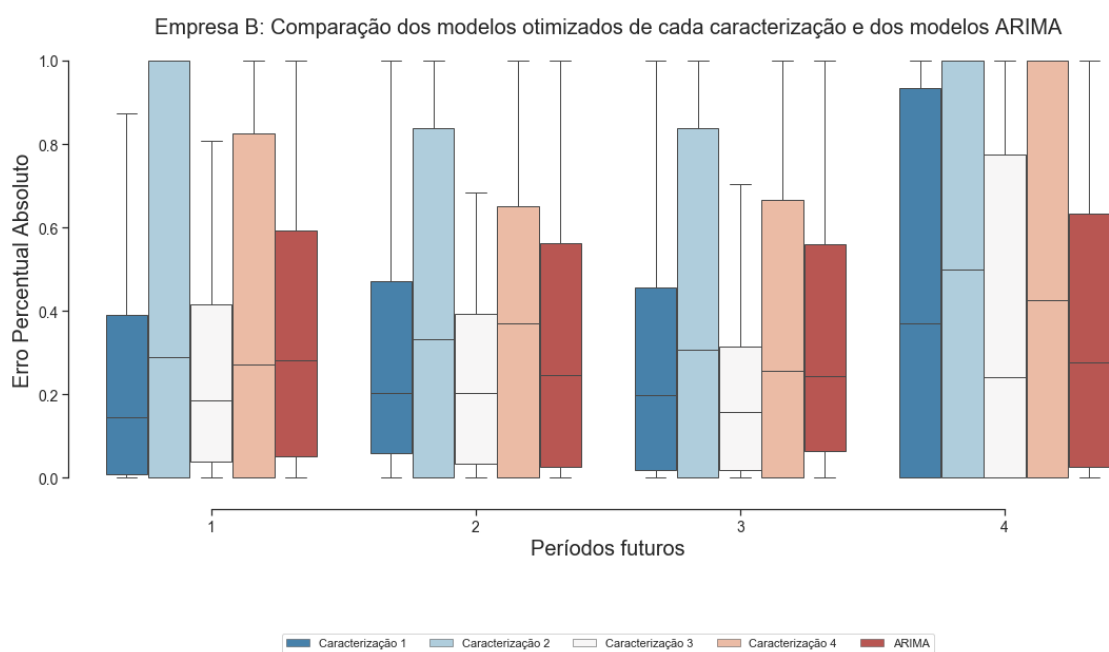


Figura 5.43 – Resultados comparados para os dados da Empresa B

Fonte: Próprio autor

Sobre a Figura 5.43 cabe mencionar que podem ser observadas altas dispersões dos erros no quarto período do horizonte para todas as caracterizações do problema. Uma investigação do fato indicou que o quarto período do horizonte para os dados da empresa

B é um período com alta ocorrência de ruptura e uma vez que as vendas observadas são nulas devido as rupturas, a comparação das previsões com as vendas reais fica prejudicada.

A Figura 5.44 apresenta uma visão alternativa dos resultados de erros médios da Figura 5.43. Os resultados das abordagens que consideram a previsão da demanda sem considerar janelas de múltiplos de movimentação e conseqüentemente definem o problema como um problema de regressão (Caracterização 1 e 3) apresentaram erros inferiores aos modelos ARIMA exceto para o quarto período do horizonte de planejamento. Já os resultados das abordagens que consideram múltiplos de movimentação (Caracterização 2 e 4), que conseqüentemente representam a demanda por variáveis discretas e definem o problema como um problema de classificação, apresentaram resultados consistentemente inferiores aos modelos ARIMA em todos os períodos. De todas as abordagens a que apresentou melhores resultados é a abordagem de previsão de demanda de um conjunto de produtos e lojas.

5.8. Sumário dos resultados

Analisando os resultados comparados para os dados das duas empresas, algumas conclusões podem ser tiradas.

1. Em geral, os modelos de aprendizado computacional tiveram melhor desempenho que os modelos ARIMA;
2. A abordagem do problema que produziu consistentemente os melhores resultados foi a abordagem três (previsão do comportamento comum de demanda);
3. A representação da demanda com vetores binários que representam os intervalos de vendas de acordo com os múltiplos de movimentação não produziu bons resultados;
4. As abordagens de modelagem conjunta das séries de vendas produziram melhores resultados que as abordagens de modelos independentes para cada série de vendas.

Empresa B: Comparação das médias de erros para cada horizonte para cada definição do problema

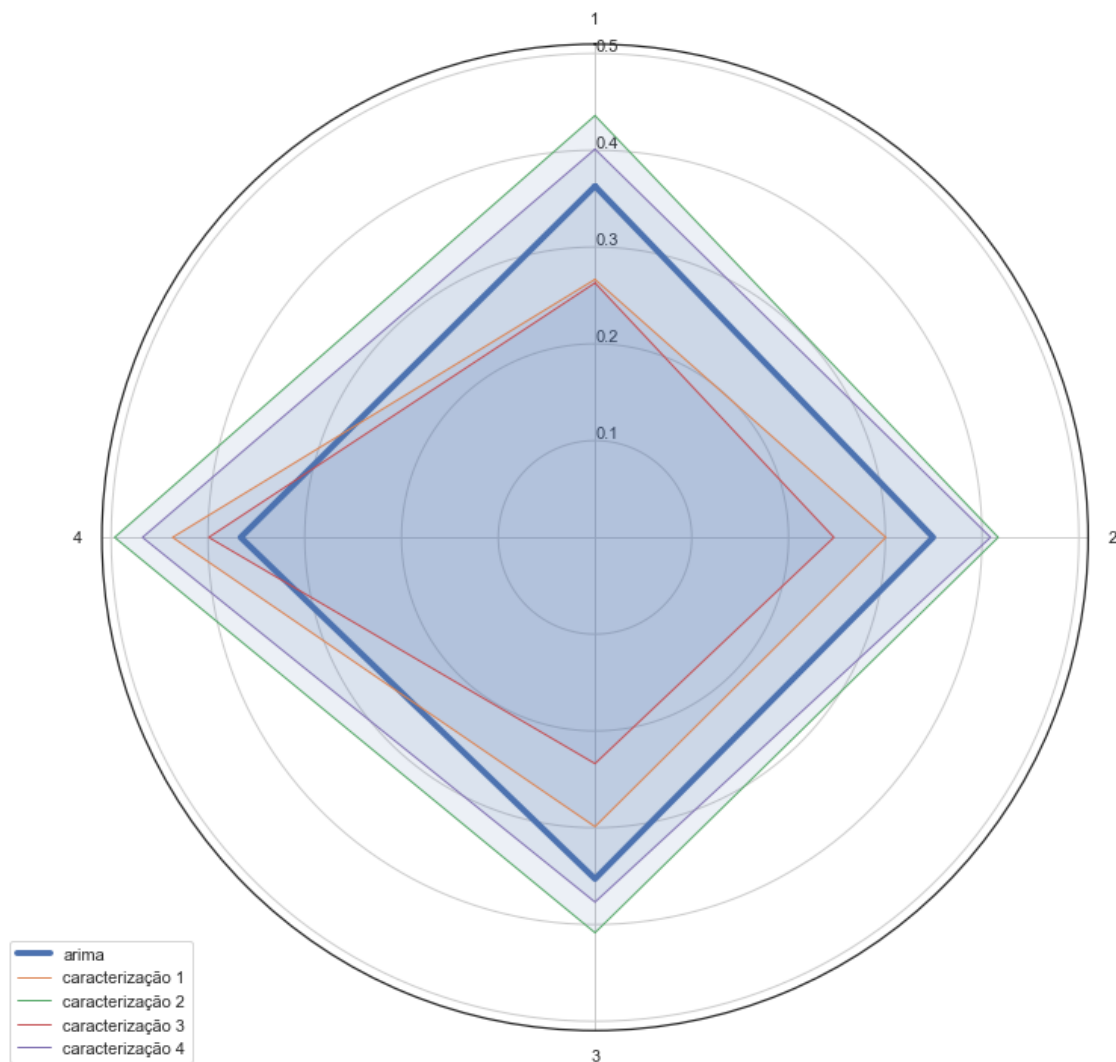


Figura 5.44 – Resultados comparados entre todas as abordagens de definição do problema para os dados da Empresa B

Fonte: Próprio autor

Vale ressaltar que a realização dessa aplicação representou de forma adequada o cenário e a problemática descritas na seção 1.1, ou seja, foram modeladas muitas séries simultaneamente e sem a possibilidade de ajustes pontuais e otimizações para cada uma delas. Isso vai de encontro ao cenário do setor varejista de bens de consumo, no qual é impraticável dedicar muito tempo de análise para modelagem convencional de séries temporais.

Os modelos ARIMA utilizados para comparação foram ajustados por meio do método de força bruta, variando as ordens dos modelos dentro de valores usuais desses parâmetros.

Certamente há possibilidade de melhorar esses modelos, mas o próprio contexto da pesquisa vai contra essa modelagem individual das séries de venda por meio de técnicas convencionais.

6. CONCLUSÕES E RECOMENDAÇÕES

O foco dessa pesquisa foi analisar o problema de previsão de demanda desagregada no contexto de cadeias varejistas do segmento de bens de consumo. A previsão de demanda desagregada consiste em realizar previsões para as vendas de produtos em lojas, ou seja, as previsões devem ser feitas no nível mais detalhado. Esse tipo de problema se encontra dentro do contexto de decisões de abastecimento de estoque de pontos de venda a afeta diretamente o nível de serviço percebido pelo consumidor: previsões abaixo das vendas resultam em rupturas de estoque e frustração de consumidores, previsões acima das vendas resultam em excessos de estoque, perdas e ocupação ineficiente do espaço das lojas.

A pesquisa teve como primeiro passo a proposição de diferentes definições válidas para o problema. A definição do problema pode variar em dois sentidos: as séries de vendas de produtos em lojas podem ser consideradas como fenômenos isolados ou como ocorrências de um mesmo fenômeno, e a demanda a ser prevista pode ser definida como uma variável contínua, ou como uma variável discreta representando janelas de valores de vendas. Assim, essa pesquisa propõe quatro definições alternativas para o problema compreendendo as combinações desses dois fatores (seção 3).

Além da definição do problema, essa pesquisa propõe uma metodologia de construção de modelos de previsão que pode ser aplicada a qualquer uma das definições do problema. A metodologia aborda técnicas de limpeza e tratamento de dados e a aplicação de modelos de aprendizado computacional para a solução do problema de previsão de demanda desagregada.

Um dos principais pontos da pesquisa é que metodologias convencionais de análise de séries temporais são incompatíveis com os desafios atuais do varejo. Isso decorre do fato de que técnicas convencionais requerem análises pontuais de cada série de vendas e também requerem tempo de pessoas com conhecimento especializado para ajuste de parâmetros de modelos. No segmento varejista a quantidade de produtos e pontos de venda (lojas) para os quais são necessários realizar análises de series de vendas é muito elevada, tornando impraticável a modelagem convencional. Por outro lado, as técnicas de modelagem por aprendizado computacional são mais compatíveis com esse cenário, pois a extração dos

padrões de demanda pode ser modelada e extrapolada a partir dos dados observados com baixa supervisão humana, tomando como premissa uma quantidade suficiente de dados.

Além disso, a alta disponibilidade de informações de vendas e estoques das cadeias varejistas, provenientes dos sistemas de gestão dessas empresas, e a complexidade dos padrões de demanda também induz a utilização de metodologias de aprendizado computacional.

A metodologia proposta no capítulo 4 recomenda formas de tratamento e representação das principais variáveis do problema (vendas, estoques, preços, campanhas de marketing e sazonalidades). As técnicas propostas abordam pontos específicos de cadeias varejistas: limpeza de *outliers*, limpeza e dados de vendas decorrentes de rupturas e estoques virtuais, identificação de *lags* de vendas, identificação de datas especiais e sazonalidades.

A metodologia também indica a aplicação de modelos de aprendizado computacional para construir modelos de previsão de demanda considerando os dados tratados. Dentro os diversos tipos de modelos foram selecionados quatro, árvores de decisão, *gradient boosting machines*, *support vector machines* e redes neurais artificiais do tipo *feed-forward*. Esses quatro modelos possuem teorias bastante distintas e por essa razão foram escolhidos. Uma busca exaustiva por todos os modelos possíveis seria impraticável.

Para avaliação de todos os tópicos propostos da pesquisa, incluindo as diferentes caracterizações do problema e a metodologia de modelagem, foi realizada uma aplicação prática com informações de duas empresas relevantes do segmento varejista nacional, uma empresa de supermercados de grande porte e uma rede de franquias também de grande porte. Para comparação de resultados foi selecionada uma técnica convencional bastante usual de previsão de vendas e amplamente utilizada por empresas do setor, a metodologia ARIMA de modelagem de séries temporais. Para essa pesquisa os modelos de comparação foram ajustados com um método de força bruta.

O primeiro resultado interessante verificado com a aplicação da metodologia é que as caracterizações do problema que definem a demanda como uma variável discreta, ou seja, como uma janela de vendas, atingem resultados inferiores em termos de erros médios e

variância de erros se comparadas com as caracterizações do problema que consideram a demanda como uma variável contínua. O segundo resultado que vale mencionar é que as caracterizações que consideram as séries de vendas de forma conjunta produzem resultados superiores em relação às caracterizações que consideram as séries de forma isolada. Foi possível concluir que a caracterização do problema que considera as séries de forma conjunta e representa a demanda como uma variável contínua é a mais adequada para o propósito de construir modelos de previsão de demanda desagregada no setor varejista, ou seja, é aquela que resulta em modelos com menores erros médios e menores dispersões de erros. Como recomendação de pesquisa futura vale a pena investigar os motivos pelos quais as caracterizações que consideram a demanda como uma variável discreta, e, portanto, definem o problema como um problema de classificação, não produziram bons resultados. Como ponto de partida nesse sentido, vale a pena investigar o espaço de distribuição das vendas e determinar se os valores podem de fato serem separados em conjuntos ou classes bem definidas.

Os resultados também indicam que a metodologia proposta apresenta resultados superiores que os modelos ARIMA utilizados como referência. Esses resultados superiores podem ser verificados tanto em termos de erros médios de previsão quanto em termos de dispersão dos erros de previsão. Isso indica que a metodologia é efetiva na criação de modelos de previsão. Os modelos ARIMA, apesar de necessitarem de análise de especialistas e intervenções manuais para sua aplicação, são efetivos na modelagem de relações lineares entre as vendas passadas e as vendas futuras. O fato de modelos de aprendizado computacional que não tomam essa linearidade como premissa, terem sido capazes de produzir previsões mais assertivas e com menor variância, indica que o fenômeno de vendas no varejo possui características não lineares que são mais bem representadas pelos modelos de aprendizado computacional considerados nessa pesquisa em relação aos modelos convencionais ARIMA. Uma linha de pesquisa que pode ser explorada é uma análise estatística de conjuntos de dados de vendas de diferentes segmentos com o objetivo de caracterizar a não linearidade das relações entre a demanda passada, as demais variáveis e a demanda futura. Isso poderia justificar a extensão da metodologia proposta nessa pesquisa para segmentos diferentes do varejo de bens de consumo.

Na aplicação prática, a metodologia foi executada com pouca supervisão e sem intervenções manuais pontuais para ajuste de modelos. Nenhuma otimização ou ajuste pontual para determinado produto ou loja foi realizado. Essa característica é importante considerando a necessidade de realizar previsões de vendas de muitos produtos em muitas lojas no segmento varejista. Com isso é possível afirmar que a metodologia proposta é eficiente para geração de modelos de previsão dadas as características do segmento varejista, indicando que tomadores de decisão podem ter confiança na utilização da metodologia proposta para construção de sistemas de previsão de demanda.

Toda a metodologia foi construída pensando no cenário de uma empresa varejista. Esse cenário possui problemas e desafios semelhantes à indústria de *Fast Moving and Consumer Packaged Goods* (FMCPG) que atende diretamente o varejo, muitas vezes por processos do tipo *Vendor Managed Inventory* (VMI) nos quais a responsabilidade pelos estoques nos pontos de venda é do próprio fabricante. Nesses casos, a responsabilidade de prever a demanda e determinar os estoques dos pontos de venda é da indústria. Uma possível extensão da pesquisa é testar a metodologia proposta em conjuntos de dados dessas indústrias de FMCPG.

Alguns sub-segmentos dentro do setor de bens de consumo possuem forte influência de fatores de moda como é o caso do mercado *fashion* e de *life style*. Esses fatores muitas vezes não são planejados e impactam negativamente a previsão de demanda. Uma extensão válida dessa pesquisa seria seu aprofundamento nesses sub-segmentos e especificação de variáveis de marketing e moda para construção de modelos de previsão de demanda.

No momento de execução dessa pesquisa as técnicas de inteligência artificial vêm se aprimorando rapidamente. Uma das linhas de desenvolvimento é chamada de *Deep Learning* ou Aprendizado Profundo. No caso de problemas com estruturas temporais e sequenciais, redes neurais com estruturas recorrentes têm demonstrado bons resultados. Exemplos de aplicação dessas redes recorrentes em problemas com estrutura sequencial são em problemas de tradução de linguagem (*machine translation*), modelagem de voz para texto e vice-versa (*speech to text*) e até mesmo previsão de séries temporais. Considerando os aprendizados dessa pesquisa, em especial a definição do problema como modelagem

conjunta das séries de vendas, uma extensão da pesquisa seria a construção de uma arquitetura de rede neural recorrente que seria alimentada com as séries de vendas de todos os produtos de uma empresa varejista e as demais variáveis explicativas para compor um modelo único de previsão de demanda.

Ao final da pesquisa entende-se que os objetivos foram alcançados: foi possível determinar a melhor definição para o problema, foi possível propor e validar uma metodologia capaz de gerar previsões de venda com pouca supervisão e a metodologia foi capaz de gerar resultados melhores que as técnicas convencionais.

Espera-se que os resultados dessa pesquisa motivem profissionais e tomadores de decisão de empresas varejistas a utilizarem a metodologia proposta em seus processos e sistemas de previsão de demanda. Com isso será possível reduzir os estoques e reduzir as rupturas, resultando em maior rentabilidade da operação, melhor nível de serviço e maior satisfação dos consumidores finais.

REFERÊNCIAS BIBLIOGRÁFICAS

- Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., Arshad, H. (2018) State-of-the art in artificial neural network applications: A survey. *Heliyon* 4 (2018) e00938. doi: 10.1016/j.heliyon.2018. e00938
- Adivar, B., Hüseyinoğlu, I. Ö. Y., Christopher, M. (2019) A quantitative performance management framework for assessing omnichannel retail supply chains. *Journal of Retailing and Consumer Services* 48 p 257-269.
- Andiojaya, A., Demirhan, H. (2019) A bagging algorithm for the imputation of missing values in time series. *Expert Systems with Applications* (129) p 10 – 26.
- Alon, A, Qi, M., Sadowski, R. J. (2001) Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services* 8 p147-156.
- Alpaydin E. Introduction to Machine Learning. The MIT press (2010).
- Alzamendi, g. a., Schlotthauer, G., Torres, M. E. (2015) State-Space Approach to Structural Representation of Perturbed Pitch Period Sequences in Voice Signals. *Journal of Voice*, 29 (6) p 682-692.
- Barman, R.B. Estimation of Default Probability for Basel II on Credit Risk (2005)
- Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. (1984). Classification and Regression Trees. Belmont, CA: Wadsworth International Group.
- Breunig, M.M., Kriegel, H.-P., Ng, R.T., and Sander, J. (2000). LOF: identifying density-based local outliers. *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD)*, Dallas, Texas.
- Boone, T., Ganeshan, R., Jain, A., Sanders, N. R., (2019) Forecasting sales in the supply chain: Consumer analytics in the big data era. *International Journal of Forecasting*, Elsevier, vol. 35(1), pages 170-180.

Brown, R.G. (1959) *Statistical forecasting for inventory control*, McGraw-Hill: New York.

Brown, R.G. (1963). *Smoothing, forecasting and prediction of discrete time series*, Englewood Cliffs, NJ: Prentice-Hall.

Carbonneau R., Laframboise K., Vahidov R. (2008) Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research* v 184 p 1140–1154.

Che, H. Chen , X., Chen, Y., (2012) Investigating effects of out-of-stock on consumer stock keeping unit choice. *Journal of Marketing Research* (49), 502–513.

Chen, C., Liu, L-H (1993) Forecasting time series with outliers. *Journal of Forecast* 1993 p 13–35

Chu C.W, Zhang, G. P. (2003) A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics*, 86 (3) p 217 – 231, 2003.

Corsten, D., Gruen, T. (2003). Desperately seeking shelf availability: An examination of the extent, the causes, and the efforts to address retail out-of-stocks. *International Journal of Retail & Distribution Management*, 31 (12), 605–617 .

De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International journal of forecasting*, 22(3), 443-473.

Doganis, P., Alexandridis, A., Patrinos, P., Sarimveis, H. (2006) Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering* (75), p 196-204.

Durbin, J. and Koopman, S. J. (2012). *Time Series Analysis by State Space Methods: Second Edition*. Oxford Statistical Science Series. OUP Oxford.

Edgeworth, F. Y. 1887. On discordant observations. *Philosophical Magazine* 23 (5), p364 - 375.

Ehrenthal, J.C.F., Honhon, D., Van Woensel, T. (2014) Demand seasonality in retail inventory management. *European Journal of Operational Research* v 238 p 527 – 539.

Fildes, R., Ma, S., & Kolassa, S. (2018). Retail forecasting: Research and practice. Lancaster University Management School. Lancaster University Working paper.

Fildes, R., Nikolopoulos, K., Crone, S. , & Syntetos, A. (2008). Forecasting and operational research: A review. *Journal of the Operational Research Society*, 59 (9), 1150–1172.

Friedman, J. (2001). Greedy boosting approximation: a gradient boosting machine. *Ann. Stat.* 29, 1189–1232. doi: 10.1214/aos/1013203451

Box, G. E. P., Jenkins G. Time Series Analysis, Forecasting and Control. Holden-Day, Incorporated, 1990.

Goodfellow I., Bengio Y., Courville A. Deep Learning, The MIT Press, 2016

Goodness, C. A, Balcilar, M., Gupta, R., Mujumdar, A. (2015) Forecasting aggregate retail sales: The case of South Africa. *Int. J. Production Economics* 160 p 66–79.

Guyon I., Elisseeff, A. (2003) An introduction to Feature and Variable Selection. *Journal of Machine Learning Research* 3 p 1157-1182.

Guo, Z. X., Wong, W. K., Li, M. (2013) A multivariate intelligent decision-making model for retail sales forecasting. *Decision Support Systems* 55, p247-255.

Hastie, T., Tibshirani, R. and Friedman, J. H. (2008). Elements of Statistical Learning 2nd edition. Springer 2008.

Winklhofer H., Diamantopoulos A., Witt S. F. Forecasting practice: A review of the empirical literature and an agenda for future research. *International Journal of Forecasting*, 12(2):193 – 221, 1996.

Holt, C.C. (1957). Forecasting seasonals and trends by exponentially weighted averages. O.N.R. Memorandum 52/1957, Carnegie Institute of Technology. Reprinted with discussion in 2004, *International Journal of Forecasting*, 20, 5–13.

Hornik, K., Stinchcombe, M., White, H. (1990) Using multi-layer feedforward networks for universal approximation, *Neural Networks* 3 p551–560.

Hu, M.J.C., 1964. Application of the adaline system to weather forecasting. Master Thesis, Technical Report 6775-1, Stanford Electronic Laboratories, Stanford, CA, June.

Huang, T., Fildes, R., & Soopramanien, D. (2019). Forecasting retailer product sales in the presence of structural breaks. *European Journal of Operations Research*, 279 (2) 459 – 470.

Hyndman R. J., Athanasopoulos G., Forecasting: Principles and Practice, Online Open-access Textbooks, <http://otexts.com/fpp/>, 2013.

Hyndman, R.J., Koehler, A.B., Snyder, R.D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18, 439–454.

James B. Boulden. Fitting the sales forecast to your firm. *Business Horizons*, 1(1): 65–72, 1958.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*. 82 (1): 35–45.

Karabati S., Tan B., Öztürk O-C (2009) A method for estimating stock-out-based substitution rates by using point-of-sale data, *IIE Transactions*, 41:5, 408-420, doi: 10.1080/07408170802512578.

Kaufman, S., Rosset S., Perlich, C. (2011) Leakage in data mining: formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data* 6(4): 556-563, 2011.

Kourentzes, N., Rostami-Tabar, B., Barrow, D. K. (2017) Demand forecasting by temporal aggregation: Using optimal or multiple aggregation levels? *Journal of Business Research* 78 p 1 – 9.

Johnson, R.A. and Wichern, D.W. (2008) Applied Multivariate Statistical Analysis, Sixth Edition. Englewood Cliffs, New Jersey: Prentice Hall.

Lee, K. C., Oh, S. B. (1996) An intelligent approach to time series identification by a neural network-driven decision tree classifier. *Decision Support Systems* 17, p 183-197.

Lorena A. C., Carvalho, A. C. P. L. F. (2007) Uma Introdução às *Support Vector Machines*. *Revista de Informática Teórica e Aplicada* v 14 n 2.

Minsky, M., S. Papert (1969), *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, MA.

Morettin, P. A., Toloi, C. M. C. *Análise de séries temporais*. São Paulo. Edgar Blucher LTDA, 2004, 535p.

Müller, K. R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B. (2001) An Introduction to Kernel-Based Learning Algorithms. *IEEE Transactions On Neural Networks*, V. 12, n. 2.

Müller K.R., Smola A.J., Rätsch G., Schölkopf B., Kohlmorgen J., Vapnik V. (1997) Predicting time series with support vector machines. In: Gerstner W., Germond A., Hasler M., Nicoud JD. (eds) *Artificial Neural Networks — ICANN'97*. ICANN 1997. Lecture Notes in Computer Science, vol 1327. Springer, Berlin, Heidelberg

Muth, J.F. (1960). Optimal properties of exponentially weighted forecasts. *Journal of the American Statistical Association*, 55, 299–306.

Myers, R. (1990), *Classical and Modern Regression with Applications*, PWS- Kent Publishing Company, Boston, MA.

Natekin, A. Knoll A. (2013) Gradient Boosting Machines, A Tutorial. *Frontiers in Neurorobotics*.

Peckham, J.O. (1963) The consumer speaks. *Journal of Marketing*.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning* 1: 81–106

Quinlan, J. R. (1993) *C4.5: Programs for machine learning*, Morgan Kauffman, São Francisco (1993).

Ramos, P., Santos, N., Rebelo, R. (2015) Performance of state space and ARIMA models for consumer retail sales forecasting. *Robotis and Computer-Integrated Manufacturing* 34 pp 151–193.

Randall, W. S., Gibson, B.J., Defee, C.C., Williams, B.D. (2011) Retail Supply Chain Management: Key priorities and practices, *The International Journal of Logistics Management* Vol. 22 No. 3, pp. 390-402.

Roberts, S.A. (1982). A general class of Holt-Winters type forecasting models. *Management Science*, 28, 808–820.

Rojas, R. Neural Networks: a systematic approach. Springer-Verlag, Berlin, 1996.

Rosenblatt, F. (1958) The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain”, *Psychological Review*, Vol. 65, pp. 386–408.

Rosenblatt, F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. *Spartan Books*, 1962.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representation by back-propagating errors. In: Rumelhart, D.E., McClelland, J.L., the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, MA.

Schary, P. B., Becker, B. W. (1978). The impact of stock-out on market share: temporal effects. *Journal of Business Logistics*, 1(1), 31-44.

Schary, Ph., & Christopher, M. (1979). The anatomy of a stock-out. *Journal of Retailing*, n 55 v 2 p 59–70.

Schölkopf B. Support Vector Learning. R. Oldenbourg Verlag Munich 1997.

Smola, A. J., Schölkopf, B. Learning with Kernels. The MIT Press, Cambridge, MA, 2002.

Silvestrini, A., Veredas, D. (2008) Temporal Aggregation of Univariate and Multivariate Time Series Models: A Survey. *Journal of Economic Surveys*, Vol. 22, Issue 3, pp. 458-497.

Singh, S., Gupta, P (2014) Comparative Study ID3, CART and C4.5 Decision Tree Algorithm: A Survey. *International Journal of Advanced Information Science and Technology V 27, N 27*.

Singh, K., Upadhyaya S. (2012) Outlier Detection: Applications And Techniques. *International Journal of Computer Science Issues*, Vol. 9, Issue 1, No 3.

Sun, J., Zuo, H., Wang, W., Pecht, M. G. (2012) Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance. *Mechanical Systems and Signal Processing*, n 28 p 585-596.

Tanaka, K. Na introduction to fuzzy logic for practical applications. New York, 1997, 148p.

Thomassey S., Fiordaliso A. (2006) A hybrid sales forecasting system based on clustering and decision trees. *Decision Support Systems* 42 (1) p408-421.

George C. Tiao (2015). Time Series: ARIMA Methods. *International Encyclopedia of the Social & Behavioral Sciences* (Second Edition), 2015

Ishibuchi, H., Okada H., Tanaka, H. (1993) Fuzzy neural networks with fuzzy weights and fuzzy biases, Proc. ICNN '93 p 1650-1655.

Tsay, R. S., Tiao, G. C. (1984) Consistent Estimates of AR Parameters and ESACF for Stationary and Nonstationary ARMA Models, *Journal of American Statistical Association* 79 p84-96.

Uçkun, C., Karaesmen, F., Savas, S. (2008) Investment in improved inventory accuracy in a decentralized supply chain. *International Journal of Production Economics*, Elsevier, vol. 113(2), p 546-566.

Vapnik, V. N. *Statistical Learning Theory*. John Wiley and Sons, 1998.

Vapnik, V. N. (1999) An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks*, v 10, n 5.

Veiga, C. P., Veiga, C. R. P., Puchalskic, W., Coelho, L. S., Tortato U. (2016) Demand forecasting based on natural computing approaches applied to the foodstuff retail segment. *Journal of Retailing and Consumer Services* 31 p 174 – 181.

Winters, P.R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6, 324–342.

Yule, G.U. (1927). On the method of investigating periodicities in disturbed series, with special reference to Wolfers sunspot numbers. *Philosophical Transactions of the Royal Society London, Series A*, 226, 267– 298.

Zeng, Y., Wu, S. State Space Models: Applications in Economics and Finance. 347p, ISBN: 978-1-4614-7788-4, 2013.

Wang, G., Gunasekaran, A., Ngai, E.W.T., Papadopoulos, T. (2016) Big data analytics in logistics and supply chain management: Certain investigations for research and applications. *Int. J. Production Economics* (176) p 98 – 110.

Zhang, G. P. (2003) Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50 p 159-175.

Zhang G., Patuwo, B. E., Hu, M. Y. (1998) Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1): 35 – 62, 1998.

Zinn W., Liu P. C. (2001) Consumer response to retail stockouts. *Journal of Business Logistics*, (22), pp. 49-71.

APÊNDICE A. CONCEITOS FUNDAMENTAIS

Este capítulo apresenta os principais conceitos de inteligência computacional necessários para a condução da pesquisa. O capítulo é dividido em três seções, sendo a primeira uma revisão dos conceitos fundamentais de inteligência computacional, a segunda uma lista não exaustiva dos principais modelos de inteligência computacional e a terceira uma discussão sobre a legitimidade dos dados para a construção de modelos de previsão.

A1. Aprendizado computacional e aprendizado estatístico

O aprendizado computacional faz referência a um tipo específico de aprendizado chamado de indução. A indução pode ser descrita como o processo de aprendizado por meio do qual se tiram conclusões generalistas a partir de exemplos de um fenômeno analisado (HAYKIN, 2009). Do ponto de vista da lógica formal, o aprendizado por indução pode ser encarado como um processo contrário ao aprendizado por dedução, no qual conclusões generalistas são desenvolvidas de forma teórica e validadas por exemplos práticos *a posteriori*.

O aprendizado por indução pode ocorrer de forma supervisionada ou não supervisionada. No caso do aprendizado indutivo supervisionado, ou simplesmente aprendizado supervisionado, há a existência de uma quantidade de exemplos pré-rotulados que representam exemplos dos conceitos que devem ser aprendidos. Esses exemplos pré-rotulados são representados por pares de entradas e saídas desejadas. Para cada entrada a saída é calculada por uma função e é comparada com a saída desejada. O aprendizado supervisionado ocorre com o ajuste da função para minimizar a diferença entre a saída da função e a saída desejada. Exemplos de problemas de aprendizado supervisionado são problemas de regressão ou classificação em que a qualidade da solução é avaliada com base numa medida de erro de regressão ou classificação.

O aprendizado indutivo não supervisionado, ou simplesmente aprendizado não supervisionado ocorre, por sua vez, sem a existência de exemplos pré-rotulados. Nesse tipo de aprendizado a abstração dos conceitos ocorre por meio de uma função de qualidade, ou seja, dado um conjunto de entradas as saídas geradas por uma função são avaliadas com base numa função arbitrária de aptidão. O aprendizado não supervisionado ocorre ao

ajustar a função de modo a maximizar a aptidão do conjunto de saídas. Um exemplo de um problema de aprendizado não supervisionado é o problema de clusterização em que a saída é avaliada por métricas de qualidade arbitrária dos clusters formados.

Os métodos de aprendizado supervisionado, são caracterizado por pares de exemplos (x_i, y_i) em que $x_i \in X$ representa uma entrada de um exemplo i e y_i representa a saída rotulada correspondente. Nenhuma restrição ou premissa é feita em relação ao domínio dos exemplos X . No entanto, muitos dos algoritmos de aprendizado computacional operam em espaços vetoriais com produto interno definido, ou seja, espaços V para os quais existe uma função f que associa quaisquer pares de vetores $\langle u, v \rangle$ a um valor escalar k respeitando as seguintes propriedades:

- $\langle v, v \rangle \geq 0$;
- $\langle v, v \rangle = 0$ se e somente se $v = 0$;
- $\langle v, u \rangle = \langle u, v \rangle$;
- $\langle v + u, w \rangle = \langle v, w \rangle + \langle u, w \rangle$;
- $\langle kv, u \rangle = k \langle v, u \rangle$;

Em muitos casos, para que os problemas sejam representados em espaços vetoriais com produtos internos, transformações são aplicadas aos exemplos de entrada. Exemplos dessas transformações são: transformação de classificações textuais em valores reais, transformações de classificações binárias em valores binários, dentre outros.

A expressão (A.1) formaliza uma transformação (Φ) genérica aplicada nos problemas de aprendizado computacional. O espaço vetorial \mathcal{H} com produto interno definido resultante da transformação recebe o nome de espaço de atributos (*feature space*). No restante deste capítulo os exemplos de entrada de um problema serão genericamente descritos assumindo que já se encontram representados adequadamente e a notação vetorial é assumida implicitamente, exceto se mencionado o contrário.

$$\begin{aligned}\Phi: X &\rightarrow \mathcal{H} \\ x &\rightarrow \vec{x} := \Phi(x)\end{aligned}\tag{A.1}$$

Além da representação adequada dos exemplos de um problema, outro requisito para grande parte dos algoritmos de aprendizado é que a definição do próprio produto interno do espaço vetorial \mathcal{H} resulte em uma medida de similaridade entre os dois exemplos. Uma função k que define o produto interno do espaço vetorial \mathcal{H} para dois exemplos recebe o nome de *kernel*.

$$\begin{aligned}k: X, X &\rightarrow \mathbb{R} \\ k(x, x') &:= k(\Phi(x), \Phi(x')) = k(\vec{x}, \vec{x}') = \langle \vec{x}, \vec{x}' \rangle\end{aligned}\tag{A.2}$$

Cabe ressaltar que em muitos problemas práticos de aprendizado há a existência de ruídos nos pares de exemplos. Esse ruído é caracterizado por exemplos rotulados de forma imperfeita decorrentes de erros amostrais e imperfeições nos dados de um problema real. Assim, uma característica desejada de um modelo de aprendizado computacional é a sua robustez frente à existência de ruído, ou seja, a sua capacidade de aprender os conceitos presentes em um conjunto de exemplos mesmo com a presença de dados imperfeitos (LORENA, CARVALHO, 2007).

Um modelo resultante da aplicação de uma técnica de aprendizado computacional é chamado de classificador ou de regressor dependendo do objetivo do problema de aprendizado. Caso o problema de aprendizado seja, dado um conjunto de entradas, rotular uma entrada dentre um conjunto de valores discretos, o problema de aprendizado é dito de classificação e o modelo resultante é um classificador. Caso o problema de aprendizado seja, dado um conjunto de entradas, gerar um valor no domínio real, o problema é chamado de regressão e o modelo resultante de regressor.

A construção de um modelo a partir de um algoritmo de aprendizado, seja um classificador ou um regressor, visa produzir um modelo com capacidade de descrever da melhor forma possível o domínio dos dados em que foi gerado. Assim, para determinar a capacidade de generalização do conhecimento aprendido por um modelo, a amostra de exemplos é em

geral dividida em uma amostra de treino e uma amostra de teste. A amostra de treino é utilizada para o aprendizado do modelo e a amostra de teste é utilizada para avaliar as taxas de acertos e erros produzidas por um modelo treinado (ALPAYDIN, 2010).

O algoritmo de aprendizado é utilizado para treinar um modelo específico para um conjunto de dados de treinamento. Há casos em que o modelo produzido se especializa nos dados presentes na amostra de treinamento e possui mal desempenho na previsão de dados fora dessa amostra. A esse fenômeno se dá o nome de sobre-treinamento ou *overfitting*. Há também o caso oposto de sub-treinamento ou *underfitting*, em que o modelo treinado possui um mau desempenho na previsão nos dados de treinamento e também na previsão dos dados de teste (ALPAYDIN, 2010).

Deseja-se então, com uma técnica de aprendizado computacional, produzir um modelo com a melhor capacidade de generalização possível. Esse conceito abstrato pode ser concretizado por meio da Teoria Estatística de Aprendizado (TEA). Essa seção descreve alguns conceitos da TEA no contexto de um problema de classificação binária conforme explicitado em Müller *et al.* (2001). Um aprofundamento dessas questões pode ser encontrado em Hastie, Tibshirane e Friedman (2008).

Sejam (\mathbf{x}_i, y_i) pares de exemplos sendo \mathbf{x} um vetor e y uma saída tal que $y_i \in \{-1, 1\}$ gerados a partir de uma distribuição de probabilidade desconhecida. Assume-se que os pares de exemplos são independentes e identicamente distribuídos de acordo com uma função de probabilidade $P(\mathbf{x}, y)$ desconhecida (VAPNIK, 1998).

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n) \in \mathbb{R}^N \times Y \rightarrow Y = \{-1, 1\} \quad (\text{A.3})$$

Um classificador f para esse conjunto de dados pode ser assumido como uma função que toma uma entrada \mathbf{x} e resulta num valor previsto para y , tal que se $f(\mathbf{x}) > 1$ o exemplo é classificado com $y = 1$ e caso contrário o exemplo é classificado com $y = -1$. O melhor classificador possível é aquele que minimiza o erro esperado, representado na expressão (A.4), em que $l(\cdot)$ é uma função de perda (*loss function*) que representa uma penalidade de erro de previsão ou erro de classificação. A expressão do erro esperado também é chamada de função de risco.

$$R[f] = \int l(f(\mathbf{x}), y) dP(\mathbf{x}, y) \quad (\text{A.4})$$

De acordo com Müller *et al.* (2001), a função de risco (A.4) não pode ser minimizada diretamente para obtenção do classificador ótimo, pois a função probabilidade $P(\mathbf{x}, y)$ é desconhecida e são conhecidos apenas dados de treinamento amostrados de $P(\mathbf{x}, y)$. Dessa forma, é necessário aproximar a função de risco por uma outra função, para que essa então seja minimizada para determinação de um classificador. Uma alternativa é a função empírica de risco que é calculada sobre os exemplos de um conjunto de dados de treinamento conforme a expressão (A.5).

$$R_{emp}[f] = \frac{1}{n} \sum_{i=1}^n l(f(\mathbf{x}_i), y_i) \quad (\text{A.5})$$

A determinação de um classificador a partir da minimização de (A.5) é chamada de indução pela minimização do risco empírico. É possível determinar condições para a função f para as quais a função de risco empírico converge para a função de risco com n suficientemente grande ($n \rightarrow \infty$). No entanto, para um caso geral a determinação do classificador f pela minimização direta da expressão (A.5) pode gerar um problema de *overfitting*.

Uma maneira de evitar o risco de *overfitting* é controlar a complexidade da classe de funções F a partir da qual f é gerada. Intuitivamente é interessante favorecer funções mais simples a funções mais complexas. Uma função mais complexa possui maior capacidade de ajuste aos pontos de uma amostra e, portanto, tem um risco maior de *overfitting*. O risco de *overfitting*, relacionado à complexidade de uma classe de funções F a partir da qual o classificador f é gerado também é chamado de risco estrutural.

O conceito de complexidade de uma classe de funções pode ser representado pela dimensão de Vapnik-Chervonenkis (dimensão VC) (VAPNIK, 1998). A dimensão VC de uma classe de funções F é uma medida da capacidade de um conjunto de funções, quanto maior a dimensão VC mais complexas são as funções e vice-versa. A dimensão VC pode ser vista como a quantidade diferente de diferentes particionamentos possíveis em um espaço n dimensional proporcionado pela classe de funções F . De acordo com Lorena e Carvalho

(2007) para um problema de classificação binário, essa dimensão é definida como o número máximo de exemplos de classes diferentes que podem ser particionados por funções contidas em F .

A Teoria do Aprendizado Estatístico (TAE) (VAPNIK, 1998) fornece alguns limites para a função de risco que relaciona a função de risco empírico e o risco estrutural. Seja um problema de classificação binário para o qual se deseja um classificador f de uma classe F de funções, h a dimensão VC da classe de funções de f e n o tamanho de dados de uma amostra de treinamento. É possível demonstrar com probabilidade $1 - \delta$ e para $n > h$ que a função de risco é limitada superiormente por uma composição do risco empírico e do risco estrutural da classe de funções de f de acordo com a expressão (A.6), em que a primeira parcela do lado direito da equação representa o risco empírico e a segunda parcela o risco estrutural.

$$R[f] \leq R_{emp}[f] + \sqrt{\frac{h \left(\ln \frac{2n}{h} \right) - \ln \frac{\delta}{4}}{n}} \quad (\text{A.6})$$

Para auxiliar a interpretação da expressão (A.6) é possível considerar dois exemplos extremos:

- (i) Pode-se escolher uma classe de funções extremamente simples (e.g. funções lineares) tal que o risco estrutural seja muito próximo de zero, no entanto isso possivelmente pode gerar um risco empírico grande (equivalente a um ajuste inadequado dos exemplos de treinamento);
- (ii) Pode-se escolher uma classe de funções extremamente complexa que consiga mapear todos os exemplos de treinamento resultando num risco empírico próximo de zero, porém um alto valor do risco estrutural.

Um classificador adequado para um problema geralmente é uma solução de compromisso entre o risco empírico e o risco estrutural. A consideração de ambas as parcelas de risco na determinação de um classificador é chamada de ‘princípio da minimização do risco estrutural’ (VAPNIK, 1998).

A expressão (A.6) é apenas um exemplo de limite fornecido pela TAE para o contexto de um problema de classificação binário sendo que existem outras definições para diferentes classes de problemas de aprendizado estatístico.

Segundo Müller *et al.* (2001), a expressão (A.6) não é utilizada em casos práticos pois em geral a dimensão VC de uma classe de funções é indefinida ou infinita. O propósito da expressão é dar uma intuição sobre a relação entre complexidade e capacidade de mapeamento de um classificador para um problema de aprendizado.

Outro conceito importante da TAE para a compreensão das técnicas de aprendizado é o conceito de margem. A margem de um exemplo em relação a um classificador induzido consiste na sua distância em relação à superfície de decisão produzida pelo classificador. A Figura A.1 ilustra esse conceito para um caso de um problema de classificação binário linearmente separável com um classificador linear induzido a partir dos dados de treinamento.

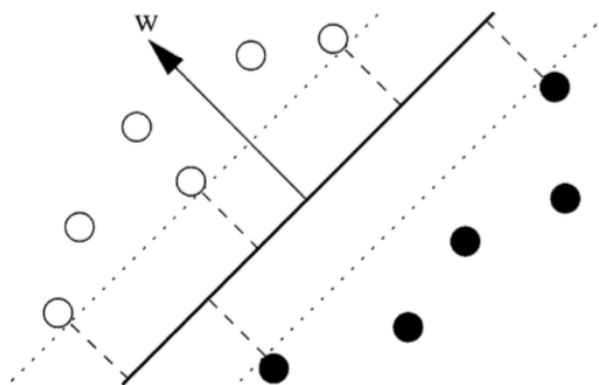


Figura A.1 – Pontos linearmente separáveis num espaço bidimensional

Fonte: Müller *et al.* (2001)

Para um problema de classificação binário em que $y_i \in \{-1, 1\}$, a margem de um exemplo pode ser dada pela expressão (A.7). Um valor incorretamente classificado por f resulta em uma margem negativa.

$$\varrho(f(\mathbf{x}_i), y_i) = y_i f(\mathbf{x}_i) \quad (\text{A.7})$$

Com a definição da margem, é possível definir o erro marginal de um classificador em relação a um conjunto de dados. Seja a função $I(q)$ igual a 1 se a proposição q é verdadeiro e igual a 0 se q é falso. Para uma determinada constante não negativa ρ o erro marginal é definido pela expressão (A.8).

$$R_\rho[f] = \frac{1}{n} \sum_{i=1}^n I(y_i f(\mathbf{x}_i) < \rho) \quad (\text{A.8})$$

Vapnik (1998) demonstra que existe uma constante c tal que, com probabilidade $1 - \theta$ e para $\rho > 0$ para a classe de funções lineares do tipo $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b$ com $\|\mathbf{x}\| \leq R$ e $\|\mathbf{w}\| \leq 1$ vale o limite definido pela expressão (A.8).

$$R[f] \leq R_\rho[f] + \sqrt{\frac{c}{n} \left(\frac{R^2}{\rho^2} \log^2 \left(\frac{n}{\rho} \right) + \log \left(\frac{1}{\theta} \right) \right)} \quad (\text{A.9})$$

Assim como na expressão (A.8), a expressão (A.9) define um limite para a função de risco do classificador com base em uma medida de erro nos dados de um conjunto de treinamento e com base em um termo de capacidade da classe de funções do classificador. Um valor alto para a constante ρ resulta num baixo valor do segundo termo da expressão (A.9), mas por outro lado pode acarretar em um alto valor do risco marginal, uma vez que torna mais difícil a separação dos exemplos. Em contrapartida, um valor baixo da constante ρ pode resultar num baixo risco marginal pois todos os exemplos são mais facilmente separados dentro da margem, porém gera um alto risco de capacidade representado pelo segundo termo da expressão.

As expressões (A.8) e (A.9) exemplificam que há uma relação de compensação (*trade-off*) entre a capacidade de representação do modelo e a capacidade de generalização. Um classificador ou regressor adequado para um problema de aprendizado computacional deve ser obtido por um balanço entre capacidade de representação dos dados de teste e a capacidade de generalização de dados fora da amostra de teste, sendo que modelos mais simples são preferíveis a modelos mais complexos com maior risco de sobre treinamento. Maiores detalhes sobre a TAE podem ser encontrados em Vapnik (1998) ou Vapnik (1999).

A2. Principais modelos de inteligência computacional

Esta seção apresenta alguns dos principais modelos ou classes de modelos de inteligência computacional. As referências não são exaustivas, e buscam apenas dar uma visão geral das técnicas disponíveis.

A.2.1. Associadores lineares e não lineares

O problema de regressão linear é amplamente estudado na teoria estatística. Esse mesmo problema pode ser descrito sob uma perspectiva de aprendizado computacional do ponto de vista de associadores lineares (Figura A.2). Associadores lineares são unidades de processamento que possuem um vetor de pesos \vec{w} com dimensão p e, dada uma entrada \vec{x} também de dimensão p , produzem uma saída escalar $\vec{w} \cdot \vec{x}$.

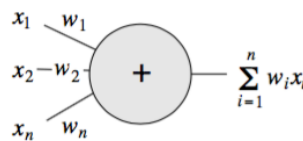


Figura A.2 – Associador Linear

Fonte: Rojas (1996)

O treinamento de um associador linear se dá por meio da apresentação de n pares ordenados de inputs e output (\vec{x}_i, y_i) . Para o treinamento do associador linear, em geral, define-se a função de erro (A.10) em que X é uma matriz $(n \times p)$ com os *inputs* e y é um

vetor n -dimensional com os *outputs* de treinamento. Caso as dimensões de X sejam não colineares, a minimização de (A.10) possui uma forma analítica fechada e o associador linear pode ser treinado por uma simples operação matricial (A.11). Caso contrário, podem ser aplicadas técnicas numéricas de otimização como o algoritmo *backpropagation* (ROJAS, 1996).

$$E(\vec{w}) = \|X\vec{w} - \vec{y}\|^2 \quad (\text{A.10})$$

$$\vec{w} = (X^T \cdot X)^{-1} X^T \vec{y} \quad (\text{A.11})$$

Formalmente, o associador linear é uma unidade de processamento que possui uma função de integração de soma e uma função de transformação de identidade ($f(x) = x$). Uma vez que tanto a operação de soma quanto a transformação identidade são operações lineares, o associador é dito linear.

Caso a função de transformação de um associador seja não linear, o associador passa a ser classificado como não linear. Um exemplo é o caso em que a função de transformação é do tipo sigmoidal $s(x)$. Nesse caso a saída do associador é dada por $s(\vec{w} \cdot \vec{x})$.

Assim como o associador linear está relacionado com o problema estatístico de regressão linear múltipla, um associador não linear com uma função de transformação sigmoidal está relacionado com o problema estatístico de regressão logística. Além desse paralelo com a teoria estatística, um associador não-linear com função sigmoide consiste em unidades de processamento convencionalmente utilizadas em redes neurais artificiais.

Assim como as redes neurais artificiais, os associadores não-lineares podem ser treinados com o uso do algoritmo *backpropagation*. Contudo, pode-se também aplicar uma transformação nos dados de output e linearizar o problema.

Sejam (\vec{x}_i, a_i) pares ordenadores de amostras de treinamento de um associador não linear. Sabe-se que os outputs a_i estão contidos no intervalo entre 0 e 1. A função de erro cuja minimização representa o treinamento do associador é dada pela expressão (A.12).

$$E(\vec{w}) = \sum_i (a_i - s(\vec{w} \cdot \vec{x}_i))^2 \quad (\text{A.12})$$

Anteriormente ao treinamento do associador não linear pode ser aplicada a transformação logit (A.13) nos outputs a_i , de modo que a função de erro passa a ser a mesma do caso do associador linear e o problema se transforma em um problema de treinamento de um associador linear.

$$a'_i = s^{-1}(a_i) = \ln \left(\frac{a_i}{1 - a_i} \right) \quad (\text{A.13})$$

$$E(\vec{w}) = \sum_i (a'_i - (\vec{w} \cdot \vec{x}_i))^2 \quad (\text{A.14})$$

Associadores não lineares são utilizados em problemas de classificação binária, em que é necessário prever se uma amostra pertence a uma classe ou não, ou ainda, prever a probabilidade de uma amostra pertencer a uma classe. Um exemplo desse tipo de problema é modelagem de risco de crédito amplamente estudado na literatura financeira (BARMAN, 2005). Nesse problema, dado um indivíduo e seus parâmetros é necessário determinar se o indivíduo é um bom pagador ou um mal pagador pela sua probabilidade de *default*.

Vale ressaltar que a teoria estatística clássica de regressão faz uma série de considerações a respeito da distribuição de probabilidade dos dados. Para a aplicação do modelo de regressão multilinear convencional assume-se que as variáveis independentes possuem uma distribuição gaussiana multivariada e são independentes entre si. Sob o enfoque de inteligência computacional, essas premissas não são necessárias, sendo de interesse apenas a capacidade de generalização do modelo treinado resultante.

Os modelos de associadores lineares e não lineares podem ser classificados como modelos paramétricos no sentido em que assumem uma forma funcional para o modelo, além de que o treinamento consiste na estimação dos parâmetros.

Myers (1990) apresenta uma visão detalhada do problema de regressão linear e não linear, bem como sua relação com associadores no contexto de aprendizado computacional.

A.2.2. Árvores de decisão

Uma árvore de decisão é um tipo de estrutura de dados composta de nós de decisão intermediários e folhas terminais que agrupam amostras ou entradas de acordo com os valores de seus atributos em diferentes sub regiões bem definidas do espaço de entrada (HASTIE *et al.*, 2008) . Uma ilustração de uma árvore de decisão pode ser encontrada na Figura A.3.

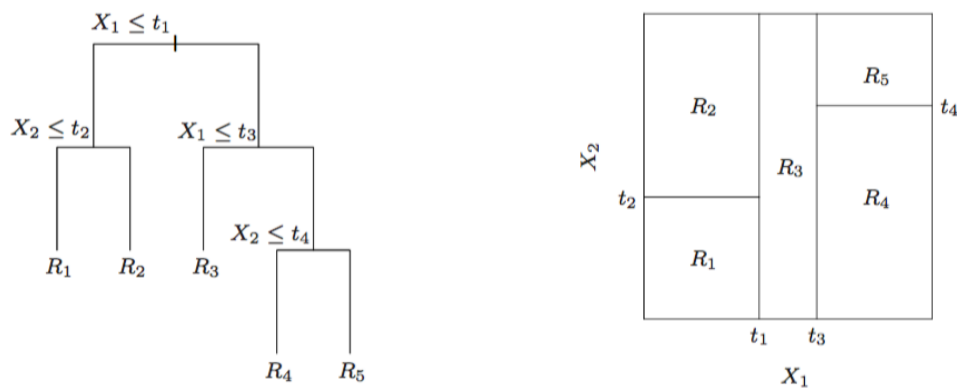


Figura A.3 – Árvore de decisão

Fonte: Hastie *et al.* (2008)

Um algoritmo de árvore de decisão é composto de regras de treinamento para determinar os nós intermediários e suas regras de decisão com a finalidade de compor um modelo preditivo para os dados de entrada. O método precisa determinar as regras de decisão em cada nó e a própria topologia da árvore de decisão. Assim, os algoritmos de árvores de decisão formam uma classe de modelos não paramétricos capazes de ajustar uma árvore de decisão aos dados de entrada (HASTIE *et al.*, 2008). O algoritmo é dito não paramétrico pois nenhuma forma funcional para a relação entre os inputs e os outputs é definida antes do treinamento da árvore.

Conforme uma entrada caminha pela árvore e conseqüentemente pelos seus nós intermediários, ela é classificada em regiões cada vez menores do espaço de entrada até atingir uma folha terminal. Cada folha terminal tem uma etiqueta ou valor que define a saída prevista para cada amostra presente na folha. Uma etiqueta é utilizada no caso de um problema de classificação e um valor é utilizado no caso de um problema de regressão.

Cada nó de decisão m implementa uma função de decisão $f_m(\vec{x})$, também chamada de função de discriminação (HASTIE *et al.*, 2008). A função f_m divide o espaço de entrada em dois subespaços. Uma árvore pode ser classificada como univariada ou multivariada de acordo com o formato das suas funções f_m :

- Caso a todas as funções f_m de uma árvore utilizem apenas uma dimensão x_j de cada entrada \vec{x} por vez, a árvore é dita univariada.
- Caso contrário, a árvore é caracterizada como multivariada.

As árvores univariadas são o caso mais comum e no restante dessa seção serão tratadas em maiores detalhes.

A função mais comumente utilizada nos nós de uma árvore de decisão univariada é a função *threshold* (também chamada de função limiar). Caso a dimensão x_j utilizada por f_m seja numérica, a função *threshold* se traduz na expressão (A.15) em que w_m é o valor limite que caracteriza a separação das amostras.

$$f_m(\vec{x}) = \begin{cases} 1 & \text{se } x_j > w_m \\ 0 & \text{caso contrário} \end{cases} \quad (\text{A.15})$$

A função limite definida por (A.15) divide o espaço de busca em duas regiões. Essa divisão recebe o nome de *split* binário pois divide o espaço de entrada em duas regiões, uma denominada “esquerda” (expressão (A.16)) e outra “direita” (expressão (A.17)). As regiões definidas por *splits* também são chamadas de galhos do nó.

Sucessivos *splits* binários no caminho de uma árvore de decisão, da raiz para as folhas, produzem subdivisões ortogonais do espaço de entrada.

$$L_m = \{\vec{x} \mid x_j > w_m\} \quad (\text{A.16})$$

$$R_m = \{\vec{x} \mid x_j \leq w_m\} \quad (\text{A.17})$$

A construção de uma árvore de decisão eficiente para um conjunto de amostras se resume no problema de determinação de uma árvore capaz de dividir corretamente as amostras de entrada com o menor número de *splits* possível, resultando na menor estrutura dentro do espaço de todas as estruturas possíveis. Esse é um problema NP-Completo conforme demonstrado por Quinlan (1986), de modo que procedimentos heurísticos podem ser utilizados para encontrar árvores com eficiência aceitável em tempo computacional igualmente aceitável.

Nesta seção são apresentados os fundamentos da metodologia de construção de árvores de decisão univariadas. Algumas implementações populares dessa metodologia são os algoritmos ID3 (QUINLAN; 1983), C4.5 (QUINLAN; 1995) e CART (BREIMAN *et al.* 1984). Uma comparação detalhada sobre os três algoritmos pode ser encontrada em Singh e Gupta (2014).

No caso de uma árvore para um problema de classificação, a qualidade de um *split* deve ser determinada por uma medida de impureza. Um nó é dito perfeitamente puro se após a divisão do espaço de entradas, todas as amostras pertencentes a um mesmo subespaço possuem a mesma classificação do atributo *target*. Dado que N_m amostras chegam ao nó m de uma árvore, a probabilidade *a priori* de uma amostra pertencer a classe C_i é dada pela expressão (A.18) em que N_m^i é o número de amostras da classe C_i dentro de N_m .

$$P(C_i | m, \vec{x}) = p_m^i = \frac{N_m^i}{N_m} \quad (\text{A.18})$$

Considerando a definição da expressão (A.18) uma possível medida de impureza de um nó é dado pela sua entropia (A.19) em que K é a quantidade de classes em que a amostra pode ser classificada de acordo com o atributo *target* do problema.

$$\mathcal{J}_m = - \sum_{i=1}^K p_m^i \log (p_m^i) \quad (\text{A.19})$$

Também no caso de *splits* binários, outra medida de impureza frequentemente utilizada é o índice de Gini (A.20).

$$\phi_m = 2p_m^m(1 - p_m^m) \quad (\text{A.20})$$

Os algoritmos de construção de árvores de decisão buscam *splits* que reduzem a impureza total da árvore, definida pela soma das impurezas das suas folhas terminais. Seja N_m a quantidade de amostras que chega até um nó m , N_{mj} a quantidade de amostras que chega ao nó m e é classificada pela regra de decisão do nó m como sendo pertencente ao galho j do split e seja N_{mj}^i a quantidade de amostras da classe i que chega ao nó m e é classificada como sendo do galho j pela regra de decisão do nó. A probabilidade *a posteriori* de uma amostra classificada pelo nó m pertencer a classe i é dada pela expressão (A.21).

$$P(C_i|m, \vec{x}, j) = p_{mj}^i = \frac{N_{mj}^i}{N_{mj}} \quad (\text{A.21})$$

A impureza total do nó m após o *split*, em termos da medida de entropia, é dada pela expressão (A.22) em que n é a quantidade total de amostras \vec{x} e os demais termos possuem definição já mencionada.

$$\mathcal{J}_m = - \sum_{j=1}^n \frac{N_{mj}^i}{N_{mj}} \sum_{i=1}^K p_{mj}^i \log (p_{mj}^i) \quad (\text{A.22})$$

O algoritmo de construção de árvores de decisão CART busca em cada iteração determinar o *split* da árvore que maximiza a redução da impureza da mesma, medida pela diferença entre a impureza do nó antes do split, dada pela expressão (A.19), e a impureza do nó após o split, dada pela expressão (A.22). O algoritmo faz otimizações locais de *splits* e constrói a árvore de decisão de forma recursiva. A determinação de um *split* compreende tanto a determinação do atributo que será utilizado para fazer o split como a função f_m do nó.

Os critérios de parada do algoritmo podem ser relativos a uma profundidade máxima de nós de decisão, a uma quantidade mínima de amostras em cada folha terminal ao atendimento de uma impureza limite θ_l ou uma combinação dos critérios anteriores. A Figura A.4 representa o pseudo-código da metodologia para a construção de uma árvore de classificação univariada.

GerarArvore(X)

Se $EntropiaNo(X) < ThetaL$

Cria folha com etiqueta equivalente a classe da maioria das amostras de X

retorna

$i \leftarrow Split(X)$

Para cada galho produzido pelo split i **faça**

Divide as entradas do nó de acordo com a função de discriminação

$GeraArvore(X_i)$

Split(X)

$MinEnt \leftarrow MAX$

Para cada atributo de x_i $i = 1, 2, 3, \dots, d$ **faça**

Se x_i é uma variável discreta com n valores possíveis **então**

Divide X em X_1, X_2, \dots, X_n de acordo com x_i

$e \leftarrow EntropiaSplit(X_1, X_2, \dots, X_n)$

Se $e < MinEnt$:

$MinEnt \leftarrow e$;

$bestf \leftarrow i$

Se x_i é uma variável numérica contínua **então**

Para todos os possíveis splits **faça**

Divide X em X_1, X_2 de acordo com x_i

$e \leftarrow EntropiaSplit(X_1, X_2)$

Se $e < MinEnt$ **então**

$MinEnt \leftarrow e$

$bestf \leftarrow i$

retorna $bestf$

Figura A.4 – Pseudo-código da metodologia de construção de uma árvore de classificação

Fonte: Alpaydin (2010)

Uma das características interessantes dos algoritmos de construção de árvores de decisão é que os mesmos são considerados como do tipo *whitebox*, no sentido em que o modelo resultante pode ser traduzido em regras de decisão e explicações explícitas para o modelo preditivo. Por conta disso, os modelos de árvores de decisão são preferidos em contextos práticos em que uma explicação para as previsões é necessária.

A construção de uma árvore de regressão é realizada da mesma maneira exceto pela medida de impureza que é substituída por outra mais adequada para o caso da regressão.

Seja $b_m(\vec{x})$ uma variável binária que indica se a amostra \vec{x} chega até o nó m , ou ainda, uma variável binária que indica se \vec{x} pertence ao subconjunto X_m de amostras que percorrem todo o caminho da árvore até o nó m . No caso de uma árvore de regressão, a impureza de um nó pode ser mensurada pelo erro quadrático médio entre a estimativa do nó g_m e os valores da variável de interesse de cada amostra. Esse erro é dado pela expressão (A.23).

$$E_m = \frac{1}{N_m} \sum_t (y^t - g_m)^2 b_m(\vec{x}^t) \quad (\text{A.23})$$

Usualmente a estimativa dada por um nó em uma árvore de regressão é dada pela média ou mediana dos valores da variável de interesse das amostras de X_m . Nesse caso a expressão (A.23) equivale à medida de variância da estimativa do nó.

Se a impureza medida em um nó é menor que um certo valor limite θ_l o nó é considerado uma folha terminal da árvore e caso contrário, deve ser dividido. Seja b_{mj} uma variável binária que indica se uma amostra que chega ao nó m segue o caminho j após o *split*. Assim, a impureza de um nó após o *split* é dada pela expressão (A.24).

$$E'_m = \frac{1}{N_m} \sum_j \sum_t (y^t - g_{mj})^2 b_{mj}(\vec{x}^t) \quad (\text{A.24})$$

O algoritmo de construção de árvores de classificação busca o split que maximiza a diferença de impureza antes e após a divisão do nó, ou seja, a diferença entre as expressões (A.23) e (A.24). O pseudocódigo apresentado na Figura A.4 pode ser modificado trocando apenas as medidas de impureza de entropia para as medidas de impureza de erro quadrático médio, resultando no pseudocódigo para construção de uma árvore de regressão.

Além das metodologias de construção de árvores de decisão existem algumas técnicas que podem ser utilizadas para melhorar o desempenho desses modelos. Uma dessas técnicas é chamada de *pruning* ou aparamento. Essa técnica consiste em remover sub-árvores de

uma árvore construída para reduzir a possibilidade de sobre-treino e aumentar a capacidade de generalização da árvore. Segundo Alpaydin (2010) a aplicação de pruning após a construção da árvore de decisão por completo (*postpruning*) apresenta melhores resultados do que aplicação de *pruning* ao longo da construção da árvore (*prepruning*).

A técnica de *postpruning* funciona da seguinte forma:

- (i) No início do algoritmo uma porção dos dados é separada (*pruning set*);
- (ii) Após a construção da árvore cada nó da árvore que compõe uma subárvore é substituído por uma folha terminal;
- (iii) Se o desempenho de previsão (classificação ou regressão com as respectivas medidas de erro) da nova árvore não é pior que a árvore sem substituição, então a subárvore é aparada e substituída pelo nó.

Um detalhamento completo dos algoritmos de construção de árvores de decisão pode ser encontrado em Hastie *et al.* (2008).

A.2.3. Máquinas de vetores de suporte

As máquinas de vetores de suporte (SVM – *Support Vector Machines*) representam uma classe de modelos baseados em técnicas de otimização em problemas linearmente separáveis. De acordo com Alpaydin (2010), as SVMs decorrem da aplicação direta dos conceitos da Teoria do Aprendizado Estatístico no contexto de problemas linearmente separáveis.

Existem formulações de SVMs para problemas de classificação binária, problemas de regressão, problemas de identificação de *outliers* e até mesmo problemas de seleção de atributos (*feature extraction*). Esta seção da pesquisa detalha inicialmente algumas formulações das SVMs no contexto de um problema de classificação binária e em seguida generaliza a formulação para os demais contextos.

Seja uma amostra de exemplos de entradas e saídas $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ em que $\mathbf{x}_i \in X$ e $y_n \in \{-1, 1\}$. Considera-se que a amostra é linearmente separável, ou seja, existe

um hiperplano definido por $\mathbf{w} \in X$ e $w_0 \in \mathbb{R}$ que respeita as expressões (A.25) e (A.26) para um par (\mathbf{x}, y) . O vetor \mathbf{w} é o vetor normal ao hiperplano. A SVM aplicada a problemas linearmente separáveis recebe o nome de SVM de margens rígidas.

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + w_0 \geq +1 \rightarrow y = +1 \quad (\text{A.25})$$

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + w_0 \geq -1 \rightarrow y = -1 \quad (\text{A.26})$$

A restrição imposta pelas expressões (A.25) e (A.26) é equivalente a expressão (A.27).

$$y(\langle \mathbf{w} \cdot \mathbf{x} \rangle + w_0) \geq +1 \quad (\text{A.27})$$

O hiperplano de separação ótimo é definido tal que a distância entre o hiperplano e os exemplos mais próximos seja máxima. Essa distância no contexto das SVMs recebe o nome de margem.

A distância entre um exemplo \mathbf{x} e o hiperplano de separação é dada pela expressão (A.28), que pode ser reescrita de forma equivalente conforme a expressão (A.29).

$$\frac{|\langle \mathbf{w} \cdot \mathbf{x} \rangle + w_0|}{\|\mathbf{w}\|} \quad (\text{A.28})$$

$$\frac{y(\langle \mathbf{w} \cdot \mathbf{x} \rangle + w_0)}{\|\mathbf{w}\|} \quad (\text{A.29})$$

É desejável que a distância mínima entre um exemplo qualquer e o hiperplano de separação seja respeitada. Seja a constante ρ a distância mínima entre um hiperplano separador e um exemplo (expressão (A.30)).

$$\frac{y(\langle \mathbf{w} \cdot \mathbf{x} \rangle + w_0)}{\|\mathbf{w}\|} \geq \rho \rightarrow \forall (\mathbf{x}, y) \quad (\text{A.30})$$

Assim, a determinação do hiperplano ótimo se resume a maximizar ρ respeitando a expressão (A.30) para todas as amostras de um conjunto de treinamento. No entanto esse problema de otimização é indeterminado pois o vetor \mathbf{w} e a constante w_0 podem ser

redimensionados mantendo-se a mesma estrutura do hiperplano de separação para infinitos valores de ρ . Para resolver esse problema fixa-se $\rho\|\mathbf{w}\| = 1$ e com isso obtêm-se a chamada representação canônica do hiperplano separador. Com essa nova restrição, a determinação do hiperplano separador ótimo se resume a resolver o problema de otimização representado pelas expressões (A.31) e (A.32) (SMOLA e SCHÖLKOPF, 2002).

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad (\text{A.31})$$

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + w_0) \geq +1 \rightarrow \forall i \quad (\text{A.32})$$

O hiperplano ótimo pode ser analisado sob uma perspectiva geométrica. Seja um exemplo \mathbf{x}_1 um vetor do hiperplano $H_1: \langle \mathbf{w} \cdot \mathbf{x} \rangle + w_0 = 1$ e \mathbf{x}_2 um vetor do hiperplano $H_2: \langle \mathbf{w} \cdot \mathbf{x} \rangle + w_0 = -1$. A projeção da distância entre esses dois vetores da direção do vetor normal ao hiperplano separador é dada pela expressão (A.33).

$$(\mathbf{x}_1 - \mathbf{x}_2) \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \frac{(\mathbf{x}_1 - \mathbf{x}_2)}{\|(\mathbf{x}_1 - \mathbf{x}_2)\|} \right) \quad (\text{A.33})$$

Uma vez que $\langle \mathbf{w} \cdot \mathbf{x}_1 \rangle + w_0 = 1$ e $\langle \mathbf{w} \cdot \mathbf{x}_2 \rangle + w_0 = -1$, substituindo na expressão (A.33), o vetor que representa a distância entre os pontos \mathbf{x}_1 e \mathbf{x}_2 é dado pela expressão (A.34) cuja norma é igual a $2/\|\mathbf{w}\|$.

$$\frac{2}{\|\mathbf{w}\|} \left(\frac{(\mathbf{x}_1 - \mathbf{x}_2)}{\|(\mathbf{x}_1 - \mathbf{x}_2)\|} \right) \quad (\text{A.34})$$

Assim, o hiperplano ótimo divide a distância entre os vetores \mathbf{x}_1 e \mathbf{x}_2 exatamente na metade, maximizando a distância mínima entre o hiperplano e um exemplo da amostra de treinamento. Essa distância ótima é dada por $1/\|\mathbf{w}\|$. A Figura A.5 ilustra a separação promovida pelo hiperplano ótimo.

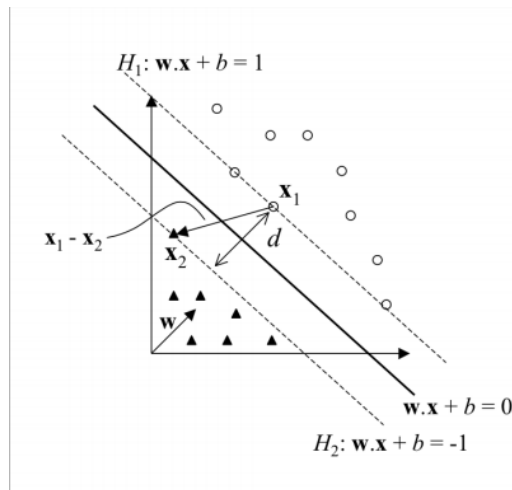


Figura A.5 – Representação geométrica em \mathbb{R}^2 do hiperplano ótimo de separação

Fonte: Lorena e Carvalho (2007)

O problema definido pelas expressões (A.31) e (A.32) é um problema de otimização quadrático cujo tamanho depende da dimensão de \mathbf{w} , ou seja, da dimensão do espaço de características das entradas. É possível alterar a formulação do problema de otimização tal que sua complexidade dependa da quantidade de exemplos da amostra através de uma relaxação lagrangeana. Adicionando as restrições (A.32) na função objetivo (A.31) com os devidos multiplicadores de Lagrange α é possível escrever a função objetivo (A.35).

$$L_p(\mathbf{w}, w_0, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + w_0) - 1] \quad (\text{A.35})$$

O problema de minimização definido por minimizar L_p é chamado de problema primal. Minimizar (A.35) implica em minimizar as variáveis primais \mathbf{w} e w_0 e maximizar as variáveis duais α . Caso uma restrição (A.32) seja violada, ou seja $y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + w_0) - 1 < 0$, a função L_p é aumentada na proporção do respectivo α_i . Ao mesmo tempo, para prevenir que $\alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + w_0) - 1]$ se torne um número negativo arbitrariamente grande, os valores de \mathbf{w} e w_0 eventualmente se ajustam para que a restrição (A.32) seja atendida (SMOLA E SCHÖLKOPF, 2002).

O problema dual de L_p se caracteriza por maximizar L_p com respeito a α garantindo que a derivada de L_p com relação a \mathbf{w} e w_0 seja nula e que $\alpha_i \geq 0$. Isso se traduz nas expressões (A.36) e (A.37). Substituindo (A.36) e (A.37) em (A.35) obtém-se a função objetivo do problema dual L_d (A.38) que depende exclusivamente de α .

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad (\text{A.36})$$

$$\frac{\partial L_p}{\partial w_0} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \quad (\text{A.37})$$

$$L_d(\alpha) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + \sum_{i=1}^n \alpha_i \quad (\text{A.38})$$

O problema dual consiste em maximizar $L_d(\alpha)$ respeitando a restrição (A.37). Observa-se que a dimensão do problema dual depende de n e não mais da dimensão do espaço dos vetores \mathbf{x} . Pode-se demonstrar que o problema dual é convexo e pode ser resolvido por técnicas de otimização quadráticas (*quadratic optimization*) (SMOLA e SCHÖLKOPF; 2002).

Seja α^* a solução do problema dual. O valor ótimo do vetor perpendicular ao hiperplano separador \mathbf{w}^* é obtido pela aplicação direta da expressão (A.36). A determinação do intercepto do hiperplano ótimo w_0^* é determinado por condições de Karush-Kühn-Tucker (KKT) que relacionam a solução de um problema dual com a solução do respectivo problema primal. De acordo com Smola e Schölkopf (2002), as condições de KKT para os problemas primal e dual do hiperplano separador implicam nas expressões (A.39).

$$\alpha_i^* [y_i (\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + w_0^* - 1)] = 0 \rightarrow \forall i \quad (\text{A.39})$$

Observa-se que os multiplicadores α_i^* podem ser diferentes de zero apenas para pontos em que $y_i (\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + w_0^* - 1) = 0$, ou seja, apenas para pontos localizados na margem do hiperplano separador. Esses exemplos para os quais $\alpha_i^* > 0$ que estão localizados na

margem do hiperplano ótimo são chamados de vetores de suporte (*SV – support vectors*). Os vetores de suporte são os dados mais informativos do conjunto de amostras e, sendo os vetores mais próximos da fronteira de decisão do hiperplano separador, são os pontos que de fato são utilizados no cálculo de uma SVM. O intercepto w_0^* é calculado pela média das expressões (A.39) para os vetores de suporte conforme a expressão (A.40) na qual SV representa o subconjunto de vetores de suporte e n_{SV} é o número de vetores de suporte.

$$w_0^* = \frac{1}{n_{SV}} \sum_{i \in SV} \frac{1}{y_i} - \langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle \quad (\text{A.40})$$

A formulação da SVM apresentada funciona apenas no caso de problemas linearmente separáveis, o que na maioria dos casos práticos não se verifica. Para considerar problemas que não são linearmente separáveis pode ser introduzido uma variável de folga nas restrições (A.41), o qual recebe o nome de SVM com margens suaves (*soft margin SVM*).

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + w_0) \geq 1 + \zeta_i \rightarrow \forall i \quad (\text{A.41})$$

Numa SVM de margens suaves, caso ζ_i seja igual a 0, a amostra \mathbf{x}_i é adequadamente separada, caso $0 < \zeta_i < 1$, a amostra é incorretamente separada porém dentro da margem e caso contrário ($\zeta_i > 1$) a amostra é incorretamente separada.

Para determinação do hiperplano ótimo de uma SVM de margens suaves, utiliza-se uma abordagem similar a SVM convencional, porém introduzindo um termo relativo as variáveis de folga na função objetivo. O problema de minimização passa a ser definido pela função objetivo (A.42) em que C é a penalidade atribuída aos erros de separação. O problema primal de uma SVM de margens suaves é minimizar (A.42) respeitando as restrições (A.41).

$$L_p = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \zeta_i \quad (\text{A.42})$$

Assim como nas SVMs com margens rígidas, o problema primal das SVMs de margens suaves pode ser transformado num problema dual cuja solução depende exclusivamente das

amostras de treinamento. Essa transformação é realizada por meio de uma relaxação lagrangeana e tomando derivadas parciais iguais a 0. Essa transformação resulta no problema de otimização quadrático definido pelas expressões (A.43), (A.44) e (A.45).

$$\max L_d(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + \sum_{i=1}^n \alpha_i \quad (\text{A.43})$$

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (\text{A.44})$$

$$0 < \alpha_i < C \rightarrow \forall i \quad (\text{A.45})$$

A solução do problema primal de uma SVM de margens suaves é obtida de forma semelhante a SVM com margens rígidas. Seja $\boldsymbol{\alpha}^*$ a solução do problema dual. O valor da variável primal \mathbf{w}^* é obtido pela aplicação direta da expressão (A.36). As condições KKT para a SVM de margens suaves são dadas pelas expressões (A.46) e (A.47).

$$\alpha_i^* [y_i (\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + w_0^* - 1 + \zeta_i^*)] = 0 \rightarrow \forall i \quad (\text{A.46})$$

$$(C - \alpha_i^*) \zeta_i^* = 0 \rightarrow \forall i \quad (\text{A.47})$$

Os vetores de suporte são as amostras para as quais $\alpha_i^* > 0$, no entanto podem existir dois tipos de vetores de suporte no caso de SVMs de margens suaves. Caso $\alpha_i^* < C$, pela equação (A.47), $\zeta_i^* = 0$ e são denominados vetores de suporte livres. Caso $\alpha_i^* = C$ o valor da variável de folga pode indicar três situações: (i) erro de classificação caso $\zeta_i^* > 1$, (ii) amostras classificadas corretamente dentro da margem caso $0 < \zeta_i^* < 1$, (iii) amostras sobre as margens caso $\zeta_i^* = 0$. Para calcular o intercepto do hiperplano w_0^* utiliza-se a média das expressões (A.46) para os vetores de suporte livres de acordo com a expressão (A.48) em que SV_{livres} é o conjunto dos vetores de suporte livres.

$$w_0^* = \frac{1}{n_{SV_{livres}}} \sum_{i \in SV_{livres}} \frac{1}{y_i} - \langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle \quad (\text{A.48})$$

As SVMs de classificação apresentadas, tanto de margens rígidas quanto de margens suaves, funcionam num contexto em que se deseja diferenciar duas classes. Para casos com múltiplas classes, por exemplo K classes, existem formulações apropriadas. No entanto, o que em geral se faz é resolver K problemas de classificação de duas classes, um para cada classe em que se determinam classificadores específicos de cada classe (ALPAYDIN, 2010).

As formulações apresentadas das SVMs de margens rígidas e suaves tratam de problemas de classificação. O mesmo conceito pode ser empregado para definir SVMs para problemas de regressão. Esse caso também é comumente chamado de *Support Vector Regressor* (SVR).

No contexto de um problema de regressão as saídas da amostra de treinamento y_i não mais representam valores no conjunto $\{-1, +1\}$, mas sim valores reais. Seja $f(\mathbf{x})$ o hiperplano com o qual se deseja representar os dados de treinamento. Ao invés do erro quadrático, comumente utilizado em problemas de regressão, a medida de erro utilizada para definição de uma SVR é chamada de erro ϵ -sensível de acordo com a expressão (A.49). A interpretação dessa medida de erro é que erros de regressão até um valor limite ϵ são tolerados pela SVR.

$$e_\epsilon(y_i, \mathbf{x}_i) = \begin{cases} 0 & \text{caso } |y_i - f(\mathbf{x}_i)| < \epsilon \\ |y_i - f(\mathbf{x}_i)| - \epsilon & \text{caso contrário} \end{cases} \quad (\text{A.49})$$

O problema de definição do hiperplano ótimo de regressão é análogo ao problema do hiperplano de classificação. A Figura A.6 ilustra conceitualmente o problema, sendo que as amostras que se encontram dentro da parte cinza possuem erro nulo de regressão e as demais apresentam erro de regressão conforme a medida de erro ϵ -sensível.

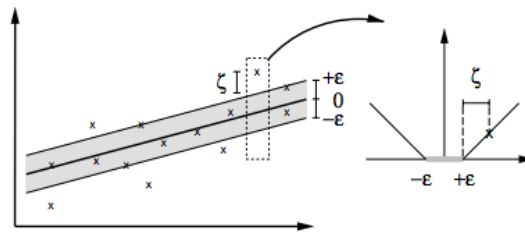


Figura A.6 – Representação geométrica em \mathbb{R}^2 do hiperplano ótimo de regressão

Fonte: Alpaydin (2010)

A expressão (A.50) representa a função objetivo do problema do hiperplano ótimo de regressão, sendo que ζ_i^+ e ζ_i^- são variáveis de folga para as diferenças positivas e negativas do erro de regressão e as expressões (A.51) representam as restrições sobre os erros de regressão.

$$\min L_p = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\zeta_i^+ + \zeta_i^-) \quad (\text{A.50})$$

$$y_i - (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + w_0) \leq \epsilon + \zeta_i^+ \rightarrow \forall i \quad (\text{A.51})$$

$$(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + w_0) - y_i \leq \epsilon + \zeta_i^- \rightarrow \forall i$$

Assim como no caso da SVM de classificação o problema de otimização primal definido pelas expressões (A.50) e (A.51) pode ser transformado em um problema de otimização dual que depende exclusivamente das amostras de treinamento. A transformação é feita através de uma relaxação lagrangeana e derivando a função objetivo resultante com relação as variáveis primais (ALPAYDIN, 2010).

O problema de otimização dual no caso de uma SVR é definido pelas expressões (A.52), (A.53) e (A.54). A função objetivo (A.52) depende apenas os pares de amostras de treinamento (\mathbf{x}_i, y_i) e é uma função dos multiplicadores de lagrange α_i^+ e α_i^- . No caso da SVR existem multiplicadores de lagrange para a parte positiva da restrição (A.51) referente a variável primal ζ_i^+ e multiplicadores para a parte negativa analogamente.

$$\max L_d(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i^+ + \alpha_i^-) (\alpha_j^+ + \alpha_j^-) \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle - \epsilon \sum_{i=1}^n (\alpha_i^+ + \alpha_i^-) - \sum_{i=1}^n y_i (\alpha_i^+ + \alpha_i^-) \quad (\text{A.52})$$

$$0 \leq \alpha_i^+, \alpha_i^- \leq C \rightarrow \forall i \quad (\text{A.53})$$

$$\sum_{i=1}^n (\alpha_i^+ - \alpha_i^-) = 0 \quad (\text{A.54})$$

Para os exemplos localizados dentro da margem ϵ , tem-se $\alpha_i^+ = \alpha_i^- = 0$ sendo esses os exemplos adequadamente descritos pelo hiperplano de regressão. Os vetores de suporte são as amostras para as quais $\alpha_i^+ > 0$ ou $\alpha_i^- > 0$. Para os vetores de suporte localizados exatamente na margem do hiperplano, os multiplicadores de lagrange são $0 < \alpha_i^+ < C$ ou $0 < \alpha_i^- < C$, sendo esses os vetores mais informativos do conjunto. Para os vetores de suporte localizados fora da margem do hiperplano, tem-se $\alpha_i^+ = C$ ou $\alpha_i^- = C$. O cálculo das variáveis primais ótimas \mathbf{w}^* e w_0^* é calculado com base na solução ótima das variáveis duais. O valor de \mathbf{w}^* é calculado conforme a expressão (A.55) que é análoga a expressão referente ao caso do hiperplano de classificação referente ao caso do hiperplano de classificação.

$$\mathbf{w}^* = \sum_{i=1}^n (\alpha_i^+ - \alpha_i^-) \mathbf{x}_i \quad (\text{A.55})$$

Para a determinação de w_0^* devem ser consideradas as condições KKT entre as soluções do problema primal e dual de otimização do hiperplano de regressão. Para os vetores de suporte tal que $0 < \alpha_i^+ < C$ valem as expressões (A.56) e para os vetores de suporte tal que $0 < \alpha_i^- < C$ valem as expressões (A.57). Pode-se calcular w_0^* com a média dos valores obtidos por essas duas expressões para os respectivos vetores de suporte.

$$w_0^* = y_i - \langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle - \epsilon \quad (\text{A.56})$$

$$w_0^* = y_i - \langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + \epsilon \quad (\text{A.57})$$

Uma vez determinados os valores ótimos do hiperplano de regressão, sua formulação passa a ser definida pela expressão (A.58).

$$f(\mathbf{x}) = \left\langle \sum_{i=1}^n (\alpha_i^+ - \alpha_i^-) \mathbf{x}_i, \mathbf{x} \right\rangle + w_0^* \quad (\text{A.58})$$

Os modelos apresentados para a o hiperplano de classificação e de regressão produzem separadores lineares, ou seja, os hiperplanos de separação e de regressão, respectivamente. Contudo, em muitos casos é necessário que os separadores produzam fronteiras de decisão não lineares para os *dados* de treinamento. Isso é possível com o uso de funções *Kernel* (vide seção A1).

Seja uma função *kernel* $k(\mathbf{x}_i, \mathbf{x}_j)$ referente a um mapeamento $\Phi(\mathbf{x})$ não linear. A dimensão do espaço vetorial resultante da aplicação da função *kernel* é em geral maior que a dimensão do espaço original de \mathbf{x} . Substituindo os produtos internos $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ nas expressões para as diferentes SVMs pela função kernel, obtêm-se a formulação das respectivas SVMs não lineares. O tipo de não linearidade depende das funções kernel utilizadas, sendo as mais comuns kernels polinomiais, de base radial, sigmoidais (SMOLA e SCHÖLKOPF, 2002).

De acordo com Alpaydin (2010) a aplicação de *kernels* aumenta a dimensão do espaço das amostras, e confere ao tipo de classificador ou regressor que está sendo buscado maior capacidade de mapeamento das amostras de treinamento. No caso específico das SVMs isso não aumenta a complexidade do problema de aprendizado, pois ao contrário de muitos algoritmos de aprendizado computacional, a complexidade na determinação dos parâmetros do classificador ou regressor depende da quantidade de amostras, e não da quantidade de dimensões das entradas da amostra de treinamento. Isso só é válido pois os problemas de otimização das SVMs são convertidos em suas formais duais.

Schölkopf (1997) apresenta maiores detalhes sobre os algoritmos de otimização das SVMs, bem como detalhes de diferentes formulações. Em Müller et al. (1997) pode ser encontradas aplicações práticas do uso de SVRs para previsão de séries temporais.

A.2.4. Redes Neurais Artificiais

As redes neurais artificiais (RNAs) compreendem todo um ramo de pesquisa com diferentes paradigmas, arquiteturas e categorizações de modelos. Abiodun *et al.* (2018) apresentam uma revisão abrangente do estado da arte em relação a pesquisa e aplicações de RNAs. Os autores sugerem uma taxonomia para as diferentes arquiteturas de RNAs (Figura A.7).

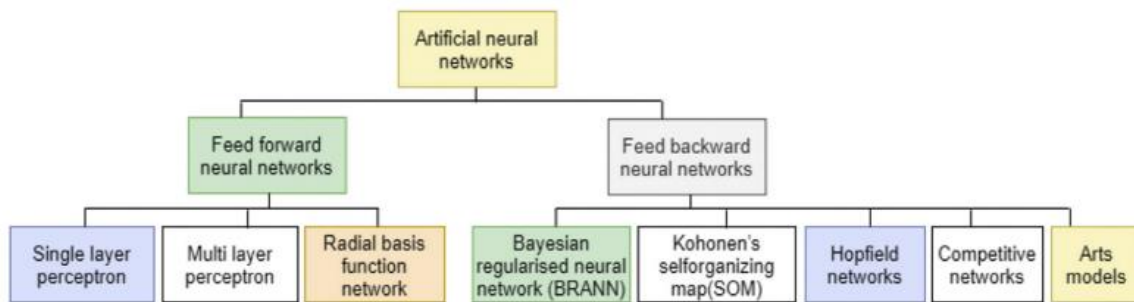


Figura A.7 – Taxonomia de modelos de RNA

Fonte: Abiodun *et al.* (2018)

Com o objetivo de introduzir os principais conceitos das RNAs, esta seção apresenta uma visão clássica de um tipo específico de RNA, chamado de *Redes Neurais FeedForward*. Maiores detalhes das diferentes arquiteturas de RNAs pode ser encontrado em Abiodun *et al.* (2018).

As redes neurais artificiais (RNAs) compreendem toda uma classe de modelos que tem como fundamento um modelo de computação baseado na estrutura do cérebro humano. Durante a década de 1940 descobertas a respeito da fisiologia de redes de neurônios abriram caminho para a proposição de modelos alternativos de computação em relação ao modelo de computação sequencial de Turing (ROJAS; 1996).

Uma RNA é composta por unidades de processamento interconectadas através de canais unidirecionais de transmissão de informação. Essa arquitetura de nós interconectados é capaz de mapear entradas e saídas (*inputs* e *outputs*) assim como uma função.

As unidades de processamento, também chamadas de neurônios ou nós, são definidas em termos de uma função primitiva que recebem *inputs* de outros neurônios e propagam um *output* para a rede (Figura A.8a). Os canais de transmissão de informação, também

denominados de sinapses ou conexões, se encarregam de transmitir *outputs* entre os nós da rede. Cada conexão possui um peso que multiplica o *output* propagado pela conexão para os nós de destino.

A Figura A.8b apresenta uma RNA sem uma arquitetura definida. A rede pode ser encarada como uma função Φ avaliada em (x, y, z) que recebe o nome de função da rede, sendo produzida pela combinação dos inputs da rede, das funções primitivas e pelos padrões de conexão da rede. Diferentes configurações dos pesos das conexões produzem diferentes funções Φ para uma mesma arquitetura da rede.

Uma RNA pode ser utilizada para aproximar funções por exemplo. Dado um conjunto de pares de exemplos de entradas e saídas desejadas de uma função que se deseja aproximar, é possível ajustar a rede para que ela represente a função desejada. O ajuste da rede se concentra na identificação dos pesos das conexões que minimizam uma medida de erro entre as saídas produzidas pela rede e os exemplos de saídas que devem ser mapeados. Esse ajuste recebe o nome de treinamento e pode ser realizado por meio de diferentes algoritmos de otimização dentre os quais se destaca o algoritmo de retropropagação (*Backpropagation*).

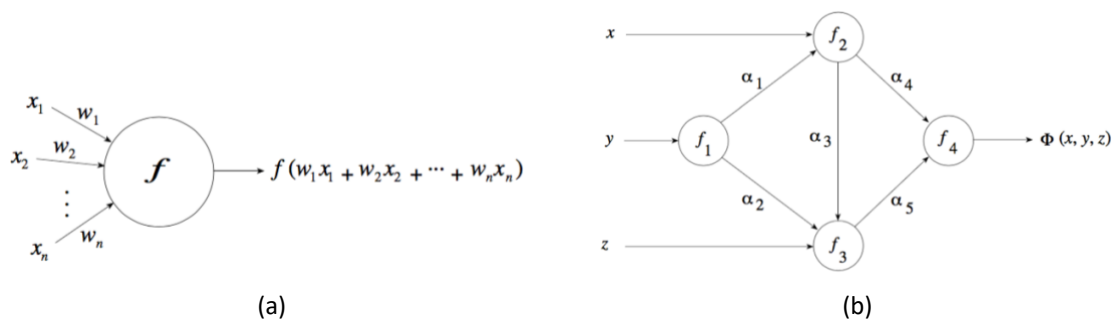


Figura A.8 – Unidade de processamento genérica (a) e rede neural artificial (b)

Fonte: Rojas (1996)

Os três elementos que definem um modelo de RNA são:

1. A estrutura dos nós: diferentes funções primitivas podem ser utilizadas nos nós, como por exemplo funções sigmoidais (tangente hiperbólica, a própria função sigmoide ou

- qualquer função com comportamento saturante), funções limite (*threshold functions*), ou ainda funções lineares (função identidade ou retificadores lineares);
2. A topologia da rede: esta propriedade é dada pela quantidade de nós, pelo padrão de interconexão entre os nós e pelo sincronismo de processamento da informação. Existem algumas denominações de topologias específicas, como é o caso das redes *feedforward*, redes recorrentes e redes síncronas e assíncronas.
 3. O algoritmo de aprendizado da rede: diferentes técnicas de otimização podem ser aplicadas para encontrar os pesos da rede que mapeiam uma função. Os mais populares são os métodos de descida em gradiente, mas em teoria, qualquer algoritmo de otimização não-linear pode ser empregado no treinamento de uma RNA.

Rojas (1996) apresenta um detalhamento histórico do desenvolvimento das redes neurais, desde sua primeira concepção até as atuais técnicas de modelagem. O autor comenta que os elementos fundamentais de um modelo computacional podem ser metaforicamente representados pela expressão (A.59).

$$computacao = \begin{matrix} \textit{armazenamento} \\ \textit{(pesos)} \end{matrix} + \begin{matrix} \textit{transmissão} \\ \textit{(topologia)} \end{matrix} + \begin{matrix} \textit{processamento} \\ \textit{(nós)} \end{matrix} \quad (A.59)$$

No caso das RNAs o armazenamento de informação se dá nos pesos da rede que armazenam todo o conhecimento da mesma, a transmissão ocorre entre os neurônios pelas conexões que definem a topologia da rede e o processamento em si ocorre nas unidades de processamento.

Um dos principais modelos de RNA é o modelo *Perceptron* Multicamadas (*Multilayer Perceptron* - MLP) proposto por Rosenblatt (1958) e aprimorado por Minsky e Pappert. Uma rede MLP é uma rede do tipo *feedforward* com múltiplas camadas de nós e com conexão completa (Figura A.9). Uma rede MLP possui um conjunto de sinais de entrada que recebem estímulos que, por sua vez, representam os *inputs* da rede. A rede possui conjuntos de nós organizados em camadas intermediárias. Todos os nós entre camadas consecutivas são conectados entre si. Na saída da rede existe um conjunto de nós de *output* que produzem a saída desejada da rede.

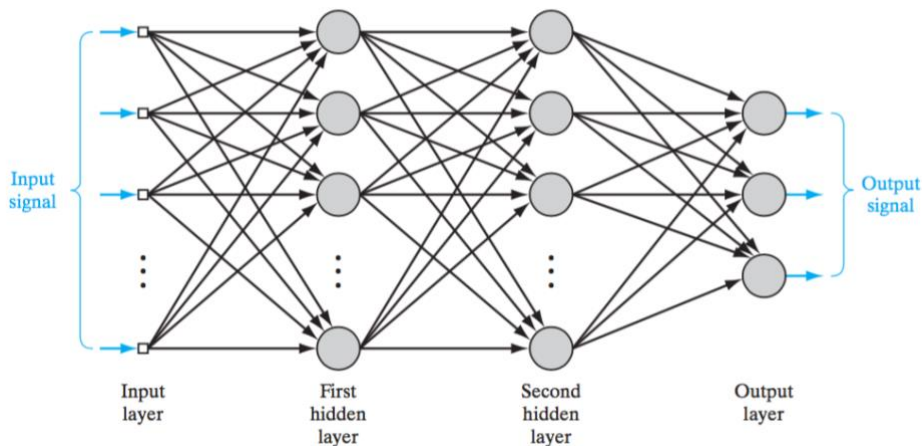


Figura A.9 – Multilayer Perceptron

Fonte: Haykin (2009)

Nesta seção será apresentado o funcionamento do algoritmo *backpropagation* para o caso de redes do tipo MLP. Todavia, o algoritmo *backpropagation* pode ser utilizado em qualquer outra arquitetura de RNA desde que as funções primitivas dos nós sejam diferenciáveis. Para uma visão mais abrangente do funcionamento do algoritmo *backpropagation* em outras arquiteturas de RNA pode-se consultar Rojas (1996) ou Haykin (2009).

Seja uma rede R do tipo MLP com funções primitivas contínuas e diferenciáveis em cada um de seus nós. Seja o conjunto de treinamento $\{\vec{x}(n), \vec{d}(n)\}_{n=1}^N$ composto por entradas \vec{x} e saídas desejadas \vec{d} . Seja $y(n)_j$ a saída produzida no neurônio j da camada de *output* pela entrada $\vec{x}(n)$. Seja w_{ji} o peso da conexão entre o neurônio i e neurônio j . O erro de aproximação da rede produzido pela entrada $\vec{x}(n)$ no neurônio de saída j é dado pela expressão (A.60) em que $d(n)_j$ representa o j -ésimo elemento do vetor de saída desejada $\vec{d}(n)$.

$$e(n)_j = d(n)_j - y(n)_j \quad (\text{A.60})$$

Seja $E(n)_j$ uma medida de erro quadrático do neurônio j da rede. O termo $1/2$ é inserido na expressão para facilitar manipulações algébricas da derivada do erro (expressão (A.61)).

$$E(n)_j = \frac{1}{2} e(n)_j^2 \quad (\text{A.61})$$

Dessa forma, o erro total da rede é dado pela soma dos erros quadráticos de todos os neurônios da camada de saída representados pelo conjunto C na expressão (A.62).

$$E(n) = \frac{1}{2} \sum_{j \in C} E(n)_j^2 \quad (\text{A.62})$$

A expressão (A.62) considera apenas uma amostra do conjunto de treinamento. Para que todos os pares de amostras sejam considerados, basta considerar a média dos erros da rede para o conjunto de amostras de acordo com a expressão (A.63).

$$E_{avg}(N) = \frac{1}{N} \sum_{n=1}^N E(n) = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} E(n)_j^2 \quad (\text{A.63})$$

Uma vez que os outputs da rede são definidos pelos pesos das conexões, as medidas de erro definidas pelas expressões (A.60), (A.61), (A.62) e (A.63) também podem ser encaradas como uma função dos pesos.

A entrada de um neurônio j é dada pela combinação de *outputs* dos m neurônios conectados ao mesmo (expressão (A.64)). A saída do neurônio j para a entrada $x(n)$ é dada pela expressão (A.65) em que φ_j representa a função primitiva do neurônio j .

$$v(n)_j = \sum_{i=0}^m w_{ji} y(n)_i \quad (\text{A.64})$$

$$y(n)_j = \varphi_j(v(n)_j) \quad (\text{A.65})$$

A derivada parcial da função de erro de aproximação com relação ao peso w_{ji} pode ser definida pela expressão (A.66) que consiste na aplicação da regra da cadeia à expressão (A.62) e cálculo das respectivas derivadas.

$$\frac{\partial E(n)}{\partial w_{ij}} = \frac{\partial E}{\partial e_j} \frac{\partial e_j}{\partial y_j} \frac{\partial y_j}{\partial v_j} \frac{\partial v_j}{\partial w_{ji}} = -e(n)_j \varphi'_j(v(n)_j) y(n)_i \quad (\text{A.66})$$

A cada iteração do algoritmo *backpropagation* são calculadas as derivadas parciais com relação a cada peso da rede e são realizadas atualizações nos pesos no sentido contrário a derivada, com o intuito de ajustar os pesos da rede para que seja atingido o mínimo da função de erro de aproximação. O ajuste aplicado a cada peso da rede é dado pela expressão (A.67) em que η é uma constante de aprendizado e $\delta(n)_j$ é denominado gradiente local do neurônio j , dado por $-e(n)_j \varphi'_j(v(n)_j)$.

$$\Delta w_{ij}(n) = -\eta \frac{\partial E(n)}{\partial w_{ij}} = \eta \delta(n)_j y(n)_i \quad (\text{A.67})$$

Da equação (A.67), percebe-se que um fator importante do algoritmo é o cálculo dos erros de aproximação dos neurônios $e(n)_j$. Existem dois casos distintos:

1. O neurônio j pertence a camada de saída: este caso é simplesmente resolvido calculando o erro como a diferença entre a saída do neurônio j com o valor desejado $d(n)_j$, dado pelo conjunto de treinamento. Em outras palavras, basta a aplicação direta da expressão (A.60);
2. O neurônio j pertence a uma camada intermediária: neste caso não há saída desejada explicitamente definida no conjunto de treinamento. O erro local do neurônio deve então ser calculado de forma recursiva dependendo dos neurônios conectados com o neurônio j de acordo com a expressão (A.68) em que o conjunto k representa os neurônios numa camada posterior a camada do neurônio j . Percebe-se que para o cálculo do gradiente local do neurônio j é necessário conhecer o gradiente local de todos os neurônios k da camada imediatamente posterior a camada do neurônio j . De posse do gradiente local do neurônio j , a mesma regra de atualização de pesos dada pela expressão (A.67) pode ser utilizada para ajustar o peso do neurônio da camada intermediária.

$$\delta(n)_j = \varphi'_j(v(n)_j) \sum_k \delta(n)_k w_{kj} \quad (\text{A.68})$$

Cabe mencionar que o algoritmo *backpropagation* pode ser executado em bateladas de amostras (*batch*) ou de forma *on-line*. Na modalidade batelada, todas as amostras são apresentadas para a rede ao mesmo tempo e nenhum ajuste de pesos é realizado. São calculados os valores dos ajustes que seriam realizados individualmente para cada amostra $\Delta w_{ij}(n)$ e uma vez que todas as amostras são processadas, calcula-se o ajuste médio (A.69). A apresentação de todas as amostras para a rede compreende uma época de treinamento. Na modalidade batelada, o algoritmo *backpropagation* realiza apenas um ajuste de pesos por época.

$$\Delta \bar{w}_{ij}(n) = \frac{1}{N} \sum_{k=1}^N \Delta w_{ij}(n) \quad (\text{A.69})$$

Na modalidade online, o algoritmo apresenta sequencialmente as entradas de treinamento da rede e a cada entrada realiza um ajuste de pesos, realizando múltiplos ajustes por época de treinamento.

Podem ser definidos diferentes critérios de parada para o algoritmo sendo os mais comuns:

1. O algoritmo para uma vez que uma quantidade de épocas pré-determinada é processada;
2. O algoritmo para se, por uma determinada quantidade de épocas consecutivas, o erro quadrático de aproximação não apresenta redução maior que um certo valor limite;
3. O algoritmo para se a norma do gradiente para o conjunto atual de pesos é menor que um certo valor limite.

Com base nas equações apresentadas, pode-se dizer que o algoritmo *backpropagation* é composto de uma forma de cálculo das derivadas parciais da função de erro em relação aos pesos da rede e de uma função de ajuste dos pesos para minimizar a função de erro. Os

passos básicos do algoritmo no caso de aprendizado em batelada são detalhados na Figura A.10 (o pseudo-código é apenas uma das formas de aplicação do algoritmo de *backpropagation* visando apresentar apenas a intuição do método). A cada iteração, duas fases de computação são realizadas:

1. Fase de propagação (*forward pass*): na primeira fase do algoritmo a rede calcula as saídas de cada neurônio com os pesos atuais. Em cada neurônio são armazenados os valores do *output* calculado, bem como a derivada da função primitiva em relação ao *input* ponderado do neurônio;
2. Fase de retropropagação (*backward pass*): na segunda fase, o erro de aproximação é retropropagado pela rede para cálculo dos valores da derivada parcial da função de erro em relação aos pesos da rede. Procede-se também nessa fase o ajuste dos pesos de acordo com a expressão (A.67).

Backpropagation(*MLP, Features_Treinamento, Target_Treinamento, Taxa_Aprendizado, Criterio_Parada*)

```
pesos <- inicializa_pesos()
saida <- MLP(pesos, Conjunto_Features_Treinamento)
delta <- saida - Conjunto_Target_Treinamento
enquanto delta > Criterio_Parada faça
    gradientes <- calcula_gradientes(pesos, delta)
    pesos <- pesos - Taxa_Aprendizado * gradientes
    saida <- MLP(pesos, Conjunto_Features_Treinamento)
    delta <- saida - Conjunto_Target_Treinamento
retorna pesos
```

Figura A.10 – Pseudo-código do algoritmo backpropagation na modalidade batelada

Fonte: Haykin (2009)

O algoritmo pode ser implementado de forma eficiente utilizando uma notação matricial. Seja \vec{y}^i o vetor com os outputs produzidos pelos neurônios da i -ésima camada da rede e seja W^i a matriz de pesos das conexões entre neurônios de camadas subsequentes i e $(i-1)$.

$$\vec{y}^i = \varphi(\vec{y}^{i-1} \cdot W^i) \quad (\text{A.70})$$

Seja D^i a matriz diagonal com as derivadas das funções primitivas de cada neurônio da camada i em relação aos inputs ponderados. Assumindo que a camada i possui l neurônios, D^i é uma matriz quadrada ($l \times l$) com termos não nulos apenas na diagonal principal.

$$D^i = \{\varphi'_i(v_i) \mid i = j\}_{l \times l} \quad (\text{A.71})$$

Seja \vec{e} o vetor de erros locais dos neurônios da camada de *output* da rede, em que cada elemento do vetor é dado por $y_j - d_j$.

$$\vec{e} = \begin{pmatrix} y_1 - d_1 \\ y_2 - d_2 \\ \dots \\ y_N - d_N \end{pmatrix} \quad (\text{A.72})$$

O vetor de gradientes locais dos neurônios da camada de output é dado pela expressão (A.73). O vetor dos gradientes locais das demais camadas da rede podem ser calculados de forma recursiva, de acordo com a expressão (A.74). A correção de pesos na forma matricial é definida pela expressão (A.75).

$$\vec{\delta}^{output} = D^{output} \cdot \vec{e} \quad (\text{A.73})$$

$$\vec{\delta}^i = D^i W^{i+1} \vec{\delta}^{i+1} \quad (\text{A.74})$$

$$\Delta W^i = -\eta \vec{\delta}^i \vec{y}^i \quad (\text{A.75})$$

As expressões (A.73), (A.74) e (A.75) permitem uma implementação eficiente do algoritmo por meio de operações matriciais.

O algoritmo *backpropagation* fornece a cada iteração uma estimativa da trajetória do gradiente da função de erro no espaço de pesos. Dessa forma, a convergência do algoritmo depende em grande parte da magnitude do ajuste dos pesos que é realizado a cada iteração. Quanto menor é a constante η , menor será a magnitude do ajuste de pesos, mais suave será

a trajetória percorrida pelo algoritmo a cada iteração e mais demorada será a convergência até um ponto de mínimo. Quanto maior a constante η maior será a magnitude do ajuste de pesos e maior a chance de rápida convergência do algoritmo, porém maior também será a chance de ser percorrida uma trajetória divergente fugindo de um ponto de mínimo.

A regra de atualização de pesos definida pela expressão (A.67) e de forma matricial pela expressão (A.75) pode ser generalizada pela regra delta generalizada para ajuste de pesos, definida pela expressão (A.76) em que α é uma constante de *momentum*. Essa regra permite constantes de aprendizado maiores com menor risco de instabilidade do algoritmo.

$$\Delta w_{ij}(n) = \alpha \Delta w_{ij}(n-1) + \eta \delta(n)_j y(n)_i \quad (\text{A.76})$$

A regra delta generalizada é uma regra iterativa de ajuste de pesos que considera o ajuste calculado da última amostra, ponderado por uma constante α . A regra pode ser reescrita de forma recursiva resultando na expressão (A.77). Pode-se observar que a regra delta generalizada é uma forma de suavização exponencial do ajuste de pesos ao longo das amostras que são apresentadas na rede.

$$\Delta w_{ij}(n) = \eta \sum_{t=0}^n \alpha^{n-t} \delta(t)_j y(t)_i \quad (\text{A.77})$$

A expansão de (A.76) em (A.77) também revela que:

1. O ajuste de pesos atual é uma soma ponderada de pesos anteriores. Para que a soma seja convergente é necessário que $0 \leq |\alpha| < 1$;
2. Quando o sinal da derivada parcial é o mesmo em iterações sucessivas, o valor do ajuste de pesos cresce. Isso significa que a convergência do algoritmo se acelera na direção de trajetórias continuamente decrescentes da função de erro;
3. Quando o sinal da derivada parcial muda entre iterações consecutivas, a magnitude do ajuste fica menor. Isso significa que a regra delta generalizada possui um efeito de

estabilizar a trajetória do algoritmo diminuindo o ajuste em regiões de oscilação da superfície da função de erro.

Além do algoritmo *backpropagation* apresentado nessa seção, existem outras modificações que podem ser realizadas assim como outros algoritmos de treinamento de RNAs. Por exemplo, podem ser utilizadas constantes de treinamento independentes por conexão η_{ij} , diferentes funções de ativação por camada, entre outras. Outros algoritmos de aprendizado baseados na segunda derivada da função de erro no espaço de pesos também podem ser aplicados. Rojas (1996) apresenta uma revisão abrangente do algoritmo *backpropagation* e suas diferentes adaptações. O autor também descreve em detalhes diferentes algoritmos de treinamento de RNAs baseados na segunda derivada da função de erro.

Zhang *et al.* (1998) apresentam uma revisão dos passos para a construção de modelos de RNA. Os autores afirmam que as definições necessárias para a construção de uma RNA para um determinado problema são:

- (i) Escolha da arquitetura: a determinação da arquitetura de uma rede envolve a definição da quantidade de neurônios em cada camada, a quantidade de camadas e as conexões entre neurônios. A quantidade de neurônios em camadas intermediárias da rede confere a habilidade de mapear relações não-lineares entre as entradas e saídas, porém um excesso de nós nas camadas intermediárias ou um excesso de camadas intermediárias pode prejudicar a capacidade de generalização da rede. Quanto a determinação da quantidade de neurônios na camada de entrada, em redes que utilizam termos atrasados de uma série temporal, a escolha da quantidade de neurônios determina a quantidade de termos autoregressivos a serem utilizados no modelo, sendo essa uma das principais atividades na modelagem de RNAs para previsão de séries temporais;
- (ii) Escolha das funções de ativação: a função de ativação confere a RNA a capacidade de modelar estruturas não-lineares de correlação entre entradas e saídas. Qualquer função diferenciável, monotonicamente crescente e limitada superior e inferiormente pode ser utilizada como função de ativação de um neurônio, sendo as mais tipicamente utilizadas: (i) a função logística ou sigmoide,

- (ii) a tangente hiperbólica, (iii) o seno ou cosseno e (iv) a função linear $f(x) = x$. Não existe base teórica para determinação da melhor função de ativação para diferentes tipos de rede, nem comprovação empírica que um tipo de função tenha melhor desempenho que outra;
- (iii) O algoritmo de treinamento: o algoritmo de treinamento de uma RNA consiste em um problema de minimização não-linear irrestrito no qual os pesos das conexões da rede são iterativamente ajustados para minimizar uma função de erro de previsão, também denominada de função de custo da RNA. Existem vários algoritmos de treinamento possíveis, apesar de não existir um método geral de otimização de problemas não-lineares que garanta o atingimento de ótimos globais em tempo computacional razoável. Sendo assim, qualquer algoritmo de treinamento da RNAs está sujeito a problemas de velocidade de convergência e parada em ótimos locais;
- (iv) Os métodos de normalização e desnormalização dos dados: a normalização de dados é um processo realizado para ajustar os valores das entradas e saídas para uma faixa em geral entre $[-1, 1]$ ou $[0,1]$. Este é um processo necessário, visto que as funções de transferência dos neurônios possuem uma característica de achatamento nos valores extremos. A normalização dos dados pode ocorrer de quatro formas diferentes: (i) normalização em cada dimensão das entradas da RNA, (ii) normalização considerando todos os valores das amostras no conjunto de dados, (iii) uma combinação de normalização por dimensão e considerando todos os valores dos dados das amostras e (iv) normalização considerando uma faixa fixa de valores pré-definida. Não existem recomendações práticas de qual o melhor método de normalização para diferentes aplicações de RNAs;
- (v) A separação dos dados em amostras de treino e teste: a separação de dados para treinamento e teste é uma etapa fundamental para modelagem de RNAs. Uma amostra de treino muito pequena pode fazer com que a rede não seja capaz de capturar os padrões presentes nos dados e uma amostra muito grande pode causar uma superespecialização (*overfitting*) da rede. Uma amostra de teste

muito pequena pode superestimar a capacidade de generalização de uma rede treinada e uma amostra de teste muito grande pode inviabilizar o treinamento efetivo da rede. Como regra prática, em geral utiliza-se 80% das amostras disponíveis para treinamento e os 20% restantes para teste. Em alguns casos, pode-se separar os dados em múltiplos conjuntos e realizar múltiplos treinamentos e testes, fazendo com que sejam apresentadas amostras diversas para treinamento da rede, como é o caso do algoritmo *K-fold* para separação de dados;

- (vi) Medidas de desempenho: não existe medida de desempenho de RNAs amplamente aceita, porém todas são de alguma forma relacionadas com a precisão das previsões da rede. As medidas mais frequentemente utilizadas são o desvio médio absoluto, a soma de erros quadráticos, a média dos erros quadráticos e o erro percentual absoluto médio.

A.2.5. Redes Neurais Fuzzy

Pode-se definir a lógica fuzzy como sendo uma ferramenta capaz de capturar informações vagas, em geral descritas em linguagem natural e convertê-las para um formato numérico que pode ser manipulado por computadores (TANAKA, 1997).

Os primeiros conceitos da lógica Fuzzy foram introduzidos por Zadeh (1965) com o objetivo de dar um tratamento matemático a definições imprecisas na teoria dos conjuntos e na lógica clássica.

Os modelos que utilizam lógica Fuzzy possuem uma inerente capacidade de interface com seres humanos, o que motivou o aparecimento de modelos de previsão de vendas com o uso desse tipo de técnica. Em especial existem casos em que as variáveis de entrada dos modelos de previsão possuem definição imprecisa, ou mesmo casos em que a resposta de um modelo de previsão deve ser dada considerando possíveis desvios decorrentes da definição das variáveis de entrada.

Ishibuchi, Okada e Tanaka (1993) propuseram um modelo de previsão denominado *Fuzzy Neural Network* (FNN). Trata-se de uma rede neural com arquitetura do tipo *feedforward*

em que os pesos das conexões da rede e os vieses são números *fuzzy* triangulares simétricos. As entradas da rede são vetores de números reais e a saída é composta por um número *fuzzy* triangular.

A função de pertinência de um número *fuzzy* triangular simétrico A pode ser definida conforme a expressão (A.78) em que a_L é o limite inferior de A, a_C é o ponto central de A e a_U é o limite superior de A. O número fuzzy A também pode ser representando pela tripla (a_L, a_C, a_U) .

$$\mu_A(x) = \begin{cases} 0 & \rightarrow x \leq a_L \\ (x - a_L) / (a_C - a_L) & \rightarrow a_L < x \leq a_C \\ (a_U - x) / (a_U - a_C) & \rightarrow a_C < x \leq a_U \\ 0 & \rightarrow x > a_U \end{cases} \quad (\text{A.78})$$

No modelo FNN, os pesos W da rede neural e os vieses Θ são números *fuzzy* triangulares simétricos definidos pelas respectivas triplas de limites inferiores, pontos centrais e limites superiores. A Figura A.11 ilustra a arquitetura da rede FNN omitindo os vieses.

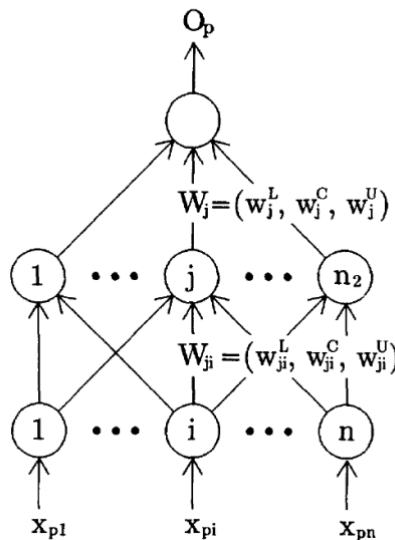


Figura A.11 – Arquitetura do modelo FNN

Fonte: Ishibuchi *et al.* (1993)

A arquitetura da rede FNN contém uma camada de *inputs* com n neurônios sendo n a dimensão do vetor de entrada da rede, uma camada intermediária com n_2 neurônios e uma camada de *output* com um neurônio de saída.

A relação entre uma amostra de entrada x_p e uma saída fuzzy O_p é dada pelo princípio da extensão de Zadeh representado pela expressão (A.79) e ilustrado pela Figura A.11.

$$\mu_{f(A)}(y) = \max_{x:f(x)=y} \mu_A(x) \tag{A.79}$$

Para a compreensão da metodologia proposta pelos autores, é necessária a compreensão dos h -níveis de um número *fuzzy*. Seja α um número real entre 0 e 1, o h -nível de um número *fuzzy* é tal que $\mu_A(x) \geq h$.

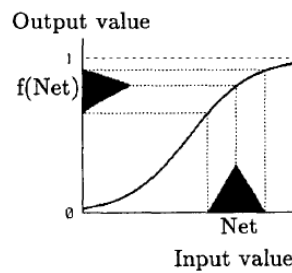


Figura A.12 – Ilustração do princípio da extensão de Zadeh

Fonte: Ishibuchi *et al.* (1993)

No modelo proposto pelos autores, assume-se que existem m pares de *inputs* e *outputs* (x_p, T_p) em que x_p é um vetor de números reais de entrada e T_p é um número *fuzzy* conhecido descrito por uma tripla (t_p^L, t_p^C, t_p^U) . Sejam $[.]_h^L$ e $[.]_h^U$ os limites inferior e superior do h -nível de um número *fuzzy* respectivamente, o erro de previsão de O_p em relação à amostra (x_p, T_p) é dado pela expressão (A.80).

$$e_{ph} = \frac{([T_p]_h^L - [O_p]_h^L)^2}{2} + \frac{([T_p]_h^U - [O_p]_h^U)^2}{2} \tag{A.80}$$

Pode ser observado que a função de erro depende do valor de h utilizado para avaliar os h -níveis do *target* e da previsão *fuzzy*. Os autores sugerem o cálculo do erro para diferentes valores de h e definem o erro de previsão de acordo com a expressão (A.81).

$$e_p = \sum_h h e_{ph} \quad (\text{A.81})$$

Com base na definição de erro de previsão do *output fuzzy* O_p em relação ao *target fuzzy* T_p os autores propõem um algoritmo de treinamento da rede FNN que visa minimizar a função de erro. O treinamento da rede FNN consiste no ajuste dos limites inferior e superior dos pesos e dos vieses da rede de acordo com a derivada do erro de previsão. Dadas duas iterações consecutivas definidas pelos índices t e $t+1$, as regras de ajuste dos pesos da rede FNN são das pelas expressões (A.82), (A.83), (A.84) e (A.85) em que η e α são constantes de aprendizado e de *momentum* respectivamente.

$$\Delta w_j^L(t+1) = -\eta h \left(\frac{\partial e_{ph}}{\partial w_j^L} \right) + \alpha \Delta w_j^L(t) \quad (\text{A.82})$$

$$\Delta w_j^U(t+1) = -\eta h \left(\frac{\partial e_{ph}}{\partial w_j^U} \right) + \alpha \Delta w_j^U(t) \quad (\text{A.83})$$

$$\Delta w_{ij}^L(t+1) = -\eta h \left(\frac{\partial e_{ph}}{\partial w_{ij}^L} \right) + \alpha \Delta w_{ij}^L(t) \quad (\text{A.84})$$

$$\Delta w_{ij}^U(t+1) = -\eta h \left(\frac{\partial e_{ph}}{\partial w_{ij}^U} \right) + \alpha \Delta w_{ij}^U(t) \quad (\text{A.85})$$

São realizados experimentos numéricos que validam o algoritmo proposto por Ishibuchi *et al.* (1993), mostrando a aderência da rede FNN a um conjunto de dados conhecidos. Os autores estendem o modelo FNN proposto para o caso em que as entradas da rede FNN são compostas por um vetor de números *fuzzy* e a saída é também um número *fuzzy*, validando as expressões de ajuste de pesos para esse caso.

A rede FNN pode ser considerada como uma extensão da RNA *fuzzy* proposta por Ishibushi, Fujioka e Tanaka (1992) que é capaz de realizar o mapeamento de *inputs fuzzy* e *outputs fuzzy* com base em um algoritmo de treinamento similar, porém com pesos e vieses compostos apenas de números reais.

A.2.6. Gradient Boosting Machines

De acordo com Natekin e Knoll (2013), muitos problemas de aprendizado computacional se resumem a construir um único modelo baseado num conjunto de dados coletados sobre um fenômeno do qual se possui pouco ou nenhum conhecimento teórico. O procedimento comumente utilizado nesses casos é ajustar um modelo não paramétrico, como uma rede neural ou uma máquina de vetor de suporte, aos dados do problema. Isso representa uma tentativa de construir um único modelo robusto considerando os dados disponíveis do problema. Uma abordagem alternativa é construir um conjunto de modelos mais simples (*ensembles*) que combinados representam de forma robusta o fenômeno de interesse.

Existem diversas formas de combinar modelos para produção de conjuntos eficientes. Dentre as formas de combinar modelos em *ensembles* existe a possibilidade de combinar modelos de forma construtiva e sequencial de modo a gradativamente especializar o modelo resultante em relação aos dados apresentados.

Nesse contexto as *gradient boosting machines* (GBMs) representam uma técnica de combinação sequencial de modelos de previsão numa estrutura sequencial para um determinado problema de aprendizado computacional, representando uma técnica construtiva de *ensembles* de modelos.

O conceito das GBMs consiste em sequencialmente aplicar modelos simples, também chamados de *weak learners* ou *base learners*, para a produção de um modelo mais robusto. A metodologia de construção de GBMs, conforme proposta por Friedman (2001), pode ser vista como um método de descida em gradiente em que cada passo consiste em ajustar uma função aos resíduos de previsão do passo anterior.

Tipicamente em um problema de aprendizado, deseja-se ajustar uma função $F(x)$ a um conjunto de pares de exemplos $\{x, y\}_{i=0}^N$ que representam instâncias de variáveis aleatórias

X e Y com distribuições de probabilidade $P(x)$ e $P(y)$ respectivamente. O objetivo do problema é encontrar uma função F^* que minimize uma função de erro ou perda (*loss function*) sob a distribuição de probabilidades de x e y . A expressão (A.86) apresenta essa definição do problema em que $L(.)$ é uma função de perda e E representa o operador de esperança.

$$F^* = \arg \min_F E_{y,x} L(y, F(x)) = \arg \min_F E_x [E_y L(y, F(x)) | x] \quad (\text{A.86})$$

As funções de erro comumente usadas são, por exemplo, o desvio quadrático e o erro absoluto para problemas de regressão.

Uma possível abordagem para o problema (A.86) é restringir a função F a uma classe de funções parametrizadas $F(x, P)$ em que P é o conjunto de parâmetros. A metodologia das GBMs tem como foco expansões aditivas para a função F da forma definida pela expressão (A.87) em que $h(.)$ representa a classe de funções de F (funções de base), a_i representa um conjunto possível de parâmetros de h e β representa um coeficiente real.

$$F(x, \{\beta_m, a_m\}_1^M) = \sum_{m=1}^M \beta_m h(x, a_m) \quad (\text{A.87})$$

A escolha de um modelo parametrizado $F(x, P)$ muda o problema (A.86) para um problema de otimização de parâmetros (A.88) em que P^* é o conjunto ótimo de parâmetros e $F^*(x) = F(x, P^*)$.

$$P^* = \arg \min_P E_x [E_y L(y, F(x, P)) | x] \quad (\text{A.88})$$

Muitos métodos de otimização realizam estimativas consecutivas dos parâmetros da função F , de modo que o conjunto de parâmetros ótimos pode ser escrito como uma sequência de incrementos, também chamados de *steps* ou *boosts*, conforme a expressão (A.89), em que p_m são os *steps* e p_0 é uma estimativa inicial dos parâmetros.

$$P^* = \sum_{m=0}^M p_m \quad (\text{A.89})$$

A solução do problema de otimização dos parâmetros pode ser encontrada por um método de descida em gradiente que é um dos métodos mais diretos e simples de otimização de funções. A cada iteração m do método, a descida em gradiente se baseia no cálculo do gradiente \mathbf{g}_m da função de perda com relação aos parâmetros a serem otimizados que no caso do problema de aprendizado são os parâmetros da função a ser ajustada (expressão (A.90)).

$$\mathbf{g}_m = \{\mathbf{g}_{jm}\} = \left\{ \left[\frac{\partial E_{y,x}L(y, F(x, P))}{\partial P_j} \right]_{P=P_{m-1}} \right\} \quad (\text{A.90})$$

O gradiente da expressão (A.90) indica a direção de máximo crescimento da função de perda em relação a variação dos parâmetros da função de ajuste. Uma vez que problema em questão é de minimização da função de perda, deve-se reduzir o valor da função de perda ajustando os parâmetros da função de ajuste na direção contrária ao gradiente (A.90). O *step* de atualização dos parâmetros da função de ajuste para redução do valor da função de perda é dado pela expressão (A.91) em que ρ_m é uma grandeza real representando a magnitude do *step*.

$$\mathbf{p}_m = -\rho_m \mathbf{g}_m \quad (\text{A.91})$$

A magnitude do *step* pode ser encontrada pela solução do problema de minimização dado pelas expressões (A.92) e (A.93), também chamado de problema de busca em linha (*line search problem*).

$$\rho_m = \arg \min_{\rho} \Phi(P_{m-1} - \rho \mathbf{g}_m) \quad (\text{A.92})$$

$$\Phi(P) = E_{y,x}L(y, F(x, P)) \quad (\text{A.93})$$

O problema de otimização (A.88) é definido no espaço de parâmetros da função de ajuste. No caso das GBMs o problema deve ser definido de forma não paramétrica no espaço de funções. De acordo com Natekin e Knoll (2013) a principal diferença das GBMs para outros métodos de aprendizado computacional é que a otimização da função de perda é realizada no espaço de funções ao invés do espaço de parâmetros.

No caso do problema definido no espaço das funções, cada $F(x)$ é considerada um parâmetro para um problema de minimização de uma função de perda e deseja-se minimizar uma função de perda (A.94) para cada valor possível de x .

$$\phi(F(x)) = E_y[L(y, F(x))|x] \rightarrow \forall x \quad (\text{A.94})$$

A solução para o problema (A.94) pode ser escrita como uma composição aditiva de funções f_m , análogas aos *steps* ou *boosts* no espaço de parâmetros, conforme a expressão (A.95) em que f_0 é uma estimativa inicial.

$$F^*(x) = \sum_{m=0}^M f_m(x) \quad (\text{A.95})$$

A busca pela função F^* pode também ser realizada por um método de descida em gradiente. Nesse caso o gradiente em uma iteração do método é definido de acordo com a expressão (A.96).

$$g_m(x) = \left[\frac{\partial \phi(F(x))}{\partial F(x)} \right]_{F=F_{m-1}} = \left[\frac{\partial E_y[L(y, F(x))|x]}{\partial F(x)} \right]_{F=F_{m-1}} \quad (\text{A.96})$$

De acordo com Friedman (2001), assumindo regularidade do espaço de x e y é possível inverter a diferenciação e a integração na expressão (A.97).

$$g_m(x) = E_y \left[\frac{\partial L(y, F(x))|x}{\partial F(x)} \right]_{F=F_{m-1}} \quad (\text{A.97})$$

Dado o gradiente $g_m(x)$ a atualização da função de ajuste é dada pela expressão (A.98) em que ρ é determinado pela solução do problema (A.99) análogo ao problema de *line search* (A.93).

$$f_m = -\rho_m g_m \quad (\text{A.98})$$

$$\rho_m = \arg \min_{\rho} E_{x,y} L(y, F_{m-1}(x) - \rho g_m(x)) \quad (\text{A.99})$$

A solução do problema definido no espaço de funções implica na minimização da função de perda por todo o espaço possível de valores de x e y . Segundo Friedman (2001), em casos práticos é inviável a solução do problema uma vez que não se conhecem as distribuições de x e y . Na verdade, o que se tem disponível em casos de problemas de aprendizado de máquina é um conjunto finito de exemplos de pares x e y .

Uma forma de resolver o problema é assumir uma classe de funções a que F deve pertencer, assumir um conjunto finito de dados com N pares de exemplos e realizar otimização de parâmetros para o problema definido no espaço de funções. O problema pode então ser representado conforme a expressão (A.100).

$$\{\beta'_m, a'_m\}_1^M = \arg \min_{\{\beta'_m, a'_m\}_1^M} \sum_{i=1}^N L\left(y_i, \sum_{m=1}^M \beta'_m h(x_i, a'_m)\right) \quad (\text{A.100})$$

O problema definido pela expressão (A.100) tem sua complexidade relacionada a quantidade de boosts m e com a quantidade de parâmetros a_m da classe de modelos. Segundo Friedman (2001), assumindo como premissa uma suavidade das distribuições de x e y , é possível utilizar uma abordagem iterativa para reduzir a complexidade. Dada uma estimativa F_{m-1} da função de ajuste, os parâmetros do próximo *boost* podem ser determinado pela solução do problema (A.101). Assim a estimativa da função de ajuste na m -ésima iteração é dada pela expressão (A.102).

$$\beta_m, a_m = \arg \min_{\beta'_m, a'_m} \sum_{i=1}^N L(y_i, F_{m-1}(x) - \beta h(x_i, a)) \quad (\text{A.101})$$

$$F_m(x) = F_{m-1}(x) + \beta_m h(x, a_m) \quad (\text{A.102})$$

Dada uma estimativa $F_{m-1}(x)$, as expressões (A.101) e (A.102) podem ser interpretadas como uma descida em gradiente em direção a $F^*(x)$ com a restrição que a direção do *step* seja dada por uma classe de funções $h(x, a)$.

Conforme definido por Friedman (2001), por construção do problema o gradiente irrestrito para o caso de um conjunto finito de dados é dado pela expressão (A.103). O conjunto dos gradientes irrestritos com sinal negativo para todos os dados da amostra resulta na direção de máxima descida $-g_m$ (expressão (A.104)) que é o gradiente N -dimensional no espaço de funções.

$$g_m(x_i) = \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F=F_{m-1}} \quad (\text{A.103})$$

$$-g_m = \{-g_m(x_i)\}_1^N \quad (\text{A.104})$$

O gradiente (A.103) é definido apenas para os dados da amostra sendo difícil de generalizar para outros valores possíveis de x . Uma alternativa para definição da direção do *step* é encontrar $h(x, a_m)$ mais paralelo a $-g_m$. Isso pode ser realizado pela solução da equação (A.105) sob os dados da amostra.

$$a_m = \arg \min_a (-g_m(x_i) - h(x_i, a))^2 \quad (\text{A.105})$$

Uma vez encontrado a_m pela solução da expressão (A.105) é possível substituir o gradiente irrestrito nas expressões de atualização da função de aproximação (A.106) e (A.107).

$$\rho_m = \arg \min_{\rho} \sum_{i=1}^N L(y_i, F_{m-1}(x) + \rho h(x_i, a_m)) \quad (\text{A.106})$$

$$F_m(x) = F_{m-1}(x) + \rho_m h(x, a_m) \quad (\text{A.107})$$

Ao invés de obter a solução do problema pelo uso iterativo das expressões (A.101) e (A.102) que dependem da verificação da hipótese de suavidade dos dados, obtêm-se a solução do problema pela aplicação iterativa das expressões (A.105), (A.106) e (A.107). Isso substitui um problema complexo de otimização de parâmetros por um problema mais simples de minimização de quadrados, com dificuldade computacional em geral menor.

Assim, para qualquer classe de funções $h(x, a)$ é possível definir o algoritmo de *Gradient Boosting* conforme a Figura A.13 em que F é um conjunto de aproximações sucessivas de F^* , a_m é o conjunto de parâmetros da m -ésima função aditiva $h(x, a_m)$. Os cálculos das direções dos *steps* de do tamanho deles são realizadas nas linhas 9 e 10 que correspondem a aplicação das expressões (A.106) e (A.107) respectivamente.

GradientBoosting (X, y, h, M)

```

 $F \leftarrow []$ 
 $a \leftarrow []$ 
 $\rho \leftarrow []$ 
 $F[0] \leftarrow \text{argmin}[\rho, \text{sum}(1, N, L(y[i], \rho))]$ 
para  $m$  entre 1 e  $M$  faça
   $\text{gradiente} \leftarrow []$ 
  para  $m$  entre 1 e  $M$  faça
     $\text{gradiente}[i] \leftarrow -g_m(x_i)$ 
     $a[m] \leftarrow \text{argmin}((a, \beta), \text{sum}(1, N, \text{gradiente}[i] - \beta * h(X[i]*a)))$ 
     $\rho[m] \leftarrow \text{argmin}(\rho, \text{sum}(1, N, L(y[i], F[m-1](x[i]) + \rho * h(X[i], a_m))))$ 
     $F[m] \leftarrow F[m-1] + \rho[m] * h(x, a[m])$ 
retorna  $F[M]$ 

```

Figura A.13 – Pseudo-código do algoritmo de *Gradient Boosting*

Fonte: Friedman (2001)

Para a construção de uma GBM utilizando o algoritmo da Figura A.13 é necessário especificar a função de perda e a classe de funções de base. Natekin e Knoll (2013) apresentam uma taxonomia para as funções de perda e para as funções base das GBMs.

Segundo os autores, as funções de perda podem ser de resposta contínua ou de resposta categórica. As funções de perda de resposta contínua são aplicadas nos casos em que o modelo que se deseja construir é um modelo de regressão. Analogamente, as funções categóricas são aplicadas nos casos em que o modelo a ser construído é um classificador.

As funções de perda comumente utilizadas são a função de perda gaussiana L2 (*L2-Gaussian Loss*) dada pela expressão (A.108), a função de perda absoluta L1 ou perda de Laplace (*Laplace L1-loss*) dada pela expressão (A.109) e a função de perda de Huber dada pela expressão (A.110) em que δ é chamado de parâmetro de robustez. A intuição da função de perda de Huber é que até um desvio δ o erro deve ser penalizado pela função L2 de perda e partir desse desvio o erro deve ser penalizado pela função de perda L1.

$$L(y, f)_{L2} = \frac{1}{2}(y - f)^2 \quad (\text{A.108})$$

$$L(y, f)_{L1} = |y - f| \quad (\text{A.109})$$

$$L(y, f)_{Huber} = \begin{cases} \frac{1}{2}(y - f)^2 \rightarrow |y - f| \leq \delta \\ \delta(|y - f| - \delta/2) \rightarrow |y - f| > \delta \end{cases} \quad (\text{A.110})$$

No caso das funções de perda de resposta categórica em que a variável de resposta y é definida dentro do conjunto $\{0, 1\}$, as funções mais utilizadas são a perda binomial (expressão (A.111)) e a perda exponencial simples (expressão (A.112)). A função de perda exponencial simples penaliza de forma muito mais severa os erros de classificação, sendo menos robusta a ruídos na amostra de dados.

$$L(y, f)_{Binomial} = \ln[1 + \exp(-(2y - 1)f)] \quad (\text{A.111})$$

$$L(y, f)_{exp} = \exp -(2y - 1)f \quad (\text{A.112})$$

As funções de base por sua vez são classificadas em três conjuntos: as de modelo linear, as de modelo de suavização e as árvores de decisão, sendo essas últimas as mais utilizadas em casos práticos de acordo com Natekin e Knoll (2013). Os autores comentam que as árvores utilizadas como *base learners* são definidas considerando uma profundidade máxima e que na maioria dos casos, árvores de estrutura simples (eg.: árvores com profundidade máxima menor que cinco) produzem bons resultados. Há casos de aplicação de árvores com profundidade máxima igual a um, chamadas de *tree stumps*, com resultados satisfatórios dentro da metodologia de GBMs.

Assim como em qualquer modelo de aprendizado computacional, as GBMs podem ser especializadas para os dados de uma amostra de treino, caracterizando uma situação de sobre-treinamento. Para mitigar o risco de sobre-treinamento e aumentar a capacidade de generalização de GBMs algumas técnicas podem ser aplicadas como por exemplo a seleção aleatória da amostra (*subsampling*).

A técnica de *subsampling* consiste em selecionar de forma aleatória apenas uma fração dos dados de treinamento a cada iteração da construção do modelo. Isso significa que cada termo do modelo aditivo resultante é submetido a uma sub amostra dos dados de treinamento, evitando assim uma especialização sequencial da GBM. A aplicação de *subsampling* por sua vez necessita da especificação de um parâmetro que representa a fração dos dados de treino que deve ser utilizada a cada iteração do algoritmo de treinamento. Esse parâmetro recebe o nome de *bag fraction*. Um valor de *bag fraction* de 0.1 significa que apenas 10% dos dados são utilizados em cada passo do algoritmo de treinamento. Natekin e Knoll (2013) recomendam o uso do parâmetro igual a 0.5 em casos em que a quantidade de dados de treinamento não representa um problema prático de manipulação de dados.

Outra técnica para evitar o sobre-treinamento de GBMs é chamada de *shrinkage*. Trata-se de uma redução sequencial do tamanho de cada *boost* de uma GBM. A cada iteração o *boost* é multiplicado por uma constante de redução λ que atenua o efeito de um *base learner*. Nesse caso, a equação de atualização da função de previsão (A.107) deve ser modificada de acordo com a expressão (A.113).

$$F_m(x) = F_{m-1}(x) + \lambda \rho_m h(x, a_m) \quad (\text{A.113})$$

Natekin e Knoll (2013) realizam um estudo empírico sobre os efeitos do *shrinkage* e afirmam que modelos construídos com o uso dessa técnica produzem GBMs com maior capacidade de generalização.

A metodologia de construção de GBMs permite diferentes configurações e apresenta resultados satisfatórios em diversos problemas práticos. Maiores detalhes sobre a metodologia podem ser encontrados em Hastie *et al.* (2008).

A3. Modelos de séries temporais

Os modelos de series temporais foram propostos inicialmente nos anos 1950 e têm sido desde então a metodologia convencional para problemas de previsão. Desde os modelos iniciais de suavização exponencial até os modelos de espaços de estado, houve muitos desenvolvimentos no ramo da estatística, sempre com a finalidade de representar da melhor forma o fenômeno gerador de uma série temporal observada.

Os modelos mais antigos que foram propostos para modelagem de séries temporais são os modelos de suavização exponencial. Tais modelos foram originados nas décadas de 1950 e 1960 decorrentes dos trabalhos de Brown (1959, 1963), Holt (1957) e Winters (1960).

Gooijer e Hyndman (2006) apresentam uma revisão abrangente da literatura sobre modelagem de séries temporais e afirmam que os modelos de suavização exponencial, apesar de bem aceitos na indústria e nos negócios, não atraíram tanto a atenção de pesquisadores e estatísticos por não possuírem fundamentos estatísticos bem definidos. A primeira fundamentação estatística para tais modelos foi proposta por Muth (1960) que mostra que a técnica de suavização exponencial simples produz previsões ótimas caso o fenômeno gerador seja um passeio aleatório com ruído branco.

Os modelos de suavização exponencial assumem que a série observada (Y_t) é resultante de um componente de nível (l_t), com componente de tendência (b_t), um ou mais componentes de sazonalidade (s_t) e um componente de ruído (e_t), conforme a expressão (A.114).

$$Y_t = f(l_t, b_t, s_t, e_t) \quad (\text{A.114})$$

Hyndman et al. (2002) propõe uma taxonomia para os modelos de suavização exponencial. O termo de tendência pode ser definido como inexistente, aditivo, multiplicativo ou amortecido. O termo de sazonalidade pode ser definido como não existente, aditivo ou multiplicativo. Dependendo da definição desses termos o modelo recebe uma categorização diferente conforme a Tabela A.1.

Tabela A.1 – Taxonomia de modelos de suavização exponencial

Tendência	Sazonalidade		
	Nenhum	Aditiva	Multiplicativa
Nenhum	NN	NA	NM
Aditiva	AN	AA	AM
Multiplicativa	MN	MA	MM
Amortecida (<i>Damped</i>)	DN	DA	DM

Fonte: Hyndman *et al.* (2002)

Como exemplos da abrangência da taxonomia, a célula NN da Tabela A.1 corresponde ao método de suavização exponencial simples, a célula NA da tabela representa o método de suavização de Holt e a célula AA representa o método de suavização de Holt-Winter.

Além da classificação da Tabela A.1, o termo de erro (e_t) pode ser definido como aditivo ou multiplicativo, totalizando 24 categorias de métodos de suavização na taxonomia proposta por Hyndman *et al.* (2002).

As expressões gerais dos métodos de suavização são dadas pelas expressões (A.115), (A.116) e (A.117) em que P_t , Q_t , R_t e T_t são os termos de suavização e α , β , γ e ϕ são constantes e m é o período da sazonalidade da série.

$$l_t = \alpha P_t + (1 - \alpha)Q_t \quad (\text{A.115})$$

$$b_t = \beta R_t + (\phi - \beta)b_{t-1} \quad (\text{A.116})$$

$$s_t = \gamma T_t + (1 - \gamma)s_{t-m} \quad (\text{A.117})$$

Os termos P_t , Q_t , R_t e T_t variam de acordo com a classificação do modelo da Tabela A.1 e de acordo com o tipo de erro do modelo, se aditivo ou multiplicativo. A Tabela A.2 detalha as expressões de P_t , Q_t , R_t e T_t para os diferentes casos.

Tabela A.2 – Termos de suavização para os diferentes modelos de suavização

ADDITIVE ERROR MODELS

Trend	Seasonal		
	N	A	M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/\ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
Ad	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$

MULTIPLICATIVE ERROR MODELS

Trend	Seasonal		
	N	A	M
N	$y_t = \ell_{t-1}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
Ad	$y_t = (\ell_{t-1} + \phi b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$

Fonte: Hyndman *et al.* (2002)

Modelos de suavização definidos conforme a Tabela A.2 também recebem o nome de modelos ETS (acrônimo de *Erros, Trend e Seasonality*) e são casos particulares de modelos de espaços de estado.

Além dos modelos de suavização, outra classe de modelos desenvolvida para lidar com fenômenos temporais, é a classe de modelos do tipo *Autoregressive Integrated Moving Average* (ARIMA). Yule (1927) foi o primeiro a propor uma visão estocástica das séries temporais, em que uma série temporal nada mais é que o acontecimento observado de um fenômeno estocástico. Desde a proposição dessa noção, inúmeros estudos foram realizados

no campo da estatística inferencial e foram também propostas as definições de processos estocásticos autorregressivos e de médias móveis.

Dentro desse contexto, nos anos 1960s Box e Jenkins (1990) realizaram a publicação de um trabalho que sumarizou o conhecimento da época. Além disso, os autores propuseram uma metodologia para identificação e modelagem de séries temporais que influenciou substancialmente a comunidade de profissionais que atuam na modelagem previsão de séries. Vale ressaltar que Roberts (1982) demonstra que os métodos de suavização exponencial lineares são casos especiais de modelos da classe ARIMA.

Quanto à equivalência entre modelos ARIMA e modelos de suavização, de acordo com Hyndman e Athanasopoulos (2013) há uma crença incorreta de que modelos ARIMA são mais gerais que modelos de suavização exponencial. Os modelos lineares de suavização de fato podem ser considerados casos especiais de modelos ARIMA, porém modelos de suavização não lineares não possuem sua contrapartida na classe de modelos ARIMA. Por outro lado, existem modelos da classe ARIMA que não possuem modelos ETS equivalentes.

Seja y_t a variável de interesse gerada por um processo estocástico. A modelagem da série assume que há uma dependência seria, caso contrário não há estrutura temporal no problema. Assim, pode-se definir y_t conforme a expressão (A.118) em que π_j são parâmetros que se relacionam com y_t de forma linear, a_t é a variável de inovação do processo ocorrida no instante t (também chamada de choques aleatórios) e c é uma constante que representa a tendência da série.

$$y_t = c + \pi_1 y_{t-1} + \pi_2 y_{t-2} + \dots + \pi_j y_{t-j} + \dots + a_t \quad (\text{A.118})$$

Comumente se assume que a_t é um processo com variáveis independentes e identicamente distribuídas (i.i.d.) e possui média zero e desvio σ^2 .

A expressão (A.118) descreve o processo estocástico gerador de y_t em função dos valores passados. Uma forma alterativa é descrever o processo em função dos choques aleatórios a_t conforme a expressão (A.119). Os parâmetros π_j e ψ_j são funcionalmente relacionados.

$$y_t = c^* + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \dots + \psi_j a_{t-j} + \dots \quad (\text{A.119})$$

As expressões (A.118) e (A.119) são desconhecidas e seus parâmetros devem ser estimados com base em amostras de dados. Em princípio, y_t pode se relacionar com seu passado remoto ou mesmo com o passado remoto de a_t , por isso é necessário escolher uma quantidade finita de parâmetros para estimar o modelo corretamente. A classe de modelos ARMA (*Auto Regressive Moving Average*) cumpre esse propósito ao definir um modelo com parâmetros finitos de ordem (P, Q) de acordo com a expressão (A.120).

$$y_t - \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} = c + a_t + \theta_1 a_{t-1} + \theta_2 a_{t-2} + \dots + \theta_q a_{t-q} \quad (\text{A.120})$$

De acordo com Tiao 2015, o modelo ARMA é utilizado para representar fenômenos estacionários, porém muitas vezes o fenômeno de interesse não exibe essa propriedade. Uma prática comum para modelar séries não estacionárias é tirar a d -ésima diferença da série até que ela se torne estacionária. Com isso o modelo da expressão (A.120) pode ser escrito conforme a (A.121) em que ∇^d representa a d -ésima diferença.

$$\nabla^d y_t - \phi_1 \nabla^d y_{t-1} + \dots + \phi_p \nabla^d y_{t-p} = c + a_t + \theta_1 a_{t-1} + \dots + \theta_q a_{t-q} \quad (\text{A.121})$$

O modelo da expressão (A.121) recebe o nome efetivamente de *Autoregressive Integrated Moving Average* (ARIMA) de ordem (p, d, q) .

A metodologia proposta por Box e Jenkins segue três passos:

1. Tentativa de especificação do modelo: o primeiro passo compreende a análise das funções autocorrelação e autocorrelação parcial da série para determinação de possíveis ordens do modelo;
2. Estimação dos parâmetros: o segundo passo contempla o cálculo dos parâmetros determinados pela ordem do modelo. Em geral os parâmetros são estimados pela maximização da verossimilhança dos dados amostrais;

3. Diagnóstico dos erros: uma vez que o modelo tem seus parâmetros estimados verificam-se as propriedades dos resíduos. Buscam-se resíduos não correlacionais, com média zero e desvio padrão constante (características representativas de um ruído branco). Caso os resíduos não passem nas verificações retorna-se ao passo 1 alterando a ordem do modelo.

Existem extensões do modelo ARIMA que consideram diferenças sazonais (SARIMA), variáveis externas ou exógenas (ARIMAX), uma combinação dos dois (SARIMAX). Existem ainda extensões heterocedásticas, ou seja, que modelam o erro do modelo como um processo ARMA em si. Esses modelos podem ser vistos em detalhes em Box e Jenkins (1990).

De acordo com Gooijer e Hyndman (2006), independente do sucesso da metodologia de Box e Jenkins, a determinação da ordem dos modelos depende do julgamento do indivíduo realizando a análise que deve analisar de forma qualitativa as funções de autocorrelação e autocorrelação parcial.

Os modelos da classe ARIMA possuem três principais pontos fracos considerando o tema dessa pesquisa:

- Consideram apenas relações lineares entre as variáveis explicativas;
- Necessitam de análises humanas e do julgamento do indivíduo responsável pela análise dos dados para determinação da ordem dos modelos, e;
- Assume a premissa de erros caracterizados por um ruído branco.

No início dos anos 1980, os modelos de espaços de estados foram utilizados para modelagem de séries temporais. As ideias dos modelos de espaços de estado foram propostas inicialmente por Kalman (1960) e fornecem uma estrutura única para as diferentes metodologias de modelagem linear de séries temporais.

Dentro do contexto de séries temporais, os modelos de espaços de estados também são chamados de modelos estruturais de séries temporais. Nesta seção são apresentados apenas os conceitos fundamentais desses modelos, sendo que maiores detalhes podem ser encontrados em Durbin e Koopman (2012).

Um modelo temporal univariado de espaço de estados pode ser escrito por meio de duas equações matriciais (expressões (A.122) e (A.123)).

$$y_t = z_t' \alpha_t + \varepsilon_t \varepsilon_t \sim N(0, \sigma_\varepsilon^2) i. i. d \quad (A.122)$$

$$\alpha_{t+1} = T_t' \alpha_t + R_t \eta_t \eta_t \sim N(0, Q_t) i. i. d \quad (A.123)$$

A expressão (A.122) é a equação da observação em que y_t é a série univariada de interesse, z_t' é a matriz de design do sistema no instante t e ε_t é uma fonte geradora de ruído no instante t . Assume-se que ε_t são variáveis independentes e identicamente distribuídas de acordo com uma distribuição normal de probabilidade com média zero e variância σ_ε^2 .

A expressão (A.123) é a equação do estado em que α_t é a matriz de estado do sistema no instante t , T_t' é a matriz de transição de estado do sistema no instante t , R_t é chamada de matriz de seleção (em geral equivalente à matriz identidade), η_t é a matriz de perturbações do estado do sistema. Assume-se que η_t é um vetor de variáveis independentes e identicamente distribuídas de acordo com uma distribuição normal multivariada com média zero e matriz de covariância Q_t .

A representação de séries temporais utilizando as expressões (A.122) e (A.123) permite modelar processos ARIMA ou mesmo processos de suavização exponencial. Apenas como exemplo, as expressões (A.124) e (A.125) representam um processo ARIMA de ordem $(p, 0, q)$ escrito por meio de equações de espaços de estado em que $m = \max(p + d, q + 1)$.

$$y_t = (1 \quad 0 \quad \dots \quad 0) z_t \quad (A.124)$$

$$z_t = \begin{pmatrix} \varphi_1 & 1 & 0 & \dots & 0 \\ \dots & 0 & 1 & \dots & 0 \\ \varphi_{m-1} & 0 & 0 & \dots & 1 \\ \varphi_m & 0 & 0 & \dots & 0 \end{pmatrix} z_t + \begin{pmatrix} 1 \\ -\theta_1 \\ \dots \\ -\theta_{m-1} \end{pmatrix} a_t a_t \sim N(0, \sigma^2) i. i. d \quad (A.125)$$

A estimação de parâmetros de modelos de espaços de estado é realizada por procedimentos numéricos de maximização da verossimilhança das observações dados os parâmetros a serem estimados.

Os modelos de espaços de estado apesar de conseguirem generalizar os conceitos das outras classes de modelos de séries temporais ainda possuem pouca utilização em problemas de previsão de demanda. Para esses modelos, ainda é necessário o julgamento do analista para proposição da estrutura das expressões (A.122) e (A.123), ou seja, os modelos de espaço de estado dificilmente são utilizados em casos em que existem muitas séries temporais que precisam ser modeladas.

As classes de modelos de suavização, modelos ARIMA e modelos de espaço de estado representam os principais modelos de séries temporais. Os modelos ARIMA ainda são os mais utilizados na prática por possuírem uma teoria estatística sólida e serem de simples compreensão.

A4. Legitimidade dos dados

Para a construção de qualquer modelo de previsão \mathbb{M} deve-se considerar um conjunto de dados legítimos para a construção e validação do modelo. Kaufman, Rosset e Perlich (2011) apresentam uma discussão sobre legitimidade de dados no contexto de mineração de dados e aprendizado computacional. Os autores definem a legitimidade de um conjunto de dados tanto em termos de atributos (*features*) quanto em termos de amostras de treinamento dos modelos.

Para uma definição formal de legitimidade de um conjunto de dados, é necessário definir dois atores no processo de construção de um modelo preditivo: (i) um cliente do modelo e (ii) um modelador. O cliente do modelo tem interesse em utilizar o modelo que será desenvolvido pelo modelador em seu contexto para prever algum tipo de grandeza de interesse (*target*). O contexto do cliente é definido por um processo aleatório $\mathcal{W}(\mathcal{X}, \mathcal{Y})$ em que \mathcal{X} é o processo aleatório de geração de amostras, ou *inputs*, e \mathcal{Y} é o processo aleatório de geração da grandeza de interesse, ou *target*. Instâncias de amostras x_i e targets y_i gerados pelos processos \mathcal{X} e \mathcal{Y} , mas relacionadas pelo processo \mathcal{W} , são ditas \mathcal{W} -relacionadas.

O objetivo do modelador é gerar um modelo preditivo a partir de um subconjunto de amostras e targets \mathcal{W} -relacionados providos pelo cliente. A esse subconjunto dá-se o nome

de conjunto de treinamento W_{tr} . Uma vez construído o modelo e entregue ao cliente, este deve ser capaz de obter uma estimativa da variável de interesse \hat{y} a partir de uma nova amostra X gerada pelo processo \mathcal{X} . A solução do problema do modelador é um modelo expresso genericamente por (A.126).

$$\mathbb{M}(X, W_{tr}) = \hat{y} \quad (\text{A.126})$$

Seja u uma variável aleatória. Outra variável aleatória v é dita legítima para modelagem preditiva de u (u -legítima) se sua realização v é observável pelo cliente no momento da previsão de u . Nesse caso pode-se escrever $v \in \text{legit}\{u\}$.

Para que um modelo \mathbb{M} seja considerado legítimo, ele deve ser construído a partir de variáveis legítimas. Para problemas definidos no eixo temporal, como é o caso do problema de previsão objeto dessa pesquisa, isso se traduz no requisito (A.127), também denominado de *no-time-machine requirement*. Esse requisito indica que todos os atributos de uma amostra devem ser conhecidos antes da geração do seu target \mathcal{W} -relacionado.

$$\text{legit}\{y\} \subseteq \{x \in \mathcal{X} | t_x < t_y\} \quad (\text{A.127})$$

A condição (A.127) abrange os atributos que podem ser utilizados para produzir um modelo legítimo. Além dos atributos, é necessário que as amostras utilizadas para treinamento do modelo também sejam legítimas. Kaufman *et al.* (2011) comentam que o uso de amostras não legítimas para parametrização de modelos preditivos são o caso mais comum de ilegitimidade de modelo causado por vazamento de dados (*data leakage*). Um exemplo de uso impróprio de dados é a utilização de amostras de validação para normalização do conjunto total de amostras.

Para solucionar esse caso de ilegitimidade, propõe-se a adição de uma segunda condição ao requisito (A.127). Para que um modelo \mathbb{M} seja construído de forma legítima, apenas amostras de treinamento e seus *targets* \mathcal{W} -relacionados podem ser utilizados. Esse requisito se traduz na expressão (A.128).

$$\forall X \in X_{tr}, X \in \text{legit}\{y\} \wedge \forall \tilde{y} \in Y_{tr} \in \text{legit}\{y\} \quad (\text{A.128})$$