UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

FERNANDO FAVORETTI VITAL DO PRADO

Solução automatizada de engenharia de características para problemas de aprendizado de máquina

São Paulo
2021

FERNANDO FAVORETTI VITAL DO PRADO

Solução automatizada de engenharia de características para problemas de aprendizado de máquina

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Metodologia e Técnicas da Computação

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 05 de outubro de 2021. A versão original encontra-se em acervo reservado na Biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a Resolução CoPGr 6018, de 13 de outubro de 2011.

Orientador: Prof. Dr. Luciano Antonio Digiampietri

São Paulo

2021

Ficha catalográfica elaborada pela Biblioteca da Escola de Artes, Ciências e Humanidades, EACH/USP,
com os dados fornecidos pelo(a) autor(a)

Prado, Fernando Favoretti Vital do
Solução automatizada de engenharia de características para problemas de aprendizado de máquina / Fernando Favoretti Vital do Prado; orientador, Luciano Antonio Digiampietri. -- São Paulo, 2021.
70 p: il.

Dissertacao (Mestrado em Ciencias) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, 2021.
Versão corrigida

1. Aprendizado computacional. I. Digiampietri, Luciano Antonio, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação:
Brenda Fontes Malheiros de Castro CRB 8-7012; Sandra Tokarevicz CRB 8-4936

Dissertação de autoria de Fernando Favoretti Vital do Prado, sob o título **"Solução automatizada de engenharia de características para problemas de aprendizado de máquina"**, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em 05 de outubro de 2021 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Luciano Antonio Digiampietri

Instituição: Universidade de São Paulo

Presidente

Prof. Dr. Flavio Soares Correa da Silva

Instituição: Universidade de São Paulo

Prof. Dr. José de Jesús Pérez Alcázar

Instituição: Universidade de São Paulo

Agradecimentos

Ao Prof. Dr. Luciano Digiampietri, obrigado pela oportunidade de trabalhar como seu orientando, da confiança depositada em mim e da compreensão de cada dificuldade durante o processo. Agradeço por ser, sobretudo, um ser humano brilhante e compreensivo, que serve como fonte de inspiração para todos seus alunos e por mostrar que com paciência e cuidado torna-se possível transformar as mais adversas situações em realidade, o sr. é o meu maior modelo acadêmico.

A Profa. Dra. Violeta Sun, pela indicação e apoio ao PPGSI , mas sobretudo pela amizade, paciência, aconselhamento e por sempre extrair o melhor de cada aluno.

A XP Investimentos e a todo o time de ciência de dados (especialmente Ber, Caio, Yama e Danilo), pela disponibilização de toda a infraestrutura para que o trabalho pudesse ser realizado, e sobretudo pelas discussões e validações realizadas por todo o caminho.

Aos meus grandes amigos de graduação Kenji, Gamino, Helô, Mandha, Juju, Cin, Hebert e Victor que me acompanharam por toda a formação em SI, e por todo o apoio durante o período de mestrado nos momentos mais difíceis, descontração e suporte mútuo. Vocês são pessoas brilhantes que merecem o mundo e é difícil colocar em palavras o quanto eu admiro cada um de vocês, sem vocês não seria possível, obrigado por tudo.

A meus amigos Adriano, Leonardo, Isabella, Evelyn, Leandro, Marta, Sérgio, Shifu Luis e a todos amigos do Kung Fu. Obrigado pela amizade de mais de uma década, por todo apoio, descontração e aconselhamento e por mais longe que possam estar, obrigado por sempre estarem por perto. Me sinto a pessoa mais sortuda do mundo por ter pessoas tão incríveis ao meu lado, contem sempre comigo.

A todos meus professores do SESI 94, Colégio Singular e principalmente da USP, que me mostraram desde sempre que o conhecimento é o bem mais valioso que se pode receber de alguém.

A Escola de Artes Ciências e Humanidades da Universidade de São Paulo, que talvez em um dos momentos atuais mais tenebrosos já vistos por esse país, se mostra como o gigante imponente e intransponível que sempre foi, não é nada menos que uma honra poder levar este nome comigo.

A minha família, tios e avós que são simplesmente as pessoas mais benevolentes e generosas que se têm conhecimento, obrigado pelo apoio, torcida e orações, espero que aonde quer que estejam saibam que sou eternamente grato por tudo.

A minha irmã Paula, minha maior fonte de inspiração durante toda minha vida, nada seria possível sem o modelo de ser humano que você é, obrigado pelos conselhos acadêmicos, cumplicidade e por todo cuidado que você sempre teve comigo, você é meu maior modelo e meu maior orgulho.

A meus pais, Artur e Maria Luiza que desde pequeno me ensinaram o real valor da educação, não imagino outra família dando a quantidade de apoio tanto estrutural quanto emocional que vocês me proporcionaram por toda a jornada, agradeço muito pelo carinho, compreensão e preocupação que vocês demonstraram por mim, minha maior motivação é um dia retornar toda a dedicação que tiveram comigo, amo vocês.

Resumo

Prado, Fernando Favoretti Vital do. **Solução automatizada de engenharia de características para problemas de aprendizado de máquina**. 2021. 70 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2021.

Atualmente, o aprendizado de máquina vem sendo amplamente utilizado para auxiliar em diferentes atividades, desde a sugestão de vídeos ou séries até no auxílio ao diagnóstico médico. O desenvolvimento de soluções envolvendo aprendizado de máquina envolve uma série de tarefas que incluem entendimento do problema, entendimento dos dados, preparação dos dados, modelagem, avaliação e verificação dos resultados. A construção de modelos de aprendizado de máquina de alta qualidade é, tipicamente, interativo e complexo, exigindo conhecimento específico e um grande esforço do executor. O aprendizado de máquina automatizado (*AutoML*) procura automatizar partes desse processo. Uma etapa importante do desenvolvimento desse tipo de solução é a engenharia de características que aplica transformações nos dados originais, tornando-os mais representativos para o modelo final. O presente trabalho atua no escopo de apresentar uma solução que automatize o processo de engenharia de características. A estratégia resultante da aplicação de técnicas de geração e seleção automatizadas de características em um arcabouço único é capaz de propiciar melhoria no desempenho de diferentes algoritmos aplicados a problemas de classificação quando comparados a um *baseline* inicial frente a quatro diferentes métricas. A solução apresentada neste trabalho provê a opção de automatização do arcabouço completo de engenharia de características, para o contexto específico de problemas de aprendizado do tipo classificação que utilizam dados tabulares.

Palavras-chaves: AutoML. Engenharia de Características. Aprendizado de máquina.

Abstract

Prado, Fernando Favoretti Vital do. **Automated feature engineering solution for machine learning problems**. 2021. 70 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, DefenseYear.

Nowadays, machine learning has been widely used to assist in different activities, from recommending videos or series to aiding in medical diagnosis. The development of solutions involving machine learning involves a series of tasks which include understanding the problem, understanding the data, preparing the data, modeling, evaluating, and verifying the results. Building high-quality machine learning models is iterative and complex, requiring specific knowledge and a great deal of effort from the performer. Automated machine learning (AutoML) seeks to automate parts of this process. An important step in the development of this type of solution is feature engineering that applies transformations to the original data, making them more representative for the final model. The present work presents an approach that automates the feature engineering process. The developed solution combines automated feature generation and selection techniques in a single framework. It was able to improve the performance of different algorithms applied to classification problems when compared with an initial baseline, considering four different metrics. The solution presented in this work provides the option of automating the complete feature engineering framework, for the specific context of classification problems that use tabular data.

Keywords: AutoML. Feature engineering. Machine learning.

Lista de figuras

Lista de quadros

Lista de tabelas

Lista de abreviaturas e siglas

MI	*Mutual Information*
JMI	*Joint Mutual Information*
HJMI	*Historical Joint Mutual Information*
AUTOML	*Automated Machine Learning*
RF	*Random Forest*
XGBoost	*Extreme Gradient Boost*
TP	Verdadeiro Positivo
FP	Falso Positivo
TN	Verdadeiro Negativo
FN	Falso Negativo
VC	Validação Cruzada
AUC	- *Area under the ROC Cuver*

Sumário

[1] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

Durante as últimas décadas, o aprendizado de máquina vem tomando um importante papel na construção de experimentos voltados a dados. Com base em informações extraídas das mais diversas fontes, novos padrões podem ser identificados, predições podem ser feitas mais facilmente e as decisões se tornam mais rápidas e efetivas (GE *et al.*, 2017; AGUIAR; GREVE; COSTA, 2017).

Trabalhos e pesquisas das mais diversas áreas fazem uso de técnicas de aprendizado de máquina e isto exige conhecimento específico, tempo e, muitas vezes, não são o foco principal do executor destes trabalhos ou pesquisas.

A criação de soluções envolvendo aprendizado de máquina pode ser definida em uma série de passos padronizados iterando pelos tópicos de entendimento do problema; entendimento dos dados; preparação dos dados; modelagem; avaliação e verificação dos resultados (SILVA; PERES; BOSCARIOLI, 2017). Em geral, o processo de construir modelos de alta qualidade é interativo e complexo, além de exigir conhecimento específico do executor e um grande esforço temporal. Em particular, quem executa métodos do tipo é constantemente desafiado com uma ampla possibilidade de decisões a serem tomadas. Por exemplo, a compreensão do problema e a maneira de se tratar os dados deve ser coerente com a ampla gama de modelos disponíveis para cada tipo de problema, em adição à necessidade de otimizar potencialmente centenas de hiper parâmetros diferentes em cada tipo de modelagem. Por fim, é necessário também decidir qual a melhor maneira de validar e medir o desempenho final de um modelo (ELSHAWI; MAHER; SAKR, 2019).

Naturalmente, a decisão do executor do processo em cada passo impacta diretamente nos resultados finais de um modelo. Nesse contexto, nos últimos anos o interesse em automatizar e democratizar o processo como um todo vem crescendo. Técnicas de aprendizado de máquina automatizado (*AutoML*) buscam automatizar partes do processo citado, buscando o desempenho ótimo dentro de uma determinada tarefa ou conjunto de dados, permitindo que o usuário não tenha que realizar, necessariamente, todos os passos durante o desenvolvimento de projetos que envolvam o aprendizado de máquina (KAUL; MAHESHWARY; PUDI, 2017b). O arcabouço de aplicações de *AutoML* é responsável por automatizar algumas partes do processo, podendo fazer com que o mesmo seja realizado com mais facilidade por não especialistas.

As aplicações de técnicas de *AutoML* obtêm relativo sucesso, quando aplicadas para algumas situações como a seleção automatizada de modelos, nas quais, dado um conjunto de características e a denotação de um problema, é possível a aplicação de técnicas automatizadas para a seleção de um conjunto de modelos ótimo, que melhor represente o problema estudado. Ou a otimização automatizada de hiper parâmetros, na qual, dado um modelo já selecionado juntamente com a esfera bem definida de um problema, é possível encontrar o melhor conjunto de parâmetros e hiper parâmetros que melhorem o desempenho geral do modelo

Na mesma esfera, profissionais e pesquisadores frequentemente descobrem que um passo central em seu trabalho é implementar uma transformação adequada em seus dados, reestruturando os mesmos em uma nova e mais compreensível representação para um possível modelo final. Esta etapa é conhecida como engenharia de características e é considerada custosa, pois requer um esforço substancial e não escalável (KAUL; MAHESHWARY; PUDI, 2017b).

A engenharia de características envolve a transformação dos dados fornecidos para representar melhor um problema de aprendizagem. Tipicamente essa transformação é executada por meio da aplicação de funções matemáticas nas próprias características do dado (MOHR; WEVER; HüLLERMEIER, 2018b), dependendo do tipo de modelo que se pretende desenvolver e de qual será sua principal tarefa (MOHR; WEVER; HüLLERMEIER, 2018a). Tal processo também faz parte do arcabouço de atuação de técnicas de *AutoML*, consistindo em diversos métodos propostos para realizar a automação do processo e focando em diferentes etapas do mesmo.

Neste contexto, este projeto propõe o desenvolvimento de um arcabouço automatizado para a etapa de engenharia de características em problemas de aprendizado de máquina que usam dados tabulares.

1.1 Objetivo

O objetivo principal do presente projeto é desenvolver uma solução que automatize o processo de engenharia de características e que tenha resultados melhores frente a processos não automatizados, tanto na questão de tempo de execução quanto resultados (por exemplo, precisão e medida F de classificadores).

1.2 Hipótese

Sabendo-se que diferentes metodologias são utilizadas para o desenvolvimento de aplicações na área de automação da etapa de engenharia de características, é possível que a combinação dos métodos existentes de engenharia de características em um método único, possa acarretar na melhoria das métricas finais calculadas frente aos modelos *baselines*.

A partir da unificação de técnicas diferentes em um método único pode ser possível encontrar uma boa interação entre as técnicas de geração e seleção de características aplicadas em um mesmo arcabouço, resolvendo problemas como o controle de características geradas frente ao custo computacional de medir a importância de cada uma para um determinado problema, entre outros.

Por exemplo, Katz, Shin e Song (2016) aplicaram a metodologia de meta aprendizagem para guiar a etapa de geração de características. Por outro lado, diferentes aplicações como aprendizado por reforço utilizada por Khurana, Samulowitz e Turaga (2018); Árvores Hierárquicas de transformações e algoritmos gulosos (KHURANA *et al.*, 2016) são amplamente utilizadas de diferentes maneiras durante a construção desta etapa.

1.3 Justificativa

Se for comprovado que a criação de um algoritmo mais robusto e mais abrangente (combinando diferentes técnicas de engenharia de características automatizada) melhora o desempenho final da solução, futuras aplicações que necessitem de etapas de engenharia de características poderão tirar proveito da técnica proposta, obtendo resultados mais assertivos de maneira simples e automatizada.

1.4 Organização deste documento

O restante desta dissertação está organizado da seguinte forma. O capítulo 2 descreve os principais conceitos utilizados neste trabalho. O capítulo 3 apresenta os resultados da revisão da literatura realizada (PRADO; DIGIAMPIETRI, 2020) com maior enfoque nos artigos considerados mais relevantes para o desenvolvimento deste trabalho. Já o capítulo 4 detalha a solução proposta, apresentando a abordagem em termos gerais da arquitetura

desenvolvida e é seguido, no capítulo 5, pelas devidas experimentações e análises dos resultados considerando cada uma das métricas analisadas contando com o apoio de testes estatísticos. Por fim, o capítulo 6 contém as considerações finais acerca do desenvolvimento, resultados e futuras aplicações do arcabouço proposto.

2 Conceitos Fundamentais

Neste capítulo são apresentados de forma resumida alguns conceitos tidos como fundamentais para esta dissertação. Estes conceitos estão focados em dois tópicos principais: aprendizado de máquina e engenharia de características.

2.1 Aprendizado de máquina

Aprendizado de máquina pode ser definido como algoritmos ou aplicações computacionais capazes de aprender a partir de um conjunto de experiências. Alguns conceitos tipicamente relacionados a aprendizado de máquina são o conjunto de experiências a partir do qual os algoritmos "aprendendem" E (conjunto de treinamento), a tarefa que está sendo aprendida T, e a métrica de desempenho que é medida P. Assim, é típico em pesquisas de aprendizado de máquina tentar melhorar o desempenho sendo medido com P na realização da tarefa T a partir de E (MITCHELL *et al.*, 1997).

O uso de aprendizado de máquina pode fornecer informações sobre estruturas e padrões a partir de um conjunto de dados, criando modelos que são capazes de prever determinados resultados ou comportamentos (REBALA; RAVI; CHURIWALA, 2019).

2.1.1 Aprendizado supervisionado

Para os problemas de aprendizado de máquina determinados como supervisionados, o problema pode ser formulado como, usando um conjunto de treinamento E, encontrar uma função $f: X \rightarrow Y$, que mapeia objetos $x \in X$ para rótulos $y \in Y$ com "bom" desempenho P (KAUL; MAHESHWARY; PUDI, 2017b).

Os tipos de problemas de aprendizado de máquina supervisionado mais comuns incluem:

- Classificação: classificar algo como pertencente a uma ou mais categorias (classes)
- Regressão: com base em dados históricos (dados de treinamento), construir modelos e usá-los para prever um valor numérico, referente ao possível valor futuro da mesma base de dados apresentada previamente.

2.2 Engenharia de características

Conforme mencionado, a engenharia de características envolve a transformação dos dados fornecidos para representar melhor um problema de aprendizagem. Tipicamente essa transformação é executada por meio da aplicação de funções matemáticas nas próprias características do dado, dependendo do tipo de modelo que se pretende desenvolver e de qual será sua principal tarefa (MOHR; WEVER; HüLLERMEIER, 2018a).

2.2.1 Seleção de características

Dentro do contexto de aprendizado de máquina supervisionado, um algoritmo de aprendizado é tipicamente representado por um conjunto de instâncias de treino, em que cada instância é descrita como um vetor de características e um vetor resposta. Por exemplo, em um problema de diagnóstico médico as características podem incluir a idade do paciente, peso, pressão sanguínea entre outras enquanto o vetor resposta irá indicar se o paciente está ou não sofrendo um problema de coração naquele momento. A tarefa do algoritmo de aprendizado, nesse caso, é de induzir (classificar) futuros casos desse mesmo problema, ou seja, o "classificar" age como um mapa do espaço de características para o conjunto de valores resposta.

Alguns algoritmos de aprendizado de máquina (como árvores de decisão) são conhecidos por se deteriorarem (perder desempenho) quando apresentados a vetores de que contêm características que não são necessárias para a realização da previsão de um dado problema. Já alguns algoritmos (como por exemplo algoritmos como Nayve-Bayes), embora sejam mais robustos a vetores de características desnecessários, podem apresentar outros tipos de problemas, por exemplo, a adição de características com alta correlação pode causar uma degradação grande no desempenho do algoritmo, mesmo se ambas as características forem relevantes para o problema.

A seleção de características tem como objetivo encontrar um subconjunto ótimo de um dado conjunto de características, selecionando apenas as características relevantes para o processo de aprendizado de máquina, dado que se um algoritmo utilizar apenas as características indicadas espera-se que ele tenha o melhor desempenho possível. A seleção de características é capaz de reduzir a complexidade de um modelo, eliminando

características que apresentam ruído ou características correlacionadas, reduzindo o erro de generalização causado pelas mesmas (OSMAN; GHAFARI; NIERSTRASZ, 2017).

Usualmente os atributos de um conjunto de dados são classificados em relevantes e irrelevantes, sendo que na literatura, comumente, esses níveis de relevância são definidos entre relevância forte e relevância fraca. Um atributo X é fortemente relevante se a remoção de X sozinho resulta em um deterioração de desempenho do classificador. Um atributo é fracamente relevante se ele não é fortemente relevante e se existe um subconjunto de características S, tal que o desempenho do classificador quando aplicado a S é pior que o desempenho quando aplicado em S U X. Uma característica também pode ser irrelevante, caso ela não seja fortemente relevante nem fracamente relevante (KOHAVI; JOHN, 1997).

Três métodos de seleção de características são mais frequentemente adotados:

- Métodos de filtro: Neste tipo de método as características são individualmente pontuadas frente a aplicações de métodos estatísticos e ranqueadas uma a uma e, após isto, um subconjunto das características mais relevantes é formado com base em uma determinada pontuação de corte (OSMAN; GHAFARI; NIERSTRASZ, 2017).
- Métodos por invólucro (*wrappers*): Métodos deste tipo dependem fortemente de um modelo de aprendizado de máquina como gerador da pontuação do conjunto de características, utilizando o próprio modelo para realizar a seleção (SHILBAYEH; VADERA, 2014).
- Métodos incorporados (*embedded*): Métodos deste tipo tentam atrelar a seleção de características juntamente com o processo de treinamento de um modelo, reduzindo a necessidade de classificar diferentes conjuntos de dados, como acontece nos métodos do tipo *wrapper* (CHANDRASHEKAR; SAHIN, 2014).

2.2.2 Geração de características

Geração de características (ou construção de características) pode ser definida como o ato de explorar interações implícitas entre as características originais de um conjunto, podendo aumentar significantemente o desempenho de uma aplicação de aprendizado de máquina. Realizada construindo novas características a partir das existentes por meio de operações matemáticas pré-definidas (QUANMING *et al.*, 2018).

Encontrar uma boa representação das características já existentes requer conhecimentos de domínio específico de acordo com a área do problema. A expertise humana, que normalmente é necessária para converter características 'cruas' em um novo conjunto de características mais úteis pode ser complementada por metodologias automáticas de construção de características. Em alguns processos, a etapa de construção de características é integrada na própria etapa de modelagem, como no caso de redes neurais onde as diversas camadas ocultas presentes na arquitetura da rede computam representações internas análogas à construção de características. Em outros processos, a construção de características é presente na etapa de pré-processamento do dado, aplicando diversos tipos de operações (automatizadas ou não) às características originais (GUYON *et al.*, 2008).

Alguns exemplos de operações de pré-processamento para a transformação de características são as unárias, que transformam uma única característica em uma nova, utilizando operações como discretizadores, normalizadores, *encodings* entre outras. Já as operações binárias transformam um par de características em uma nova a partir de um operador matemático (adição, divisão etc.). Operações de agrupamento nas quais características fontes de um determinado conjunto são agrupadas e a partir delas uma ação é calculada a partir das características separadas, por exemplo agrupamento por média ou desvio padrão e, por fim, operadores escalares ou lineares aplicam regras pré definidas a uma ou mais características, traduzindo-as para melhor entendimento de um modelo.

Alguns destes métodos não alteram o espaço dimensional dos dados (como operações unárias), enquanto outras aumentam ou reduzem o mesmo espaço. A construção de características é um dos passos chave no processo de análise de dados, conduzindo amplamente o sucesso de qualquer passo subsequente estatístico ou de aprendizado de máquina. Em particular é interessante nunca perder informações no processo de construção de características, então pode ser interessante manter as características originais juntamente ao conjunto de dados trabalhados, sempre arriscando o erro por ser muito inclusivo do que o risco de se perder informação. Normalmente um processo de seleção de características é aplicado logo após a etapa de construção das mesmas (GUYON *et al.*, 2008).

2.3 Entropia e informação mútua

Entropia é uma medida de incerteza para uma variável aleatória. Seja X uma variável aleatória discreta com função de densidade de probabilidade *p(x)*, a entropia *H(X)*, expressa em *bits* se dá por:

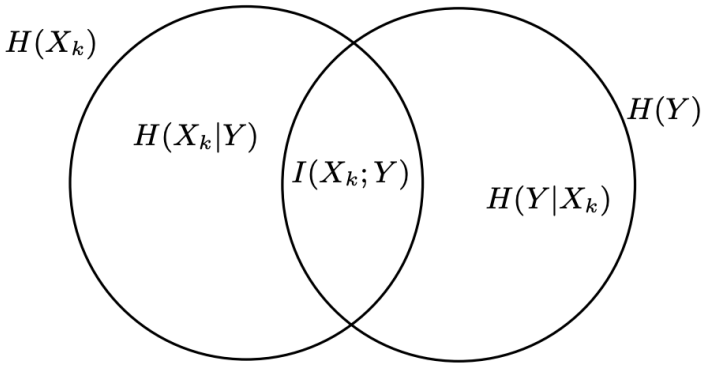
$$H(x) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i). \quad (1)$$

A Entropia busca quantificar quanta informação existe em uma variável aleatória, somando a probabilidade de cada evento ocorrer multiplicada pelo logaritmo da probabilidade de cada evento. Uma distribuição desbalanceada tem baixa entropia enquanto uma distribuição de probabilidade onde os eventos têm a mesma chance de ocorrer apresenta uma entropia maior. Por exemplo, a entropia da distribuição de probabilidades de uma moeda comum é de 1 *bit* (EDWARDS, 2008).

Informação mútua é uma das várias medidas usadas para mensurar quanto uma variável aleatória tem representatividade sobre outra, ou também apresentada como a redução da incerteza sob uma variável aleatória dado que temos conhecimento de outra. Uma quantidade alta de informação mútua indica uma alta redução nesta incerteza enquanto uma baixa informação mútua indica uma pequena redução. Caso a mensuração seja igual a zero significa que as duas variáveis aleatórias são independentes (LATHAM; ROUDI, 2009). Para entender a informação mútua recorremos a definição de entropia e de entropia relativa, que por sua vez é uma medida de distância entre duas distribuições, funcionando como uma medida de ineficiência em assumir uma determinada distribuição q quando a verdadeira distribuição seria p.

Considerando duas variáveis aleatórias X e Y com probabilidade conjunta *p(x, y)*, a informação mútua *I(X:Y)* é a entropia relativa entre as distribuições conjuntas. A figura 1 mostra a relação entre entropia e informação mútua. Enquanto cada círculo se refere a diferentes classes em uma característica, a informação mútua (*I(X:Y)*) é construída através da entropia (*H(Xk)*) e a entropia condicional (*H(Xk| Y)*), ou seja, a área demarcada pela informação mútua representa a informação compartilhada entre a característica X e o valor predito Y.

Figura 1 – Informação Mútua



Fonte: (GOCHT; LEHMANN; SCHöNE, 2018)

2.4 Aprendizagem automatizada

A partir do nome, pode se denotar que a aprendizagem automática ou automatizada (*AutoML*) representa a interseção da automação e do aprendizado de máquina. A combinação das duas áreas se tornou um tópico influente e muito pesquisado nos últimos anos.

Com a definição apresentada anteriormente para o aprendizado de máquina, AutoML pode se traduzir também como uma ferramenta capaz de realizar aprendizado gerando um bom desempenho com base em dados de entrada E e dada uma tarefa T. Por outro lado, pesquisas tradicionais de aprendizado de máquina têm um maior enfoque em desenvolver novas ferramentas de aprendizado e não se importam muito em quão fáceis e aplicáveis as mesmas serão (QUANMING *et al.*, 2018).

Ferramentas de AutoML não apenas objetivam ter desempenho tão bom quanto as ferramentas de aprendizado tradicional, mas também necessitam que tal desempenho seja atingido sem a assistência humana (ou minimizando essa assistência) durante todo processo e dentro de um limite temporal e computacional. A equação 2 formaliza essa definição (QUANMING *et al.*, 2018).

$$\max_{configurações} \ de\ desempenho\ das\ ferramentas\ de\ aprendizado$$
$$sujeito\ a \begin{cases} Sem\ assistência\ humana \\ Tempo\ e\ poder\ computacional\ limitados \end{cases} \tag{2}$$

AutoML normalmente se refere a automação dos processos de preparação de dados

(engenharia de características), treinamento de modelos ou otimização de hiper-parâmetros de modelos, onde o número de opções possíveis para cada um desses passos no processo pode variar drasticamente dependendo do tipo do problema. AutoML permite que executores do processo de aprendizado de máquina automaticamente construam o arcabouço (ou parte do) de resolução de problemas de aprendizado de máquina para cada passo tentando criar, ao final, algoritmos de aprendizado com alto desempenho para um dado problema. (DAS; CAKMAK, 2018)

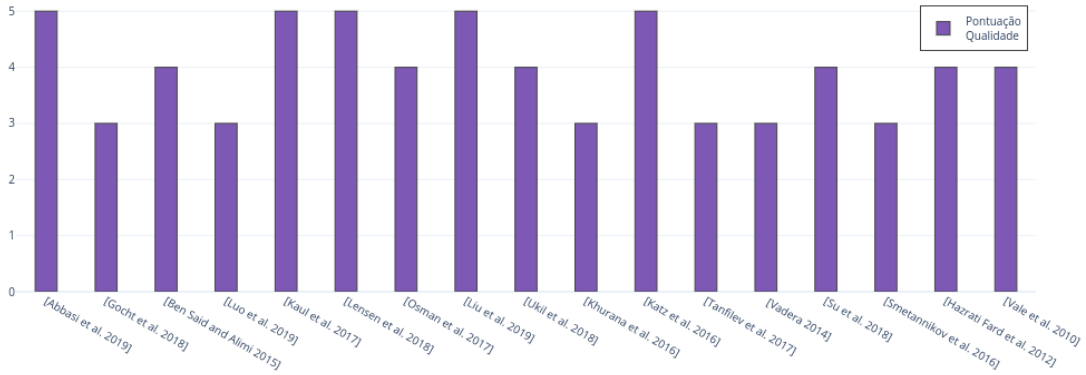
3 Revisão Bibliográfica

Este capítulo resume a revisão bibliográfica realizada com intuito de compreender o estado-da-arte dos métodos automatizados de aprendizado de máquina, com enfoque em técnicas automatizadas de engenharia de características. Para tal, foi realizada uma revisão sistemática (metodologia que visa ao entendimento profundo de uma área ou assunto de conhecimento, por meio de um processo metódico (KITCHENHAM *et al.*, 2009)) acerca dos mesmos. A busca considerou as duas principais bibliotecas digitais de artigos científicos na área de computação, ACM Digital Library e IEEExplore, restrita aos idiomas inglês e português. Um artigo com a presente revisão sistemática foi publicado no Simpósio Brasileiro de Sistemas de Informação 2020 (PRADO; DIGIAMPIETRI, 2020). A tabela 1 apresenta características básicas de todos os artigos extraídos durante a realização da mesma.

Adicionalmente, a fim de unificar a leitura completa e análise da qualidade de cada trabalho, a figura 2 apresenta a relação de qualidade e prioridade de leitura. A próxima seção apresenta os principais destaques em relação aos trabalhos selecionados, enfocando nos diferentes métodos identificados e tendências observadas.

Figura 2 – Gráfico de qualidade calculada e prioridade de leitura.

Qualidade do artigo x Prioridade de leitura.



Fonte: Prado e Digiampietri (2020)

Tabela 1 – Resumo do formulário de extração de dados dos artigos selecionados

Referência	Conjunto de Dados	Método	Algoritmos e Técnicas	Validação	Métrica de avaliação
Abbasi et al. 2019 (ABBASI; HUSSAIN; FAISAL, 2019)	bases textuais	Seleção de Características	Lógica Fuzzy e Algoritmos clássicos de aprendizado	N	Precisão, Revocação e Medida F
Gocht et al. 2018 (GOCHT; LEHMANN; SCHöNE, 2018)	5 conjuntos públicos (Arcene; Dexter; Dorothea; Gisette; Madelon)	Seleção de Características	JMI;HJMI; Informação Mútua e Entropia	N	Redução de erro frente a *baseline*
Ben Said and Alimi 2015 (SAID; ALIMI, 2015)	6 bases de dados clássicas UCI (svmguide3; german; magic04; splice; spambase; a8a)	Seleção de Características	Combinação de RAND e Petrun	N	Número de erros cometidos na escolha de uma característica relevante
Luo et al. 2019 (LUO *et al.*, 2019)	5 conjuntos públicos: Bank; Adult; Credit; Employee; Criteo	Geração de Características	Algoritmo guloso (Beam Search); Field Wise logistic regression; mini-batch gradient descent	N	AUC
Kaul et al. 2017 (KAUL; MAHESHWARY; PUDI, 2017a)	25 conjuntos públicos UCI	Seleção e Geração de Características	Geração de características por Regressão por pares; Seleção de características por ganho de informação	VC	Acurácia
Lensen et al. 2018 (LENSEN; XUE; ZHANG, 2018)	7 conjuntos públicos: Iris; Wine; WDBC; Dermatology; Vehicle; Image. Seg.; Movement Libras	Seleção de Características	Algoritmos Genéticos	N	Acurácia
Osman et al. 2017 (OSMAN; GHAFARI; NIERSTRASZ, 2017)	5 Códigos java Open Source (Eclipse JDT Core; Eclipse PDE UI; Equinox; Mylyn; Lucene)	Seleção de Características	Regressores linear e Poisson; Lasso, Ridge e Elastic Net para regularização	N	Acurácia
Liu et al. 2019 (LIU *et al.*, 2019)	1 conjunto públicos	Seleção de Características	Aprendizado por Reforço; Meta Aprendizagem	N	Acurácia e Medida F

Continua na próxima página

Referência	Conjunto de Dados	Método	Algoritmos e Técnicas	Validação	Métrica de avaliação
Osman et al. 2017 (OSMAN; GHAFARI; NIERSTRASZ, 2017)	5 Códigos java Open Source (Eclipse JDT Core; Eclipse PDE UI; Equinox; Mylyn; Lucene)	Seleção de Características	Regressores linear e Poisson; Lasso, Ridge e Elastic Net para regularização	N	Acurácia
Liu et al. 2019 (LIU *et al.*, 2019)	1 conjunto públicos	Seleção de Características	Aprendizado por Reforço; Meta Aprendizagem	N	Acurácia e Medida F
Ukil et al. 2018 (UKIL *et al.*, 2018)	sinais fisiológicos (eletrocardiogramas, fonocaridogramas)	Seleção de Características	Algoritmo guloso e relaxamento (*relax-greedy/*), filter, wrapper, *heterogeneou integration function'*	VC	Acurácia e Medida F
Khurana et al. 2016 (KHURANA *et al.*, 2016)	8 conjuntos públicos (Australian Credit; Svmguide3; Svmguide1; Ionosphere; Pima Diabetes; German Credit; Weather; Energy)	Seleção e Geração de Características	Algoritmo Guloso; árvore de Transformações	VC	Acurácia
Katz et al. 2016 (KATZ; SHIN; SONG, 2016)	25 conjuntos públicos	Seleção e Geração de Características	Meta Aprendizagem;Ganho de Informação	VC	Redução de erro frente a *baseline* (métrica própria baseada em AUC)
Tanfilev et al. 2017 (TANFILEV; FILCHENKOV; SMETANNIKOV, 2017)	115 bases de dados publicas para treino e 75 para experimentação	Seleção de algoritmos e Características	Meta Aprendizagem, *ensemble*, Ranqueamento e Agregação de Ranking	N	métrica de ranqueamento e agregação AEARR *(Aggregated Extended Adjusted Ratio of Ratios)*
Vadera 2014 (SHILBAYEH; VADERA, 2014)	26 conjuntos de dados públicos	Seleção de Características	Algoritmos Clássicos (árvores e Nayve Bayes)	N	Acurácia

Continua na próxima página

Referência	Conjunto de Dados	Método	Algoritmos e Técnicas	Validação	Métrica de avaliação
Osman et al. 2017 (OSMAN; GHAFARI; NIERSTRASZ, 2017)	5 Códigos java Open Source (Eclipse JDT Core; Eclipse PDE UI; Equinox; Mylyn; Lucene)	Seleção de Características	Regressores linear e Poisson; Lasso, Ridge e Elastic Net para regularização	N	Acurácia
Su et al. 2018 (SU *et al.*, 2018)	Base *Kdd99 network*	Seleção de Características	*Learning Automata*	N	Acurácia
Smetannikov et al. 2016 (SMETANNIKOV; DEYNEKA; FILCHENKOV, 2016)	Vetores de DNA, totalizando 246 bases de dados	Seleção de Características	Melif para Seleção de Características (*coordinate descent*); Meta Aprendizagem para otimização;SVM para mensurar resultados	*one-vs-all*	AUC
Hazrati Fard et al. 2012 (FARD; HAMZEH; HASHEMI, 2012)	3 bases de dados públicas (Arcene; Madelon; Spambase)	Seleção de características	Aprendizado por Reforço; Simulações de Monte Carlo, UCT (*Upper Confidence Tree*), UCG (*Upper Confidence Graph*	N	*Gain ration*, *FUSE*, *FSU* e *RELIEF*
Vale et al. 2010 (VALE; FEITOSA-NETO; CANUTO, 2010)	2 conjuntos artificialmente gerados, cerca ded 600 atributos em cada, com ruído e características ruins	Seleção de Características	Aprendizado por Reforço	VC	Acurácia

Fonte: Prado e Digiampietri (2020)

3.1 Discussão e análise

Conforme o objetivo principal da revisão, que é o de identificar e analisar as metodologias, técnicas e arcabouços existentes para a realização da etapa de engenharia de características automatizada dentro do escopo de problemas envolvendo a aplicação de aprendizado de máquina, foi observado que o processo de automação da etapa é realizado de diferentes maneiras, como mostrado na tabela 1.

Cada solução pode ser construída de diferentes maneiras, podendo ser responsáveis apenas pela etapa de seleção de características, apenas pela geração de características ou responsáveis pela aplicação de ambas, formando um arcabouço completo de engenharia de características. Dado isto, as estratégias e detalhamento das principais metodologias utilizadas são apresentadas separadamente nas subseções a seguir.

3.1.1 Seleção automatizada de características

A seleção de características é um passo importante em problemas de aprendizado de máquina, tendo como principais objetivos realizar a remoção de características irrelevantes ou redundantes, retornando um subconjunto de dados mais simplificado e por sua vez, otimizado, frente as características presentes originalmente.

É uma técnica amplamente utilizada no escopo de problemas de aprendizado de máquina, nos quais tipicamente é utilizada como etapa de pré-processamento dos dados com enfoque na redução do tempo de treinamento, melhoria na generalização - o que diminui a probabilidade de *overfitting* - através da simplificação dos dados de entrada.

Dentre os trabalhos analisados com enfoque puramente na seleção de características, grande parte cita os desafios da utilização de metodologias do tipo invólucro e incorporados uma vez que tais métodos necessitam da aplicação de diversas iterações de algoritmos de aprendizado de máquina ou da exposição de uma base não filtrada a um algoritmo de aprendizado. Em ambos os casos, o tempo computacional necessário e o consumo de recursos como processamento e memória são extremamente custosos.

Uma alternativa a tais métodos são os métodos de filtragem, que por sua vez ocorrem antes da etapa de modelagem o que reduz drasticamente a quantidade de experimentos necessários para se chegar a uma base final. Gocht, Lehmann e Schöne (2018) chegam a

conclusão que um bom algoritmo de filtragem deve ter as seguintes propriedades: uma maneira automática de finalização do processo após todas as características significativas haverem sido selecionadas; uma forma paralela de finalização baseada em uma quantidade máxima de características selecionadas desejadas pelo usuário do sistema; e não apresentar necessidade de múltiplas iterações do algoritmo principal de aprendizagem.

A partir dos três tópicos expostos, os autores do trabalho apresentam uma proposta de extensão ao já conhecido algoritmo *Joint Mutual Information* (JMI) (YANG; MOODY, 1999b) que adiciona ao algoritmo de informação mútua a capacidade de evitar a seleção de características irrelevantes levando em consideração todas as características já selecionadas e, além disso, adicionando condições de parada baseadas na qualidade da informação adicionada perante a seleção de cada característica. A seleção de características utilizando a técnica de JMI necessita que o usuário especifique o número final de características desejadas, caso esse número seja desconhecido é possível que o mesmo seja estimado realizando diversas iterações, selecionando N características, testando com o algoritmo de aprendizado subsequente e a partir do cálculo de erro repetir o processo que faz com que, dependendo do número de características presentes na base de dados, se torne uma abordagem muito custosa.

Para a criação do método os autores expandem o algoritmo de JMI permitindo que o mesmo guarde informações histórica de todas as características já selecionadas adicionando um termo novo a equação já existente além de um controle para parada do algoritmo caso uma nova característica selecionada não apresente informação adicional acima de um determinado limiar de corte.

A nova equação, chamada de HJMI, é colocada a teste frente a cinco bases conhecidas, oriundas do *Nips Feature Selection Challenge*, uma competição de seleção de características com objetivo de resolver problemas de aprendizado de máquina utilizando o menor número possível de características. Os resultados mostram que na maioria dos casos de teste os autores conseguiram manter as informações relevantes assim como o desempenho dos algoritmos de aprendizado utilizando um menor número de características frente a algoritmos tradicionais de seleção.

Diversas outras técnicas de seleção de características podem ser conferidas na tabela 1, por exemplo, paradigmas de aprendizado por reforço e meta aprendizagem (VALE; FEITOSA-NETO; CANUTO, 2010),(SHILBAYEH; VADERA, 2014), métodos de negociação baseados em teoria dos jogos (SAID; ALIMI, 2015), o uso de conjuntos *fuzzy*

para realização da seleção (ABBASI; HUSSAIN; FAISAL, 2019), além de métodos mais simplistas como regressões regularizadas (OSMAN; GHAFARI; NIERSTRASZ, 2017) e aplicações utilizando algoritmos genéticos (LENSEN; XUE; ZHANG, 2018).

3.1.2 Geração automatizada de características

Ao se apresentar um conjunto de dados novo a um algoritmo de aprendizado de máquina, para diversos problemas não se obtém resultados satisfatórios utilizando apenas as características puras (inicias) da base. A etapa de geração de características atua expandindo o espaço atual de características de uma base bruta, criando novas características a partir da interação das já existentes, buscando criar uma melhor representação dos dados para a etapa final de modelagem.

O processo de geração de características manual é custoso em quesito temporal e dificilmente o realizador da tarefa consegue explorar todas as possibilidades existentes, podendo perder oportunidades valiosas de possíveis otimizações do espaço de características. Neste contexto surge a geração automatizada de características.

O tópico de geração de características (aplicada de maneira isolada) não é tão explorado no contexto de *AutoML*, sendo que apenas um artigo em toda a revisão realiza a aplicação da técnica. Luo *et al.* (2019) apresentam uma aplicação de geração automatizada de características que pode ser utilizada de forma simples, mesmo por usuários sem conhecimento específico. O algoritmo de geração utilizado trás um enfoque maior na possibilidade da geração de características de maior ordem, ou seja, quando as características geradas são expostas a mais de uma rodada de transformações.

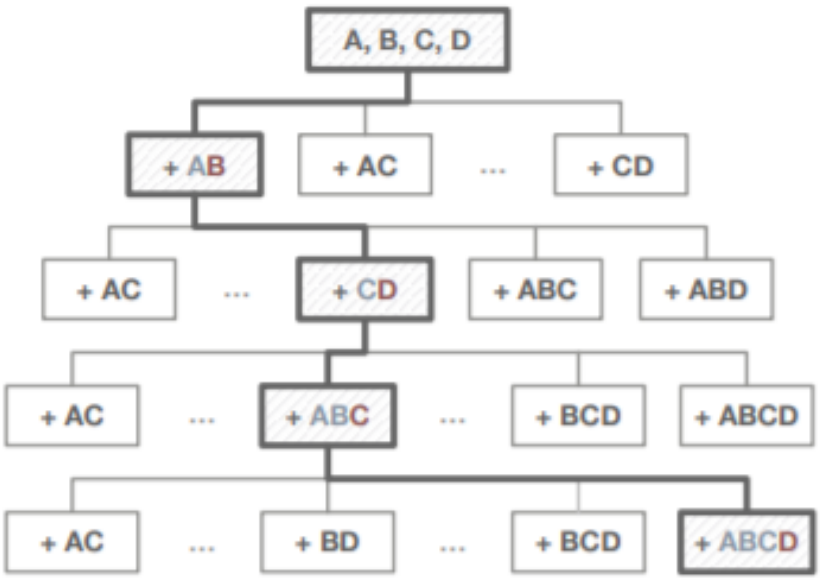
O primeiro ponto importante a se observar é o tipo de transformação sugerida pelos autores. Transformações do tipo força bruta, nas quais os dados são percorridos por diversas iterações de operações matemáticas como, por exemplo, somar ou multiplicar duas características em busca de uma terceira, são computacionalmente custosas para serem realizadas bem como para terem sua qualidade mensurada. A técnica utilizada pelos autores, chamada de *Feature Crossing*, aplica mudanças nas características transformando-as em vetores binários por meio da técnica de codificação de variáveis, popularmente conhecida como *One-Hot-Encoding*. Com o vetor em mãos, cada elemento de cada vetor

pode ser combinado com outros elementos advindos de uma característica diferente por meio de operações de conjunção lógica.

Com as transformações realizadas, outra dificuldade encontrada é a medição de desempenho de cada nova característica criada. Para contornar isto, os autores propõem um método que controla diretamente a geração, guiando o algoritmo para que apenas características relevantes sejam criadas. A técnica de pesquisa por feixe (*Beam Search*) é utilizada pelos autores. Nela as transformações dos vetores binários partem da base inicial e as demais são criadas com o auxílio de uma estrutura de árvore.

Com o conjunto de dados inicial e sem transformações representando a raiz da árvore, todas as transformações do nível são realizadas e mensuradas sendo que apenas a melhor delas é utilizada e adicionada como novo nó na árvore de transformações atual. A figura 3 representa este funcionamento, onde cada letra se refere a uma característica original da base e cada combinação de letras representa uma transformação no vetor binário de duas ou mais características, tornando-se fácil a visualização da proposta dos autores sobre a criação de características de alta ordem.

Figura 3 – Representação da aplicação de pesquisa por feixe



Fonte: Luo *et al.* (2019)

Todas as características construídas têm seu efeito mensurado a partir da aplicação do algoritmo de regressão logística, sendo que em cada iteração apenas o parâmetro relativo a nova característica construída é otimizado, poupando processamento.

Em resumo trata-se de um algoritmo guloso, que funciona expandindo apenas os nós mais promissores em busca de um subconjunto otimizado de características, até que uma condição de parada pré-determinada seja atingida. Os experimentos finais mostram resultados positivos, além da aplicação ser construída para se aproveitar de um ambiente totalmente paralelizável e escalável.

3.1.3 Engenharia automatizada de características

Apesar da grande maioria dos trabalhos listados na tabela 1 ter um enfoque maior na etapa de seleção e geração de características aplicadas separadamente, é necessária uma visão mais geral e abrangente de todas as técnicas para a concretização dos objetivos deste trabalho em específico. Dado isto, os trabalhos considerados mais importantes, mesmo que em menor número, são aqueles que permitem a junção das técnicas de seleção e geração de características em um arcabouço único, possuindo suporte a validação por meio de métricas diversas de avaliação, comprovando seus resultados.

A criação de um arcabouço completo de engenharia de características necessariamente precisa apresentar a capacidade de lidar com todas as etapas oriundas dos métodos de seleção e geração, assim como nos demais processos necessários, como limpeza da base e pré-processamento. Por estes motivos, três artigos se destacaram com esta completude.

Khurana *et al.* (2016) apresentam um arcabouço que automatiza a engenharia de características para problemas de aprendizado de máquina do tipo supervisionado, buscando aplicar a menor quantidade possível de transformações que maximizem o desempenho do modelo final. De uma maneira similar a pesquisa por feixes utilizada por Luo *et al.* (2019), os autores propõem a aplicação de uma árvore, na qual, inicialmente partindo-se das características originais, cada nó filho representa uma transformação diferente atrelada às características de seu nó pai. Tais transformações são realizadas de maneiras distintas por algoritmos que percorrem a árvore de formas diferentes e que, de maneira gulosa, escolhem o melhor caminho até que a métrica alvo pare de obter incrementos significativos.

As transformações realizadas são categorizadas em dois tipos diferentes: unárias e binárias. Transformações unárias são operações matemáticas aplicadas diretamente em uma ou mais características como logaritmos, exponenciações, multiplicações, normalizações entre outras, já as transformações binárias são realizadas com origem em agregações.

Com as sucessivas transformações aplicadas é natural que o número de características aumente significativamente, o que pode acarretar em problemas futuros de desempenho na etapa de seleção de características. Para mitigar tais problemas, os autores aplicam diversas etapas de seleção de características diretamente em cada nó da árvore de transformações, agindo como uma espécie de poda da árvore e limitando o crescimento da base como um todo. Ao final, uma última etapa de seleção é realizada no nó com maior desempenho, deixando a árvore pronta para a etapa final de modelagem.

Katz, Shin e Song (2016) apresentam uma solução para o problema de custo de mensuração de características geradas. A partir de uma base de dados, a ferramenta chamada *ExploreKit* atua gerando características candidatas por meio da combinação de características previamente existentes. Para tal, uma série de operadores são utilizados, sendo eles operadores unários, que aplicam transformações a partir de uma única característica, binários que combinam pares de características a partir de operações matemáticas comuns e, por fim, operadores de alta ordem que funcionam a partir do agrupamento e resumo de um grupo de características.

O sistema funciona gerando características sem uma determinada trava, o que acarreta na criação de milhares de novas variáveis, tornando a mensuração da capacidade individual de cada variável uma tarefa não trivial. Para contornar o problema, os autores propões uma abordagem de meta características, na qual cada característica construída tem suas informações tabeladas e atreladas ao seu desempenho perante adição no conjunto original de dados e mensuradas por um modelo de apoio. Este processo é custoso, porém é realizado de maneira separada do algoritmo principal e não precisa ser repetido.

Com tais informações sobre as características criadas, as mesmas são utilizadas para a criação de um algoritmo de aprendizado de máquina de apoio que é treinado separadamente e adquire a capacidade de prever se uma característica com determinado comportamento tem alta ou baixa probabilidade de ser uma boa característica preditora para um problema. O resultado da aplicação desta metodologia estima uma maneira bastante rápida de se avaliar o desempenho de um grupo grande de novas características criadas.

Como resultado final da aplicação do método em diversas bases de dados, o trabalho mostra uma redução no erro final de 17,4% a 29,3% em problemas do tipo classificação tanto frente ao *baseline* quanto quando comparado com diferentes versões da própria ferramenta desenvolvida.

Outro trabalho que se destaca na construção de um arcabouço completo de engenharia de características é o artigo de Kaul, Maheshwary e Pudi (2017b). Os autores apresentam um arcabouço baseado em regressões que funciona descobrindo padrões implícitos nos dados, baseando-se na maneira em que cada característica se relaciona entre si, levando também em consideração a classe a ser predita.

O algoritmo proposto trabalha por etapas. Na primeira etapa é realizado o pré-processamento automatizado do conjunto de dados utilizado a métrica de ganho de informação para desconsiderar características abaixo de um determinado limiar. A seguir, o algoritmo inicia a etapa de geração de características, encontrando pares de características correlacionados e, a partir destes, cria modelos de regressão utilizando uma característica para prever outra, em que a predição realizada é trabalhada como nova característica. Durante o último passo proposto, é executada a seleção final de características utilizando métricas de ganho de informação e estabilidade, eliminando características pouco informativas.

A avaliação dos resultados do trabalho segue a mesma linha dos demais estudados, a partir de um conjunto de bases de dados distintas em problemas do tipo classificação, variando as qualidades de cada conjunto de dados. Para a métrica de erro final, oito algoritmos de aprendizado de máquina diferentes são utilizados contando com KNN, Redes Neurais, Regressões (linear e logística), SVM e árvores de decisão. De acordo com os autores, os resultados apresentam, em média, um aumento de 10% em termos de acurácia após a etapa de geração de características e de 13% com a junção de todas as etapas, além de utilizar cerca de 10 vezes menos varáveis do que o espaço de características original.

3.1.4 Ameaças à validade

Duas importantes ameaças à validade identificadas na revisão realizada são relativas a metodologia de pesquisa utilizada. Apenas duas fontes de pesquisa foram utilizadas (ACM e IEEE), o que diminui a abrangência de artigos que poderiam ter sido considerados válidos para a revisão, diminuindo assim a completude da análise. Além disto, foram revisados apenas trabalhos na língua inglesa, podendo haver trabalhos interessantes em outras línguas não considerados na revisão.

3.2 Considerações finais sobre a revisão

No geral, considerando o escopo da revisão realizada, assim como suas limitações e critérios aplicados para inclusão e exclusão de artigos, observa-se uma quantidade relativamente pequena de trabalhos que automatizam o arcabouço completo da engenharia de características, sendo que a grande maioria limita-se apenas a aplicação das técnicas de seleção ou geração de características isoladamente.

Entretanto, mesmo nos trabalhos que se limitam a aplicação de apenas uma parte do arcabouço é possível verificar a existência de uma diversa gama de métodos que validam o objetivo primordial da revisão realizada, onde a mesma se trata justamente em identificar o estado-da-arte das técnicas e ferramentas existentes para aplicação da etapa de engenharia de características de maneira automatizada.

Os artigos destacados aqui, mostram-se mais importantes de acordo com os objetivos deste trabalho devido a resolução diferenciada dos problemas mais comumente encontrados, sendo eles a dificuldade em controlar o espaço de variáveis na geração de características, a dificuldade em medir a importância de cada característica construída durante a geração ou encontrar um bom ponto de parada para a etapa de seleção de características com potencial.

Igualmente valiosos são os artigos que, em minoria, implementam o arcabouço completo de engenharia de características, aplicando ambas as etapas necessárias para a automação completa. A partir dos mesmos pode-se observar o que é proposto hoje como estado-da-arte em trabalhos do gênero.

Por fim, a pouca quantidade de artigos valida a oportunidade existente para o desenvolvimento de um novo arcabouço para automatizar a engenharia de características, que pode se beneficiar das técnicas revisadas e aplicadas separadamente e unificando-as possibilitando desta maneira a melhora dos resultados finais.

4 Solução automática para engenharia de características

Este capítulo apresenta a estratégia para a construção da solução automática para engenharia de características em problemas de aprendizado de máquina. O intuito é dispor, em uma única solução, as etapas de geração de características e seleção de características.

Na literatura observada foram propostos diversos trabalhos individualizados e especialistas em cada uma destas aplicações, sendo que poucos abrangem o arcabouço completo. Nos parágrafos abaixo serão explicados os trabalhos escolhidos como base para a criação da solução principal, com destaque para as contribuições individuais para que o arcabouço como um todo, oriundo de dois artigos distintos funcione como uma peça única.

O capítulo segue organizado em duas partes principais: a primeira (Seção 4.1) diz respeito ao projeto geral do arcabouço proposto, com a implementação da solução descrita em pseudocódigo e a descrição do funcionamento da interação de cada etapa proposta; a segunda (Seção 4.2) descreve a proposta de avaliação do desempenho da aplicação do arcabouço.

4.1 Modelagem proposta

A figura 4 representa uma visão geral do processo proposto por este trabalho. Nela, estão organizados os elementos que fornecem a informação prévia necessária para o funcionamento do arcabouço e como essa informação é utilizada nas etapas posteriores do processo.

A parte inicial do processo de construção do arcabouço se dá com um conjunto de bases previamente selecionadas (apresentadas na Seção 5), tais bases se comportam de maneiras diferentes e são constituídas com diversas imperfeições. Esses conjuntos de dados são divididos em treino (70%), validação (20%) e teste (10%) e submetidos a uma primeira etapa contendo funções de pré-processamento de dados.

A primeira seção de pré-processamento da informação existe por duas principais razões. A primeira é permitir a padronização de cada conjunto de dados, principalmente para a criação do primeiro *baseline* que é constituído com o dado exposto ao mínimo trabalho possível. A segunda motivação deriva de outros trabalhos apresentados na seção 3

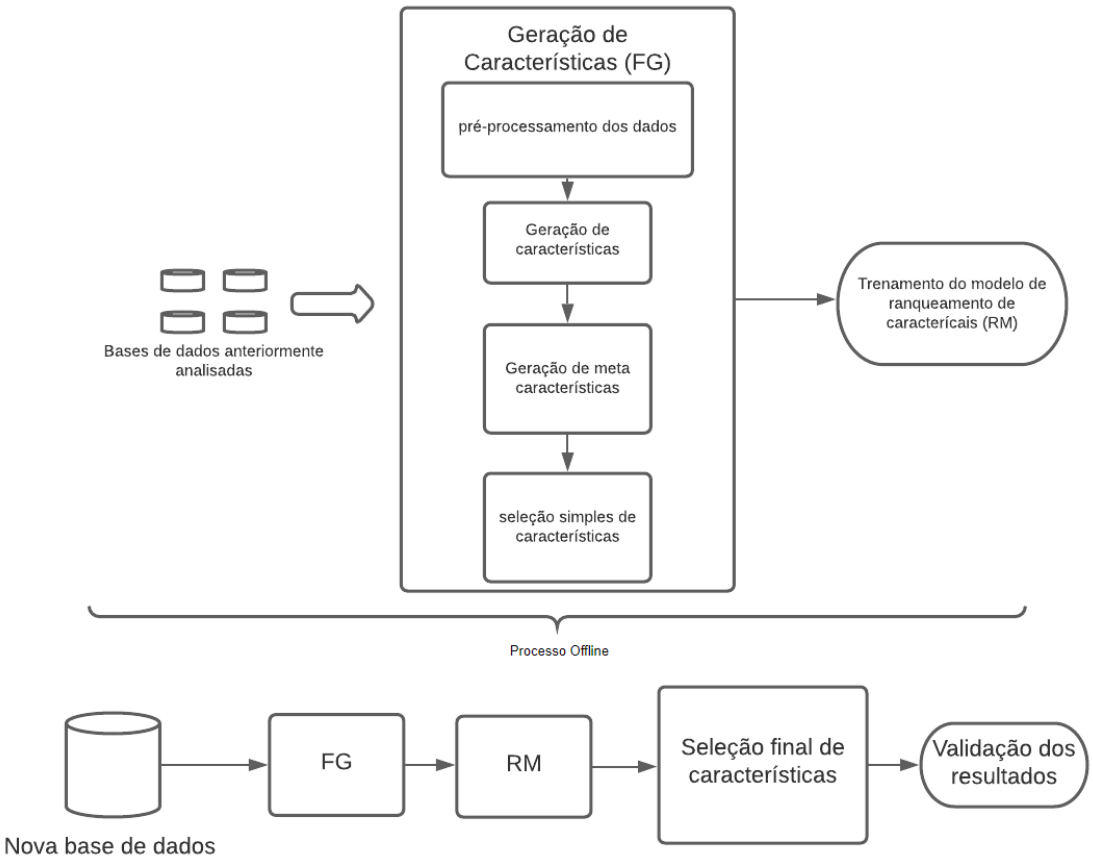


Figura 4 – Implementação do arcabouço de geração de características de Katz, Shin e Song (2016) com seleção de atributos final apresentada por Gocht, Lehmann e Schöne (2018)

onde diferentes autores apresentam técnicas iniciais de limpeza dos dados para facilitar a posterior geração de características.

Durante o passo de pré-processamento dos dados, iniciamos com a criação de um perfil de cada base, entendendo a distribuição de cada variável, removendo variáveis com pouca ou nenhuma variância; a existência de valores faltantes, onde, caso sua proporção seja maior que 80% a variável em análise é removida, entre 20%-80% a variável é preenchida com valores específicos indicando a falta de informação e, abaixo de 20% a variável é preenchida com a média das demais informações presentes; a incidência de variáveis muito correlacionadas também é um fator de remoção de características onde cada par de características é medido e caso positivo para alta correlação (maior que 0,8 absoluto) um dos pares é removido aleatoriamente; por fim a incidência de variáveis categóricas, que são transformadas utilizando a técnica de *target encoding* que apresenta bons resultados para modelos do tipo árvores de decisão. (PARGENT; BISCHL; THOMAS, 2019)

Com os conjuntos pré-processados viabiliza-se a implementação da tarefa de automação do restante do arcabouço. O principal interesse do trabalho está na redução concisa do erro gerado a partir da aplicação de algoritmos de aprendizado de máquina em conjuntos de dados automaticamente preparados, além também do tempo de execução, por isso, uma parte importante desta proposta acontece de maneira prévia.

A partir das bases iniciais pré-processadas, cada conjunto de treino, teste e validação derivado das mesmas é exposto ao restante do arcabouço. Dispondo da base pré-processada iniciamos o processo de geração automatizada de características. As seguintes etapas de geração de características automatizadas, ranqueamento de características e criação de meta características seguem a aplicação descrita em Katz, Shin e Song (2016), onde são realizadas três tipos de operações entre as características: operações unárias (normalização e discretização), binárias entre todos os pares de características (operações matemáticas de soma, subtração, multiplicação e divisão) e operações de agrupamento utilizando como operador as mesmas operações binárias. Vale notar que no passo anterior utilizamos a técnica de *target encoding* o que transforma variáveis categóricas em variáveis contínuas que, por sua vez, quando expostas a etapa de discretização criam-se oportunidades para trabalhar com mais agrupamentos. Com a geração de características realizada temos a criação de um vasto conjunto de dados Fi com até dezenas milhares de características a serem trabalhadas. Com o enfoque em diminuir esse vasto conjunto de dados é aplicado um filtro básico, baseado em Ganho de Informação (IG) onde toda característica com IG maior que zero (KAUL; MAHESHWARY; PUDI, 2017a) é levada para a próxima etapa, reduzindo assim espaço de características.

Durantes os demais experimentos feitos durante este trabalho, uma característica latente de todos os artigos implementados era a dificuldade e demora em analisar a grande quantidade de características geradas automaticamente. Devido a isto, observamos as mais diversas estratégias de ranqueamento de características, alguns autores enfrentam este problema limitando o número de características geradas como em Luo *et al.* (2019) usando algoritmos de suporte para diminuir o espaço de busca durante a criação de características. No arcabouço utilizado por este trabalho, permite-se que o espaço de características inicial (Fi) aumente dezenas ou centenas de vezes pois o processo de ranqueamento de características é feito de forma prévia e não influencia no tempo de aplicação da solução para uma nova base de dados.

Para o ranqueamento das novas características e escolha das características mais relevantes, é realizado o cálculo de um conjunto diverso de meta-características tanto em relação a base de dados original F quanto em relação a *Fi*. As meta características geradas se dividem em dois grupos, os quais inicialmente englobam informações gerais sobre F contendo informações sobre tamanho da base, estatísticas derivadas dos valores originais; resultados iniciais aplicados a base original ou seja, como um modelo simples se comporta inicialmente frente a base, informações de entropia entre as colunas e informações sobre diversidade de características onde são aplicadas testes estatísticos (teste T e testes de chi-quadrado) entre todos os pares de caraterísticas. O segundo grupo de meta-características criados diz respeito as novas características geradas no passo anterior, onde são aplicados testes estatísticos (teste T e testes chi-quadrado) e cálculos de entropia para cada variável, além disso são criadas relações com as 'características pais' juntamente com o conjunto de operadores utilizado na criação de cada nova características gerada, contendo informações como estatísticas descritivas e tipo de dados e, por fim, o mesmo conjunto de testes estatísticos é aplicado utilizando como pares as características pais e todas as características filhas geradas a partir deles.

Algoritmo 1 Geração de características e meta características

Entrada: X_i, y_i, y_pos ▷ base de dados, var resposta, classe positiva
Saída: F_i, M_f ▷ Características geradas, bases de meta características

function GERACAO_CARACTERISTICAS(X_i, y_i, y_pos)
    F,y$\leftarrow$ $limpeza\_inicial(X_i, y_i, y\_pos)$ ▷ Limpeza inicial da base
    $F_i \leftarrow F$
    for $F_k in F_i$ **do**
        $F_i \leftarrow$F$_i$ + geracao_operador_unario(F$_k$)
        $F_i \leftarrow$F$_i$ + geracao_operador_binario(F$_k$)
        $F_i \leftarrow$F$_i$ + geracao_operador_agrupamento(F$_k$)
    $F_i \leftarrow selecao\_simples\_IG(F_i)$ ▷ Seleção simples por ganho de informação
    $m_f \leftarrow \emptyset$
    for Fk in Fi **do**
        $M_f \leftarrow$M$_f$ + gera_mc_basic(F$_k$) ▷ Meta características gerais de F_k
        $M_f \leftarrow$M$_f$ + gera_mc_model(F$_k$) ▷ Meta características resultantes de modelo
        $M_f \leftarrow$M$_f$ + gera_mc_entropy(F$_k$, y) ▷ Meta características IG
        $M_f \leftarrow$M$_f$ + gera_mc_diversity(F$_k$, y) ▷ Meta características Testes de hipótese
    return F_i, M_f

Ao final deste processo é formada uma única base de tamanho $NcxNm$ de meta características que foi construída a partir de diferentes tipos de bases de dados iniciais (em

que *Nc* representa o número de características geradas *Nm* o número de meta características calculadas). Aqui é aplicada uma etapa simples de aprendizado de máquina, na qual um classificador do tipo floresta é treinado que visa a generalizar, a partir de quais informações de uma meta características (calculadas a partir de $\{F\ U\ Fi\}$) transformam a característica sendo representada em uma variável considerada como boa preditora, para ser adicionada em um modelo final. O algoritmo 4.1 apresenta este passo e é aplicado a todas as bases resultantes da etapa *offline* do arcabouço

Algoritmo 2 Preparo base para modelo de ranqueamento de característica

Entrada: F, F_i, y ▷ Base original limpa, Base de dados pós geração de características, meta características

Saída: $Base\_M_r$ ▷ base preparada para modelo de ranqueamento de características

1: **function** PREPARA_MODELO_RANQUEAMENTO(F_i, y, $\vartheta = 0.75$)
2: $error\_dim\_list \leftarrow \emptyset$
3: $baseline\_model \leftarrow RandomForestClassifierCV$(F,y) ▷ Classificador RF inicial com validação cruzada
4: $baseline\_auc \leftarrow calc\_auc$(baseline$_m odel$) ▷ AuC RF em base original
5: **for** $F_k in F_i$ **do**
6: $tmp\_model \leftarrow RandomForestClassifierCV$((F+F$_k$), y)
7: $tmp\_auc \leftarrow calc_auc$(tmp$_m odel$)
8: $error\_dim \leftarrow baseline\_auc$ - tmp_auc
9: $error\_dim\_list \leftarrow error\_dim\_list + error_dim$
10: $lim\_error\_quantile \leftarrow quantile$(error_dim_listm,$\vartheta$) ▷ Filtragem quantil 0.75 de diminuição de erro
11: $lista\_y\_result \leftarrow \emptyset$
12: **for** $k \leftarrow 0$ to tam(error_dim_list) **do**
13: **if** $error\_dim\_list[k] \leq lim\_error\_quantile$ **then**
14: $y[k] \leftarrow 1$
15: **else**
16: $y[k] \leftarrow 0$
17: $Base\_M_r \leftarrow F_k + y$
18: **return** $Base\_M_r$

Para isto, antes é necessário realizar a criação da variável resposta para a modelagem (apresentada no algoritmo 2), utilizando diminuição no erro (medido por AUC) decorrente da adição de uma nova característica frente ao desempenho de um classificador do tipo floresta aplicado na base original (contando com as transformações da etapa de pré-processamento). Esta medição é realizada de forma bastante custosa usando técnicas do tipo invólucro (*wrapper*), nas quais, a partir de (Fi), são realizadas *Nc* rodadas de medição escolhendo as características consideradas como positivas, ou seja, que apresentam

diminuição no erro inicial acima de um determinado limite (baseado na distribuição de todos os erros), ou negativas caso contrário.

A aplicação do modelo construído na etapa anterior frente a uma nova base de dados submetida ao mesmo processo, permite analisar um conjunto extenso de características geradas sem o grande custo de algoritmos do tipo invólucro ou de outras técnicas do gênero, permitindo uma seleção de características inicial mais rápida. Adicionalmente, como parte importante deste trabalho o modelo em questão é treinado com enfoque em melhorar a sua revocação, permitindo que o modelo seja menos preciso, mas que idealmente selecione a maior parte das características com potencial de melhorar o desempenho da classificação, o que acarreta em uma seleção inicial de características de forma mais branda.

Quando uma nova base é apresentada para o arcabouço, todos os passos de tratamento, pré-processamento, geração de características e geração meta-características são realizados e o modelo de ranqueamento aplicado gerando uma quantidade viável de características para a etapa de seleção de características final.

Para a etapa final de seleção de características, pontos importantes sobre as maneiras convencionais (filtragem, invólucro e incorporados) de realizar a prática foram levados em consideração. A aplicação de métodos incorporados, em que a ideia é adicionar e remover características em cada iteração de um modelo de aprendizagem de máquina, em um cenário no qual existem centenas de características mesmo após a filtragem, torna-se muito custosa, assim como métodos de invólucro que exigiriam o apoio de diversas iterações de um algoritmo de aprendizagem de máquina para pontuar características (mesma técnica utilizada para a construção do modelo de ranqueamento) se tornam agora impraticáveis para o usuário final do arcabouço proposto neste trabalho tanto pelo alto custo computacional e temporal quanto pela necessidade de parametrização. Em contraste, métodos do tipo filtro não necessitam de diversas iterações de modelos, uma vez que os mesmos são aplicados antes do algoritmo de aprendizado de máquina em si (BLUM; LANGLEY, 1997), porém a maior parte desses métodos precisa da definição de alguns parâmetros do usuário, por exemplo, o número de características a serem selecionadas. Uma outra maneira é validar o processo de seleção realizado a partir da iteração com os resultados frente a aplicação de algoritmos de aprendizado de máquina, mas tal possibilidade retira a maior vantagem dos algoritmos do tipo filtragem.

Baseado nisso, chega-se a conclusão que um algoritmo de seleção de características que se enquadre nos moldes deste trabalho, precisa conter a menor quantidade de

parâmetros possíveis solicitados ao usuário e não necessitar de reavaliação por um classificador externo. O objetivo de todo algoritmo de seleção de características é, justamente, selecionar características que proveem maior informação sobre o objetivo a ser predito, na maior parte das vezes utilizando algoritmos referentes a teoria da informação.

Dadas as condições, o algoritmo HJMI (GOCHT; LEHMANN; SCHöNE, 2018) é apresentado como uma extensão do método de informação mútua (VERGARA; ESTÉVEZ, 2014) e também uma extensão dos métodos de informação mútua conjunta (YANG; MOODY, 1999a) ambos os métodos suportam apenas um número pré-definido de características, que caso seja desconhecido precisa ser estimado a priori. Esta estimação é realizada por uma sequência de testes com diversos subconjuntos de características frente a algoritmos de aprendizado de máquina, calculando o erro de cada subconjunto e assim, chegando a uma estimação do número ideal de características. Dependendo do tamanho da base e do algoritmo de aprendizado de máquina, esta abordagem pode se tornar bastante cara. Enquanto o JMI, por iteração, determina o quanto uma única característica (Xk) representa acerca da classe a ser predita (Y) e das demais característica já selecionadas (S) permitindo, ao final, selecionar novas características que adicionam mais informação sobre Y levando em conta as (S), o mesmo não leva em conta a quantidade de informação que S carrega com o tempo. O algoritmo HJMI adiciona o termo à equação que representa toda a informação sobre as características já selecionadas em S, além da adição de um critério de parada para o seletor principal de características em que, caso uma nova característica adicionada não aumente a informação acumulada em S por um determinado limiar, o processo de seleção é encerrado. A etapa final do arcabouço de engenharia de características é apresentada no algoritmo 4.1.

Algoritmo 3 Arcabouço de engenharia de características

Entrada: X_i, y_i, y_pos
Saída: $modelo\_final, metricas\_finais, predicoes$

1: F_i, $M_f \leftarrow geracao\_caracteristicas(\mathrm{X}_i, y_i, y\_pos)$
2: $caracteristicas\_mr \leftarrow predicao\_modelo\_ranqueamento(\mathrm{M}_f)$
3: $F_i \leftarrow \mathrm{F}_i[caracteristicas\_mr]$ ▷ Características selecionadas pelo modelo
4: $base\_final \leftarrow selecao\_hjmi(\mathrm{F}_i)$ ▷ Seleção final por HJMI
5: RandomForestClassifierCV($base\_final$)
6: **return** $modelo\_final, metricas\_finais, predicoes$ =0

4.2 Avaliação dos resultados

A avaliação dos desempenhos das diferentes abordagens é realizada utilizando quatro métricas para problemas de classificação. Três delas descritas a seguir e a quarta apresentada na sequência:

- Precisão:

$$\text{Precisão} = \frac{tp}{tp + fp}$$

- Revocação

$$\text{Revocação} = \frac{tp}{tp + fn}$$

- Acurácia

$$\text{Acurácia} = \frac{tp + tn}{tp + tn + fp + fn}$$

Onde TP, FP, TN e FN se referem a classificação verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo, respectivamente. Uma última métrica também será utilizada baseada na curva composta pelas medidas TP e FP por tais medidas frente a diferentes limiares (Curva ROC), sendo que a área formada embaixo desta curva é denominada *AUC* (*Area under the ROC curve*) que também é uma medida utilizada para medir o desempenho de classificadores.

A avaliação é realizada de forma a se comparar as predições obtidas em diversas configurações do experimento, inicialmente serão mensurados os resultados referentes ao baseline que serão utilizados como base de comparação para os resultados do gerador de características e seletor de características (aplicados separadamente) e, por fim, comparado ao resultado frente a aplicação completa do arcabouço de engenharia de características proposto.

5 Experimentos e resultados

Neste capítulo são apresentados os experimentos executados para validar a abordagem de engenharia de características desenvolvida neste trabalho. A fim de organizar a apresentação do conteúdo, inicialmente é disposta uma seção para apresentar o treinamento do modelo auxiliar de ranqueamento de características, após, as demais seções de resultados dos experimentos e suas bases de dados e, por fim, as considerações finais acerca dos resultados obtidos.

5.1 Modelo de ranqueamento de características

Como apresentado no capítulo anterior, o modelo de ranqueamento de características é uma parte essencial do arcabouço completo de engenharia de características, uma vez que o mesmo consegue filtrar um grande espaço de características para um espaço menor e aplicável as necessidades do trabalho. Todas as bases utilizadas para a criação e validação do modelo em questão estão integralmente disponíveis na web[1]. Todos os dados utilizados podem ser encontrados no formato tabular, sendo este o único tipo de dado tratado neste trabalho, desconsiderando dados como áudios, vídeos e imagens. Para a criação do modelo foi utilizado um subconjunto das bases apresentadas por Katz, Shin e Song (2016), por se tratarem de bases que apresentam boa variedade em diversos aspectos como tamanho, número de atributos, número de exemplos, tipos de características, balanceamento de classes (razão entre quantidade de classes denominadas positivas e negativas) e valores faltantes. A tabela 2 denota as características das bases utilizadas. A seguir são descritas as siglas utilizadas na tabela 2 para descrever os conjuntos de dados.

1. Base: Nome da base em questão
2. NE: Número de exemplos (tamanho da base)
3. NC: Número de classes (no caso de tarefas classificação)
4. BC: Balanceamento das classes (todas as bases são relativas a problemas de classificação)
5. NNum: Quantidade de características do tipo numérico

[1] OpenML - datasets: ⟨https://www.openml.org/home⟩, acessado em 02/2021

Tabela 2 – Descrição dos conjuntos de dados utilizados no modelo de ranqueamento

Base	NE	NC	% BC	NNum
Heart	270	13	44,4%	46%
Horse	368	22	22,0%	31,8%
Cancer	569	30	37,2%	100%
Puma8	8192	8	49,7%	100%
Puma32	8192	32	49,7%	100%
Poker	1025010	10	49,8%	100%
Seismic bumps	2584	6	6,5%	77%
Vehicle_Norm	98528	100	59,0%	100%
Indian liver	585	10	28,6%	90%
Credit	690	15	44,4%	40%

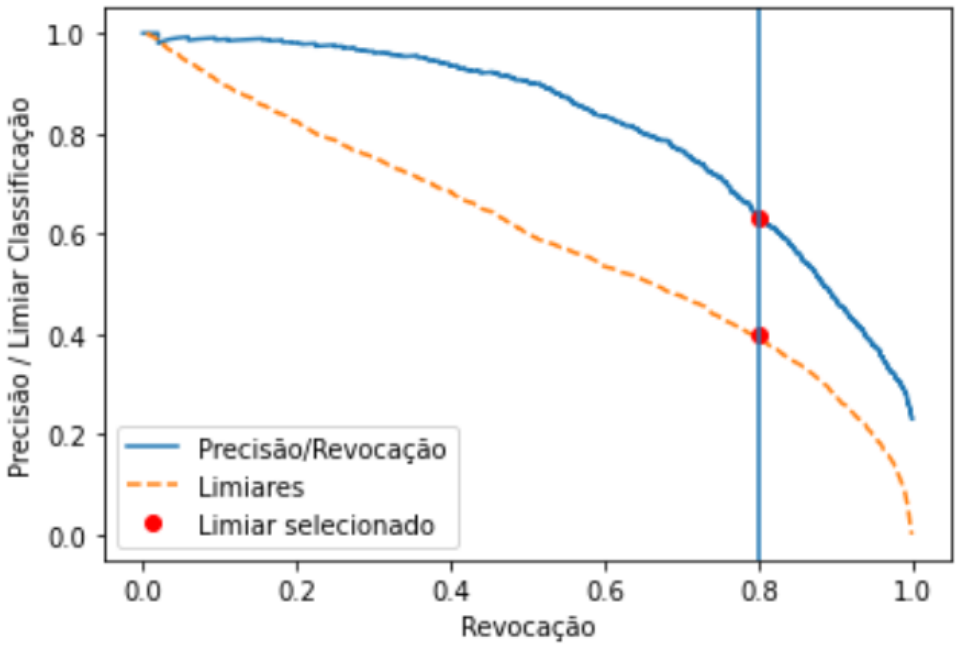
Fonte: Subconjunto das bases apresentadas em Katz, Shin e Song (2016)

Todas as bases apresentadas na tabela 2 são preparadas para o modelo de ranqueamento seguindo exatamente os mesmos passos apresentadas durante o capítulo anterior. Assim, cada base passa pelos processos de limpeza inicial que lida com características faltantes, correlacionadas e categóricas e, em seguida, são divididas em conjuntos de treino, validação e teste para aplicação do restante do arcabouço, onde ocorre a etapa de geração de características.

Seguindo a descrição do processo discutido no capítulo 4, para a criação do modelo de ranqueamento de características, meta características são geradas a partir de cada atributo presente em cada base tratada e são combinadas formando uma base tabular única de tamanho final 43.037 (soma total do número de características geradas de todas as bases tratadas) x 113 (número de meta características criadas em cada base), contando ainda com a adição de uma coluna para a variável resposta (Y), baseada na diminuição do erro frente a um processo de adição de características por invólucro. Apresentando, ao final, um balanceamento de 19% para características consideradas como promissoras.

Para o treinamento do modelo, a base foi repartida em subconjuntos de treino (70%), validação (10%) e teste (20%) e apresentada para um processo de validação cruzada de gramatura 10, no qual cada subconjunto criado é exposto a um modelo do tipo *XGBoost* (CHEN; GUESTRIN, 2016) com seus principais parâmetros focados para a melhora da métrica de revocação. Após a validação cruzada, o treinamento do modelo final é realizado para a base inteira e a partir das análises dos resultados utilizando a curva de precisão e revocação na figura 5 foi selecionado um limiar adequado aos objetivos deste trabalho para determinação da classe positiva.

Figura 5 – Gráfico da relação entre precisão e revocação.



Fonte: Fernando Favoretti Vital do Prado, 2021

Na figura 5 a linha tracejada laranja representa os diferentes valores de limiares que podem ser escolhidos para classificar as probabilidades geradas pelo modelo de ranqueamento a fim de determinar uma classe positiva. A linha contínua azul representa a relação entre precisão e revocação e os pontos vermelhos juntamente com a linha vertical representam os valores selecionados.

O valor de limiar selecionado de 0,4 consegue manter uma boa relação entre precisão e revocação (priorizada), mantendo as métricas com aproximadamente 0,65 e 0,81 respectivamente

5.2 Resultados do arcabouço em etapas separadas

Para realização dos experimentos finais, foram apresentados ao arcabouço 15 bases de dados que consistem em bases de acesso e uso público igualmente disponíveis na web em formato integral. Os dados utilizados nessa seção seguem as mesmas diretrizes de formato dos dados utilizados para treinamento do modelo de ranqueamento de características, apresentando diversidade em número de exemplos, número de características, número de características numéricas e balanceamento das classes. A limpeza inicial das bases assim como os outros processos de preparação dos dados são realizados da mesma maneira como

descrito no capítulo 4, as bases então são separadas em conjuntos de treinamento (70%), validação (20%) e teste (10%) e essa separação é mantida até a execução completa do arcabouço.

Tabela 3 – Descrição dos conjuntos de dados utilizados para validação dos resultados

Base	NE	NC	% BC	NNum
Normao	34465	119	28,5%	100,0%
Credit-g	1000	21	70,0%	33,3%
Sylvine	5124	21	50,0%	100,0%
Phoneme	5404	6	70,6%	100,0%
Cardiotocography	2126	36	77,7%	100,0%
Diabetes	768	9	34,8%	88,8%
Titanic	891	12	38,8%	58,3%
Bank Marketing	45211	17	88,3%	0,47%
Pc1	1109	22	6,6%	95,4%
Housing	20640	8	18,5%	40,0%
Ailerons	13750	40	42,0%	100,0%
Ecoli	336	8	42,5%	87,5%
Shuttle	58000	10	78,5%	100,0%
Waveform	5000	41	33,4%	41,0%
Blood Transfusion	748	5	83,3%	100,0%

Fonte: OpenML Datasets[2]

A tabela 3 apresenta as propriedades originais de cada base de dados utilizadas para validação dos resultados. A fim de realizar a criação de um *baseline* cada base foi inicialmente exposta a um algoritmo do tipo floresta aleatória, sem nenhuma espécie de otimização do espaço de hiper parâmetros, e contando apenas com a limpeza inicial das bases para que se tornasse possível a execução de um algoritmo de aprendizagem de máquina. Para a avaliação, como citado anteriormente, todas as etapas contaram com aplicação de validação cruzada em 10 partes (*10-fold cross validation*) a fim de verificar a existência de possíveis vieses e as pontuações comparadas são pontuações obtidas no conjunto final de teste. A análise dos resultados foi realizada em termos de precisão, revocação, medida F e AUC. A tabela 5 apresenta as medidas obtidas inicialmente.

As métricas de desempenho apresentadas em cada etapa de processamento foram calculadas a partir dos resultados do algoritmo de Floresta Aleatória (HO, 1995), contando com suas configurações padrão adotadas pela biblioteca utilizada, *sklearn*[3] e apresentadas na tabela 4. Buscamos, com a aplicação dos parâmetros padrões do algoritmo, realizar

[3] sklearn-RF: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html⟩

as comparações da forma mais pura possível, sem executar processos de otimização dos parâmetros (*tuning*) para cada problema em consideração.

Tabela 4 – Configuração padrão do algoritmo de Floresta Aleatória utilizado

Parâmetro	Valor	Descrição
n_estimators	100	Número de árvores utilizadas durante criação da floresta
criterion -g	Gini	Critério de avaliação da qualidade da quebra dos nós
max_depth	None	Profundidade máxima de cada árvore na floresta
min_samples_split	2	Número mínimo de exemplos para divisões de cada nó
min_samples_leaf	1	Número mínimo de exemplos em cada folha da árvore
Limiar de classe	0.5	Limiar utilizado para quebra das probabilidades preditas em classes

Fonte: Fernando Favoretti Vital do Prado, 2021

Tabela 5 – Descrição do desempenho de classificador *baseline* do tipo floresta aleatória para as configurações iniciais da base

Base	AUC Original	Precisão Original	Revocação Original	Medida F Original
Normao	0,98	0,93	0,88	0,85
Credit-g	0,78	0,64	0,54	0,82
Sylvine	0,94	0,91	0,91	0,92
Phoneme	0,85	0,74	0,74	0,85
Cardiotocography	0,97	0,96	0,88	0,96
Diabetes	0,81	0,75	0,69	0,56
Titanic	0,89	0,79	0,77	0,72
Bank Marketing	0,88	0,82	0,52	0,93
Pc1	0,87	0,72	0,52	0,09
Housing	0,89	0,85	0,68	0,53
Ailerons	0,91	0,83	0,82	0,78
Ecoli	0,98	0,96	0,96	0,95
Shuttle	0,98	0,98	0,99	0,99
Waveform	0,93	0,84	0,81	0,75
Blood Transfusion	0,74	0,73	0,74	0,84

Fonte: Fernando Favoretti Vital do Prado, 2021

Nota-se que mesmo perante as configurações padrões adotadas, grande parte das bases de dados já apresentou resultados considerados bons, o que diminui o espaço de busca por possíveis otimizações e, consequentemente, dificulta o trabalho das etapas de seleção e geração de características uma vez que o arcabouço fica mais exposto a potenciais problemas como a possibilidade de características relevantes não serem selecionas ou as chances de realizar a geração de características ruidosas.

A fim de comparação e comprovação da eficácia e necessidade de cada etapa desenvolvida no arcabouço, a tabela 6 apresenta a aplicação das etapas separadamente,

com as devidas comparações com os resultados obtidos com o *baseline*, além dos resultados finais obtidos com a aplicação completa do arcabouço. A seguir são descritas as siglas utilizadas na tabela.

1. ID: Id para identificação única de um conjunto
2. Conjunto: Nome do conjunto de dados utilizado.
3. Resultado: Medida utilizada para comparação dos resultados obtidos.
4. Resultado Original: Valores das medidas no experimento *baseline*
5. Apenas Seleção: Medidas observadas frente a aplicação do seletor de características, separadamente.
6. Apenas Geração: Medidas observadas frente a aplicação do gerador de características, separadamente
7. Arcabouço Completo: Medidas observadas frente a aplicação do arcabouço completo de engenharia de características

Tabela 6 – Comparação de resultados atingidos

ID	Conjunto	Resultado	Resultado Original	Apenas Seleção	Apenas Geração	Arcabouço Completo
1	Normao	AUC	0,98	0,97	0,98	0,97
		Precisão	0,93	0,93	0,93	0,91
		Revocação	0,88	**0,89**	**0,90**	**0,90**
		Medida F	0,85	**0,86**	**0,88**	**0,87**
2	Credit-g	AUC	0,78	0,77	0,77	0,77
		Precisão	0,64	**0,70**	**0,68**	**0,72**
		Revocação	0,54	**0,57**	**0,56**	**0,57**
		Medida F	0,82	**0,84**	**0,83**	**0,84**
3	Sylvine	AUC	0,94	**0,95**	**0,96**	**0,96**
		Precisão	0,91	0,87	**0,92**	0,91
		Revocação	0,91	0,87	0,91	0,91
		Medida F	0,91	0,86	0,92	0,91
4	Phoneme	AUC	0,85	0,85	**0,87**	**0,86**
		Precisão	0,74	0,72	0,73	0,73
		Revocação	0,74	0,72	**0,75**	0,73
		Medida F	0,85	0,83	0,83	0,83
5	Cardio.	AUC	0,97	**0,98**	**0,98**	**0,98**
		Precisão	0,96	0,95	**0,97**	**0,98**
		Revocação	0,88	0,85	**0,95**	**0,95**
		Medida F	0,96	0,95	**0,98**	**0,98**
6	Diabetes	AUC	0,81	0,80	**0,82**	**0,82**
		Precisão	0,75	0,75	0,73	0,73
		Revocação	0,69	0,67	**0,68**	**0,69**
		Medida F	0,56	0,53	0,55	**0,57**
7	Titanic	AUC	0,89	0,88	0,88	0,81
		Precisão	0,79	0,79	**0,81**	**0,80**
		Revocação	0,78	0,78	**0,79**	**0,79**
		Medida F	0,72	**0,73**	**0,74**	**0,74**
8	Bank Marketing	AUC	0,88	0,87	0,87	0,85
		Precisão	0,82	0,79	0,77	0,76
		Revocação	0,52	**0,53**	**0,64**	**0,60**
		Medida F	0,93	0,93	**0,94**	**0,95**

Fonte: Fernando Favoretti, 2021

ID	Conjunto	Resultado	Resultado Original	Apenas Seleção	Apenas Geração	Arcabouço Completo
9	Pc1	AUC	0,87	0,79	**0,88**	**0,88**
		Precisão	0,72	0,46	0,72	**0,80**
		Revocação	0,52	0,50	0,52	**0,54**
		Medida F	0,09	0,00	0,09	**0,17**
10	Housing	AUC	0,90	0,87	0,88	0,87
		Precisão	0,85	0,85	0,83	0,85
		Revocação	0,68	0,68	**0,72**	**0,70**
		Medida F	0,53	0,53	**0,60**	**0,57**
11	Ailerons	AUC	0,91	0,92	**0,93**	**0,93**
		Precisão	0,83	0,81	0,83	**0,85**
		Revocação	0,82	0,80	**0,83**	**0,85**
		Medida F	0,78	0,76	**0,83**	**0,83**
12	Ecoli	AUC	0,98	0,96	0,98	0,98
		Precisão	0,96	0,91	0,96	0,93
		Revocação	0,96	0,92	0,96	0,94
		Medida F	0,95	0,92	0,95	0,94
13	Shuttle	AUC	0,98	0,96	**0,99**	**0,99**
		Precisão	0,98	0,95	0,97	0,97
		Revocação	0,99	0,92	0,96	0,96
		Medida F	0,99	0,91	0,95	0,93
14	Waveform	AUC	0,93	0,91	0,93	0,92
		Precisão	0,84	0,83	0,84	0,82
		Revocação	0,81	0,81	**0,84**	**0,84**
		Medida F	0,75	0,74	**0,81**	**0,80**
15	Blood Transfusion	AUC	0,74	0,71	**0,76**	**0,76**
		Precisão	0,73	0,72	**0.75**	**0,74**
		Revocação	0,74	0,72	0,74	0,74
		Medida F	0,84	0,82	**0,85**	**0,85**

Fonte: Fernando Favoretti, 2021

Para os testes, um arcabouço inicial foi desenvolvido para facilitar a construção das bases de pontuações de maneira simples, contando com uma biblioteca própria. A metodologia de divisão de bases, construção dos modelos e aplicação da pontuação seguiu as etapas descritas no início do presente capítulo.

Dividindo a análise dos resultados em partes, inicialmente com foco nos resultados obtidos apenas com a aplicação da seleção de características, pode-se perceber que na maioria dos casos não foram obtidas alterações ou melhoras em nenhuma das medições realizadas. Em determinadas situações, a pequena perda de desempenho observada nos resultados pode vir a compensar, dado a considerável diminuição de características no conjunto de dados final usado para modelagem, o que acarreta em um menor tempo necessário para execução do algoritmo de aprendizado de máquina, entretanto, para os

objetivos deste trabalho e de acordo com a definição de *AutoML*, consideramos que estamos a procura de uma solução que maximize o desempenho das ferramentas de aprendizado com completude, ou seja, tanto em tempo de execução como desempenho dos resultados.

Nos poucos casos em que houve algum tipo de melhora com a aplicação da seleção de características, destaca-se o conjunto *Credit-g* que apresentou o melhor comportamento dentre os demais, o que indica que o algoritmo utilizado teve sucesso em separar apenas as características mais relevantes para o modelo final. Embora pequena, a diminuição na métrica de AUC indica que, no geral, para diversos níveis de limiares aplicados na previsão de probabilidade do classificador, temos uma piora do classificador. Em contraponto, a melhora nas demais métricas de Precisão, Revocação e Medida F mostram que o classificador se comportou melhor para um determinado limiar, o tornando um classificador mais específico. Para o conjunto de dados *Cardiotocography* observa-se um comportamento oposto, o aumento da métrica de AUC juntamente com a diminuição das demais métricas nos mostra que o classificador está se comportando de maneira mais generalista, ou seja, aplicando todos os valores de limiares nas previsões de probabilidade conseguimos encontrar diversos valores que tornam o classificador melhor no geral, mas para o valor específico utilizado na medição (igual para todos os classificadores treinados) houve queda de desempenho.

Para a etapa de geração de características, observamos um maior número de sucessos frente ao *baseline* original, sendo que a grande maioria não obteve queda de desempenho. Com o mesmo comportamento da etapa anterior, vemos melhorias expressivas em algumas medidas como por exemplo nas bases de dados *Bank Marketing*, *Titanic*, *Credit-g* e *Pc1* o que demonstra que foi possível extrair mais informações das características existentes através das diversas transformações aplicadas durante o processo de geração de características aplicado. Em um único caso, no conjunto *Cardiotocography*, foram observadas melhoras em todas as medições o que ressalta a necessidade e a importância da etapa de geração aplicada.

Como a geração de características aumenta muito o espaço de variáveis a serem trabalhadas, e dado que estamos usando todos os parâmetros padrão do algoritmo de floresta aleatória descritos na tabela 4, o algoritmo, nessas configurações, seleciona sempre todas as características durante a criação das árvores da floresta. Por se tratar de um algoritmo do tipo árvore, conseguimos com facilidade acessar as características consideradas mais importantes pelo modelo apresentadas na coluna NCRF da tabela 8, valor importante

para denotar a diferença da utilização de um método do tipo invólucro (*wrappers*) como foi utilizado pelo modelo de florestas aleatórias frente a estratégia de seleção de características adotada para este trabalho e apresentada a seguir.

Pode-se perceber que, para os conjuntos *1*, *3* e *4*, a aplicação da seleção de característica teve um sucesso perceptível. A maior ênfase aqui vai para o conjunto *3*, que apresentava uma proporção de 0,92 de características propositalmente irrelevantes, nesse caso, ou seja, frente a casos com a existência de muito ruído, uma simples seleção de características aumenta consideravelmente o desempenho de aplicações de aprendizado de máquina. Já para o caso *1* e *2*, o número excessivo de características frente ao número de exemplos de treino também é filtrado e, ao diminuir o mesmo, o desempenho tende a crescer, pois as chances de *overfitting* e do uso de ruído (não proposital) também diminuem.

Para os conjuntos que não apresentaram melhora, inclusive apresentando piora em alguns casos, mostra-se a necessidade de uma solução mais robusta de engenharia de características caso o objetivo final seja sempre melhorar efetivamente as medidas de desempenho. Um ponto importante a ser levantado é que embora algumas medidas possam apresentar defasagem frente aos *baselines*, a diminuição do espaço de características diminui, consequentemente, o tamanho geral do conjunto de dados e o tempo de treinamento. Dependendo do contexto de aplicação da classificação, esta relação de custo-benefício pode ser considerada positiva (por exemplo, acurácia um pouco menor, mas solução produzida muito mais rapidamente).

5.3 Resultados do arcabouço completo

A análise apresentada na seção anterior mostra a aplicação e comportamento de cada etapa do arcabouço de maneira individual. A comparação de métricas em problemas de aprendizado de máquina exige adequação ao domínio e contexto de cada situação, o que torna uma tarefa não trivial assumir que o algoritmo A se sai melhor que o algoritmo B se ambos apresentam resultados diferentes para métricas igualmente diferentes.

Devido a isto, para a análise dos resultados finais obtidos frente a aplicação completa do arcabouço proposto, cada métrica de estudo será abordada separadamente levando em consideração os valores de medição apresentados na tabela 6.

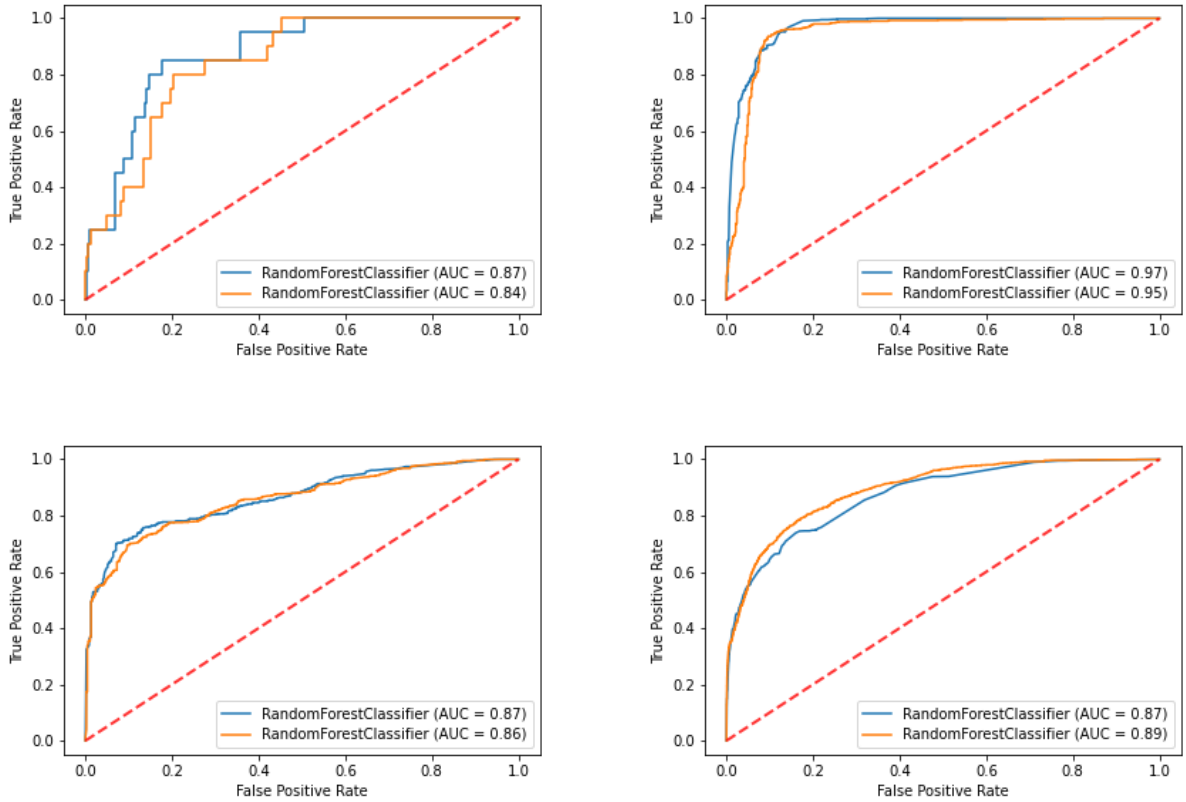
5.3.1 Análise dos resultados: AUC

A métrica AUC é utilizada para visualização da incidência de verdadeiros positivos e falsos positivos para todos os valores de cortes de probabilidades previstas em um problema de classificação. Apesar de ser uma métrica que possibilite uma visão mais geral sobre o desempenho do algoritmo em diferentes situações, como inerentemente devemos escolher um valor de corte em aplicações reais, o valor apresentado pela métrica em si pode não importar muito em determinadas situações.

Em nossos experimentos 60% das bases de dados utilizadas para comparação apresentaram melhorias em torno da métrica em relação ao *baseline*. Para melhor compreensão do comportamento da métrica utilizamos o gráfico da curva ROC, as linhas contínuas laranja representam o classificador *baseline* e azul representam o classificador aplicado após o arcabouço de engenharia de características, apresentando o desempenho de ambos os classificadores para a métrica de AUC, enquanto a linha vermelha tracejada representa a atuação hipotética de um classificador aleatório.

Na figura 6, os dois primeiros quadros são relativos às bases de dados Pc1 e Sylvine, nas quais ocorreram melhoras em termos de AUC. Observa-se que para a grande maioria dos pontos de corte a linha referente ao classificador final (azul) realmente se comporta de forma superior ao classificador *baseline* o que demonstra uma melhora consistente na métricas. Para o terceiro gráfico, relativo à base de dados Sylvine, embora a métrica final apresente melhora, a existência de alguns cruzamentos entre as linhas apresentadas no gráfico implica em que para determinados pontos de corte ambos os classificadores podem apresentar o mesmo desempenho, o que não torna possível afirmamos que um classificador é (sempre) melhor que o outro, isso agora depende da relação entre precisão e revocação desejada pelo usuário final. O último gráfico, referente a base de dados Housing, apresenta uma piora consistente na métrica de AUC.

Figura 6 – Comparação de curvas ROC para um grupo de bases, da esquerda para direita: Pc1, Sylvine, Phoneme e Housing



Fonte: Fernando Favoretti vital do Prado, 2021

5.3.2 Análise dos resultados: Precisão e Revocação

A partir da definição da precisão podemos claramente perceber que, ao avaliar a métrica especificamente estamos avaliando, a partir das predições positivas quantas são positivas de fato. É uma boa métrica para selecionar modelos quando o custo de se obter um resultado do tipo Falso Positivo é alto.

Por outro lado, a revocação diz respeito a quantas instâncias verdadeiramente positivas nosso modelo conseguiu capturar e classificar corretamente como tal, ou seja, ao contrário da métrica de precisão a revocação é uma boa métrica para priorizar a escolha do modelo quando temos um custo alto caso aconteça a ocorrência de predições do tipo Falso Negativo

Dados os aspectos de ambas as métricas e o escopo apresentado por este trabalho no qual, em tese, não temos acesso às reais necessidades do usuário de cada base, não é possível levar a discussão a finco da escolha do melhor modelo, por isto vamos recorrer puramente

as métricas. Em 40% dos casos apresentados na tabela 6 tivemos melhora em ambas as medidas o que significa que o arcabouço completo de engenharia de características conseguiu melhorar ambos os aspectos necessários para a criação de um modelo de aprendizado de máquina mais próximo do nível ótimo.

Nos casos restantes (onde não existiu melhora em ambas as métricas) a métrica de revocação melhorou em 80% dos casos e no único caso que não apresentou melhora expressiva, os resultados alcançados pelo classificador *baseline* e o classificador oriundo do arcabouço completo tiveram o mesmo valor. Observando a métrica de precisão igualmente para os casos restantes, não foi obtida melhora unicamente dessa métrica em nenhum dos casos.

Novamente, é difícil apontar melhorias consistentes observando as métricas separadamente dado que não sabemos a natureza do problema de cada conjunto de dados e nem dos futuros usuários do sistema.

5.3.3 Análise dos resultados: Medida F

A medida F deriva da interação entre as medidas de Precisão e Revocação observadas anteriormente, ela é útil para casos nos quais é desejado um melhor balanceamento para ambas as métricas, além de obter bom desempenho frente a diferentes tipos de balanceamento de dados. Levando em consideração a natureza dos objetivos deste trabalho, a medida F é uma boa métrica para realizar a comparação final dos classificadores, uma vez que estamos trabalhando com diferentes tipos de bases tanto na questão de necessidade do problema quanto na questão do balanceamento de classes.

Em nossos experimentos, 80% das comparações realizadas apresentaram melhoras na medida F, o que denota que o classificador construído após a aplicação do arcabouço de engenharia de características obteve maior sucesso em balancear as métricas de precisão e revocação vistas anteriormente, criando um modelo mais equilibrado. Nota-se também que, apesar da melhora, em alguns casos como com as bases de dados Pc1 e Housing, que de acordo com a tabela 3 são as bases de dados com o menor balanceamento entre os valores das classes, ainda apresentam valores bastante baixos para a medida F.

5.3.4 Análise dos resultados: Testes estatísticos

Para a avaliação dos resultados obtidos nos experimentos, foi aplicado o teste U de Mann-Whitneym (GIBBONS; CHAKRABORTI, 2020) sob a premissa de que estamos trabalhando com duas distribuições independentes, ou seja, dois grupos não pareados e de mesmo tamanho para verificar se ambos pertencem a mesma população.

O teste foi aplicado sobre as distribuições de probabilidades preditas para todos os conjuntos de dados do *baseline* comparando-as com as probabilidades preditas para os mesmos conjuntos após a aplicação do arcabouço completo, individualmente.

Tabela 7 – Comparação de resultados atingidos: Teste de Mann-Whitney

ID	Conjunto	P Valor
1	Normao	0,137
2	Credit-g	0,022
3	Sylvine	0,092
4	Phoneme	0,018
5	Cardio	0,001
6	Diabetes	0,026
7	Titanic	0,354
8	Bank Marketing	0,001
9	Pc1	0,124
10	Housing	0,003
11	Ailerons	0,001
12	Ecoli	0,273
13	Shuttle	0,002
14	Waveform	0,023
15	Blood Transfusion	0,001

Fonte: Fernando Favoretti, 2021

Aplicando um limiar padrão para o nível de significância do teste (mostrado a tabela 7 como p valor) de 0,05, podemos rejeitar ou não a hipótese nula que, no caso, diz respeito a existência ou não de diferenças significativas entre as probabilidades advindas das predições realizadas para cada base, antes e depois da aplicação do arcabouço completo.

A partir dos valores dispostos na tabela 7 nota-se que a maioria dos resultados obtidos rejeitam a hipótese nula, apresentando p valor menor que 0,05 concluindo que existem diferenças significativas entre as distribuições de probabilidades dos classificadores, mostrando que os resultados positivos, principalmente nos conjuntos de dados 2, 5, 6 e 8 obtidos observados tabela 6 são, de fato, melhores que a base de comparação.

Entretanto, para as bases de dados 1, 7, 9 e 12 os p valores obtidos não são suficientes para rejeição da hipótese nula, ou seja, mesmo que para algumas bases foram atingidos melhores desempenhos com a aplicação do arcabouço completo, não podemos afirmar, com segurança, que essa melhora de fato é fruto da eficiência do arcabouço ou de algum outro fato aleatório.

5.3.5 Análise dos resultados: Transformações aplicadas às bases

Com a aplicação do arcabouço completo de engenharia de características é necessário observar como as bases de dados sofreram alterações durante sua aplicação e o que tais alterações implicam no resultado final. A tabela 8 representa tais transformações, com a seguinte identificação de seus dados:

1. ID: Id para identificação única de um conjunto.
2. Conjunto: Nome do conjunto de dados utilizado.
3. NC: Número de características da base.
4. NCS: Número de características selecionadas após a aplicação do seletor principal, separadamente.
5. NCG: Número de características geradas pelo arcabouço.
6. NCRF: Número de características selecionadas com a utilização do algoritmo de floresta aleatória.
7. NCIG: Número de caraterísticas selecionadas previamente pelo filtro de ganho de informação do arcabouço.
8. NCMR: Número de características selecionadas previamente pelo modelo de ranqueamento do arcabouço.
9. NCSC: Número de características selecionadas ao final da aplicação do arcabouço completo.

É fácil observar o quanto o número de características cresce durante a etapa de geração de características aplicada, principalmente nas bases de dados que apresentam mais características do tipo numérico, como os conjuntos *Normao* e *Cardiotocography*, e que possibilitam por sua vez a aplicação de um número maior de transformações.

Com a aplicação dos arcabouços de seleção e geração separadamente, as bases apresentaram uma diferença expressiva na seleção de características realizadas para

Tabela 8 – Resultados - Configurações dos conjuntos de dados frente a aplicação de cada etapa do arcabouço

ID	Conjunto	NC	NCS	NCG	NCRF	NCIG	NCMR	NCSC
1	Normao	119	15	60387	654	43663	1866	28
2	Credit-g	21	13	5518	520	3302	474	31
3	Sylvine	21	9	10412	970	8622	970	18
4	Phoneme	6	5	640	280	603	229	18
5	Cardio.	36	19	26590	374	21332	1426	31
6	Diabetes	9	8	1645	374	1522	273	30
7	Titanic	12	10	1695	1456	722	25	30
8	Bank Marketing	17	9	3393	358	1928	95	20
9	Pc1	22	11	3719	456	3398	744	26
10	Housing	10	4	517	196	342	129	30
11	Airelons	40	12	16116	548	11480	1673	27
12	Ecoli	8	6	1247	37	1121	47	22
13	Shuttle	10	8	1640	32	1121	47	22
14	Waveform	41	12	41450	2562	36795	3595	29
15	Blood Transfusion	5	3	242	156	205	25	20

Fonte: Fernando Favoretti, 2021

modelagem (NCS x NCRF x NCMR x NCSC). Aqui pode-se observar principalmente que a seleção feita pela aplicação de modelos do tipo floresta aleatória (NCRF), em média, diminui, frente as características geradas, cada conjunto de dados em 73%, enquanto a seleção completa feita pelo arcabouço diminui em média cada conjunto em 98%.

Outro fator importante a ser levado em consideração é o tempo gasto em cada etapa. A tabela 9 apresenta os resultados coletados para cada base de dados utilizada em cada etapa do arcabouço completo, com as seguintes colunas:

1. ID: Id para identificação única de um conjunto.
2. Conjunto: Nome do conjunto de dados utilizado.
3. TMF : Tempo gasto para geração das meta características.
4. TIG: Tempo gasto para aplicação do filtro de ganho de informação.
5. TSF: Tempo gasto para seleção final de características.

A partir da tabela, nota-se que, no pior caso, o arcabouço completo utilizou de pouco menos de 4 horas para execução total, enquanto no melhor caso, para a base de menor tamanho, todo o processo é executado em menos de 5 minutos. Embora não seja possível a criação de uma base de comparação computando a quantidade de tempo necessário para realizar o processo de forma manual, sabe-se que a aplicação do processo

Tabela 9 – Resultados - Tempo para execução de cada etapa do arcabouço

ID	Conjunto	TMF	TIG	TSF
1	Normao	01:19:17h	00:06:08h	02:34:05h
2	Credit-g	00:03:01h	00:00:08h	00:07:58h
3	Sylvine	00:07:24h	00:00:19h	00:10:23h
4	Phoneme	00:00:17h	00:00:03h	00:05:43h
5	Cardio.	00:26:16h	00:01:02h	00:28:02h
6	Diabetes	00:00:55h	00:00:02h	00:04:22h
7	Titanic	00:00:53h	00:00:13h	00:03:34h
8	Bank Marketing	00:01:55h	00:00:13h	00:03:34h
9	Pc1	00:02:08h	00:00:05h	00:07:56h
10	Housing	00:00:13h	00:00:02h	00:05:45h
11	Ailerons	00:12:28h	00:00:44h	00:57:35h
12	Ecoli	00:00:36h	00:00:02h	00:09:18h
13	Shuttle	00:00:52h	00:00:08h	00:22:46h
14	Waveform	00:56:58h	00:02:32h	01:41:35h
15	Blood Transfusion	00:00:14h	00:00:06h	00:01:12h

Fonte: Fernando Favoretti, 2021

de modelagem para problemas de aprendizado de máquina é complexo e temporalmente custoso.

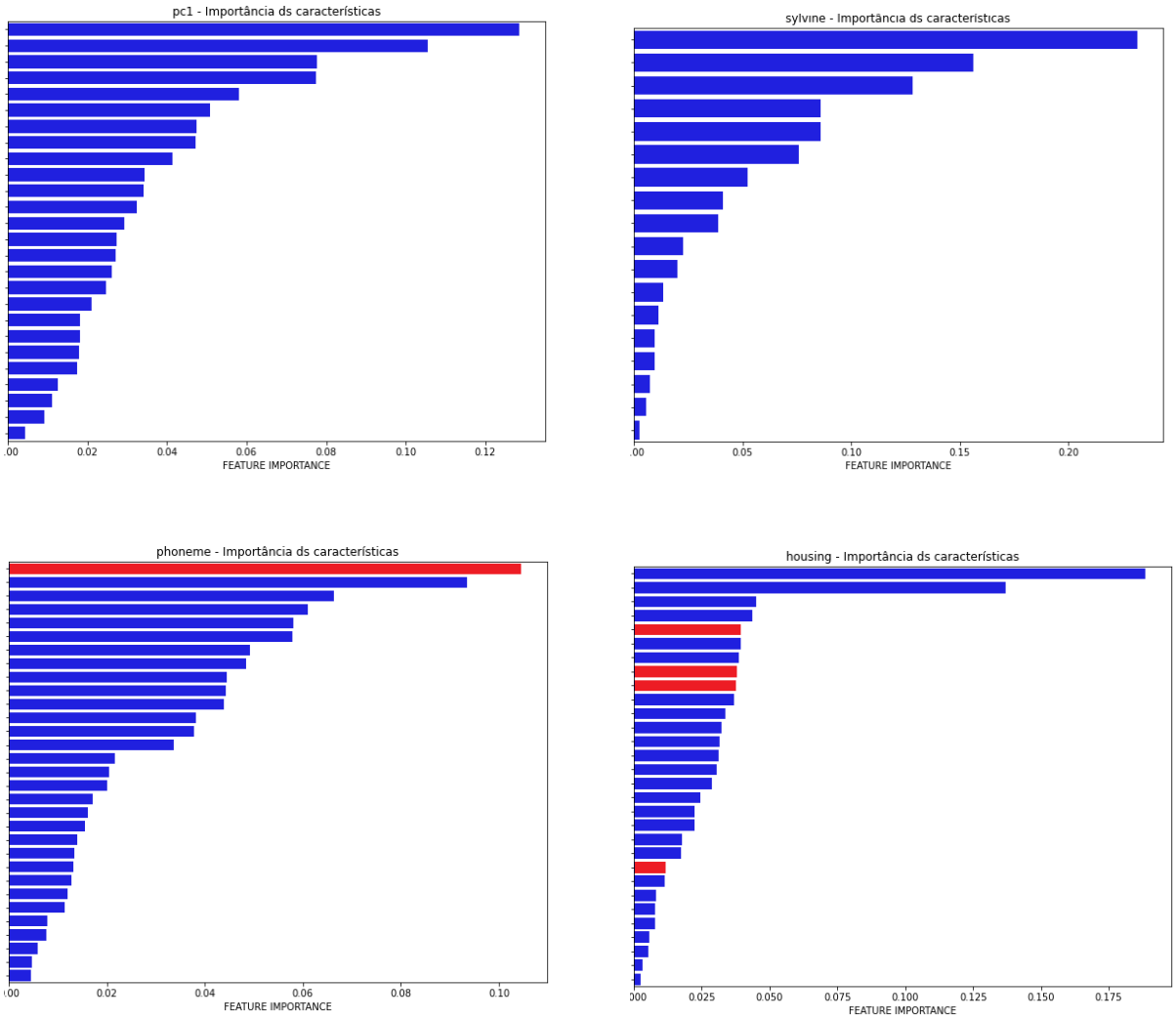
Depois de construídas, selecionar as características principais é uma das etapas mais custosas de todos os trabalhos discutidos no capítulo 3. O grande ganho temporal da aplicação ocorre justamente nessa etapa devido à aplicação feita pelo modelo de ranqueamento. Como o mesmo é exposto a outras bases e assim treinado para selecionar as características potencialmente mais relevantes, a aplicação da predição se dá em questão de milissegundos - por isso o mesmo não é expresso na tabela 9 - diminuindo muito o número de características que o seletor principal precisa lidar, como visto na coluna NCMR da tabela 8.

Com a aplicação da modelagem final utilizando algoritmos do tipo árvore, no caso florestas aleatórias, podemos ter fácil acesso à importância de cada característica utilizando uma técnica baseada em impureza, calculando a redução total no coeficiente Gini trazido pela adição de uma determinada características no modelo, quanto maior, mais importante é a característica. Essa técnica é conhecida como Importância de Gini (MENZE *et al.*, 2009).

Na figura 5.3.5 são representadas as importâncias das características utilizadas em quatro conjuntos de dados com comportamento diferente. Em azul estão as características

sintéticas criadas utilizando o processo de geração de características e em vermelho as características originalmente presentes em cada base de dados.

Figura 7 – Importância das características em cada base, da esquerda para direita: Pc1, Sylvine, Phoneme e Housing



Fonte: Fernando Favoretti vital do Prado, 2021

A maior presença das características geradas (azul) para todas as bases de dados mostra a importância do processo do arcabouço completo, capaz de extrair informações com maior sinal para a etapa de modelagem ao se comparar com as características originais. Nota-se também que, o pior caso, no qual o algoritmo final acarretou em piora das métricas frente ao nosso *baseline* é aquele em que se observa o maior número de características originais apresentadas ao modelo final, sugerindo que o nível de informação carregado por tais características é suficiente para obter-se bons resultados em uma etapa de modelagem, sem a necessidade de aplicar transformações diversas.

5.4 Considerações finais

Ao analisar os resultados obtidos nos experimentos executados foi possível identificar que a aplicação do arcabouço completo de engenharia de características, quando comparado tanto com a aplicação do mesmo em partes (seleção e geração aplicadas separadamente) quanto ao *baseline*, permite, na maior parte dos casos, a obtenção de melhoras concisas nas métricas utilizadas para avaliar o desempenho da classificação. Entretanto, cabe ao executor do processo a escolha apropriada da métrica dentro da natureza de cada situação a ser trabalhada no âmbito de problemas que envolvem o uso do aprendizado de máquina.

O uso do arcabouço separadamente apresenta o comportamento esperado, com nenhuma melhora ou uma piora clara quando os conjuntos originais foram expostos apenas ao seletor de características. Essa aplicação separada pode funcionar bem caso as bases de dados apresentem algum tipo de ruído ou diversas características irrelevantes, mas com a aplicação do pré-processamento de todas as bases, muitos desses pontos são previamente tratados o que diminui o poder do método de seleção por si só.

Por outro lado, a geração de características aplicada separadamente acarreta, em média, bons resultados uma vez que os algoritmos utilizados para a medição final conseguem selecionar por si só características consideradas relevantes o que mitiga levemente o possível ruído criado pelas operações sintéticas da etapa. Sendo assim, como desvantagem tem-se o número elevado de características que devem ser apresentadas ao modelo e mantidas por todo o arcabouço, o que aumenta consideravelmente o tempo de execução da modelagem e a complexidade do problema, podendo acarretar na criação de ruído ou de características irrelevantes.

A adição do modelo de ranqueamento para a seleção inicial de características melhora significativamente o tempo em que as características candidatas são comparadas e selecionadas, em comparação com os trabalhados revisados. A única troca ocorre entre desempenho do modelo e separação das características mais relevantes em um dado período de tempo em que, mesmo que o modelo de ranqueamento tenha sido otimizado para a métrica de revocação, ainda podem ocorrer falsos negativos que seriam relevantes para as etapas conseguintes.

Por fim, vale destacar a natureza não paramétrica do arcabouço, facilitando muito o uso para experimentação ou em situações que seja necessário realizar uma busca rápida

por possíveis soluções otimizadas em comparação com a original. O sistema se mostrou capaz de manter um comportamento estável para bases de diferentes tipos, tamanhos e balanceamentos.

6 Conclusão

O objetivo principal deste trabalho foi desenvolver uma solução única que automatize o processo de engenharia de características e que tenha resultados melhores frente a processos não automatizados e outros *baselines* automatizados, tanto na questão de tempo de execução quanto em uma série de métricas da área. Com a criação do arcabouço, um conjunto de experimentos foi realizado e organizado de maneira separada para que seja possível compreender a importância de cada etapa disponível no mesmo.

Defende-se que os objetivos deste trabalho foram atingidos, uma vez que com o arcabouço criado a análise dos resultados mostra uma melhora concisa em 80% dos experimentos realizados (de acordo com a métrica de medida F). Nota-se que a escolha de uma métrica deve depender inerentemente da natureza de cada problema, não sendo possível chegar a conclusão final que o arcabouço encontre soluções otimizadas para todas as situações. Cabe, a seu utilizador final, definir a melhor métrica de acompanhamento para realizar as devidas comparações.

Diante dos resultados obtidos, a hipótese delineada para este trabalho foi confirmada, pois, a partir dos resultados obtidos com a execução dos experimentos comprovou-se que a união, em um único método, de diferentes metodologias de automação da etapa de engenharia de características, criadas para agir separadamente em diferentes contextos, consegue atingir bons resultados e ser executada de maneira simples.

Este trabalho apresenta uma contribuição potencial para a área de aprendizado de máquina no geral, tanto na evolução da pesquisa no aspecto de metodologias de automação de engenharia de características (dentro do escopo de *AutoML*), quanto para a possível aplicação do trabalho em diferentes pesquisas acadêmicas ou profissionais, uma vez que a solução disponibilizada permite a aplicação simples da etapa de engenharia de características em problemas supervisionados de classificação.

O escopo deste projeto limita-se a aplicação do mesmo para problemas supervisionados, restringindo-se ao estudo de técnicas clássicas na área de aprendizado de máquina (principalmente para algoritmos baseados em árvores), não levando em conta, por exemplo, metodologias de aprendizado profundo.

A ferramenta de automação criada e disponibilizada em formato de código aberto não teve sua aplicabilidade e facilidade de uso mensuradas, pois isto não é parte primordial do objetivo principal do projeto.

6.1 Trabalhos futuros

Embora o arcabouço tenha sido construído de forma inteiramente paralela, o mesmo ainda funciona utilizando de recursos de uma única máquina, que necessariamente deve possuir grande quantidade de memória e capacidade de processamento. Todos os experimentos apresentados aqui foram realizados em máquinas virtuais dispondo de 128GB RAM e 32 processadores, o que permite a rápida execução de experimentos mais complexos. Sabendo desse limitador, um potencial ajuste futuro pode se dar na forma da utilização de computação distribuída fazendo uso de uma linguagem como spark, o que aumenta a capacidade de escala da solução.

Para a evolução da solução em si, a adaptação do arcabouço para problemas supervisionados do tipo regressão se torna completamente possível, uma vez que a mesma pode ser beneficiada dos artefatos disponíveis para este trabalho como no código fonte disponível em Python e no modelo de ranqueamento de características já treinado.

Referências[1]

ABBASI, B. Z.; HUSSAIN, S.; FAISAL, M. I. An automated text classification method: Using improved fuzzy set approach for feature selection. In: *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. [S.l.: s.n.], 2019. Citado 2 vezes nas páginas 26 e 31.

AGUIAR, M.; GREVE, F.; COSTA, G. Um arcabouço flexível para integração de análise preditiva e prescritiva, com atuação. In: *Anais do XIII Simpósio Brasileiro de Sistemas de Informação*. [S.l.: s.n.], 2017. p. 174–181. Citado na página 14.

BLUM, A. L.; LANGLEY, P. Selection of relevant features and examples in machine learning. *Artificial intelligence*, Elsevier, v. 97, n. 1-2, p. 245–271, 1997. Citado na página 42.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. *Computers & Electrical Engineering*, Elsevier, v. 40, n. 1, p. 16–28, 2014. Citado na página 20.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: ACM. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.], 2016. p. 785–794. Citado na página 46.

DAS, S.; CAKMAK, U. M. *Hands-On Automated Machine Learning: A beginner's guide to building automated machine learning systems using AutoML and Python*. [S.l.]: Packt Publishing Ltd, 2018. Citado na página 24.

EDWARDS, S. *Elements of Information Theory, Thomas M. Cover, Joy A. Thomas, John Wiley & Sons, Inc.(2006)*. [S.l.]: Pergamon, 2008. Citado na página 22.

ELSHAWI, R.; MAHER, M.; SAKR, S. Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287*, 2019. Citado na página 14.

FARD, S. M. H.; HAMZEH, A.; HASHEMI, S. Proposing a reinforcement learning based approach for feature selection. In: *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*. [S.l.: s.n.], 2012. Citado na página 28.

GE, Z.; Song, Z.; Ding, S. X.; Huang, B. Data mining and analytics in the process industry: The role of machine learning. *IEEE Access*, 2017. Citado na página 14.

GIBBONS, J. D.; CHAKRABORTI, S. *Nonparametric statistical inference*. [S.l.]: CRC press, 2020. Citado na página 58.

GOCHT, A.; LEHMANN, C.; SCHöNE, R. A new approach for automated feature selection. In: *2018 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2018. Citado 6 vezes nas páginas 8, 23, 26, 29, 38 e 43.

GUYON, I.; GUNN, S.; NIKRAVESH, M.; ZADEH, L. A. *Feature extraction: foundations and applications*. [S.l.]: Springer, 2008. v. 207. Citado na página 21.

HO, T. K. Random decision forests. In: IEEE. *Proceedings of 3rd international conference on document analysis and recognition*. [S.l.], 1995. v. 1, p. 278–282. Citado na página 48.

[1] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

KATZ, G.; SHIN, E. C. R.; SONG, D. Explorekit: Automatic feature generation and selection. In: IEEE. *2016 IEEE 16th International Conference on Data Mining (ICDM)*. [S.l.], 2016. p. 979–984. Citado 8 vezes nas páginas 8, 16, 27, 34, 38, 39, 45 e 46.

KAUL, A.; MAHESHWARY, S.; PUDI, V. Autolearn — automated feature generation and selection. In: *2017 IEEE International Conference on Data Mining (ICDM)*. [S.l.: s.n.], 2017. Citado 2 vezes nas páginas 26 e 39.

KAUL, A.; MAHESHWARY, S.; PUDI, V. Autolearn—automated feature generation and selection. In: IEEE. *2017 IEEE International Conference on Data Mining (ICDM)*. [S.l.], 2017. p. 217–226. Citado 4 vezes nas páginas 14, 15, 18 e 35.

KHURANA, U.; SAMULOWITZ, H.; TURAGA, D. Feature engineering for predictive modeling using reinforcement learning. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2018. Citado na página 16.

KHURANA, U.; TURAGA, D.; SAMULOWITZ, H.; PARTHASRATHY, S. Cognito: Automated feature engineering for supervised learning. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. [S.l.: s.n.], 2016. Citado 3 vezes nas páginas 16, 27 e 33.

KITCHENHAM, B.; BRERETON, O. P.; BUDGEN, D.; TURNER, M.; BAILEY, J.; LINKMAN, S. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, Elsevier, v. 51, n. 1, p. 7–15, 2009. Citado na página 25.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. *Artificial intelligence*, Elsevier, v. 97, n. 1-2, p. 273–324, 1997. Citado na página 20.

LATHAM, P. E.; ROUDI, Y. Mutual information. *Scholarpedia*, v. 4, n. 1, p. 1658, 2009. Citado na página 22.

LENSEN, A.; XUE, B.; ZHANG, M. Automatically evolving difficult benchmark feature selection datasets with genetic programming. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. New York, NY, USA: ACM, 2018. (GECCO '18), p. 458–465. ISBN 978-1-4503-5618-3. Disponível em: ⟨http://doi.acm.org/10.1145/3205455.3205552⟩. Citado 2 vezes nas páginas 26 e 31.

LIU, K.; FU, Y.; WANG, P.; WU, L.; BO, R.; LI, X. Automating feature subspace exploration via multi-agent reinforcement learning. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: ACM, 2019. (KDD '19), p. 207–215. ISBN 978-1-4503-6201-6. Disponível em: ⟨http://doi.acm.org/10.1145/3292500.3330868⟩. Citado 2 vezes nas páginas 26 e 27.

LUO, Y.; WANG, M.; ZHOU, H.; YAO, Q.; TU, W.-W.; CHEN, Y.; DAI, W.; YANG, Q. Autocross: Automatic feature crossing for tabular data in real-world applications. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: ACM, 2019. (KDD '19), p. 1936–1945. ISBN 978-1-4503-6201-6. Disponível em: ⟨http://doi.acm.org/10.1145/3292500.3330679⟩. Citado 5 vezes nas páginas 26, 31, 32, 33 e 39.

MENZE, B. H.; KELM, B. M.; MASUCH, R.; HIMMELREICH, U.; BACHERT, P.; PETRICH, W.; HAMPRECHT, F. A. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, Springer, v. 10, n. 1, p. 1–16, 2009. Citado na página 61.

MITCHELL, T. M. *et al.* Machine learning. McGraw-hill New York, 1997. Citado na página 18.

MOHR, F.; WEVER, M.; HüLLERMEIER, E. Ml-plan: Automated machine learning via hierarchical planning. *Machine Learning*, v. 107, n. 8, p. 1495–1515, 2018. Citado 2 vezes nas páginas 15 e 19.

MOHR, F.; WEVER, M. D.; HüLLERMEIER, E. Ml-plan: Automated machine learning via hierarchical planning. *Machine Learning*, Springer, 2018. Citado na página 15.

OSMAN, H.; GHAFARI, M.; NIERSTRASZ, O. Automatic feature selection by regularization to improve bug prediction accuracy. In: *2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE)*. [S.l.: s.n.], 2017. Citado 5 vezes nas páginas 20, 26, 27, 28 e 31.

PARGENT, F.; BISCHL, B.; THOMAS, J. *A benchmark experiment on how to encode categorical features in predictive modeling*. Tese (Doutorado) — M. Sc. Thesis, Ludwig-Maximilians–Universitat Munchen, pp12, 2019. Citado na página 38.

PRADO, F. F. V. do; DIGIAMPIETRI, L. A systematic review of automated feature engineering solutions in machine learning problems. In: *Simpósio Brasileiro de Sistemas de Informação (SBSI 2020)*. [S.l.: s.n.], 2020. Citado 3 vezes nas páginas 16, 25 e 28.

QUANMING, Y.; MENGSHUO, W.; HUGO, J. E.; ISABELLE, G.; YI-QI, H.; YU-FENG, L.; WEI-WEI, T.; QIANG, Y.; YANG, Y. Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306*, 2018. Citado 2 vezes nas páginas 20 e 23.

REBALA, G.; RAVI, A.; CHURIWALA, S. *An Introduction to Machine Learning*. [S.l.]: Springer, 2019. Citado na página 18.

SAID, F. B.; ALIMI, A. M. Anofs: Automated negotiation based online feature selection method. In: *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*. [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 26 e 30.

SHILBAYEH, S.; VADERA, S. Feature selection in meta learning framework. In: *2014 Science and Information Conference*. [S.l.: s.n.], 2014. Citado 3 vezes nas páginas 20, 27 e 30.

SILVA, L.; PERES, S.; BOSCARIOLI, C. *Introdução a Mineração de Dados com aplicações em R*. [S.l.]: GEN LTC, 2017. 495 p. ISBN 9788535284478. Citado na página 14.

SMETANNIKOV, I.; DEYNEKA, A.; FILCHENKOV, A. Meta learning application in rank aggregation feature selection. In: *3rd International Conference on Soft Computing Machine Intelligence (ISCMI)*. [S.l.: s.n.], 2016. Citado na página 28.

SU, Y.; QI, K.; DI, C.; MA, Y.; LI, S. Learning automata based feature selection for network traffic intrusion detection. In: *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. [S.l.: s.n.], 2018. Citado na página 28.

TANFILEV, I.; FILCHENKOV, A.; SMETANNIKOV, I. Feature selection algorithm ensembling based on meta-learning. In: *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. [S.l.: s.n.], 2017. Citado na página 27.

UKIL, A.; SAHU, I.; PURI, C.; MUKHERJEE, A.; SINGH, R.; BANDYOPADHYAY, S.; PAL, A. Automodeling: Integrated approach for automated model generation by ensemble selection of feature subset and classifier. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018. Citado na página 27.

VALE, K. O.; FEITOSA-NETO, A.; CANUTO, A. M. P. Using a reinforcement-based feature selection method in classifier ensemble. In: *2010 10th International Conference on Hybrid Intelligent Systems*. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 28 e 30.

VERGARA, J. R.; ESTÉVEZ, P. A. A review of feature selection methods based on mutual information. *Neural computing and applications*, Springer, v. 24, n. 1, p. 175–186, 2014. Citado na página 43.

YANG, H.; MOODY, J. Feature selection based on joint mutual information. In: CITESEER. *Proceedings of international ICSC symposium on advances in intelligent data analysis*. [S.l.], 1999. v. 1999, p. 22–25. Citado na página 43.

YANG, H. H.; MOODY, J. E. Data visualization and feature selection: new algorithms for nongaussian data. In: *NIPS*. [S.l.: s.n.], 1999. v. 12. Citado na página 30.