INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia Associada à Universidade de São Paulo

Modelo preditivo de infecção hospitalar utilizando aprendizado de máquina

PATRÍCIA PEDROSA MOREIRA MENDES

Dissertação apresentada como parte dos requisitos para obtenção do Grau de Mestre em Ciências na Área de Tecnologia Nuclear - Aplicações

Orientador:
Prof. Dr. Mário Olímpio de Menezes

São Paulo
2023

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia Associada à Universidade de São Paulo

Modelo preditivo de infecção hospitalar utilizando aprendizado de máquina

Versão Corrigida
Versão Original disponível no IPEN

PATRÍCIA PEDROSA MOREIRA MENDES

Dissertação apresentada como parte dos requisitos para obtenção do Grau de Mestre em Ciências na Área de Tecnologia Nuclear - Aplicações

Orientador:
Prof. Dr. Mário Olímpio de Menezes

São Paulo
2023

Como citar:

PEDROSA MOREIRA MENDES, P. . ***Modelo preditivo de infecção hospitalar utilizando aprendizado de máquina***. 2023. 99 f. Dissertação (Mestrado Profissional em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN, São Paulo. Disponível em: ‹http://repositorio.ipen.br› (data de consulta no formato: dd/mm/aaaa)

Ficha catalográfica elaborada pelo Sistema de geração automática da Biblioteca IPEN, com os dados fornecidos pelo(a) autor(a).

Pedrosa Moreira Mendes, Patricia
Modelo preditivo de infecção hospitalar utilizando aprendizado de máquina / Patricia Pedrosa Moreira Mendes; orientador Mário Olimpio de Menezes. -- São Paulo, 2023.
99 f.

Dissertação (Mestrado Profissional) - Programa de Pós-Graduação em Tecnologia Nuclear (Aplicações) -- Instituto de Pesquisas Energéticas e Nucleares, São Paulo, 2023.

1. Aprendizado de máquina. 2. Infecção Hospitalar. 3. Variáveis categóricas . 4. Hiperparâmetros. I. Olimpio de Menezes, Mário, orient. II. Título.

FOLHA DE APROVAÇÃO

Autor: Patrícia Pedrosa Moreira Mendes

Título: Modelo preditivo de infecção hospitalar utilizando aprendizado de máquina

Dissertação apresentada ao Programa de Pós-Graduação em tecnologia Nuclear da Universidade de São Paulo para obtenção o título de mestre em Ciências.

Data: __/__/___

Banca Examinadora

Prof. Dr. _____

Instituição: _____Julgamento: _____

Prof. Dr. _____

Instituição: _____Julgamento: _____

Prof. Dr. _____

Instituição: _____Julgamento: _____

Prof. Dr. _____

Instituição: _____Julgamento: _____

Prof. Dr. _____

Instituição: _____Julgamento: _____

AGRADECIMENTOS

A Deus por sua providência em dar sentido e meta às minhas escolhas, e por sua ação constante em minha vida.

Aos meus pais Jacilda e Gercimar, e minhas irmãs, pelo carinho e apoio durante minha trajetória de vida pessoal, acadêmica e profissional.

Ao meu esposo Felipe e minha filha Marcela, por toda a companhia e palavras de incentivo, me ajudaram a acreditar que poderia chegar neste momento

Ao Professor Dr. Mário Olímpio de Menezes pela atenção, orientação e boa vontade durante o decorrer deste projeto.

Por fim, gostaria de agradecer a Fundação São Francisco Xavier e Hospital Márcio Cunha pela oportunidade e fornecimento dos dados para que este projeto fosse possível de se realizar.

RESUMO

Cada vez mais o aprendizado de máquina vem ganhando espaço na área da saúde devido à sua capacidade de melhorar a predição de doenças e auxiliar profissionais na condução dos tratamentos clínicos. A infecção hospitalar é o evento negativo mais comum para pacientes hospitalizados e continua a se constituir em séria ameaça à segurança dos pacientes. O objetivo deste trabalho foi encontrar uma técnica de aprendizado de máquina otimizada e eficiente que possa prever efetivamente a condição da infecção hospitalar, identificando os principais fatores responsáveis por esta condição. Neste trabalho, usamos seis técnicas de aprendizado de máquina, os algoritmos utilizados no trabalho foram Random Forest, Regressão logística, KNN, Adaboost, Bagging e XGBoost; também foram empregadas técnicas modernas de explicabilidade a estes algoritmos. Nesse processo, os dados foram divididos em dados de treino e de teste, os modelos foram treinados em um primeiro momento com os hiperparâmetros padrões, em um segundo momento os modelos foram treinados com hiperparâmetros aprimorados. Os modelos que apresentaram as melhores métricas foram o XGBoost e Random Forest, o XGBoost apresentou o melhor resultado em todas as métricas, exceto na Precisão, o Random Forest obteve o segundo melhor resultado na acurácia e na precisão, na validação cruzada o resultado foi o mesmo que o XGBoost. Para a explicabilidade do modelo foi utilizada a biblioteca SHAP, foi avaliado como o valor de cada variável influenciou no resultado alcançado pelo modelo preditivo XGBoost, SHAP apontou como mais importante as variáveis: NR_DIA_INTERNADO (quantidade de dias de internação), CD_DOENCA_PRINCIPAL_E (CID-10 Classificação internacional de doenças), DS_PROC_PRINCIPAL_E (Procedimento principal durante internação) e QT_DIAS_SONDA_VESICAL (Dias que o paciente ficou com sonda vesical). O estudo mostrou-se viável à adoção de aprendizado de máquina nas rotinas da pesquisa em saúde, no trabalho da comissão de infecção hospitalar e nas iniciativas de inovação nas instituições de saúde no Brasil.

Palavras-chave: Aprendizado de máquina. Infecção Hospitalar. Explicabilidade.

ABSTRACT

Machine learning is increasingly gaining ground in the health area due to its ability to improve disease prediction and assist professionals in conducting clinical treatments. Hospital infection is the most common negative event for hospitalized patients and continues to pose a serious threat to patient safety. The objective of this work was to find an optimized and efficient machine learning technique that can effectively predict the condition of nosocomial infection, identifying the main factors responsible for this condition. In this work, we used six machine learning techniques, the algorithms used in the work were Random Forest, Logistic Regression, KNN, Adaboost, Bagging and XGBoost; modern explainability techniques were also used for these algorithms. In this process, the data were divided into training and test data, the models were trained in a first moment with standard hyperparameters, in a second moment the models were trained with improved hyperparameters. The models that presented the best metrics were XGBoost and Random Forest, XGBoost presented the best result in all metrics, except for Precision, Random Forest obtained the second best result in accuracy and precision, in cross-validation the result was the same as XGBoost. For the explanation of the model, the SHAP library was used, it was evaluated how the value of each variable influenced the result achieved by the predictive model XGBoost, SHAP pointed out as the most important variables: NR_DIA_INTERNADO (number of days of hospitalization), CD_DOENCA_PRINCIPAL_E (ICD-10 International classification of diseases), DS_PROC_PRINCIPAL_E (Main procedure during hospitalization) and QT_DIAS_SONDA_VESICAL (Days that the patient had a urinary catheter). The study proved to be feasible for the adoption of machine learning in health research routines, in the work of the hospital infection committee and in innovation initiatives in health institutions in Brazil.

Keywords: Machine learning. Hospital Infection. Explainability.

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AED Análise Exploratória de Dados
AM Aprendizado de Máquina
AUROC Area Under Receiver Operating Characteristic
BD Banco de Dados
CEP Comitê de Ética em Pesquisa
CCIH Controle de Infecção Hospitalar
CSV Comma-Separated Values
EHRs Electronic Health Records
FN Falso Negativo
FP Falso Positivo
IA Inteligência Artificial
IH Infecção Hospitalar
IRAS Infecção Relacionada à Assistência à Saúde
LIME Locally Interpretable Modelagnostic Explanations
LR Logistic Regression
ML Machine Learning
NLP Natural Language Processing
KNN K-nearest neighbors
RES Registros Eletrônicos na Saúde
SHAP Shapley Additive Explanations
SGBD Sistema Gerenciador de Banco de Dados
RF Random Forest
RL Regresão Logística
VC Validação Cruzada
VN Verdadeiros Negativos
VP Verdadeiros Positivos
XAI Inteligência Artificial Explicável
XGBoost eXtreme Gradient Boosting package
ZB Zettabyte

SUMÁRIO

1. INTRODUÇÃO

A infecção hospitalar, é toda infecção adquirida durante a internação hospitalar ou então relacionada a algum procedimento ou dispositivo utilizado durante a assistência à saúde em hospitais, podendo manifestar-se inclusive após a alta. Desde 1998, o termo infecção hospitalar tem sido substituída por Infecção Relacionada à Assistência à Saúde (IRAS). Essa mudança abrange não só a infecção adquirida no hospital, mas também aquela relacionada a procedimentos feitos em ambulatório, durante cuidados domiciliares e à infecção ocupacional adquirida por profissionais de saúde (médicos, enfermeiros, fisioterapeutas, entre outros) (BRASIL, 1998).

Embora o nome tenha mudado, a comunidade médica, pesquisadores e a sociedade ainda utilizam o termo Infecção Hospitalar. Neste trabalho, optaremos por utilizar o termo infecção hospitalar, em consonância com a terminologia amplamente adotada.

A grande maioria das infecções hospitalares são causadas principalmente da relação de desequilíbrio entre três fatores, os quais incluem a condição clínica do paciente, a virulência e inóculo dos micro-organismos e fatores relacionados à hospitalização, tais como procedimentos invasivos, condições do ambiente e atuação do profissional de saúde.

Os recentes avanços nos sistemas de inteligência artificial (IA) na medicina e nos cuidados de saúde oferecem oportunidades extraordinárias em diversas áreas de profundo interesse social, ao mesmo tempo em que levantam questões e limitações substanciais. Isso requer uma consideração cuidadosa sobre sua implementação e como elas podem afetar, e até mesmo alterar, definições básicas no contexto médico (GÓMEZ-GONZÁLEZ ET AL, 2020).

Diante da gravidade do problema da infecção hospitalar, em 1997 foi instituído no Brasil a Lei Federal n° 6.431, onde obriga a existência da Comissão de Controle de Infecção Hospitalar (CCIH). Em 1998, o Ministério da Saúde publicou a Portaria n° 2.616, regulamentando a criação das CCIH, essa portaria define critérios para organização, bem como para o diagnóstico das infecções hospitalares, orientações sobre a vigilância e recomendações sobre a higiene

das mãos. Ainda segundo a portaria, o regulamento deve ser adotado em todo o território nacional nas atividades hospitalares de assistência à saúde.

O controle de infecção hospitalar constitui um dos parâmetros para garantir a qualidade do cuidado prestado. A CCIH é o setor responsável por planejar, elaborar, implementar, manter e avaliar o Programa de Controle de Infecção Hospitalar com o objetivo de reduzir ao máximo possível a incidência das infecções hospitalares.

Para este trabalho foram utilizados os dados do hospital Márcio Cunha que é um hospital geral de alta complexidade, com 548 leitos que atende a uma população de mais de 800 mil habitantes no leste de Minas Gerais e presta serviços nas áreas de ambulatório, pronto-socorro, internação, terapia intensiva, terapia renal substitutiva e medicina diagnóstica. Conta com cerca de 500 médicos em 50 especialidades, atende a pacientes do SUS – Sistema Único de Saúde e convênios, sendo o 5º em números de internações pelo SUS, em Minas Gerais.

Os dados médicos dos pacientes do hospital Márcio Cunha são armazenados como registros eletrônicos de saúde, também conhecido *electronic health records* (EHRs), o EHRs utilizado pela instituição é o Tasy da Philips que contém funcionalidades básicas, como dados demográficos do paciente, anotações médicas, avaliações de enfermagem, prescrição médica, resumos de alta, relatórios de radiologia, resultados de testes diagnósticos e entradas de pedidos de medicamentos.

Atualmente a equipe da CCIH do hospital Márcio Cunha utiliza os registros eletrônicos de saúde para realizar a análise das informações e identificação dos pacientes com infecção hospitalar, a análise é feita através da leitura das informações estruturadas, como, diagnóstico, exames laboratoriais, sinais vitais e prescrição de antibióticos. A análise contempla também a avaliação de campos não estruturados, como por exemplo as evoluções médicas que são anotações livres em campos abertos que são importantes para avaliação do quadro do paciente.

A partir da leitura do prontuário eletrônico do paciente a CCIH registra no sistema as infecções confirmadas, na ficha de ocorrência informam a data da infecção, data da origem, clinica, setor de internação do paciente, topografia da infecção e o sítio.

O processo de leitura do prontuário é um processo complexo e moroso e que demanda a atuação de diversos profissionais da área de saúde, além de ser um processo suscetível a erro.

Pesquisas em saúde podem se beneficiar do emprego de técnicas de aprendizado de máquina (AM) para verificar a combinação de variáveis que melhor predizem um determinado resultado, bem como verificar seus valores de corte. Em um estudo recente sobre aprendizado de máquina na área da saúde, o termo aprendizado de máquina foi definido como um conjunto de técnicas e métodos que auxiliam na identificação e detecção de padrões em dados médicos. Os métodos de AM criam uma variedade de modelos preditivos, usando um algoritmo ou mais (LUZ et al., 2020).

Apesar da grande relevância no uso de aprendizado de máquina, ainda temos os desafios com a implementação dos modelos preditivos, pois os *EHRs* podem ser inconsistentes e ruidosos, podem conter muitos valores ausentes e frequentemente incluem campos de texto não estruturados, um conjunto de dados desbalanceado também é um problema comum que pode influenciar o desempenho do modelo, exigindo, portanto, uma estratégia aprimorada para lidar com essa preocupação. No entanto, o próprio fato de esses dados estarem disponíveis eletronicamente em grandes volumes oferece potencial para aplicação de AM, inclusive no campo do gerenciamento de infecções. (AL-AHMARI; NADEEM, 2021; LUZ et al., 2020).

Este trabalho foi submetido e aprovado pelo CEP (Comitê de Ética em Pesquisa) do Hospital Márcio Cunha, na plataforma Brasil, registro 17987119.3.0000.8147, aprovado em 10/01/2020.

O trabalho está organizado em cinco capítulos, além desta Introdução, teremos a seguir a apresentação o Referencial Teórico, Metodologia utilizada, Resultados e Discussão e por fim as Conclusões.

1.1 Motivação

A infecção hospitalar no Brasil é um problema de saúde significativo e tem sido alvo de preocupação e esforços para prevenção e controle. Dados de estudos epidemiológicos e relatórios governamentais indicam que a taxa de infecções hospitalares no país ainda é elevada.

Segundo a OMS, estima-se que 1 milhão de pacientes morrem em decorrência de infecções hospitalares no mundo por ano. No Brasil, a infecção hospitalar mata 45 mil por ano, considera-se que a taxa de infecção atinja 14% das internações no Brasil.

O uso de técnicas de aprendizado de máquina em pesquisas na área da saúde traz benefícios importantes ao examinar a combinação de variáveis que melhor predizem um resultado específico e estabelecer seus valores de referência. O emprego do aprendizado de máquina na saúde permite a identificação e detecção de padrões em dados médicos, bem como a criação de vários modelos preditivos, por meio da aplicação de um ou mais algoritmos. Isso amplia as possibilidades de análise e contribui para uma compreensão mais profunda e precisa dos dados e do contexto clínico.

Além disso, desenvolver modelos de classificação para prever infecção hospitalar pode trazer benefícios significativos, incluindo a detecção precoce, melhoria na tomada de decisão, redução de custos e recursos, e melhoria na qualidade dos cuidados de saúde.

Diante deste contexto, o tema tem a sua relevância, já que, a infecção hospitalar é uma das principais causas de mortalidade entre pacientes hospitalizados e as ações desenvolvidas para o seu controle têm grande importância na promoção da saúde em geral.

1.2 Objetivos

Esta pesquisa tem o objetivo de desenvolver modelos de classificação que possam auxiliar na previsão de infecção hospitalar, a partir do quadro clínico do paciente, utilizando-se de aprendizado de máquina por meio de seis algoritmos de classificação: Regressão Logística, Random Forest, Adaboost, Bagging, KNN e XGBoost. Isso será feito com base em informações do quadro clínico de

cerca de 178.481 mil pacientes internados entre março de 2015 e dezembro de 2019, além de passar pelas etapas de um projeto de *data science*, este trabalho também apresentará no capitulo de resultados e discussão sobre a explicabilidade do modelo.

Os algoritmos de aprendizado de máquina orientados por inteligência artificial têm o potencial de oferecer contribuições significativas e aprimorar a eficiência das respostas relacionadas à infecção hospitalar. Esses algoritmos têm a capacidade de melhorar os modelos de prognóstico amplamente utilizados em hospitais ao redor do mundo, permitindo prever de forma mais precisa os resultados de saúde associados às infecções hospitalares em diferentes ambientes geográficos e sistemas de saúde. Isso possibilita uma abordagem mais efetiva na prevenção e tratamento dessas infecções.

2. REFERENCIAL TEÓRICO

Neste capitulo é apresentado os conceitos teóricos relacionados a inteligência artificial e aprendizado de máquina na saúde, de forma a apresentar os conceitos relevantes para o entendimento e desenvolvimento da pesquisa.

Primeiramente são apresentados com maior clareza os conceitos de *big data, data science* e inteligência artificial como aprendizado máquina. Na sequência são apontados os conceitos de linguagem de programação Python e algumas de suas bibliotecas. Em seguida são expostas as definições de métricas para avaliação da performance do modelo.

2.1 Big data

O conceito de *Big Data* refere-se a conjuntos volumosos e complexos de dados que são analisados e processados, a fim de obter informações que podem levar a uma mudança nas formas tradicionais de análise de dados. O crescimento do *Big Data* teve início na era digital, devido ao surgimento e evolução dos computadores, internet e tecnologia capazes de coletar dados.

Para uma aplicação ser considerada de *Big Data*, é preciso que esteja de acordo com algumas características chamadas de V's do Big Data. Os 5 Vs mais conhecidos do *Big Data* são: Volume, Variedade, Velocidade, Veracidade e Valor.

- ✓ Volume: Refere-se à enorme quantidade de dados gerados e coletados todos os dias de várias fontes..
- ✓ Variedade: refere-se à natureza diversa dos dados gerados por diferentes fontes, que incluem dados estruturados, como bancos de dados, dados não estruturados, como texto, imagens e vídeos, e dados semiestruturados, como XML e JSON.
- ✓ Velocidade: refere-se ao processamento em tempo real, ou seja, velocidade na criação dos dados, na transferência, armazenamento e análise.
- ✓ Veracidade : Refere-se à qualidade e confiabilidade dos dados. Com big data, geralmente há muito ruído e inconsistências nos dados, o que torna difícil garantir sua precisão..

- ✓ Valor: refere-se aos potenciais *insights* e valor que podem ser derivados da análise de *big data*. Ao analisar e extrair *insights* significativos de *big data*, as organizações podem tomar melhores decisões, melhorar suas operações e impulsionar a inovação.

De acordo com Saha e Srivastava (2014), estudos mostraram que dados de baixa qualidade são predominantes em grandes bancos de dados e na Web. Como dados de baixa qualidade podem ter sérias consequências nos resultados das análises de dados, a importância da veracidade, o quarto 'V' do *big data*, está sendo cada vez mais reconhecido.

Devido à imensa quantidade e velocidade dos dados, é essencial compreender e, se necessário, corrigir de forma escalável e oportuna os dados incorretos. Com a diversidade de dados provenientes de várias fontes, não é possível especificar regras de qualidade de dados antecipadamente; é necessário permitir que os "dados falem por si mesmos" para descobrir a semântica subjacente aos dados (SAHA; SRIVASTAVA, 2014).

Apesar dos desafios, o uso de *big data* tem crescido em todas as áreas da ciência nos últimos anos. Existem três áreas propicias para o uso de *big data* em saúde: medicina de precisão, registros eletrônicos de saúde e a internet das coisas. O uso de *big data* na área da saúde trará importantes ganhos em termos de dinheiro, tempo e vidas e precisa ser ativamente defendido por cientistas de dados e epidemiologistas (FILHO; PORTO, 2015).

2.2 Ciência de Dados

De acordo com Escovedo e Koshiyama (2020), o conceito de ciência de dados (*data science*) refere-se a coleta de dados de várias fontes para fins de análise, com o objetivo de extrair valor dos dados e apoiar a tomada de decisões, utilizando geralmente grandes quantidades de dados, de forma sistematizada.

De um lado temos o *big data* que é o conjunto de dados, do outro a ciência de cados que é a ciência que os estuda. Quase sempre, além do olhar para os dados passados para entender o comportamento dos mesmos, deseja-se também realizar análises de forma preditiva, usando por exemplo técnicas de aprendizado de máquina.

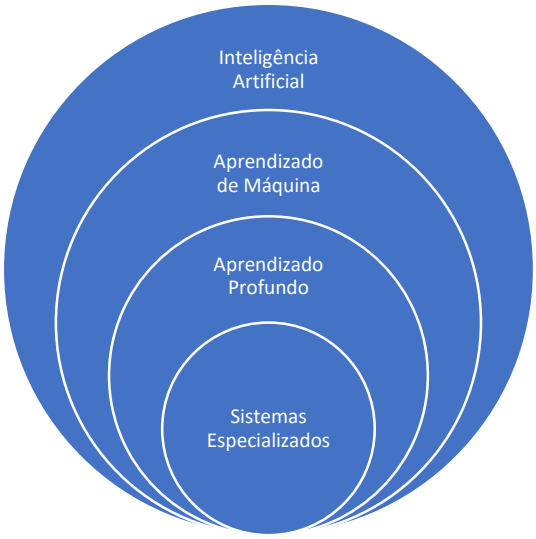
Em resumo, ciência de dados não é uma ferramenta, mas sim um conjunto de métodos com o objetivo de apoiar decisões de negócios baseadas em dados (ESCOVEDO; KOSHIYAMA, 2020).

2.3 Inteligência Artificial

O desenvolvimento da inteligência artificial tem experimentado vários ciclos de avanços desde seu início em 1950. Atualmente, o que impulsiona o avanço da inteligência artificial são: dados, processadores e algoritmos de aprendizado. Desde 2010, a quantidade de dados produzidos no mundo atingiu o nível de zettabyte (ZB).

Apesar de não ser um conceito novo, o conceito de inteligência artificial ainda necessita de uma definição com ampla aceitação. Segundo MISHRA (2022), inteligência artificial é um ramo da ciência da computação que se refere à capacidade das máquinas de realizar tarefas, das mais simples às mais complexas, de forma similar aos seres humanos. Para isso, consultam uma base pré-configurada e repetem padrões. O aprendizado de máquina, por sua vez, tem a ver com a habilidade de aprender, em uma simulação do cérebro humano (MISHRA, 2022).

Figura 1 - Relação entre aprendizado de máquina, aprendizado profundo e inteligência artificial



Fonte: MISHRA, 2022, tradução a autora.

2.4 Aprendizado de máquina

O aprendizado de máquina (AM) é uma área da inteligência artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Ao serem expostas a novos dados, elas se adaptam a partir dos cálculos anteriores e os padrões se moldam para oferecer respostas confiáveis.

Na prática em vez de programar regras em um computador e esperar o resultado, com o aprendizado de máquina, a máquina aprenderá essas regras por conta própria, sem a necessidade de intervenção humana contínua (MONARD; BARANAUSKAS, 2003).

Na área da saúde, a pesquisa de inteligência artificial está se tornando cada vez mais focada na aplicação de técnicas de aprendizado de máquina a problemas complexos (CHALLEN et al., 2019). Com a disponibilidade crescente de dados relevantes para o desenvolvimento de pesquisas em saúde, esses algoritmos têm o potencial de melhorar a predição do desfecho clínico por capturar relações complexas nos dados de pacientes e aprendendo de forma autônoma.

Nesse contexto, estudos relacionados com algoritmos de aprendizado de máquina têm sido utilizados na modelagem preditiva de desfechos de interesse para a saúde, como suporte ao diagnostico, previsões de risco (sepse por exemplo) e análise de tomografias computadorizadas de pacientes com câncer.

Embora existam milhares de artigos aplicando algoritmos de aprendizado de máquina na área da saúde, e estes demonstraram alguns resultados impressionantes, ainda assim seu valor clínico não foi percebido, alguns experimentos falham, demonstrando que os modelos de dados e a qualidade dos dados ainda não são robustos o suficiente para muitos domínios de problemas do mundo real da saúde onde o grau de incerteza é alto, além disso, ainda existe a falta de uma compreensão clara de como quantificar o benefício ou garantir a segurança do paciente e preocupações crescentes sobre os aspectos éticos e impacto médicos legal (CHALLEN et al., 2019).

2.4.1 Tipos de Algoritmos

Os algoritmos são categorizados em três tipos, sendo eles: aprendizagem supervisionada, não supervisionada e por reforço.

Na aprendizagem de máquina supervisionada, um sistema é treinado com dados rotulados, consistindo em pares de entradas e saídas. Os rótulos definem um conjunto de dados em um ou mais grupos, como por exemplo, pacientes "Com Infecção" ou “Sem Infecção”. O sistema aprende com esses dados durante o treinamento para prever os rótulos de novos dados. O objetivo do algoritmo é construir um classificador que possa determinar corretamente a classe de novos exemplos que não tenham o rótulo da classe (MONARD; BARANAUSKAS, 2003).

No aprendizado de máquina não supervisionados, o algoritmo treina a máquina utilizando informações que não são classificadas nem rotuladas, permitindo que o algoritmo atue sobre essas informações sem orientação. A tarefa da máquina passa por agrupar informações não classificadas de acordo com semelhanças, padrões e diferenças, sem qualquer treino prévio de dados, normalmente, é necessária uma análise para determinar o que cada agrupamento significa no contexto do problema que está sendo analisado.

Em algoritmos por reforço, a máquina é estimulada a descobrir através de testes do tipo tentativa e erro quais ações terão para a máquina uma maior recompensa ou também penalidades pelas ações que executa. O agente aprende a atingir uma meta em um ambiente incerto e potencialmente complexo. Com esta abordagem é possível ensinar um sistema a priorizar hábitos em detrimento de outros, tomando a melhor decisão diante de diferentes situações.

2.4.2 Métricas para avaliação de performance

A avaliação da performance de um algoritmo de AM em um determinado conjunto de dados é realizada por meio da mensuração do quão bem as predições decorrentes do modelo ajustado reproduzem o valor observado para resposta de interesse. As métricas de desempenho são úteis para determinar se o modelo atende aos requisitos definidos, ou seja, se ele resolve ou não o

problema proposto. Além disso, elas também permitem detectar problemas como overfitting e underfitting (RASCHKA; MIRJALILI, 2019).

Nesta seção são apresentadas as métricas utilizadas neste trabalho para avaliar os modelos e analisar as predições realizadas pelos mesmos, são elas: matriz de confusão, acurácia, precisão, *recall*, f1-score e AUC ROC.

2.4.2.1 Matriz de confusão

Matriz de confusão é uma representação tabular que permite analisar o desempenho de um modelo de classificação. Essa matriz é construída com base nas classes verdadeiras dos dados e nas classes previstas pelo modelo (RASCHKA; MIRJALILI, 2019).

A matriz indica quantos exemplos existem em cada grupo conforme a seguir:

- Verdadeiros Positivos (VP): classificação correta da classe positivo;
- Falso Negativo (FN): tem condição positiva e o teste é negativo.
- Falso Positivo (FP): tem condição negativa e o teste é positivo.
- Verdadeiros Negativos (VN): classificação correta da classe negativo.

Tabela 1 - Matriz de confusão

		Detectada	
		Sim	**Não**
Real	**Sim**	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Autora da dissertação

Os valores são organizados em uma matriz, onde as classes reais são representadas nas linhas e as classes previstas pelo modelo estão nas colunas.

A matriz de confusão permite visualizar o desempenho do modelo em termos de acertos e erros para cada classe. Com base nesses valores, várias métricas de desempenho podem ser calculadas, como precisão, *recall*, acurácia e F1-score, proporcionando uma análise mais completa do desempenho do modelo de classificação.

2.4.2.2 Acurácia

A acurácia é considerada uma das métricas mais simples e importantes, representada pela equação ((VP + VN) / (VP + VN + FP + FN)), é a relação de acertos positivos e negativos dividido pela quantidade total de predições, indica uma performance geral do modelo, assim a acurácia demonstra dentre todas as classificações, quantas o modelo classificou corretamente.

Para Raschka e Mirjalili (2019), se considerarmos um cenário em que existem 10 amostras para classificar pacientes como desenvolvedores de infecção ou não, e o modelo acerta 9 das 10 amostras, a acurácia será de 90%. No entanto, a acurácia pode se tornar insuficiente em cenários em que há uma probabilidade desigual entre as classes. Por exemplo, se apenas um em cada dez pacientes desenvolve uma infecção, o modelo pode simplesmente chutar que ninguém irá desenvolver uma infecção e, naturalmente, acertar 90% das vezes. Portanto, outras métricas, como *recall*, se tornam essenciais.

2.4.2.3 Precisão e *recall*

Dada pela equação (VP / (VP + FP)), a precisão é uma métrica que avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos, dentre todas as classificações de classe Positivo que o modelo fez, quantas estão corretas. A precisão é indicada para casos em que um falso positivo é considerado um problema mais crítico, como por exemplo, sistemas de detecção automática de spam, uma mensagem importante ser considerada spam pode causar prejuízos ao usuário do sistema.

Por sua vez, o *recall* (VP / (VP + FN)) é a proporção de verdadeiros positivos entre todos os indivíduos cuja resposta de interesse foi, de fato, observada. *Recall*, também conhecida como sensibilidade, pode ser usada em uma situação em que os falsos negativos são considerados mais prejudiciais que os falsos positivos, na área da saúde o *recall* é de grande relevância, já que classificar paciente doente como saudável pode trazer grandes riscos ao paciente.

Isoladamente, a precisão e o recall raramente fornecem informações significativas, ambas são perspectivas limitadas do desempenho de um classificador. Tanto a precisão quanto o recall podem falhar em diferenciar classificadores com bom desempenho de certos tipos de classificadores com desempenho insatisfatório (HACKELING, 2017).

Hackeling (2017) também destaca que um classificador trivial poderia facilmente obter uma pontuação de recall perfeita, prevendo positivo para cada instância. Por exemplo, considerando um conjunto de testes com 10 exemplos positivos e 10 exemplos negativos, um classificador que prevê positivo para todos os exemplos alcançaria um *recall* de 1 (um). Da mesma forma, um classificador que prevê negativo para todos os exemplos, ou um que faz apenas previsões falsamente positivas e verdadeiramente negativas, teria uma pontuação de *recall* de 0 (zero). Além disso, um classificador que prevê corretamente apenas uma única instância como positiva alcançaria uma precisão perfeita. É importante notar que essas métricas por si só não fornecem uma visão abrangente do desempenho de um classificador e devem ser consideradas em conjunto com outras métricas para uma avaliação mais completa.

2.4.2.4 F1-Score

De acordo com Hackeling (2017), a medida F1 é calculada como a média harmônica das pontuações de precisão e *recall*. Essa métrica penaliza classificadores com precisão desbalanceada e baixas pontuações de *recall*, como o classificador trivial que sempre prevê a classe positiva. Um modelo com precisão perfeita e *recall* perfeito alcançará uma pontuação F1 de 1. A medida F1 é especialmente útil quando se deseja equilibrar a importância tanto da precisão quanto do recall em um modelo, proporcionando uma visão mais completa do desempenho geral do classificador.

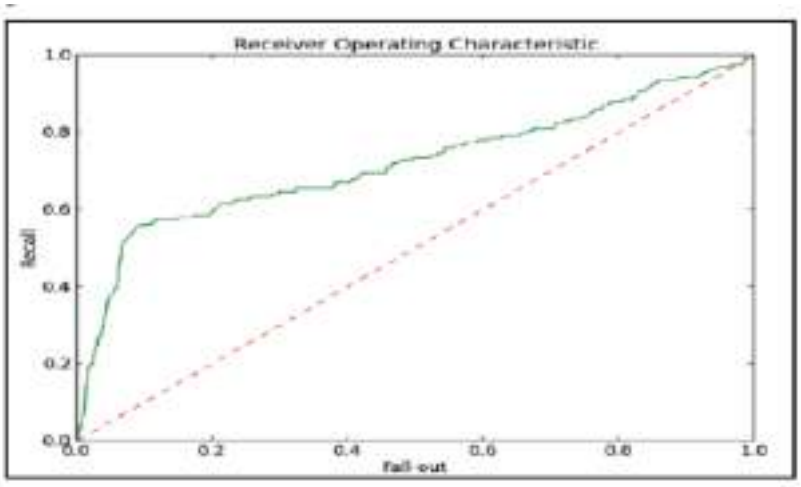
O F1-Score pode ser obtido com a equação (2 * ((Precisão * Recall) / (Precisão + Recall))).

2.4.2.5 Curva AUC e ROC

Conforme mencionado por Hackeling (2017), a curva ROC (Receiver Operating Characteristic) é uma ferramenta que permite visualizar o desempenho de um classificador. Ao contrário da métrica de precisão, a curva ROC é mais robusta diante do desbalanceamento das classes em conjuntos de dados. Além disso, ao contrário da precisão e do *recall*, a curva ROC ilustra o desempenho do classificador para diferentes valores de limite de discriminação. As curvas ROC traçam o *recall* do classificador em relação à sua taxa de falsos positivos, também conhecida como Fall-Out. O Fall-Out é calculado como o número de falsos positivos dividido pelo número total de negativos. Essa métrica fornece informações sobre o desempenho do classificador em relação a classificações incorretas da classe negativa.

Ainda de acordo com Hackeling (2017), a AUC (Área Sob a Curva) é um valor único que resume o desempenho esperado do classificador a partir da curva ROC. A AUC é calculada como a área sob a curva ROC. Na figura a seguir, podemos observar uma linha tracejada que representa um classificador que faz previsões aleatórias, cuja AUC é igual a 0,5. Por outro lado, a curva contínua representa um classificador que supera a suposição aleatória, como exemplificado na Figura 2. A AUC é uma medida útil para comparar o desempenho de diferentes classificadores, pois quanto maior a AUC, melhor o desempenho do classificador na distinção entre as classes.

Figura 2 - Curva ROC



Fonte: HACKELING (2017, p.100)

2.5 Linguagem de programação Python

Python é uma linguagem de programação de alto nível, interpretada multiparadigma, sendo uma linguagem poderosa para processamento de dados, com uma curva de aprendizagem curta, além de uma grande comunidade de usuários, oferecendo grandes quantidades de bibliotecas e recursos. Python é utilizada em diversas áreas do desenvolvimento como: desenvolvimento web, desktop e ciência de dados.

Apesar da simplicidade, o Python é utilizado em diversos projetos de software. Existem vários frameworks e bibliotecas para se trabalhar com interface gráfica, cálculos matemáticos complexos, banco de dados, desenvolvimento web e outras tecnologias. Por ser uma linguagem puramente interpretada, Python pode ser executada em qualquer ambiente que possua o interpretador instalado.

Além da simplicidade e da grande quantidade de fóruns de discussão, outra vantagem em relação ao Python é que ele pode ser usado através do Google Collaboratory que é um serviço de armazenamento em nuvem voltados à criação e execução de códigos em Python, diretamente em um navegador, sem a necessidade de nenhum tipo de instalação de software em uma máquina.

2.6 Bibliotecas de Programação

Em ciência da computação, uma biblioteca é uma coleção de subprogramas usados no desenvolvimento de software. Incluem dados de configuração, documentação, dados de ajuda, modelos de mensagens, código pré-escrito e sub-rotinas. A biblioteca é também um resumo de implementações de comportamento, quando é utilizada uma linguagem de programação e aplica uma biblioteca, é realizado o acesso a um conjunto de funções que já foram escritas por outros desenvolvedores.

2.6.1 Pandas

O Pandas é uma biblioteca de código aberto para análise e manipulação de dados, construída sobre a linguagem de programação Python, possibilita a

capacidade de trabalhar com dados do tipo planilha, permitindo carregar, manipular, alinhar e combinar dados rapidamente (CHEN, 2018).

O Pandas permite importar dados de vários formatos de arquivo, como valores separados por vírgula, JSON, SQL, Microsoft Excel. O Pandas permite várias operações de manipulação de dados, como fusão, remodelagem, seleção, bem como limpeza de dados e recursos de transformação de dados.

O Pandas disponibiliza estruturas de dados e ferramentas para tratar algoritmos que foram idealizados para facilitar o processo de tratamento e análise de informações utilizando a linguagem Python.

2.6.2 Numpy

NumPy, é uma biblioteca Python que é usada principalmente para realizar cálculos em matrizes multidimensionais. O *NumPy* fornece um grande conjunto de funções e operações de biblioteca que ajudam os programadores a executar facilmente diversos cálculos numéricos (NUMPY, 2019).

O pacote fornece um objeto de matriz multidimensional de alto desempenho e ferramentas para trabalhar com essas matrizes. É o pacote fundamental para computação científica com Python. Além de seus usos científicos óbvios, o *NumPy* também pode ser usado como um contêiner multidimensional eficiente de dados genéricos.

2.6.3 Scikit-learn

Scikit-learn é uma biblioteca de aprendizado de máquina de código aberto da linguagem Python que oferece suporte ao aprendizado supervisionado e não supervisionado. Ele também fornece várias ferramentas para ajuste de modelos, pré-processamento de dados, seleção de modelos, avaliação de modelos e muitas outras utilidades.

Os principais recursos do *scikit-learn* são:

• Análise de algoritmos de aprendizado supervisionado: Possui modelos lineares generalizados, como por exemplo, regressão linear, máquinas de vetores de suporte, árvores de decisão e métodos bayesianos.

• Validação cruzada: é uma técnica para avaliar modelos de aprendizado de máquina por meio de treinamento de vários modelos em subconjuntos de dados de entrada disponíveis e avaliação deles no subconjunto complementar dos dados. Esta técnica é usada para detectar sobreajuste, ou seja, a não generalização de um padrão.

• Algoritmos de aprendizado não supervisionado: há uma grande variedade de algoritmos de aprendizado de máquina, começando com agrupamento, análise de fator, análise de componente principal e redes neurais não supervisionadas.

2.6.4 MatplotLib

O *Matplotlib* é uma biblioteca de visualização de dados e biblioteca de plotagem 2-D do Python. É a biblioteca de plotagem mais popular e amplamente usada na comunidade Python. Ele vem com um ambiente interativo em várias plataformas.

Matplotlib pode ser usado em scripts Python*, shells* Python e Ipython, notebook *jupyter*, servidores de aplicativos web. O *matplotlib* é usado para criar plotagens, gráficos de barras, gráficos de setores circulares, histogramas, gráficos de dispersão, gráficos de erro, espectros de potência e gráficos de tronco.

Várias bibliotecas de terceiros podem ser integradas aos aplicativos *matplotlib*. Como *seaborn, ggplot* e outros kits de ferramentas de projeção e mapeamento.

2.6.5 Seaborn

Seaborn é uma biblioteca de visualização de dados Python baseada em *matplotlib* e intimamente integrada com as estruturas de dados *numpy* e pandas. *Seaborn* tem várias funções de plotagem orientadas a conjuntos de dados que operam em quadros de dados e matrizes que contêm conjuntos de dados inteiros.

O *seaborn* executa internamente as funções de agregação estatística e mapeamento necessárias para criar gráficos informativos que o usuário

deseja. É uma interface de alto nível para a criação de gráficos estatísticos bonitos e informativos, essenciais para a exploração e compreensão dos dados.

Os gráficos de dados *seaborn* podem incluir gráficos de barras, gráficos de pizza, histogramas, gráficos de dispersão e gráficos de erro.

As bibliotecas *matplotLib* e s*eaborn* foram utilizadas principalmente na etapa da análise exploratória deste projeto.

3 METODOLOGIA

Este capítulo caracteriza a pesquisa e os procedimentos empregados para sua realização. Em resumo, descreve como a pesquisa foi executada.

3.1 Caracterização da pesquisa

De acordo com as classificações de pesquisa, trata-se de uma pesquisa aplicada, descritiva e quantitativa. Do ponto de vista de sua natureza a pesquisa é aplicada, pois visa gerar conhecimento para aplicação prática com foco em um problema específico. Em relação a abordagem, a pesquisa é quantitativa, pois pretende-se medir o índice de acerto do método gerado.

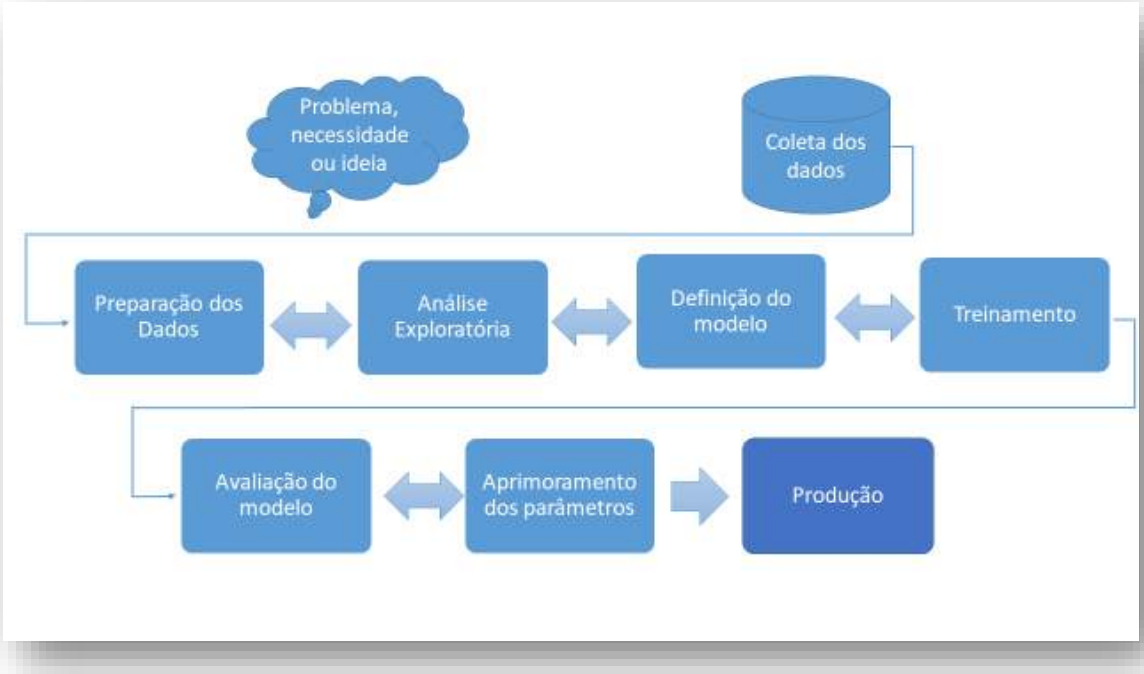
O estudo foi realizado utilizando os dados históricos de março de 2015 a dezembro de 2019, respeitando-se as seguintes etapas metodológicas: definição do problema, objetivo do estudo, critérios de inclusão/exclusão e tratamento e análise dos dados.

3.2 Procedimento metodológico

Para a execução do trabalho proposto, optou-se pela a abordagem metodológica utilizando um compilado de melhores práticas relacionados com projetos de *data science*, o trabalho foi estruturado em etapas de forma linear, cada etapa é revisitada de acordo com a necessidade do projeto, assim o projeto é realizado de maneira mais interativa, a figura 3 ilustra as fases que podem ser resumidas em 9 etapas:

1. Problema, necessidade ou ideia;
2. Coleta dos dados;
3. Preparação dos dados;
4. Análise exploratória;
5. Definição do modelo;
6. Treinamento;
7. Avaliação do modelo;
8. Aprimoramento dos hiperparâmetros;
9. Produção.

Figura 3 - Etapas projeto data Science



Fonte: ESCOVEDO; KOSHIYAMA, 2020

3.2.1 Problema, necessidade ou ideia

A primeira etapa do projeto é o mapeamento da necessidade ou ideia, nesta etapa deve-se ter em mente o problema que se deseja resolver, e, em seguida, os objetivos devem ser definidos, assim como relacionar as perguntas que desejamos responder (ESCOVEDO; KOSHIYAMA, 2020).

No contexto hospitalar pode-se observar os desafios, dificuldades e a importância do tema da infecção hospitalar. É uma questão que requer atenção constante, pois afeta diretamente a segurança e o bem-estar dos pacientes.

As infecções hospitalares representam uma séria ameaça à segurança dos pacientes, aumentando a morbidade, o tempo de internação, os custos de tratamento e a taxa de mortalidade. Além disso, essas infecções também podem contribuir para o desenvolvimento de resistência antimicrobiana, o que torna o tratamento mais desafiador.

Ao observar a relevância do tema da infecção hospitalar e o processo manual de leitura do prontuário para identificar e confirmar casos de infecção, torna-se relevante a necessidade de desenvolver modelos de classificação que possam auxiliar na previsão de infecção hospitalar. Esses modelos podem aproveitar técnicas de AM para analisar uma variedade de variáveis e padrões nos dados médicos, a fim de identificar precocemente os pacientes em risco de desenvolver infecções durante a internação hospitalar. Isso possibilitaria intervenções preventivas mais rápidas e eficazes, contribuindo para a redução das taxas de infecção, a melhoria da segurança do paciente e a otimização dos recursos de saúde.

3.2.2 Coleta dos dados

A segunda etapa consiste em extrair e coletar os dados para resolver os problemas levantados na etapa anterior, nesta etapa é importante entender quais os tipos de dados irão pautar o projeto, dados estruturados versus dados não estruturados, dados de banco de dados versus planilhas.

Nesta etapa foram coletados dados de março de 2015 a dezembro de 2019, resultando em um total de 331.905 registros, destes, 94.505 tinham a identificação de infecção hospitalar confirmada.

Para a coleta foi utilizado o banco de dados oracle da instituição, os dados das tabelas foram exportados, cada tabela gerou um arquivo csv. Antes de importar os arquivos, eles foram salvos na codificação UTF-8.

Os arquivos foram importados utilizando o *google colaboratory*, que é um serviço de armazenamento em nuvem de notebooks voltados à criação e execução de códigos em Python e também em R, diretamente em um navegador, sem a necessidade de nenhum tipo de instalação de software em uma máquina. Foi utilizada a biblioteca pandas para transformar os arquivos csv em um *dataframe*.

Os dados coletados não são de domínio público, com isto, foi necessário utilizar os mesmos de acordo com a lei geral de proteção de dados (LGPD).

3.2.3 Preparação dos Dados

A preparação de dados é a parte mais importante de um projeto de aprendizado de máquina. A preparação de dados é o ato de transformar dados brutos em um formato apropriado e com a qualidade desejável para modelagem (BROWNLEE, 2020).

A partir dos dados coletados é necessário tratá-los antes de iniciar as análises, é nesta etapa que são realizadas as atividades de preparação dos dados. Essa etapa é a mais demorada e trabalhosa do projeto, segundo Escovedo e Koshiyama (2020) estima-se que consuma pelo menos 70% do tempo total do projeto, pode ser necessário nesta etapa remover ou complementar dados faltantes ou dados duplicados, corrigir ou amenizar dados discrepantes (*outliers*) e desbalanceamento entre classes, selecionar as variáveis e instâncias mais adequadas para compor os modelos que serão construídos nas etapas seguintes.

Com a base de dados importada, uma das técnicas utilizadas foi a seleção de recursos, esta técnica tem com o objetivo identificar e remover variáveis redundantes ou altamente correlacionadas do modelo.

Quando as variáveis são altamente correlacionadas, elas podem fornecer informações redundantes. As variáveis redundantes não contribuem significativamente para o poder preditivo do modelo, mas podem aumentar a complexidade computacional e o risco de *overfitting*. Remover ou reduzir variáveis redundantes pode ajudar a simplificar o modelo sem perder muita precisão preditiva.

Outra técnica aplicada nesta etapa foi a análise dos dados ausentes, este é um desafio comum no aprendizado de máquina e na análise de dados. Quando certas observações ou atributos estão ausentes, isso pode afetar o desempenho e a precisão dos modelos. Neste trabalho foram aplicadas técnicas de exclusão e imputação, na imputação utilizou-se a substituição dos valores ausentes pela média dos valores disponíveis para o atributo. Dados não estruturados e com informações de baixa relevância sem qualquer padrão também foram excluídos para não prejudicar a performance do algoritmo.

Além das técnicas apresentadas, foram necessárias também analises em relação aos dados longitudinais e *outliers*.

Embora estruturas de dados longitudinais ofereçam flexibilidade e compatibilidade em determinados cenários, elas também apresentam algumas desvantagens potenciais como, complexidade e tamanho da base de dados.

Em relação a complexidade, em aprendizado de máquina, estruturas de dados longitudinais podem exigir etapas adicionais para criar novas variáveis ou recursos agregados. Criar novas variáveis com base nas existentes ou incorporar informações de múltiplas variáveis pode envolver transformações e operações de agregação mais complexas.

No que tange ao tamanho da base de dados, estruturas de dados longitudinais podem resultar em um conjunto de dados maior em comparação com estruturas de dados amplas. Como cada observação é representada por várias linhas em um formato longo, o tamanho geral do conjunto de dados pode ser maior, especialmente se houver muitas variáveis ou medições repetidas. Isso pode afetar o uso da memória e o tempo de processamento, principalmente ao lidar com grandes conjuntos de dados.

Diante do exposto, com o objetivo de reduzir a complexidade e diminuir o uso de memória e tempo de processamento, foi realizada a transformação dos dados longitudinais em dados transversais utilizando a função *crosstab* da biblioteca pandas.

No decorrer da preparação dos dados foram identificados possíveis *outliers,* que são pontos de dados que se desviam significativamente do restante das observações em um conjunto de dados. Eles podem ocorrer devido a vários motivos, como erros de digitação ou erros de entrada de dados. Lidar com *outliers* é importante no aprendizado de máquina porque eles podem ter um impacto significativo no desempenho e na precisão dos modelos.

Para identificar os *outliers* foi utilizada a biblioteca de visualização de dados *seaborn* que provê uma interface de alto nível para construção de gráficos estatísticos atrativos e informativos, foi utilizada a inspeção visual plotando os dados usando gráficos de dispersão, caixa e histogramas para ajudar a identificar possíveis *outliers* visualmente, desta forma é possível ver pontos de dados que estão longe da maioria das observações.

As aplicações das técnicas citadas acima estão detalhadas no capitulo Resultados e Discussão.

3.2.4 Análise Exploratória

A quarta etapa consiste em estudar e validar hipóteses, fazendo análises descritivas dos dados procurando *insights*, aplicando estudos estatísticos afim de entender os dados coletados, de forma a obter uma compreensão abrangente dos dados, identificar problemas de qualidade de dados, detectar padrões, *outliers* e possíveis relacionamentos entre variáveis.

Considerado um dos passos cruciais para as análises em bases de dados, este processo foi desenvolvido por John W. Tukey. Tukey que fez sua contribuição unindo pensamento estatístico aos processos de transformação e exploração de dados. “A análise exploratória de dados nunca pode ser toda a história, mas nada mais pode servir como a pedra fundamental” (TUKEY, 1977).

John Tukey introduziu vários conceitos fundamentais na análise exploratória de dados (AED) que tiveram um impacto duradouro no campo. Na distribuição de dados Tukey introduziu técnicas como gráficos de caixa para visualizar a distribuição de dados, identificar valores discrepantes e avaliar medidas de tendência central e dispersão. Também defendeu o uso de técnicas gráficas para explorar e analisar dados. Ele enfatizou o poder das visualizações em revelar padrões, relacionamentos e anomalias nos dados. O trabalho de Tukey influenciou o desenvolvimento de várias técnicas de visualização, incluindo gráficos de dispersão, diagrama de ramo e folha e histogramas.

Tukey reconheceu a utilidade de transformar dados para revelar padrões ou relacionamentos ocultos. Ele introduziu técnicas como transformações logarítmicas e transformações de potência para normalizar dados distorcidos ou estabilizar a variância, tornando mais fácil interpretar e modelar os dados.

Por fim, as técnicas gráficas desempenham um importante papel na análise exploratória, sendo uma das formas mais eficientes de apresentação de dados, através dos gráficos é possível analisar as relações entre as variáveis e uma análise descritiva que quantifique o grau de inter-relação entre elas.

Enquanto as tabelas fornecem uma ideia mais precisa e possibilitam um tratamento mais rigoroso aos dados, os gráficos são mais indicados em situações cujo objetivo é dar uma visão mais rápida e fácil das variáveis às quais se referem os dados.

A AED foi realizada utilizando ferramentas de visualização, como bibliotecas pandas, *matplotlib* e *seaborn*, essas bibliotecas oferecem ampla variedade de funções e métodos para suporte à manipulação de dados, visualização e análise estatística durante o processo de AED.

3.2.5 Definição do modelo

A quinta etapa consiste em elencar os modelos possíveis e passíveis para o problema que desejamos resolver, a escolha do melhor modelo depende de qual é o objetivo do problema que estamos tratando, além disto é necessário a análise de alguns fatores, como: complexidade, recursos computacionais, tempo e quantidade de dados. Abaixo segue uma breve descrição dos algoritmos utilizados neste trabalho.

3.2.5.1 Regressão Logística

A regressão logística (RL) é uma técnica estatística que tem como objetivo produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica, frequentemente binária, em função de uma ou mais variáveis independentes contínuas e/ou categóricas.

Segundo (MENARD, 2002), a regressão logística utiliza a função logit para modelar a probabilidade de um evento ocorrer ou não, dada uma combinação linear das variáveis independentes. A função logit transforma a probabilidade em uma escala logarítmica, permitindo a modelagem em termos de logitos.

A regressão logística é aplicável a uma ampla variedade de campos, incluindo ciências sociais, saúde e economia. Além disso, a regressão logística é particularmente útil quando a variável de resposta é dicotômica e quando se deseja entender o efeito das variáveis independentes na probabilidade de ocorrência do evento de interesse.

Uma das vantagens da regressão logística é que ela é um algoritmo relativamente simples e interpretável. O modelo produz coeficientes para cada variável de entrada, permitindo fácil interpretação de seu impacto na variável

alvo. Isso o torna útil quando há necessidade de entender a relação entre os preditores e o resultado.

No estudo “Logistic regression technique is comparable to complex machine learning algorithms in predicting cognitive impairment related to post intensive care syndrome”, (WU et al., 2023) avalia o desempenho de modelos de aprendizado de máquina e compara com a técnica de regressão logística na predição de comprometimento cognitivo relacionado à síndrome pós-terapia intensiva (PICS-CI). Sete diferentes modelos de AM foram considerados, árvore de decisão, random forest, XGBoost, rede neural, naive bayes e máquina de vetores de suporte (SVM), e comparados com regressão logística. A capacidade discriminativa foi avaliada pela curva AUROC, gráficos de cinto de calibração e o teste de Hosmer-Lemeshow foi usado para avaliar a calibração. Todos os modelos de AM mostraram bom desempenho (intervalo AUC: 0,822–0,906). O modelo naive bayes teve a AUC mais alta (0,906 [95% CI 0,857–0,955]), que foi ligeiramente superior, mas não significativamente diferente da de regressão logística (0,898 [95% CI 0,847–0,949]). Em dados de baixa dimensão, a regressão logística produziu um desempenho tão bom quanto outros modelos complexos de AM para prever comprometimento cognitivo após internação na UTI.

3.2.5.2 Random Forest

Random forest, também conhecido como floresta aleatória, é um método de aprendizado de máquina que pode ser utilizado para tarefas de classificação, regressão e outras tarefas que operam construindo uma multiplicidade de árvores de decisão (HO, 1995)

O algoritmo *random forest* cria uma floresta de um modo aleatório, a “floresta” que ele cria é uma combinação (*ensemble*) de árvores de decisão. Com a criação e a combinação de várias arvores de decisão obtém-se uma predição com mais exatidão e maior estabilidade (BREIMAN, 2001).

Na tarefa de classificação, o r*andom forest* realiza a predição por meio de um processo de votação, levando em consideração as classificações individuais de cada árvore. Dessa forma, a classe mais frequente entre as árvores é escolhida como a predição final.

No caso de problemas de regressão, o *random forest* realiza a predição calculando a média das previsões geradas por cada árvore do conjunto. Essa média é então utilizada como o valor de regressão final.

Essa abordagem de votação e média nas previsões das árvores individuais permite que *random forest* seja capaz de fornecer resultados confiáveis tanto em tarefas de classificação quanto em problemas de regressão.

De acordo com (BREIMAN, 2001), ao lidar com problemas de diagnóstico médico e recuperação de documentos, nos quais há um grande número de variáveis de entrada, muitas vezes centenas ou milhares, cada uma contendo apenas uma pequena quantidade de informação, um classificador de árvore única terá apenas uma precisão ligeiramente melhor do que uma escolha aleatória de classe. No entanto, ao combinar árvores cultivadas utilizando recursos aleatórios, é possível obter uma precisão aprimorada.

Em outras palavras, quando se trata de problemas complexos com um grande número de variáveis, uma única árvore de decisão não é suficiente para alcançar resultados precisos. No entanto, ao usar o algoritmo *random forest*, que combina múltiplas árvores de decisão e seleciona aleatoriamente um subconjunto de recursos para cada árvore, é possível aumentar significativamente a precisão do modelo.

Ao cultivar e combinar essas árvores, o algoritmo *random forest* consegue extrair informações úteis de diferentes características e criar um modelo robusto e confiável. Isso torna *random forest* uma abordagem eficiente para resolver problemas complexos com muitas variáveis de entrada, fornecendo resultados mais precisos do que uma única árvore de decisão.

Em suma, o algoritmo *random forest* é uma ferramenta poderosa, eficiente e flexível, embora apresente algumas limitações. Uma de suas principais vantagens é a capacidade de lidar com diferentes tipos de dados de entrada, incluindo variáveis binárias, categóricas e numéricas. Isso permite que o algoritmo seja aplicado em uma ampla variedade de problemas, adaptando-se às características específicas de cada conjunto de dados. Essa flexibilidade é uma das razões pelas quais *random forest* é amplamente utilizado em diversas áreas, proporcionando resultados confiáveis e precisos.

No estudo “Death risk and the importance of clinical features in elderly people with COVID-19 using the Random Forest Algorithm” (LIMA et al., 2021)

treina um classificador do tipo *random forest* para estimar o risco de óbito em idosos (com mais de 60 anos) diagnosticados com COVID-19. Ele utiliza uma “feature” deste classificador, chamada feature_importance para identificar os atributos (principais fatores de risco) relacionados com o desfecho final (cura ou óbito) através do ganho de informação. O algoritmo K-fold Cross Validation, com K=10, foi usado para avaliar tanto o desempenho do modelo quanto a importância das características clínicas. Segundo este estudo algoritmo *random forest* classificou corretamente 78,33% dos idosos, com AUC de 0,839. A idade avançada é o fator que representa maior risco de evolução para óbito. Além disso, a principal comorbidade e sintoma também identificados, foram, respectivamente, doença cardiovascular e saturação de oxigênio ≤95%.

3.2.5.3 AdaBoost

O algoritmo *AdaBoost (Adaptive Boosting)* é um algoritmo de aprendizado de máquina supervisionado usado para melhorar a precisão de modelos de classificação fracos, com o *adaBoost*, um classificador fraco é treinado utilizando um conjunto de dados de treinamento ponderado. Um classificador fraco é um modelo de aprendizado de máquina que tem um desempenho um pouco melhor do que uma escolha aleatória, como uma árvore de decisão simples. Ele pode fazer previsões, mas sua precisão é limitada.

O algoritmo *adaBoost* é adaptável no sentido de que as classificações subsequentes são ajustadas para favorecer as instâncias classificadas incorretamente por classificações anteriores. Isso significa que o algoritmo dá maior importância e foco nas instâncias mal classificadas, permitindo que o próximo classificador fraco concentre-se nelas e tente corrigir os erros cometidos anteriormente. Dessa forma, o *adaBoost* aprende iterativamente com os erros anteriores e busca melhorar a precisão geral do modelo combinado.

A melhoria na previsão ocorre através da redução do viés e da variância. O algoritmo *adaBoost* atinge isso atribuindo pesos diferentes aos exemplos classificados corretamente e incorretamente. Além disso, ele constrói uma combinação linear de classificadores fracos, visando minimizar a perda exponencial associada a todas essas combinações (SCHAPIRE; FREUND, 2013).

Dessa forma, o *adaBoost* é capaz de adaptar-se aos erros cometidos pelos classificadores fracos anteriores e focar nas instâncias mais desafiadoras, melhorando gradualmente a precisão do modelo final.

No estudo “Machine Learning Applied to Kidney Disease Prediction: Comparison Study” Rabby utiliza dez técnicas de aprendizado de máquina mais populares para prever doenças renais, entre as técnicas comuns do estudo estão o Random Forest, Regressão Logística, AdaBoost e o KNN. Rabby utilizou um conjunto de dados de forma online, os dados possuíam 400 registros e 25 variáveis, os dados foram divididos em 80% para treinamento e 20% para teste, para os dados de treinamento e teste, a precisão dos modelos de predição foram muito altas e os modelos que tiveram a maior precisão foram Random Forest, Decision Tree e Gaussian Naive Bayes, este fato ocorreu também para a *recall* e especificidade (RABBY et al., 2019). Os resultados apresentados foram: Regressão logística 97,50% de precisão e 96% de *recall*; Adaboost 98,75% de precisão e 97% de *recall*; Random Forest 100% de precisão e *recall* de 100%; KNN 71,25% de precisão e 56% de pontuação de *recall*.

3.2.5.4 Bagging

O algoritmo de *Bagging (Bootstrap Aggregating)* é uma técnica utilizada em aprendizado de máquina para melhorar a precisão e estabilidade de modelos preditivos. Ele se baseia na combinação de vários modelos de aprendizado (chamados de classificadores ou estimadores base) para gerar uma predição final.

O algoritmo *bagging* é um método para gerar várias versões de um preditor e usá-los para obter um preditor agregado. As médias de agregação sobre as versões ao prever um resultado numérico e faz um voto de pluralidade ao prever uma classe. As múltiplas versões são formadas fazendo réplicas *bootstrap* do conjunto de aprendizagem e usá-los como novos conjuntos de aprendizagem. Testes em conjuntos de dados reais e simulados usando classificação e as árvores de regressão e a seleção de subconjuntos na regressão linear mostram que o *bagging* pode proporcionar ganhos substanciais em precisão (BREIMAN, 1996).

Ao criar modelos base (classificadores ou estimadores) em diferentes amostras *bootstrap* e combinar suas predições, *bagging* reduz *recall* do modelo a pequenas variações nos dados de treinamento. Isso resulta em um modelo final mais estável e com menor probabilidade de *overfitting* em comparação com um único modelo.

Ao combinar as predições de vários modelos base, o *bagging* é capaz de capturar diferentes aspectos e padrões dos dados, desta forma ele consegue melhorar a capacidade de generalização e o desempenho do modelo.

De acordo com Breiman as principais vantagens do uso de *bagging* são: redução da variância do modelo, melhoria na estabilidade e robustez das predições, melhoria na capacidade de generalização e no desempenho do modelo final, aplicabilidade em diversos algoritmos de aprendizado de máquina e por fim a facilidade de implementação.

Bagging foi aplicado com sucesso em vários domínios, incluindo classificação, regressão e detecção de anomalias. Sua capacidade de reduzir a variância, lidar com conjuntos de dados complexos e melhorar a precisão do modelo o torna uma escolha popular no aprendizado conjunto para obter um melhor desempenho preditivo.

No estudo “A Comparative Study of Medical Data Classification Methods Based on Decision Tree and Bagging Algorithms” Chau e Shin fazem a comparação dos algoritmos de arvore de decisão e *bagging* para identificar doença cardíaca de pacientes. Foram realizados 3 experimentos, o experimento 1 utilizou o algoritmo C4.5 de árvore de decisão, o experimento 2 usou o *bagging* com árvore de decisão C4.5 e o experimento 3 usou o *bagging* com *naive bayes* (MY CHAU TU; DONGIL SHIN; DONGKYOO SHIN, 2009). Neste estudo o modelo *bagging* com *naive bayes* apresentou o melhor resultado com precisão de 82,50% e *recall* 72,02%, em seguida *bagging* com árvore de decisão com precisão de 79,53% e *recall* 73,72% e por último árvore de decisão C4.5 com precisão de 78,93% e *recall* 72,02%.

3.2.5.5 K-Nearest Neighbors

O KNN (*K-nearest neighbors,* ou “K-vizinhos mais próximos”) é um algoritmo de aprendizado de máquina supervisionado que é usado

principalmente para classificação e regressão. O KNN foi desenvolvido a partir da necessidade de realizar análises discriminantes quando estimativas paramétricas confiáveis de densidades de probabilidade são desconhecidas ou difíceis de determinar.

Os métodos do vizinho mais próximo são baseados nos rótulos dos K exemplos mais próximos no espaço de dados. Como métodos locais, as técnicas de vizinho mais próximo são conhecidas por serem fortes no caso de grandes conjuntos de dados e baixas dimensões.

Variantes para classificação multi-rótulo, regressão e configurações de aprendizado semi-supervisionado permitem a aplicação a um amplo espectro de problemas de aprendizado de máquina. A teoria da decisão fornece informações valiosas sobre as características dos resultados de aprendizado do KNN (KRAMER, 2013).

Essa abordagem é baseada na suposição de que amostras semelhantes tendem a ter rótulos semelhantes. O valor de k representa o número de vizinhos mais próximos que serão considerados para a classificação.

O KNN tem como principais vantagens a simplicidade, facilidade de implementação e capacidade de lidar com limites de decisão. Sua simplicidade e flexibilidade o tornam uma ferramenta valiosa em muitos problemas de classificação e regressão em aprendizado de máquina.

O KNN é particularmente útil quando o limite de decisão é altamente irregular, quando há uma grande quantidade de dados de treinamento disponíveis ou quando a distribuição de dados subjacentes é desconhecida. No entanto, é importante considerar o impacto da escolha do valor de K e da métrica de distância, pois estes podem afetar significativamente o desempenho do algoritmo.

No estudo “Exploratory study on classification of lung cancer subtypes through a combined K-nearest neighbor classifier in breathomics”, (WANG et al., 2020) propõe um método combinado, integrando o classificador KNN a técnica de sobreamostragem SMOTE e métodos de redução de recursos, para investigar a capacidade da respiração exalada de distinguir adenocarcinoma (CA) de pacientes com carcinoma de células escamosas (CEC) no câncer de pulmão. O desempenho de classificação do método proposto foi comparado com os resultados de quatro algoritmos de classificação sob diferentes combinações

de SMOTE e métodos de redução de características. O resultado indicou que o classificador KNN combinando SMOTE e métodos de redução de características foi o método mais promissor para discriminar CA de pacientes com CEC e obteve a maior área média sob a curva característica de operação do receptor (0,63) e média geométrica média (58,50) quando comparado a outros classificadores.

3.2.5.6 XGBoost

O XGBoost é a abreviatura de *eXtreme Gradient Boosting Package*. É uma biblioteca de otimização de gradiente projetada para ser altamente eficiente, flexível e portátil (CHEN; HE, 2019).

O XGBoost é uma implementação de código aberto, é um algoritmo baseado em árvores com aumento de gradiente. O aumento de gradiente é um algoritmo de aprendizagem supervisionada que tenta prever com precisão uma variável de destino. Para isso, combina as estimativas de um conjunto de modelos mais simples e mais fracos (CHEN; GUESTRIN, 2016).

O seu desempenho é atribuído por causa do tipo de aprimoramento de gradiente, proporcionando duas grandes melhorias: acelerar a construção da árvore e propor um novo algoritmo distribuído para pesquisa de árvore.

De maneira robusta, o XGBoost trabalha com uma variedade de tipos de dados, relacionamentos e distribuições, e o grande número de hiperparâmetros que podem ser aperfeiçoados e ajustados para um cenário mais apropriado. Essa flexibilidade faz do XGBoost uma escolha consistente para problemas de regressão e classificação (binária e multiclasse).

O XGBoost e o *random forest* diferem no modo como as árvores são construídas, a ordem e a maneira como os resultados são combinados. Foi demonstrado que o XGBoost tem melhor desempenho que o *random forest* se os parâmetros forem ajustados cuidadosamente.

No estudo “Encoding Techniques for High-Cardinality Features and Ensemble Learners” Johnsn e Khoshgoftaar avaliam o desempenho de classificação de cinco técnicas de codificação para variáveis categóricas utilizando os algoritmos *Random Forest*, XGBoost, Light-GBM e CatBoost. A proposta deste estudo é utilizar a variável categórica Hcpcs2 que é uma identificação padronizada de produtos e serviços de saúde, nos testes realizados

foram observados ganhos utilizando a variável (JOHNSON; KHOSHGOFTAAR, 2021). Segundo este estudo os modelos XGBoost e LightGBM apresentam melhor desempenho geral com AUC de 87,15% e 84,86%, respectivamente. O XGBoost e LightGBM superam os modelos CatBoost e *Random Forest*, o *Random Forest* teve o menor desempenho com um AUC máximo de 80,34%.

No estudo “Predicting urinary tract infections in the emergency department with machine learning” (TAYLOR et al., 2018) treinou, validou e comparou modelos preditivos baseados em aprendizado de máquina para infecção do trato urinário em um grande conjunto diversificado de pacientes no pronto-socorro. Foi desenvolvido modelos para previsão de infecção do trato urinário com seis algoritmos de aprendizado de máquina usando informações demográficas, sinais vitais, resultados de laboratório, medicamentos, histórico médico anterior, queixa principal e resultados de exames físicos e históricos estruturados. Como resultado os modelos de aprendizado de máquina apresetaram uma área sob a curva variando de 0,826 a 0,904. Os modelos XGBoost completo e reduzido demonstraram especificidade muito melhor quando comparados ao diagnóstico ou administração de antibióticos com diferenças de especificidade de 33,3 (31,3-34,3) e 29,6 (28,5-30,6), ao mesmo tempo que demonstraram *recall* superior em comparação com o diagnóstico com diferenças de *recall* de 38,7 (38,1–39,4) e 33,2 (32,5–33,9).

3.2.6 Treinamento

O treinamento de algoritmo refere-se ao processo de ensinar um algoritmo de aprendizado de máquina a fazer previsões ou executar uma tarefa específica com base nos dados de entrada, nesta fase o algoritmo é alimentado com entradas e suas respectivas saídas desejadas, esta fase é fundamental não apenas para preparar a máquina, mas também para aprimorar constantemente suas habilidades de predição.

Para realizar o treinamento dos algoritmos foi necessário transformar as variáveis categóricas em variáveis numéricas, isso por que os algoritmos citados não conseguem realizar o treinamento de variáveis categóricas na linguagem Python, a técnica de transformação das variáveis categóricas em variáveis numéricas também é conhecida como *encoder*.

Existe um espectro de técnicas para usar variáveis categóricas em aprendizado de máquina. Encontrar a técnica certa pode ter um impacto significativo no desempenho de um modelo. Determinadas técnicas são simples, é importante na escolha da técnica avaliar além da qualidade, a complexidade e o tempo de execução (HANCOCK; KHOSHGOFTAAR, 2020).

A biblioteca *scikit-learn category encoders* possui sete técnicas para codificar variáveis categóricas, no estudo “Pesquisa sobre dados categóricos para redes neurais” John Hancock e Taghi Khoshgoftaar apresentam uma compreensão dos métodos atuais para a aplicação de redes neurais em dados qualitativos.

Neste trabalho, citaremos quatro técnicas da biblioteca *scikit-learn* com um breve resumo, as quatro técnicas são: O*ne Hot, Generalized Linear Mixed Model Encoder (GLMM), Target Encoder* e *CatBoost.*

A codificação *one hot* é uma técnica que requer pouco trabalho para ser usada, ela é simples de implementar e tem um tempo de execução eficiente. Contudo, uma das desvantagens da codificação *one hot* é que ela consome de forma significativa recursos de armazenamento, em casos em que um conjunto de dados tenham uma variável categórica com *n* valores, estes *n* valores são transformados em *n* colunas (HANCOCK; KHOSHGOFTAAR, 2020). O conjunto de dados utilizado neste trabalho possui variáveis de alta dimensionalidade, com isto, seria inviável utilizar a codificação *one hot*.

A codificação *Generalized Linear Mixed Model Encoder* pode ser interpretada como um modelo linear misto generalizado que utiliza da variável alvo de forma aleatória para transformar a variável categórica em variável numérica (PARGENT, 2019). Em testes realizados com esta técnica, foi observado baixa performance em tempo de execução do modelo, além disto, não foi observado ganhos significativos em comparação com as demais técnicas, desta forma, a técnica GLMM foi descartada.

O CatBoost é um codificador categórico supervisionado baseado no valor da variável alvo que suporta alvos binomiais e contínuos. A codificação baseada no valor da variável alvo é uma técnica popular usada para codificação categórica. Ele substitui um recurso categórico pelo valor médio do alvo correspondente a essa categoria no conjunto de dados de treinamento combinado com a probabilidade do alvo em todo o conjunto de dados. Mas isso

introduz um vazamento de alvo, pois o alvo é usado para prever o destino. Esses modelos tendem a ser superajustados e não generalizam bem em circunstâncias invisíveis.

O CatBoost é semelhante à codificação de alvo, mas também envolve um princípio de ordenação para superar esse problema de vazamento de destino. Ele usa o princípio semelhante à validação de dados de séries temporais. Os valores da estatística de alvo dependem do histórico observado, ou seja, a probabilidade do alvo para o recurso atual é calculada apenas a partir das linhas anteriores a ele (JOHNSON; KHOSHGOFTAAR, 2021).

Target encoder é uma técnica de codificação bayesiana que também utiliza a variável alvo para realizar a codificação. Os dados são agrupados e, em seguida, uma média do destino é calculada para o agrupamento (PARGENT, 2019).

Tabela 2 - Transformação variáveis categóricas em numéricas usado Target e One hot

Dados Categóricos

id	color	target
1	red	1
2	blue	0
3	green	1

Target Encoding

id	color	target
1	0.012308	1
2	0.012308	0
3	0.290323	1

One-hot encoding

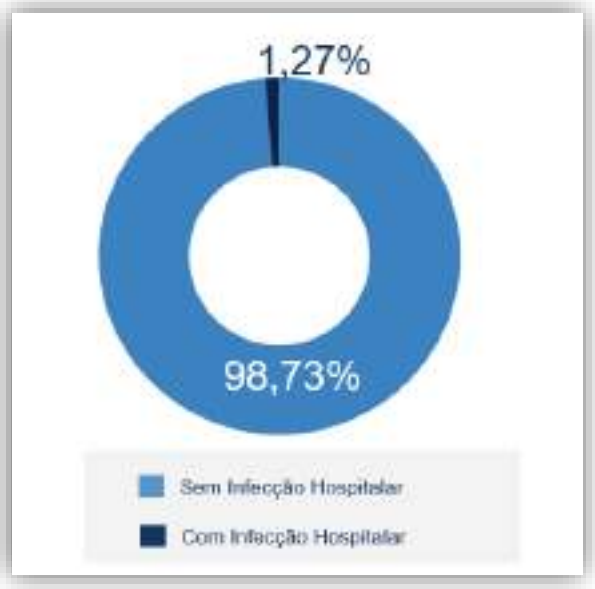
id	color_red	color_blue	color_green	target
1	1	0	0	1
2	0	1	0	0
3	0	0	1	1

Fonte: Autora da dissertação

Em um primeiro momento, o treinamento do algoritmo foi realizado com 80% da base de dados coletada, a base possui 178.481 registros, destes, 2.274 tinham o registro de infecção hospitalar, representando 1,27%, ou seja, os dados de treinamento ficaram com 142.784 e os dados de teste 35.697.

Podemos observar que nos dados de treino temos 140.965 registros com o rotulo 0 (Sem Infecção) e 1.819 registros com o rotulo 1 (Com Infecção), isso demostra que a base de dados está desbalanceada.

Gráfico 1 - Distribuição dos dados de infecção hospitalar



Fonte: Autora da dissertação

Este problema é muito comum, os dados do mundo real geralmente têm problemas de desbalancemanto de classes. Nesse cenário, os algoritmos de aprendizado costumam ser tendenciosos para as classes majoritárias, tratando as minoritárias como discrepantes ou ruídos (BASGALL et al., 2019).

Para lidar com este problema, técnicas de pré-processamento foram desenvolvidas para balancear a distribuição de classes. Isso pode ser alcançado suprimindo instâncias majoritárias (*undersampling*) ou criando exemplos minoritários (*oversampling*). Em relação aos métodos de sobreamostragem, um dos mais difundidos é o algoritmo SMOTE, que cria exemplos artificiais de acordo com a vizinhança de cada instância de classe minoritária (BASGALL et al., 2019).

Neste trabalho foi utilizada a técnica de sobreamostragem SMOTE (*Synthetic Minority Over-sampling Technique*) que é um algoritmo heurístico de sobreamostragem. Ele gera amostras artificiais da classe minoritária interpolando instâncias existentes que estão próximas umas das outras.

Em suma, o processo de treinamento do algoritmo é iterativo e geralmente requer experimentação, ajuste fino e refinamento para atingir o desempenho desejado. Envolve uma combinação de conhecimento de domínio, exploração de dados e compreensão das capacidades e limitações dos algoritmos para desenvolver um modelo eficaz e preciso.

3.2.7 Avaliação do modelo

A sétima etapa é a avaliação do modelo, esta etapa tem como objetivo verificar se a máquina realmente foi capaz de aprender, e não apenas de memorizar respostas anteriores.

A avaliação permite testar se após o treinamento a máquina está suficientemente capacitada. Com base na configuração designada, a avaliação realizará testes diante de dados que ela jamais havia visto. Sendo assim, a ideia é que a avaliação seja uma representação de como a máquina pode performar em um contexto real.

Algumas métricas de desempenho foram usadas para avaliar os diferentes modelos de algoritmos: Matriz de Confusão, Acurácia, Precisão, *Recall* (Sensibilidade), F1-Score, média da Validação Cruzada e Curva AUROC.

Cada métrica tem suas peculiaridades que devem ser levadas em consideração na escolha de como o modelo de classificação será avaliado.

A precisão pode ser usada em uma situação em que os Falsos Positivos são considerados mais prejudiciais que os Falsos Negativos.

O *recall* é usado em situações em que os Falsos Positivos são considerados mais prejudiciais que os Falsos Negativos, em trabalhos relacionados a área de saúde o ideal é que o modelo encontre todos os pacientes doentes, mesmo que classifique alguns saudáveis como doentes (situação de Falso Positivo) no processo. Ou seja, o modelo deve ter alto *recall*, pois classificar pacientes doentes como saudáveis pode ser danoso.

3.2.8 Aprimoramento dos hiperparâmetros

O aprimoramento dos hiperparâmetros é a etapa do processo de aprendizado de máquina que tem como objetivo otimizar a escolha dos

parâmetros a fim de identificar a melhor combinação de valores possíveis que podem ser utilizados como parâmetro de entrada para o algoritmo que será utilizado (HO, 1995). Esta fase é importante para identificar valores que afetam diretamente a acurácia do modelo e o tempo de treinamento necessário.

Um dos principais parâmetros a serem analisados é o quanto a linha de aprendizado da máquina é alterada de acordo com a informação adquirida no procedimento anterior. É indicado nesta etapa testar as possibilidades para analisar melhor o aprendizado de máquina e ensaiar formas de como ele pode ser aprimorado.

Em aprendizado de máquina a maioria dos algoritmos apresenta um ou mais hiperparâmetros que controlam a complexidade, ou seja, o equilíbrio entre viés e variância do modelo ajustado. Estes parâmetros, denominados parâmetros de sintonização ou hiperparâmetros, podem ser diretamente especificados antes do ajuste do modelo preditivo, pois não são diretamente estimados pelos dados de treinamento, ou otimizados por validação cruzada, pois não há uma fórmula analítica disponível para o cálculo do seu valor apropriado (KUHN; JOHNSON, 2013).

São exemplos de hiperparâmetros o número k de vizinhos mais próximos, utilizados pelo algoritmo KNN, e o número m de número de instâncias no nó folha presentes no algoritmo de árvores de decisão. Como os hiperparâmetros apresentam relação com a complexidade de um modelo preditivo, escolhas inadequadas podem resultar em sobreajuste (*overfitting*) e performance ruim do modelo em novas observações.

Efetivamente, uma métrica é escolhida para avaliar o erro de predição, como o MSE (em problemas de regressão) e a AUC ROC (em problemas de classificação) e, para um dado algoritmo, essa métrica é avaliada em uma lista de valores candidatos ao hiperparâmetro por validação cruzada, com o objetivo de selecionar aquele que resulte em um modelo que minimiza o erro de predição (KUHN; JOHNSON, 2013).

Neste trabalho foi utilizado o GridSearchCV que é um framework do scikit-learn, o GridSearchCV foi aplicado nos 6 algoritmos. As métricas foram novamente aplicadas para comparar com as métricas dos modelos anteriores.

3.2.9 Produção

Seguindo com a última etapa do projeto, a etapa de produção, nela são aplicados ao modelo os dados de treino, dados estes que não foram utilizados nas etapas anteriores. Nesta fase o modelo será efetivamente utilizado para responder as perguntas para as quais foi treinado.

4 RESULTADOS E DISCUSSÃO

Nesta seção será apresentado todas as fases da pesquisa e seu respectivo resultado.

4.1 Coleta de Dados

A coleta de dados foi realizada utilizando a base de dados do Hospital Márcio Cunha no período de março de 2015 a dezembro de 2019, foram coletados dados de todos os pacientes internados nas unidades 1 e 2 da instituição, os dados dos pacientes que não contraíram infecção também foram coletados. A instituição utiliza EHS (registros eletrônicos de saúde) desde 2002 e em março de 2015 foi realizada a migração do sistema, a utilização de um prontuário eletrônico possibilitou a realização da pesquisa.

Foram coletados dados das tabelas:

- Atendimento do paciente: Número do atendimento, sexo, unidade de federação, idade, clínica de internação, caráter de internação (Eletivo ou Urgência), número do leito, setor de atendimento, estabelecimento do atendimento, data de internação, data da alta, data da previsão de alta, motivo da alta, setor da alta, observação de alta e quantidade de dias internados.
- Convênio acomodação: Descrição do convênio e tipo de acomodação (Enfermaria e Apartamento).
- Diagnóstico: Diagnóstico Principal e Secundário.
- Dispositivo: Dispositivos utilizados durante a internação (exemplo: sonda vesical, ventilação mecânica, cateter), data de instalação, data retirada e motivo da retirada.
- Ficha de infecção hospitalar: Número da ficha, código da topografia, descrição da topografia da infecção, sítio da infecção, nome do médico, clínica da topografia, setor origem infecção, estabelecimento origem infecção.
- Sinais Vitais: data sinal vital, pressão arterial sistólica, pressão arterial diastólica, frequência cardíaca, frequência Respiratória, temperatura, nível consciência.

Para a coleta foi utilizado o banco de dados oracle da instituição, os dados das tabelas foram exportados, cada tabela gerou um arquivo csv. Antes de importar os arquivos, eles foram salvos na codificação UTF-8.

Os arquivos foram importados utilizando o *Google Colaboratory*, cada tabela citada acima gerou um *dataframe*, com isto, foi necessário a fusão das tabelas para gerar um *dataframe* único, esta fusão resultou em uma tabela com 76 variáveis e 331.905 registros, destes 94.505 tinham a identificação de infecção hospitalar confirmada.

4.2 Preparação dos dados

Na preparação de dados, o primeiro passo foi realizar a validação dos dados importados, foram confrontados os dados importados e os dados contidos na base da instituição, para isso foi realizado o comparativo com os indicadores do hospital e uma breve análise com a enfermeira responsável pelo setor de infecção hospitalar.

Após essa primeira análise, foi realizado na sequência a seleção de recursos, utilizando a biblioteca pandas foi possível identificar colunas por tipos de dados e detectar e manipular valores ausentes, nesta etapa foram excluídas colunas com alto percentual de informações nulas e que eram irrelevantes para o modelo, como, data previsão de alta, município IBGE, endereço, idade em meses, médico responsável, observação do atendimento, tipo atendimento SUS, informações de convênio, tipo de sangue (mais de 77% desta informação estava nula). Colunas com alta correlação também foram excluídas, como, suspeita de infecção, data de infecção, dispositivos invasivos associados a infecção, setor origem infecção.

Nesta etapa também foi realizado o agrupamento das informações para que seja melhor interpretado pelo modelo, como exemplo temos a informação do motivo alta, na base importada tínhamos diversos tipos de alta, Alta Curado, Alta Melhorado, Óbito Com Declaração, Óbito 72 horas, estas informações foram transformadas para somente duas: Alta e Óbito.

Tabela 3 - Descrição motivo de alta

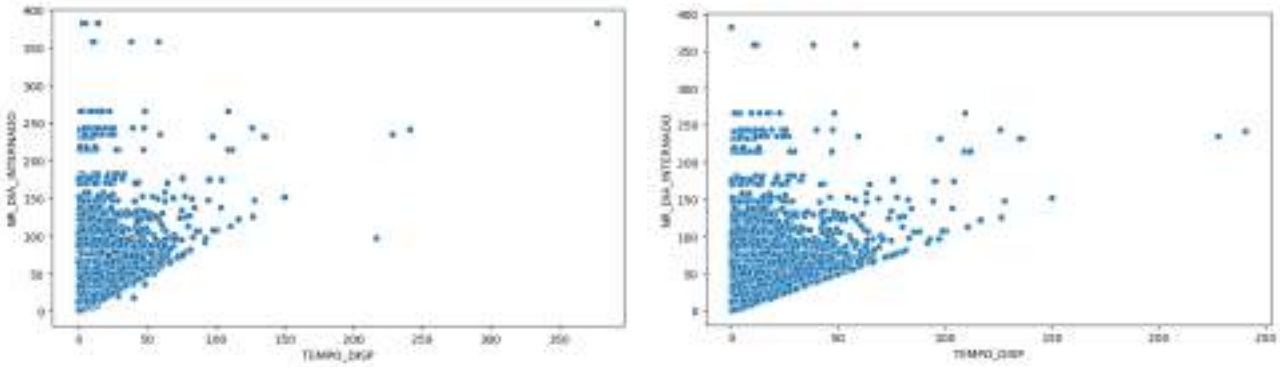
Alta melhorado
Alta da mãe/puérpera e do recém-nascido
Óbito com declaração fornecida pelo médico assistente
Alta curado
Alta com previsão de retorno para acompanhamento do paciente
Alta da mãe/puérpera e permanência do recém-nascido
Encerramento Administrativo
Alta a pedido
Alta por outros motivos
Óbito com declaração fornecida pelo Instituto Médico Legal - IML
Cancelado
Transferência para outro estabelecimento
Alta do paciente agudo em psiquiatria
Alta da mãe/puérpera e óbito recém-nascido
Óbito com declaração fornecida pelo Serviço de Verificação de Óbito - SVO
Alta da mãe/puérpera com óbito fetal
Alta por Evasão
Permanência por processo de doação de orgãos, tecidos e células - doador morto
Transferência para internação domiciliar
Permanência por intercorrência
Permanencia por características próprias da doença
Permanência por processo de doação de orgãos, tecidos e células - doador vivo
Óbito da mãe/puérpera e permanência do recém-nascido
Óbito da gestante e do concepto

Fonte: Autora da dissertação

Outro ponto trabalhado foi a identificação de possíveis casos atípicos (*outliers*). Para identificar os *outliers* foi utilizada a biblioteca de visualização de dados *seaborn*, com o gráfico de dispersão, representado pelo gráfico 2, podemos verificar pontos de dados que estão longe da maioria das observações, ao detalhar o entendimento destes casos, identificamos que o tempo que o paciente permaneceu com um dispositivo invasivo estava maior que o tempo de internação.

Para confirmar que os dados identificados tratavam-se de *outliers* foi realizada a validação das informações em conjunto com a enfermeira responsável pela CCIH do hospital Márcio Cunha, segundo a responsável, isto acontecia pois em alguns casos não era registrado a retirada do dispositivo no momento da alta, para resolver esta situação, foi atualizado o tempo de dispositivo para o mesmo tempo de internação para os registros que estavam nesta situação. Como este problema tinha sido detectado anteriormente, foi realizada uma melhoria no sistema de gestão do Hospital Márcio Cunha em 2018 para que registrasse automaticamente a retirada do dispositivo no momento da alta do paciente, caso este ainda estivesse em aberto.

Gráfico 2 -Comparativo tempo dispositivo invasivo



Fonte: Autora da dissertação

Além das técnicas citadas acima, foi necessário também realizar a transformação dos dados, os dados longitudinais *'long data structure'* foram transformados para estrutura de dados amplos ou transversais *'broad data structure'*, onde cada paciente tem um registro de dados, independentemente do número de medições ao longo do tempo, a figura 4 representa o tipo destas estruturas.

Figura 4 - Exemplo de estrutura de dados

'long' data structure

ID	Y	time	X_4
1	3.5	1	1
1	3.7	2	1
1	3.9	3	1
1	3.0	4	1
1	3.2	5	1
1	3.2	6	1
2	4.1	1	1
2	4.1	2	1
.			
.			
N	5.0	5	2
N	4.7	6	2

'broad' data structure

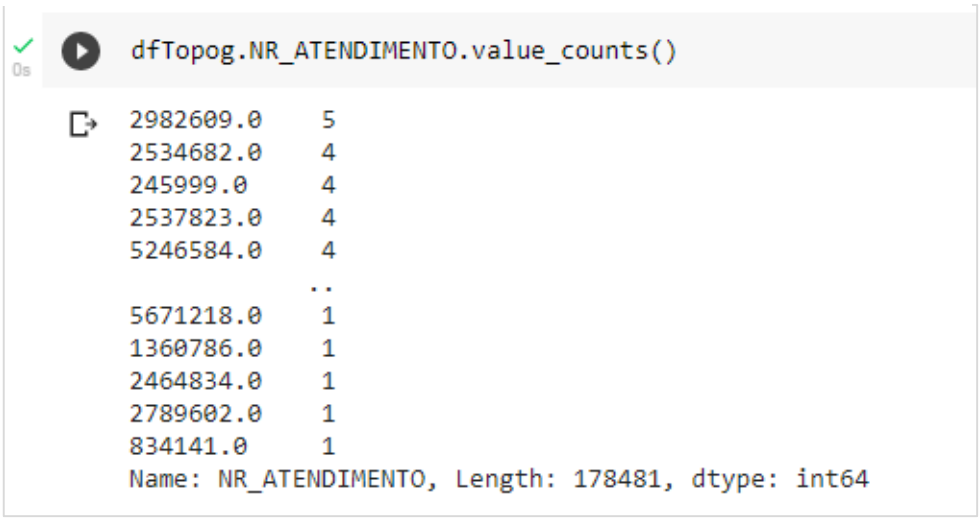
ID	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}	Y_{16}	X_4
1	3.5	3.7	3.9	3.0	3.2	3.2	1
2	4.1	4.1	4.2	4.6	3.9	3.9	1
3	3.8	3.5	3.5	3.4	2.9	2.9	2
4	3.8	3.9	3.8	3.8	3.7	3.7	1
.							
.							
N	4.0	4.6	4.7	4.3	4.7	5.0	2

Fonte: Mark J. van der Laan, 2018

Para a transformação dos dados foram avaliados os atributos que se repetem ao longo do período de internação, entre eles estão os dados relacionados com a topografia das infecções e os dispositivos invasivos, ou seja, os dados se repetem, pois, o paciente pode contrair mais de um tipo de infecção ou utilizar vários dispositivos invasivos.

Conforme a figura 5 podemos observar que o atendimento 2982609 possui 5 registros (linhas), ou seja, em momentos diferentes o paciente contraiu 5 infecções, a coluna NR_ATENDIMENTO identifica o registro do atendimento do paciente.

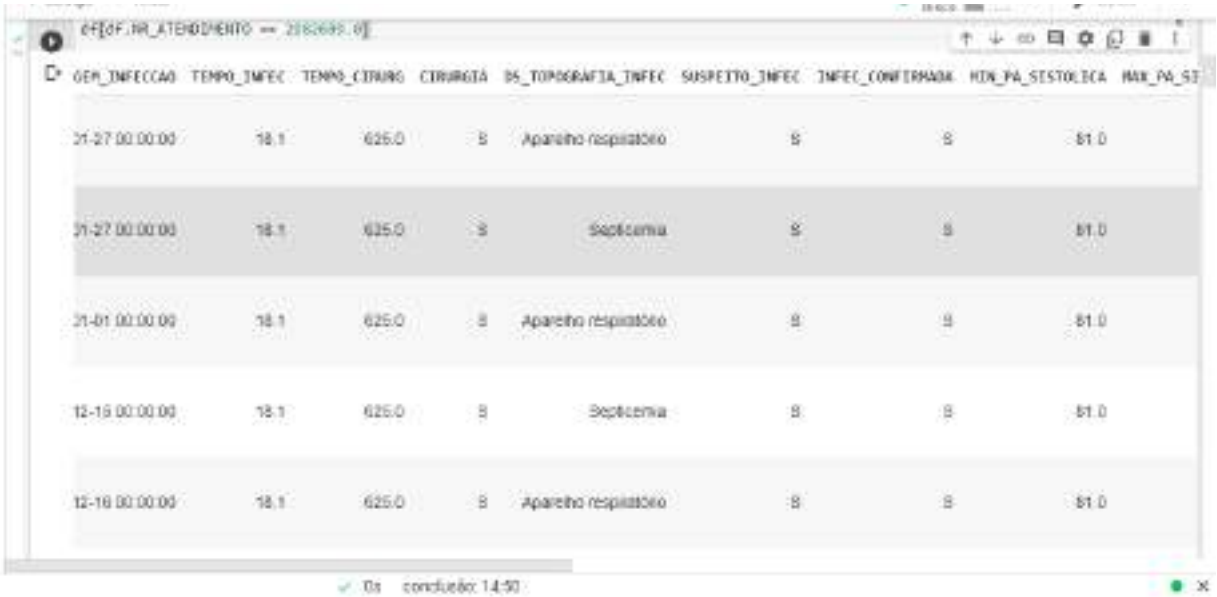
Figura 5 - Atendimentos que possuem mais de um registro de infecção



Fonte: Autora da dissertação

Na figura 6 visualizamos os dados detalhados, onde o paciente obteve infecção do aparelho respiratório e septicemia, este paciente apresentou este quadro se repetindo em momentos diferentes o que gerou 5 registros.

Figura 6 - Detalhamento do atendimento que contraiu infecções



Fonte: Autora da dissertação

Para a transformação dos dados de linhas em colunas foi utilizada a função *crosstab* da biblioteca pandas, que pode ser usada para agrupar duas ou mais variáveis e realizar cálculos para um determinado valor para cada grupo. A função recebe duas ou mais listas, pandas ou colunas de *dataframe* e retorna uma frequência de cada combinação por padrão, desta forma para cada atendimento temos apenas um registro (uma linha), assim simplificando o processo para aplicação do modelo de aprendizado de máquina.

Na figura 7 é apresentado os dados após a utilização do *crosstab*, os registros de aparelho respiratório transformaram-se em INFEC_RESPIRATORIO e septicemia em INFEC_SEPSE, a função contabilizou a quantidade de vezes que foi apresentada a infecção.

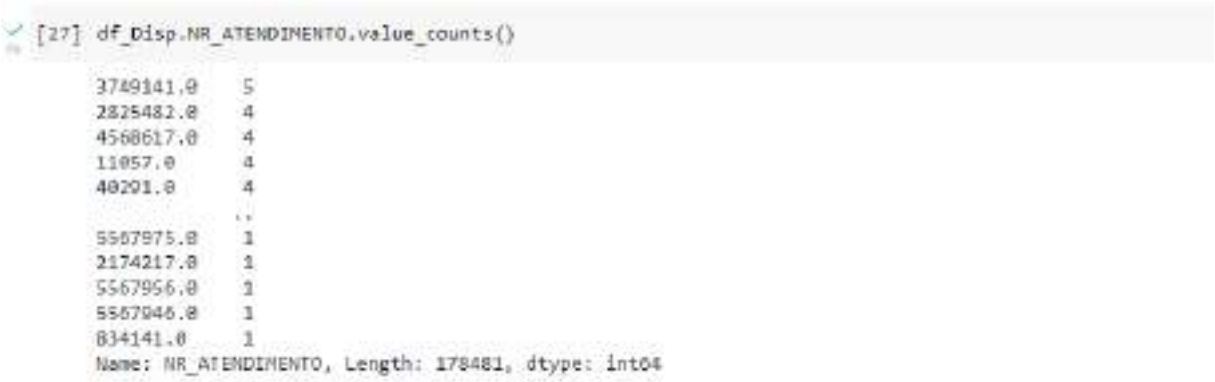
Figura 7 - Dataframe após a transformação dos dados em colunas

Fonte: Autora da dissertação

Observação: As colunas foram renomeadas para melhor organização do código.

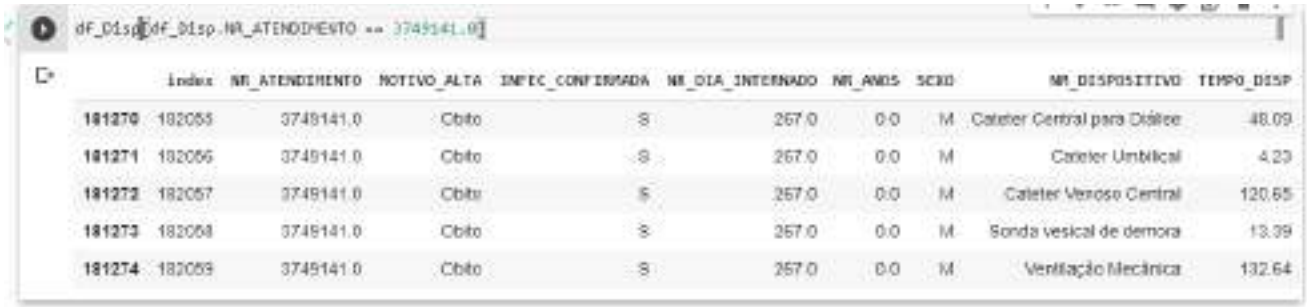
Para os dados de dispositivos foi utilizada a mesma análise feita para as topografias das infecções, foi verificado os atendimentos que tinham mais de um dispositivo, utilizaremos como exemplo o atendimento 3749141 que possui 5 registros, conforme figura 8 e 9.

Figura 8- Atendimentos com mais de um registro de dispositivo



Fonte: Autora da dissertação

Figura 9 - Detalhamento do atendimento que utilizou dispositivo



Fonte: Autora da dissertação

Neste caso também foi utilizada a função *crosstab* da biblioteca pandas para realizar a transformação dos registros de linhas em coluna, desta forma para cada atendimento temos apenas um registro (uma linha).

Figura 10 - Dataframe após a transformação dos dados em colunas

Fonte: Autora da dissertação

O tratamento realizado nos *dataframes* de infecção hospitalar e dispositivos invasivos foram feitos em tabelas diferentes, desta forma, o projeto possuía quatro *dataframes* , o *dataframe* principal com os dados demográficos dos pacientes, o *dataframe* com informações das infecções associadas aos dispositivos, o *dataframe* com os dados das infeções e o *dataframe* dos dispositivos invasivos, estes dois últimos com a transformação dos dados em colunas. Com os 4 *dataframes* já estruturados foi realizada a unificação dos dados em uma única tabela.

Figura 31 - Merge para unificação da tabela final

Fonte: Autora da dissertação

Figura 12 -Tabela final com um único registro por atendimento

Fonte: Autora da dissertação

Em suma, três critérios foram observados para ajudar a garantir que os dados sejam adequadamente estruturados e sejam facilmente analisados pela maioria dos sistemas de gerenciamento de dados e software de análise.

- ✓ Cada linha deve representar uma única observação, ou seja, registro.
- ✓ Cada coluna deve conter apenas um único valor, de forma que, não tenha unidades na célula com os valores ou várias medições em uma única célula, como por exemplo, gênero, idade, número de dias internados, etc.
- ✓ Deve haver apenas uma coluna para cada tipo de informação.

Inicialmente, na primeira etapa da coleta de dados, a fusão das tabelas importadas resultou em uma tabela com 76 variáveis e 331.905 registros, destes 94.505 tinham a identificação de infecção hospitalar confirmada. Após a etapa de preparação de dados, com seleção de recursos, onde foram excluídas as variáveis com alta correlação, colunas com alto percentual de informações nulas e transformação dos dados longitudinais em dados transversais, tivemos como resultado final 36 variáveis e 178.481 registros, destes 2.274 tinham a identificação de infecção hospitalar confirmada.

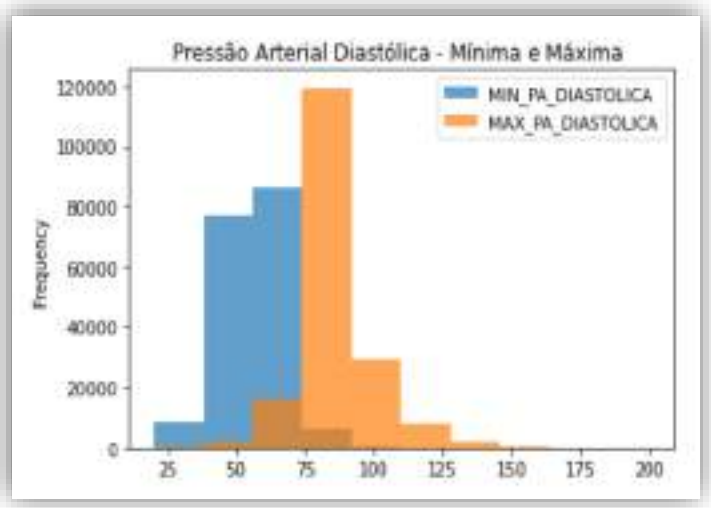
4.3 Análise Exploratória

Neste trabalho utilizou-se dos gráficos para compreensão dos dados, para os dados de sinais vitais foi utilizado o gráfico de histograma para mostrar a distribuição das frequências mínima e máxima, a escolha dos valores mínimos e máximo se deve ao alto volume destes dados pois no período de internação do paciente é gerado vários registros de sinais vitais, em alguns casos a coleta

destes dados é realizada de forma automática com a integração dos equipamentos de sinais vitais com o sistema de gestão, no Hospital Márcio Cunha estes dados são integrados de 2 em 2 horas, gerando um alto volume de dados, analisando estes dados em conjunto com a enfermeira responsável pelo setor de Infecção Hospitalar, optou-se por utilizar o valor mínimo e máximo de cada paciente durante o período de internação.

O gráfico 3 mostra os dados de pressão arterial diastólica (PAD), através do gráfico podemos observar que a PAD exibe dados entre 20 e 200, o paciente em estado normal apresenta PAD de 60 a 85, onde concentra-se a maior parte dos dados.

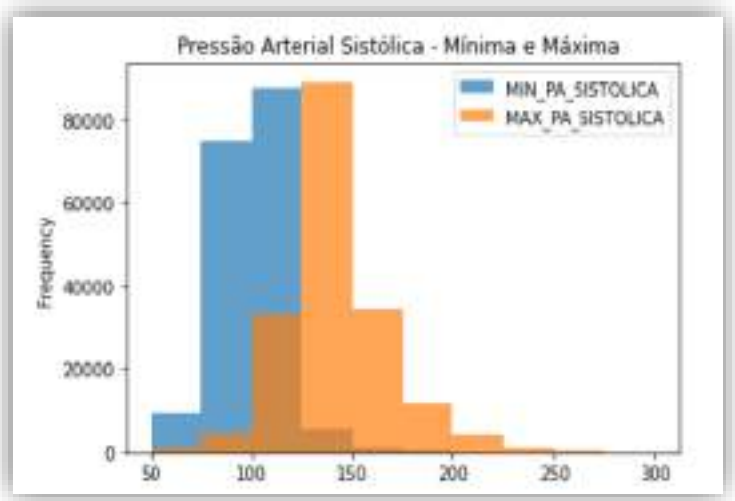
Gráfico 3 - Histograma de Frequência Pressão Arterial Diastólica



Fonte: Autora da dissertação

O gráfico 4 mostra os dados da pressão arterial sistólica (PAS), através do gráfico podemos observar a PAS com valores entre 50 e 300, o paciente em estado normal apresenta valores entre 90 e 130.

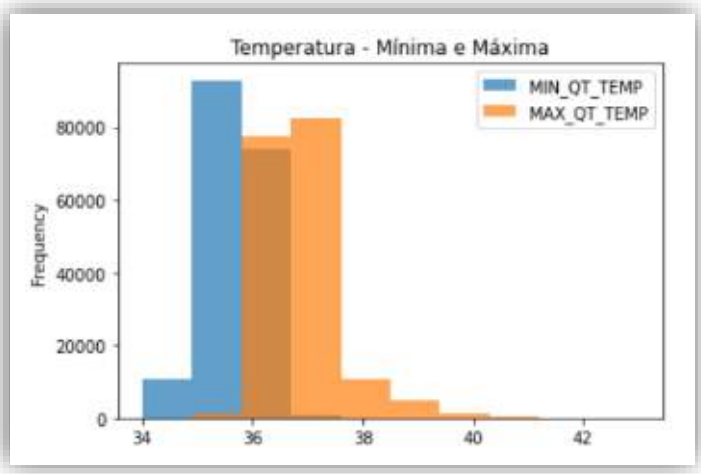
Gráfico 4 - Histograma de Frequência Pressão Arterial Sistólica



Fonte: Autora da dissertação

No gráfico 5 temos o gráfico com a frequência dos valores mínimo e máximo de temperatura corporal, os dados exibidos são entre 34 e 43, sendo considerado temperatura corporal normal entre 35 e 37,4.

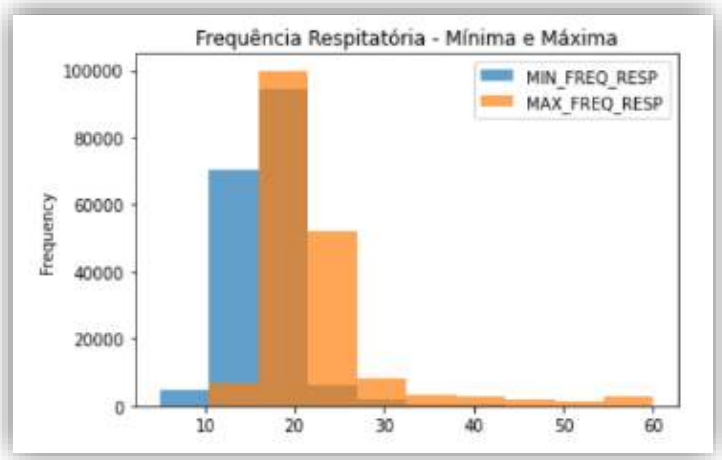
Gráfico 5 - Histograma de Frequência da Temperatura Corporal



Fonte: Autora da dissertação

A frequência respiratória é representada pelo gráfico 6, onde podemos observar os valores no gráfico entre 5 e 60, sendo considerado normal entre 12 e 25.

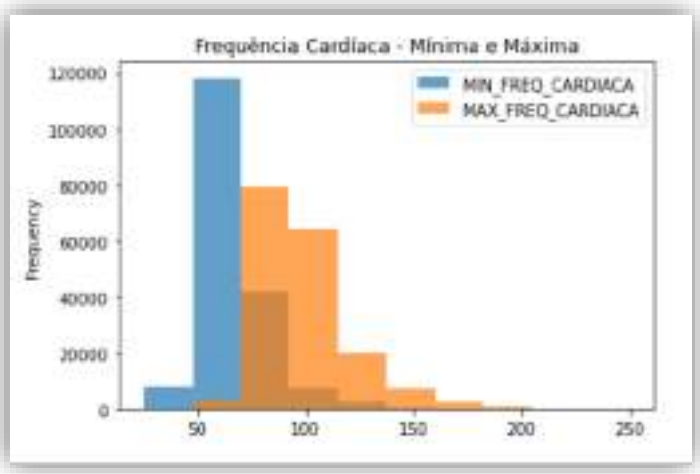
Gráfico 6 - Histograma de Frequência Respiratória



Fonte: Autora da dissertação

A frequência cardíaca é representada pelo gráfico 7, no gráfico podemos observar os valores entre 25 e 250, para a frequência cardíaca é considerado valor normal entre 60 bpm e 100 bpm.

Gráfico 7 - Histograma de Frequência Cardíaca



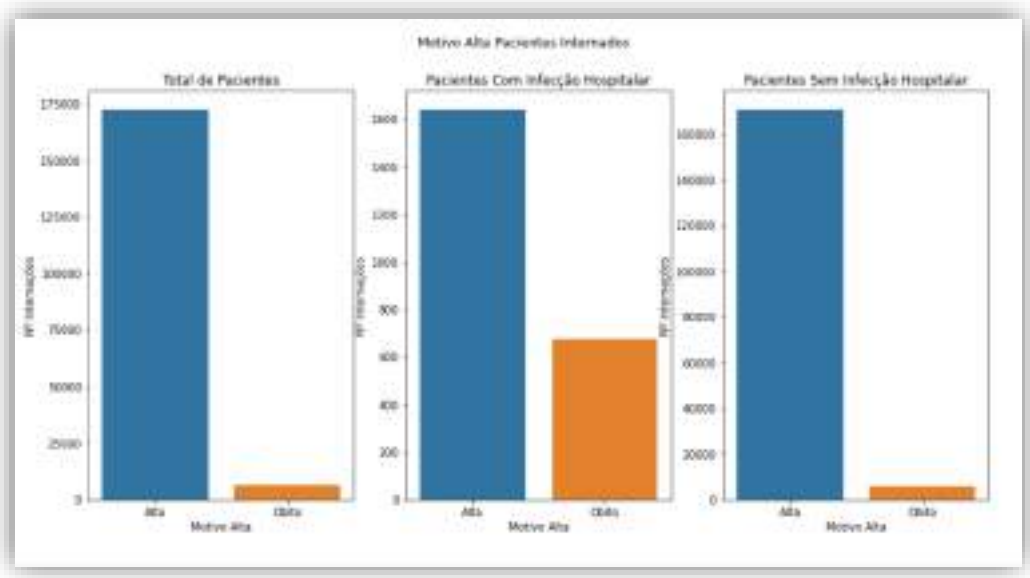
Fonte: Autora da dissertação

Em todos os gráficos, podemos observar que os valores estão concentrados nos valores de referência, ou seja, os valores considerados normais em condições saudáveis. As alterações nos sinais vitais podem ser causadas por diversos fatores, porém são indicativos relacionados a saúde do

paciente, os valores mínimos e máximos foram tratados retirando os *outliers*, além disto, os dados foram validados pela médica e pela enfermeira responsáveis pelo controle de Infecção Hospitalar da instituição onde foram coletados os dados.

Para realizar um comparativo dos dados de motivo de alta foi utilizado o gráfico de barras na vertical representando a quantidade de pacientes internados, no gráfico 8 podemos observar que comparando ao total de pacientes internados a quantidade de Óbito é consideravelmente pequeno em relação a quantidade de Alta, enquanto que o número de Óbito aumenta consideravelmente quando buscamos somente os dados de pacientes que contraíram infecção hospitalar, quando buscamos o dados apenas de pacientes que não tiveram infecção hospitalar, o gráfico fica equivalente ao total de pacientes.

Gráfico 8 - Comparativo motivo alta pacientes internados que contraíram ou não infecção

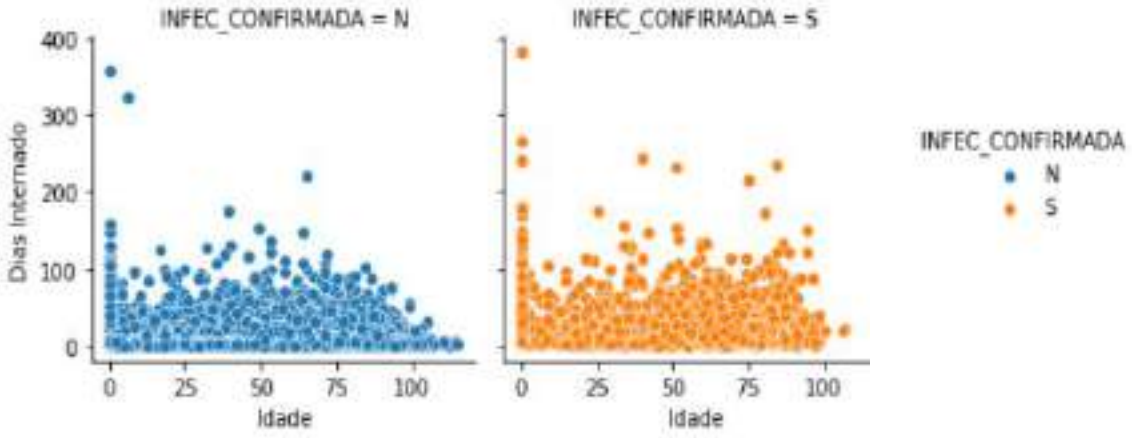


Fonte: Autora da dissertação

Para avaliar a correlação entre as variáveis idade, número de dias internados e infecção confirmada (S Sim ou N Não) utilizamos o gráfico de dispersão conforme mostra o gráfico 9. No gráfico podemos observar uma pequena variação em relação ao aumento da idade e número de dias internados para pacientes que contraíram infecção hospitalar, o que faz total sentido, pois a infecção hospitalar aumenta o tempo de internação do paciente e com o

avanço da idade existe um maior risco de complicações o que pode levar ao maior tempo de internação e maior exposição de forma a contrair infecção hospitalar.

Gráfico 9 - Correlação entre idade, dias internados e paciente com infecção confirmada

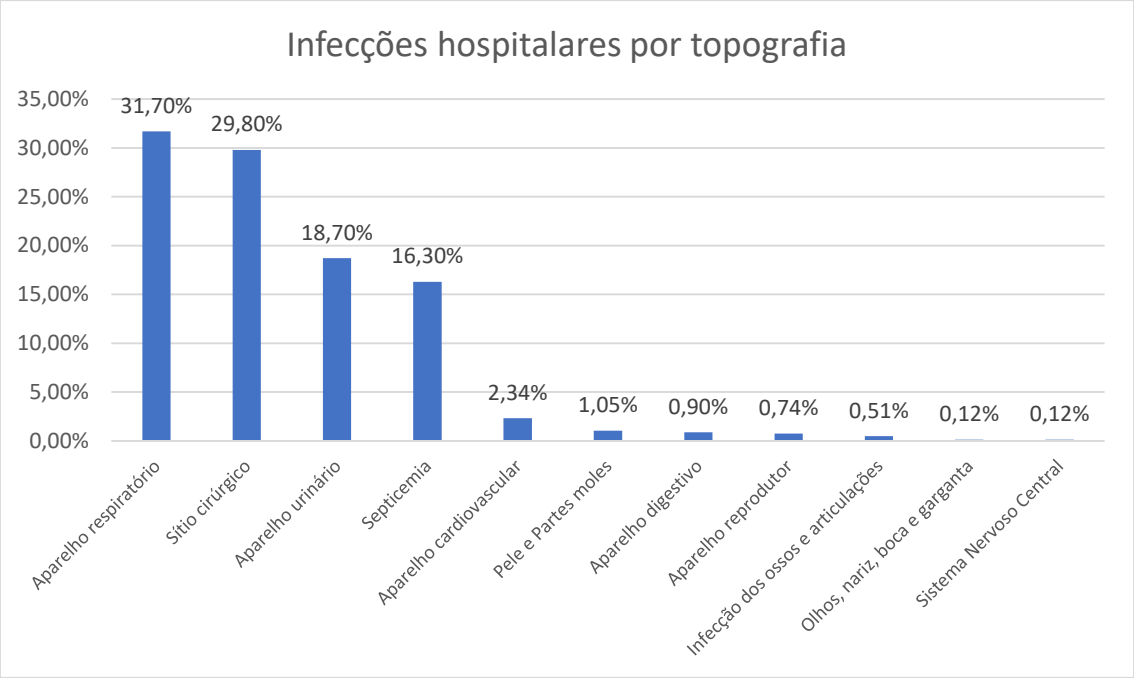


Fonte: Autora da dissertação

4.3.1 Dados Topografia das Infecções

Para melhor organização do estudo, nesta seção serão apresentados os gráficos relacionados com a topografia de todas as infecções registradas no período de 2015 à 2019, assim identificando a incidência das principais infecções hospitalares.

Gráfico 10 - Infecções hospitalares por topografia



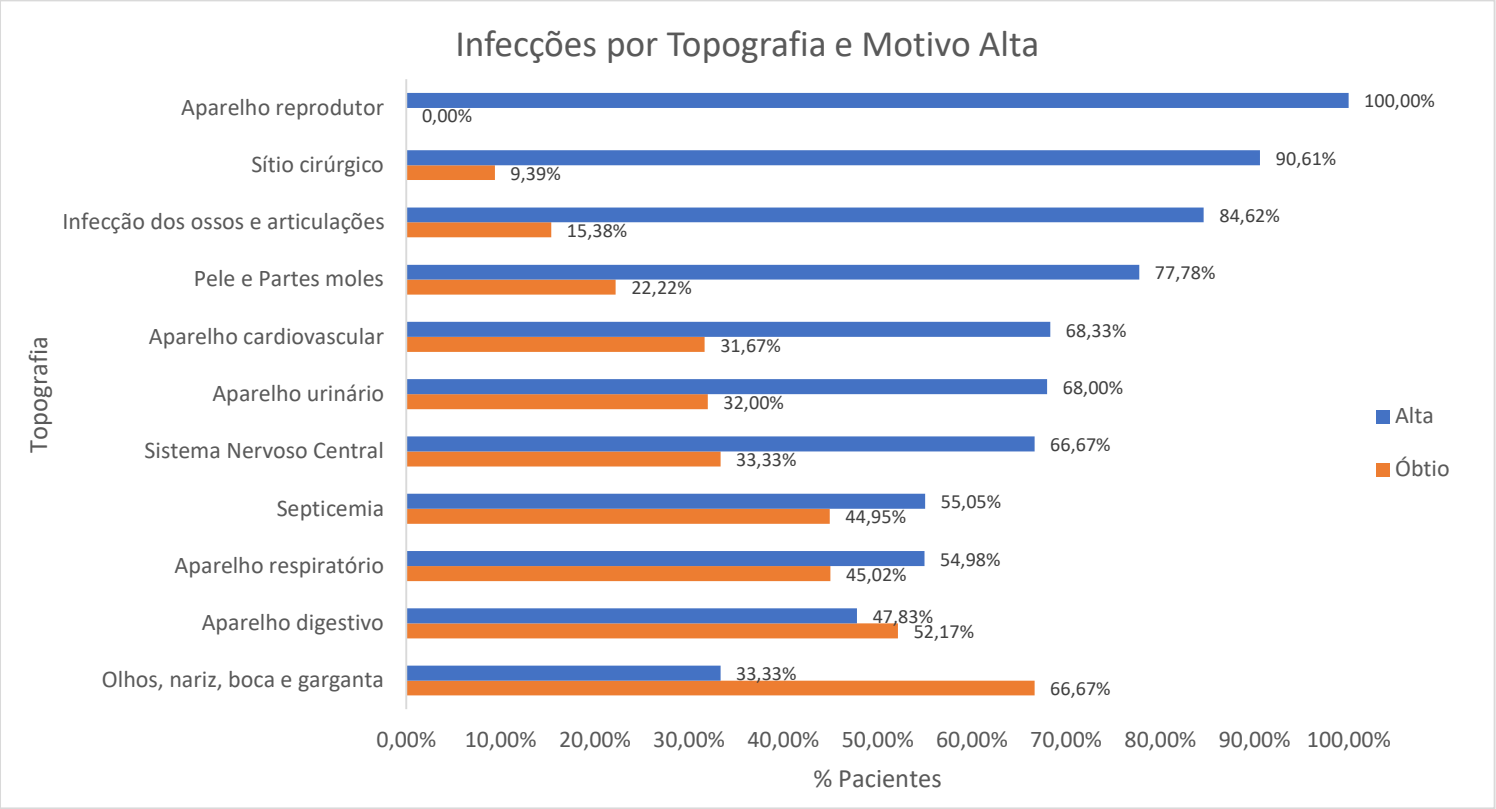
Fonte: Autora da dissertação

Observa-se que a topografia de maior incidência é do aparelho respiratório com 31,7%, em seguida sítio cirúrgico com 29.8%, aparelho urinário com 18,7% e septicemia com 16,3%.

No próximo gráfico é apresentada as infecções por topografia e por motivo de alta, um ponto importante a ser salientado é que os registros de infecção são realizados de acordo com a data que a infecção foi adquirida, com isto, um mesmo paciente pode ter mais de uma infecção.

No gráfico 11 pode-se observar que as infecções de olhos, nariz e garganta possuem a maior taxa de óbito com 66,67%, em seguida o aparelho digestivo com 52,17% e na sequencia o aparelho respiratório com 45,02%, os dois primeiros tipos de infecção com maior taxa de óbito são infecções que não são comuns, infecção de olhos, nariz, boca e garganta e infecção do aparelho digestivo representam menos de 1% do total das infecções. Já a infecção respiratória que é a terceira com maior taxa de óbito, é a como maior incidência, 31,7% do total de infecções conforme o gráfico 8.

Gráfico 11 - Comparativo motivo alta pacientes que contraíram infecção



Fonte: Autora da dissertação

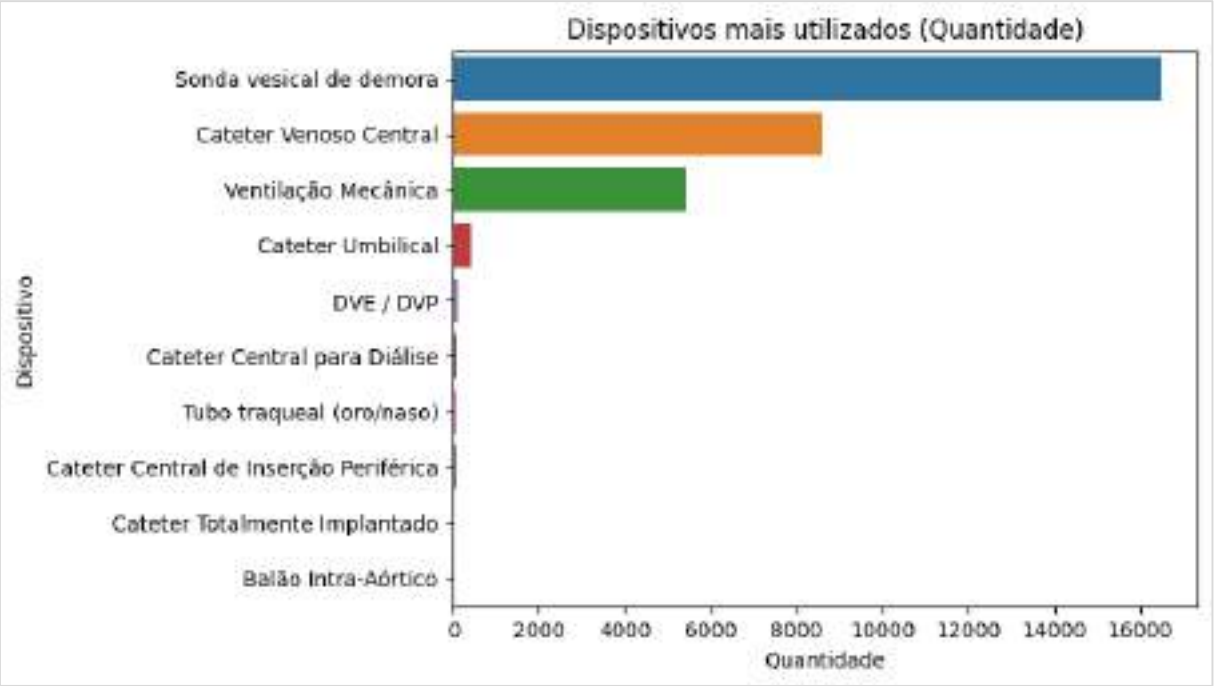
4.3.2 Dados Dispositivos Invasivos

Dispositivos médicos são utilizados no cuidado de saúde para monitorização ou para intervenção em âmbito hospitalar.

A utilização destes dispositivos nem sempre é inócua e implica que os utilizadores conheçam de modo aprofundado as características e indicações de cada um deles e que os utilizem tendo por base tanto uma análise de custo-efetividade como de custo-benefício (PINA et al., 2010).

Estudos apontam a relação entre o uso de dispositivos invasivos e a infecção hospitalar, com isto, esta informação é altamente relevante para este estudo. No gráfico 12 são apresentados os dispositivos mais utilizados em termos de quantidade de inserções; de acordo com o gráfico o dispositivo mais utilizado é a sonda vesical de demora, seguido pelo cateter venoso central e ventilação mecânica.

Gráfico 122 - Gráfico dispositivos mais utilizados (quantidade de inserções)

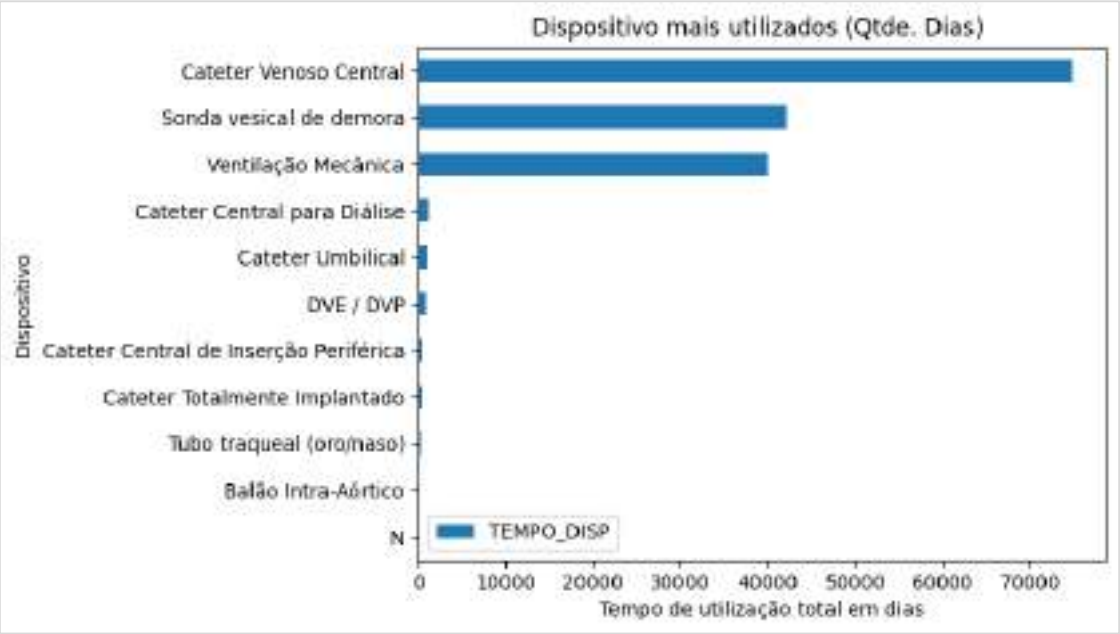


Fonte: Autora da dissertação

Apesar da relevância desta informação, estudos indicam que o tempo de utilização destes dispositivos podem acarretar danos à saúde do paciente, com isto, somente a informação de utilização é insuficiente. Nos dados coletados foram utilizados os campos data da instalação e data da retirada para o cálculo do tempo que o paciente permaneceu com o dispositivo, também foi realizada a soma dos dispositivos para os casos em que o paciente durante o período de internação utilizou o mesmo dispositivo diversas vezes, para este cálculo foi utilizado o tempo em dias.

Após este cálculo observou-se que os dispositivos mais utilizados em termos de dias é o cateter venoso central, em seguida sonda vesical de demora e ventilação mecânica, conforme gráfico 13.

Gráfico 133 - Gráfico dispositivos mais utilizados (quantidade de dias)



Fonte: Autora da dissertação

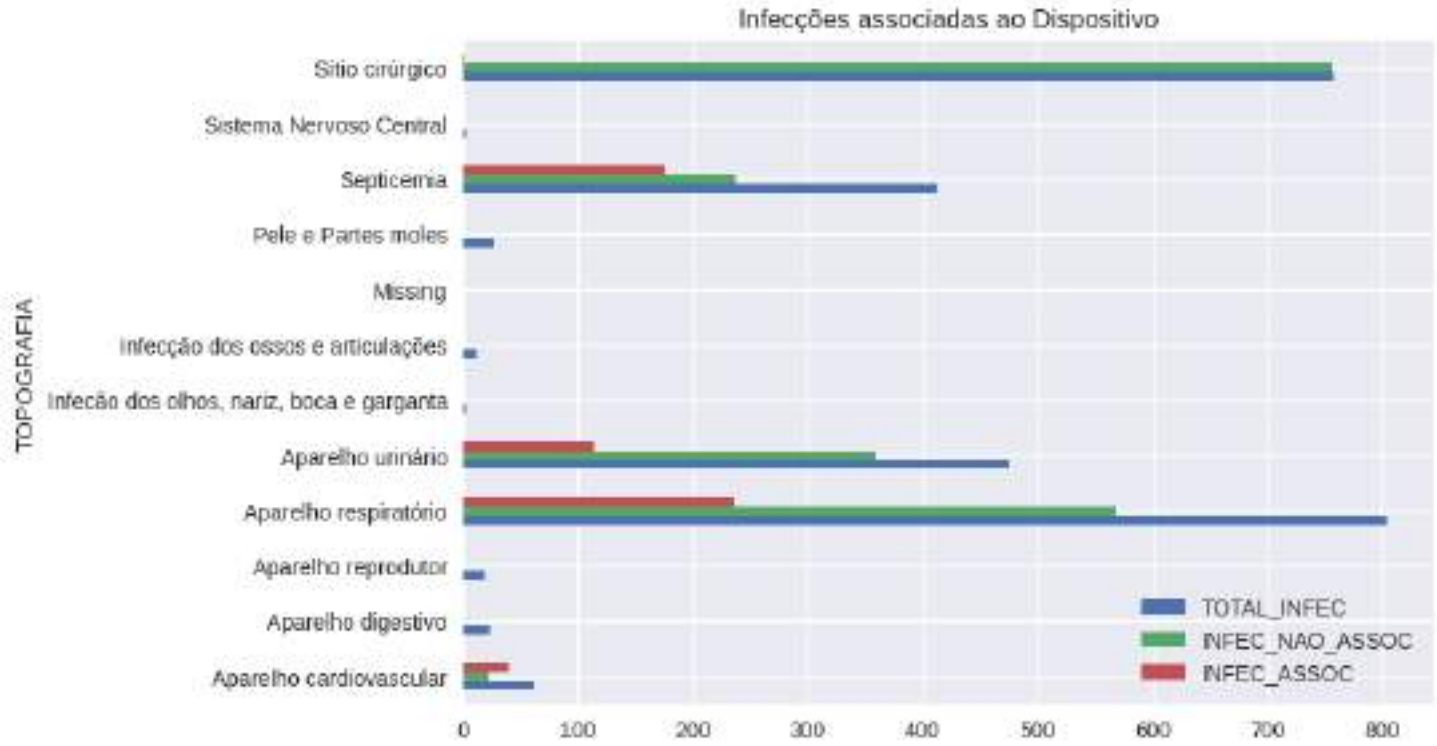
No gráfico 14 é apresentada a correlação entre as variáveis tempo dispositivo, número de dias internados, motivo da alta (Alta e Óbito) e infecção confirmada (S Sim ou N Não). No gráfico podemos observar que para as infecções confirmadas o tempo de dispositivo e o número de dias internados aumenta em relação aos pacientes que não tiveram infecção, o que faz total sentido, pois pacientes que permanecem mais tempo internados ou utilizam dispositivos invasivos por mais tempo, têm maiores chances de contraírem infecção hospitalar.

Gráfico 144 - Correlação entre tempo de dispositivo, dias internados e motivo da alta

Fonte: Autora da dissertação

Enquanto no gráfico 13 foram apresentados os dispositivos mais utilizados em termos de tempo, no gráfico 15 apresentamos as infecções associadas aos dispositivos, na coluna em azul temos o total de infecções de cada topografia, verde as infecções que não foram associadas aos dispositivos e em vermelho as infecções associadas, podemos observar que as topografias do aparelho respiratório e septicemia são as que possuem um maior número de infecções associada a dispositivo.

Gráfico 155 - Infecções associadas ao dispositivo



Fonte: Autora da dissertação

4.4 Estudo e definição do modelo

Para definição do modelo é necessário a análise de alguns fatores, como: complexidade, recursos computacionais, tempo e quantidade de dados, além disto, foram selecionados os algoritmos mais utilizados em trabalhos relacionados com a área de saúde.

Incialmente, foi realizado o treinamento do modelo utilizando seis algoritmos, são eles: Regressão logística, *Random Forest, Adaboost, Bagging*, KNN e XGBoost, o resultado de cada algoritmo será apresentado nas seções seguintes.

4.5 Treinamento

Para efetuar o treinamento dos algoritmos, foi preciso converter as variáveis categóricas em variáveis numéricas. Isso se deve ao fato de que os algoritmos mencionados não conseguem realizar o treinamento utilizando variáveis categóricas na linguagem Python.

Na transformação das variáveis categóricas em variáveis numéricas, os codificadores CatBoost e Target demostraram a mesma performance, para avaliar qual seria mais adequado a este trabalho foi feita a análise da importância das variáveis, mais conhecido como regularização, a regularização pode ser usada para reduzir a quantidade de variáveis em um modelo, assim ter um modelo simples e generalizável, o qual pode prever valores rapidamente e com boa acurácia.

Na regularização foi observado que algumas variáveis foram apontadas em comum para os codificadores Target, Catboost e GLMMEnconder. Na tabela 4 é apresentada todas as variáveis e a importância para cada codificador.

Para este estudo foi utilizado o Target Encoder, a ideia geral deste método é levar em consideração a variável alvo, possui como vantagens: requer esforço mínimo, cria apenas uma coluna para qualquer número de categorias, por fim, é o esquema de codificação mais utilizado na competição Kaggle, a desvantagem é que trabalha apenas para aprendizado supervisionado. Em comparação com o CatBoost não houve diferenças significativas.

Tabela 4 - Análise de importância das variáveis por encoder

Todas as Variáveis	Atributos importantes usando CatBoost	Atributos importantes usando Target Encoder	Atributos importantes usando GLMM
CD_DOENCA_PRINCIPAL_E	CD_DOENCA_PRINCIPAL_E	CD_DOENCA_PRINCIPAL_E	
CIRURGIA		CIRURGIA	CIRURGIA
DS_ACOMODACAO_E	DS_ACOMODACAO_E		DS_ACOMODACAO_E
DS_CARATER_INTERNACAO	DS_CARATER_INTERNACAO	DS_CARATER_INTERNACAO	DS_CARATER_INTERNACAO
DS_CLINICA_E	DS_CLINICA_E		
DS_CONVENIO_E	DS_CONVENIO_E		
DS_DOENCA_PRINCIPAL_E	DS_DOENCA_PRINCIPAL_E	DS_DOENCA_PRINCIPAL_E	
DS_PROC_PRINCIPAL_E	DS_PROC_PRINCIPAL_E	DS_PROC_PRINCIPAL_E	
DS_SETOR_ENTRADA_E	DS_SETOR_ENTRADA_E		
DS_TIPO_ATENDIMENTO			
INFEC_ASSOC_DISP1_E	INFEC_ASSOC_DISP1_E	INFEC_ASSOC_DISP1_E	INFEC_ASSOC_DISP1_E
INFEC_ASSOC_DISP2_E	INFEC_ASSOC_DISP2_E		INFEC_ASSOC_DISP2_E
INFEC_ASSOC_DISP3_E	INFEC_ASSOC_DISP3_E		INFEC_ASSOC_DISP3_E
INFEC_CABECA			
INFEC_CARDIO			
INFEC_CIRURGICO	INFEC_CIRURGICO	INFEC_CIRURGICO	INFEC_CIRURGICO
INFEC_DIGESTIVO			
INFEC_OSSOS_ARTIC			
INFEC_PELE_PARTES_MOLES			
INFEC_REPRODUTOR			
INFEC_RESPIRATORIO	INFEC_RESPIRATORIO	INFEC_RESPIRATORIO	INFEC_RESPIRATORIO
INFEC_SEPSE		INFEC_SEPSE	
INFEC_SIST_NERVOSO			
INFEC_URINARIO	INFEC_URINARIO	INFEC_URINARIO	INFEC_URINARIO
MAX_FREQ_CARDIACA		MAX_FREQ_CARDIACA	MAX_FREQ_CARDIACA
MAX_FREQ_RESP	MAX_FREQ_RESP	MAX_FREQ_RESP	MAX_FREQ_RESP
MAX_PA_DIASTOLICA	MAX_PA_DIASTOLICA	MAX_PA_DIASTOLICA	MAX_PA_DIASTOLICA
MAX_PA_SISTOLICA		MAX_PA_SISTOLICA	MAX_PA_SISTOLICA
MAX_QT_TEMP	MAX_QT_TEMP	MAX_QT_TEMP	MAX_QT_TEMP
MIN_FREQ_CARDIACA	MIN_FREQ_RESP	MIN_FREQ_RESP	
MIN_FREQ_RESP		MIN_PA_DIASTOLICA	MIN_FREQ_RESP
MIN_PA_DIASTOLICA		MIN_PA_SISTOLICA	MIN_PA_DIASTOLICA
MIN_PA_SISTOLICA	MIN_PA_SISTOLICA		MIN_PA_SISTOLICA
MIN_QT_TEMP		MIN_QT_TEMP	MIN_QT_TEMP
MOTIVO_ALTA		MOTIVO_ALTA	
NM_DISPOSITIVO1_E	NM_DISPOSITIVO1_E	NM_DISPOSITIVO1_E	NM_DISPOSITIVO1_E
NM_DISPOSITIVO2_E	NM_DISPOSITIVO2_E		NM_DISPOSITIVO2_E
NM_DISPOSITIVO3_E	NM_DISPOSITIVO3_E		NM_DISPOSITIVO3_E
NR_ANOS			
NR_DIA_INTERNADO	NR_DIA_INTERNADO	NR_DIA_INTERNADO	NR_DIA_INTERNADO
QT_DIAS_BALAO_INTRA_AORTICO			
QT_DIAS_CATETER_CENTRAL_PERIF			
QT_DIAS_CATETER_DIALISE			
QT_DIAS_CATETER_TOT_IMPLANTADO			
QT_DIAS_CATETER_UMBILICAL			
QT_DIAS_CATETER_VENOSO_CENTRAL	QT_DIAS_CATETER_VENOSO_CENTRAL	QT_DIAS_CATETER_VENOSO_CENTRAL	QT_DIAS_CATETER_VENOS
QT_DIAS_DVE_DVP			
QT_DIAS_SONDA_VESICAL	QT_DIAS_SONDA_VESICAL	QT_DIAS_SONDA_VESICAL	QT_DIAS_SONDA_VESICAL
QT_DIAS_TUBO_TRAQUEAL			
QT_DIAS_VENTILACAO_MECANICA	QT_DIAS_VENTILACAO_MECANICA	QT_DIAS_VENTILACAO_MECANICA	QT_DIAS_VENTILACAO_MEC
SETOR_ORIGEM_INFEC_E	SETOR_ORIGEM_INFEC_E	SETOR_ORIGEM_INFEC_E	SETOR_ORIGEM_INFEC_E
SEXO		SEXO	SEXO
SUSPEITO_INFEC			
TEMPO_CIRURG	SUSPEITO_INFEC	SUSPEITO_INFEC	SUSPEITO_INFEC
TEMPO_INFEC	TEMPO_INFEC	TEMPO_INFEC	TEMPO_INFEC

Fonte: Autora da dissertação

Durante a fase de treinamento, outro aspecto abordado foi o desbalanceamento presente na base de dados. Com a disparidade nos registros, em que 98,73% correspondiam a pacientes sem infecção hospitalar e apenas 1,27% correspondiam a pacientes com infecção hospitalar, foi necessário recorrer à técnica de sobreamostragem SMOTE. Para aplicar essa técnica, foram utilizados os parâmetros padrão, nos quais o SMOTE gera exemplos sintéticos da classe minoritária, buscando balancear o conjunto de dados. Ele cria novas amostras sintéticas por meio de interpolação entre amostras existentes da classe minoritária. Primeiro ele identifica a classe minoritária a qual

necessita de sobreamostragem, depois para cada amostra da classe minoritária é identificado seus k vizinhos mais próximos da mesma classe. Por último, é gerada as amostras sintética, onde para cada amostra da classe minoritária, seleciona aleatoriamente um de seus k vizinhos mais próximos. Em seguida, é criada uma amostra sintética combinando características da amostra selecionada e da amostra original. As amostras sintéticas são criadas ao longo do segmento de linha que conecta as duas amostras. Depois de aplicada a técnica, os registros se igualaram para os rótulos 0 e 1 conforme figura 13.

Figura 13 - Aplicação técnica SMOTE

Fonte: Autora da dissertação

Os resultados do treinamento e da avaliação são apresentados no tópico seguinte.

4.6 Avaliação dos modelos

Na avaliação do modelo foram utilizadas as métricas: matriz de confusão, acurácia, precisão, recall (sensibilidade), f1-score, média da validação cruzada e curva AUROC.

A seguir são apresentados os resultados do desempenho dos algoritmos utilizando o hiperparâmetro padrão de acordo com as métricas citadas:

Regressão Logística é o classificador mais popular em aprendizado máquina, obtivemos 99,77% de precisão e *recall* de 92,21%, a acurácia foi de 92,10%. A tabela 5 apresenta mais detalhes.

Tabela 5 - Classificador Regressão Logística

Métrica	Hiperparâmetro Padrão
Matriz de confusão	[32498] [2744] [76] [379]
Acurácia	92,10%
Precisão	99,77%
Recall	92,21%
F1-Score	95,84%
Validação Cruzada (média)	88,22%

Fonte: Autora da dissertação

O princípio central do *adaBoost* é ajustar uma sequência de aprendizes fracos (ou seja, modelos que são apenas um pouco melhores do que suposições aleatórias, como pequenas árvores de decisão) em versões repetidamente modificadas dos dados. O *adaBoost* obteve 99,79% de precisão e 89,56% de *recall* e 89,50% de acurácia.

Tabela 6 - Classificador AdaBoost

Métrica	Hiperparâmetro Padrão
Matriz de confusão	[31563] [3679] [65] [390]
Acurácia	89,50%
Precisão	99,79%
Recall	89,56%
F1-Score	94,40%
Validação Cruzada (média)	98,72%

Fonte: Autora da dissertação

Um classificador *bagging* é um meta-estimador de conjunto que ajusta os classificadores base em subconjuntos aleatórios do conjunto de dados original e, em seguida, agrega suas previsões individuais (por votação ou por média) para formar uma predição final. *Bagging* apresentou 99,27% de precisão, 99,13% de *recall* e 98,40% de acurácia. A tabela 7 apresenta mais detalhes.

Tabela 7 - Classificador Bagging

Métrica	**Hiperparâmetro Padrão**
Matriz de confusão	[34937] [305] [257] [198]
Acurácia	98,40%
Precisão	99,27%
Recall	99,13%
F1-Score	99,20%
Validação Cruzada (média)	99,16%

Fonte: Autora da dissertação

No algoritmo *random forest*, a 'floresta' é construída com árvores de decisão pelo método ensemble. Este algoritmo usa o método de ensacamento para treinamento em dados. *random forest* obteve 99,36% de precisão neste conjunto de dados com *recall* de 99,18% e 98,60% de acurácia. Este é até agora foi o modelo que possui a melhor acurácia para este conjunto de dados. A Tabela 8 tem todas as outras avaliações.

Tabela 8 - Classificador Random Forest (RF)

Métrica	**Hiperparâmetro Padrão**
Matriz de confusão	[34954] [288] [226] [229]
Acurácia	98,60%
Precisão	99,36%

Recall	99,18%
F1-Score	99,27%
Validação Cruzada	99,37%

Fonte: Autora da dissertação

O algoritmo *K Nearest Neighbor* (KNN) apesar de sua simplicidade, obteve bons resultados em um grande número de problemas de classificação e regressão, incluindo dígitos manuscritos e cenas de imagens de satélite. Sendo um método não paramétrico, muitas vezes tem um bom desempenho em situações de classificação onde o limite de decisão é muito irregular, neste projeto o KNN apresentou precisão de 99,38% e *recall* de 94,78%, a acurácia foi de 94,30%.

Tabela 9 - Classificador KNN

Métrica	**Hiperparâmetro Padrão**
Matriz de confusão	[33401] [1841] [208] [247]
Acurácia	94,30%
Precisão	99,38%
Recall	94,78%
F1-Score	97,03%
Validação Cruzada (média)	97,02%

Fonte: Autora da dissertação

O XGBoost trabalha com uma variedade de tipos de dados, e o grande número de hiperparâmetros que podem ser aperfeiçoados e ajustados para um cenário mais apropriado traz mais flexibilidade, isso faz do XGBoost uma escolha consistente para problemas de regressão e classificação. O XGBoost apresentou precisão de 96,59%, *recall* de 77,15% e acurácia de 96,80%.

Tabela 10 - Classificador XGBoost

Métrica	Hiperparâmetro Padrão
Matriz de confusão	[34227] [1015] [121] [334]
Acurácia	96,80%
Precisão	96,59%
Recall	77,15%
F1-Score	85,78%
Validação Cruzada (média)	97,70%

Fonte: Autora da dissertação

Para facilitar a avaliação dos classificadores, foi gerada a tabela 11 com um comparativo da acurácia dos seis modelos, o *random forest* foi o algoritmo que obteve a melhor acurácia com 98,60%, na sequência *bagging* com acurácia de 98,40% e XGBoost com 96,80%.

Tabela 11 - Acurácia dos seis modelos

Modelos	Valor
Random Forest	**98,60%**
Bagging	**98,40%**
XGBoost	**96,80%**
KNN	**94,30%**
Regressão Logística	**92,10%**
Adaboost	**89,50%**

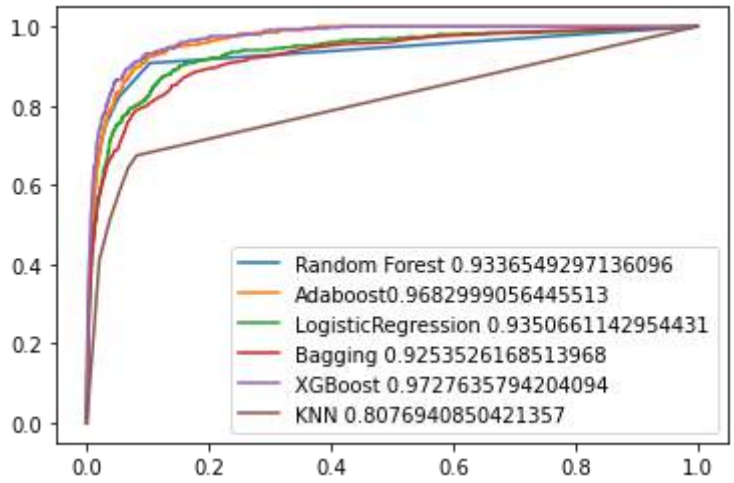
Fonte: Autora da dissertação

As curvas AUC e ROC estão entre as métricas mais utilizadas para a avaliação de um modelo de aprendizado de máquina. Neste trabalho foi utilizada a função *roc_curve* que é um gráfico que ilustra o desempenho de um sistema classificador binário à medida que seu limite de discriminação é variado. E a função *predict_proba*, de acordo com a documentação do *scikitlearn*, as probabilidades de classe previstas de uma amostra de entrada são calculadas

como a média ponderada das probabilidades de classe previstas dos classificadores no conjunto.

No gráfico 16 é apresentada a Curva ROC, nela o algoritmo o XGBoost apresenta o melhor resultado com o valor de 97,27%, em seguida o *adaboost* com 96,82%, regressão logística com 93,50% e em quarto lugar *random forest* com 93,36%.

Gráfico 166 - Curva AUC e ROC



Fonte: Autora da dissertação

4.7 Aprimoramento dos hiperparâmetros

Para o aprimoramento dos hiperparâmetros foi utilizado o *GridSearchCV* que é uma classe fornecida pelo framework *scikit-learn* para ajuste de parâmetros que é implementado por estimadores. As Instruções geralmente definem um dicionário para armazenar os parâmetros que precisam ser pesquisados primeiro, e o *GridSearchCV* realizará todos os ajustes de modelo necessários e produzirá os melhores parâmetros. Ajustar o objeto *GridSearchCV* não apenas procura os melhores parâmetros, mas também obtém um novo modelo de treinamento ajustado automaticamente dos melhores parâmetros de desempenho de validação cruzada em todos os conjuntos de treinamento. Os melhores parâmetros são armazenados no atributo *best_params_* e a melhor precisão para validação cruzada é armazenada no atributo *best_score_* (SHUAI; ZHENG; HUANG, 2018).

Abaixo é apresentada a utilização da função *GridSearchCV* e a avaliação do modelo utilizando os hiperparâmetros de acordo com a indicação da função, esta análise foi feita para cada algoritmo trabalhado neste projeto.

O algoritmo *AdaBoost Classifier* possui 5 hiperparâmetros, os mais relevantes são: *base_estimator, n_estimators, e learning_rate*.

- *base_estimator*: O estimador base é o algoritmo utilizado para treinar os modelos. O padrão é a árvore de decisão com a profundidade da arvore igual a 1.
- *n_estimators*: É o número máximo de estimadores de modelos que será utilizado para o treino. O valor padrão é 50.
- *learning_rate*: É a taxa de aprendizado, ou seja, o peso aplicado a cada classificador em cada iteração de reforço. Uma taxa de aprendizado mais alta aumenta a contribuição de cada classificador. Seu valor padrão é igual 1.

Inicialmente os hiperparâmetros foram passados de acordo com o que é sinalizado como mais relevante na documentação do algoritmo, foram utilizados os hiperparâmetros *n_estimators* igual a 5 e *learning_rate* igual 0.1, o resultado foi apresentado na seção 4.6 avaliação dos modelos.

Com o *GridSearchCV* foram dados 3 valores *n_estimators* e 3 valores para *learning_rate*, com isto, haverá 9 combinações possíveis.

Figura 44 - Hiperparâmetros AdaBoost passados para o GridSearchCV

```
#Grid Search
parametros = {'n_estimators': [1, 5, 10],
              'learning_rate': [0.1, 1, 2]}
```

Fonte: Autora da dissertação

Após a criação do objeto e treinamento do grid, utilizamos o atributo *best_params_* para obter os melhores hiperparâmetros, o atributo retornou que o melhor resultado é produzido com *n_estimators* igual a 10 e *learning_rate* igual a 1.

Figura 55 - AdaBoost - Melhores resultados usando o atributo best_score

Fonte: Autora da dissertação

Após a etapa de análise do resultado da função *GridSearchCV*, foi realizado o treinamento do modelo com *n_estimators* igual a 10 e *learning_rate* igual a 1, podemos observar que houve melhora na maioria das métricas, a acuraria passou de 89,50% para 93,10%, na métrica *recall* tivemos uma aumento de 89,56% para 93,32%, já na validação cruzada tivemos uma perda de 0,05%. A tabela 12 demonstra a comparação utilizando os hiperparâmetros originais (*n_estimators* = 5 e *learning_rate* = 0.1) e com o aprimoramento dos hiperparâmetros (*n_estimators* = 10 e *learning_rate* = 1).

Tabela 12 - Adaboost - Comparação das métricas utilizando aprimoramento dos hiperparâmetros

Métrica	Hiperparâmetro Padrão	Aprimoramento dos Hiperparâmetros
Matriz de confusão	[31563] [3679] [65] [390]	[32888] [2354] [113] [342]
Acurácia	89,50%	93,10%
Precisão	99,79%	98,97%
Recall	89,56%	93,32%
F1-Score	94,40%	96,06%
Validação Cruzada (média)	98,72%	98,67%

Fonte: Autora da dissertação

O algoritmo *Bagging Classifier* possui 11 hiperparâmetros, neste trabalho foram utilizados os hiperparâmetros padrões, para aprimoramento dos hiperparâmetros foram utilizados três, sendo eles, *n_estimators*, *max_samples* e *max_features*.

- *n_estimators*: É o número de estimadores de base no conjunto que devem ser criados. O número de estimadores deve ser analisado, já que um número muito grande pode demorar muito para executar e um número muito pequeno pode não levar a um resultado satisfatório.
- *max_samples*: Este parâmetro controla o número máximo de amostras para treinar cada estimador base.
- *max_features*: Este parâmetro controla o número máximo de colunas para treinar cada estimador base.

Na função *GridSearch* utilizamos 3 valores para os 3 hiperparâmetros, *max_samples*, *max_features* e *n_estimators*, isto gerou 27 combinações possíveis.

Figura 66 - Hiperparâmetros Bagging passados para o GridSearchCV

```
#Grid Search
parametros = {'max_samples': [0.1, 0.5, 1.2],   #padrão 1
              'max_features': [0.1, 0.5, 1.2],  #padrão 1
              'n_estimators': [5, 8, 12]}       #padrão 10
```

Fonte: Autora da dissertação

Com um número maior de combinações foi possível observar um tempo superior de execução do treinamento do grid, comparando com a execução de 9 combinações, este teve um tempo médio de 1 minuto, enquanto o último 20 minutos de execução.

Para identificar os melhores hiperparâmetros foi utilizado o atributo *best_params_,* com ele obteve-se como resultado *max_samples* igual a 0.5, *max_features* igual 0.5 e *n_estimators* igual 12.

Figura 77 - Bagging - Melhores resultados usando o atributo best_score

```
# Imprime os parâmetros que produziram o ",best_score_",
grid.best_params_

{'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 12}
```

Fonte: Autora da dissertação

Após a etapa de análise do resultado da função *GridSearchCV*, foi realizado o treinamento do modelo com o resultado *max_samples* igual a 0.5, *max_features* igual 0.5 e *n_estimators* igual 12, podemos observar que houve melhora em todas as métricas, sendo que a acurácia apresentou uma diferença maior, sendo de 0,20%.

A tabela 13 demonstra a comparação utilizando os hiperparâmetros padrões (*max_samples* = 1, *max_features* = 1 e *n_estimators* = 10) e com o aprimoramento dos hiperparâmetros (*max_samples* = 0.5*, max_features* = 0.5 e *n_estimators* = 12).

Tabela 13 - Bagging - Comparação das métricas utilizando aprimoramento dos hiperparâmetros

Métrica	Hiperparâmetro Padrão	Aprimoramento dos Hiperparâmetros
Matriz de confusão	[34937] [305] [257] [198]	[34983] [259] [246] [209]
Acurácia	98,40%	98,60%
Precisão	99,27%	99,30%
Recall	99,13%	99,27%
F1-Score	99,20%	99,28%
Validação Cruzada (média)	99,16%	99,24%

Fonte: Autora da dissertação

O algoritmo XGBoost, possui 22 hiperparâmetros, para aprimoramento dos hiperparâmetros foram utilizados dois, sendo eles, *max_depth* e *min_child_weight*. Os hiperparâmetros citados têm um impacto significativo na complexidade do modelo, portanto é necessário regula-los em conjunto para garantir que não haja sobreajuste.

O hiperparâmetro *objective* foi utilizado com o valor *binary:logistic* e não passou pelo aprimoramento, pois ele é selecionado de acordo com a classificação do problema, no caso deste trabalho temos um problema de classificação binária. Abaixo segue detalhamento dos hiperparâmetros que são utilizados para o aprimoramento:

- *Max_depth*: Indica a profundidade máxima da árvore usada pelo modelo, ou seja, a quantidade máxima de nós que podem haver da raiz até uma folha.
- *Min_child_weight*: É a soma mínima do peso da instância necessária em um nó filho, ou quantidade mínima de amostras se todas tiverem peso 1, necessário para cada árvore criar um novo nó. Um valor pequeno vai permitir o algoritmo criar nós que correspondem a uma quantidade menor de amostras, fazendo assim o modelo ficar mais complexo.

Na função *GridSearch* utilizamos 3 valores para os hiperparâmetros, *max_depth*, e *min_child_weight*, isto gerou 9 combinações possíveis, conforme a figura abaixo.

Figura 88 - Hiperparâmetros XGBoost passados para o GridSearchCV

```
#Grid Search
parametros = {'max_depth': [8, 10, 12],        #padrão 3
              'min_child_weight': [1, 5, 10]}  #padrão 1
```

Fonte: Autora da dissertação

Após a criação do objeto e treinamento do grid, o atributo *best_params_* retornou que o melhor resultado é produzido com *max_depth* igual a 12 e *min_child_weight* igual a 1.

Figura 9 - XGBoost - Melhores resultados usando o atributo best_score

```
# Imprime os parâmetros que produziram o ".best_score_".
grid.best_params_
```

```
{'max_depth': 12, 'min_child_weight': 1}
```

Fonte: Autora da dissertação

O modelo foi treinado com o hiperparâmetro *max_depth* igual a 12 e o hiperparâmetro *min_child_weight* permaneceu igual 1, já que este é o valor padrão. Analisando a tabela 14, pode-se observar que houve melhora em todas as métricas, dentre os ganhos, podemos destacar a *recall* e o f1-score, onde tiveram um ganho de 22,35% e 13,60% respectivamente.

Tabela 14 - XGBoost - Comparação das métricas utilizando aprimoramento dos hiperparâmetros

Métrica	Hiperparâmetro Padrão	Aprimoramento dos Hiperparâmetros
Matriz de confusão	[34227] [1015] [121] [334]	[35065] [177] [257] [198]
Acurácia	96,80%	98,80%
Precisão	96,59%	99,27%
Recall	77,15%	99,50%
F1-Score	85,78%	99,38%
Validação Cruzada (média)	97,70%	99,34%

Fonte: Autora da dissertação

A Regressão Logística possui 15 hiperparâmetros, destes os mais relevantes são: *solver*, *penalty* e C. Abaixo segue detalhamento de cada hiperparâmetro:

- *Solver*: É o algoritmo utilizado no problema de otimização. O padrão é 'lbfgs'. É importante considerar alguns aspectos para escolher os melhores valores, para conjuntos de dados pequenos *liblinear* é uma boa escolha, enquanto 'sag' e 'saga' são mais rápidos para grandes. Por outro lado, lbfgs tem um desempenho relativamente bom em comparação com outros métodos e economiza muita memória, no entanto, às vezes pode ter problemas com a convergência.
- *Penalty*: Este hiperparâmetro visa reduzir o erro de generalização do modelo, e tem como objetivo desencorajar o aprendizado de um modelo mais complexo, para evitar o risco de sobreajuste. As opções são: {'l1', 'l2'*, 'elasticnet', 'none'*}. Os hiperparâmetros solver e *penalty* andam juntos e possuem restrições, conforme a tabela abaixo, com isto, neste trabalho utilizou-se o *penalty* padrão l2 que são suportados tanto pelo solver lgfgs e *liblinear*, desta forma não foi utilizado o parâmetro *penalty* no aprimoramento dos hiperparâmetros.

Tabela 15 - Penalty aceitos no hiperparâmetro solver

solver	penalty
lbfgs	'l2', 'none'
liblinear	'l1', 'l2'

Fonte: Autora da dissertação

- C: Também conhecido como força de regularização, tem como objetivo controlar a força da penalidade (*penalty*), ou seja, a força de regularização funciona com a penalidade de regular o sobreajuste. Valores menores especificam uma regularização mais forte e valores mais altos informam ao modelo para dar peso alto aos dados de treinamento.

Na função *GridSearch* utilizou-se 2 valores para o hiperparâmetro solver e 5 valores para o hiperparâmetro C, conforme figura abaixo.

Figura 20 -Hiperparâmetros da Regressão Logística passados para o GridSearchCV

```
#Grid Search
parametros = {'solver': [ 'lbfgs', 'liblinear'],  #padrão lbfgs
              'C': [100, 10, 1.0, 0.1, 0.01]}     #padrão 1.0
```

Fonte: Autora da dissertação

O atributo *best_params_* foi utilizado para identificar os melhores hiperparâmetros, pode-se observar como resultado C igual a 10 e solver igual a *liblinear*.

Figura 101 - Regressão Logística - Melhores resultados usando o atributo best_score

```
# Imprime os parâmetros que produziram o ".best_score_".
grid.best_params_
```

```
{'C': 10, 'solver': 'liblinear'}
```

Fonte: Autora da dissertação

Novamente o modelo foi treinado com os hiperparâmetros aprimorados, sendo solver com o valor *liblinear* e C com o valor igual a 10, o modelo apresentou melhora em todas as métricas, porém está melhora não foi de grandes percentuais como pode-se observar nos demais algoritmos, a métrica que obteve um percentual maior de ganho foi a validação cruzada com 2,85%.

Tabela 16 - Regressão Logística - Comparação das métricas utilizando aprimoramento dos hiperparâmetros

Métrica	**Hiperparâmetro Padrão**	**Aprimoramento dos Hiperparâmetros**
Matriz de confusão	[32498] [2744] [76] [379]	[32900] [2342] [56] [399]
Acurácia	92,10%	93,30%
Precisão	99,77%	99,83%
Recall	92,21%	93,35%
F1-Score	95,84%	96,48%
Validação Cruzada (média)	88,22%	91,07%

Fonte: Autora da dissertação

O algoritmo *Random Forest* possui 18 hiperparâmetros, com base em estudos e análises realizadas, foi considerado os hiperparâmetros padrões para a primeira execução do algoritmo e para aprimoramento dos hiperparâmetros foram utilizados três, *n_estimator, max_depth* e *min_samples_split*, a utilização de mais hiperparâmetros nesta fase implica em recurso computacional e tempo de execução.

- *n_estimators*: o algoritmo *random forest* é um grupo de muitas árvores de decisão, o parâmetro *n_estimator* controla o número de árvores dentro do classificador. Utilizar muitas árvores para ajustar um modelo ajuda a obter um resultado mais generalizado, porém isso pode aumentar a complexidade de tempo do modelo. O número padrão de estimadores é 100.
- *max_depth*: Este hiperparâmetro estabelece a altura máxima até a qual as árvores dentro da floresta podem crescer. É um dos hiperparâmetros

mais importantes quando se trata de aumentar a precisão do modelo, à medida que aumentamos a profundidade da árvore, a precisão do modelo aumenta até um certo limite, mas então começará a diminuir gradualmente devido ao ajuste excessivo do modelo. É importante definir seu valor de forma adequada para evitar sobreajuste. O valor padrão é definido como Nenhum, isso significa que os nós dentro da árvore continuarão a crescer até que todas as folhas se tornem puras ou todas as folhas contenham menos que *min_samples_split* (outro hiperparâmetro).

- *min_samples_split*: especifica a quantidade mínima de amostras que um nó interno deve conter para se dividir em outros nós. Com um valor muito baixo de *min_samples_splits* a árvore continuará a crescer e poder sofrer de sobreajuste. Ao aumentar o valor de *min_samples_splits*, pode-se diminuir o número total de divisões, limitando assim o número de hiperparâmetros no modelo e, assim, ajudar a reduzir o sobreajuste no modelo. É indicado manter o valor *min_samples_split* entre 2 e 6. O valor padrão é definido como 2.

Na função *GridSearch* utilizamos 2 valores para os 3 hiperparâmetros, *n_estimators, max_depth* e *min_samples_split,* isto gerou 8 combinações possíveis.

Figura 112 - Hiperparâmetros do Random Forest passados para o GridSearchCV

```
#Grid Search
parametros = {'n_estimators': [200, 300],       #padrão 100
              'max_depth': [10, 50],            #padrão  None
              'min_samples_split': [4, 6]}      #padrão 2
```

Fonte: Autora da dissertação

Assim como nos modelos anteriores, foi utilizado o atributo *best_params_* para identificar os melhores hiperparâmetros, com ele obteve-se como resultado *n_estimators* igual a 200, *max_depth* 50 e *min_samples_split* 4.

Figura 23 - Random Forest - Melhores resultados usando o atributo best_score

```
# Imprime os parâmetros que produziram o ".best_score_".
grid.best_params_
```

```
{'max_depth': 50, 'min_samples_split': 4, 'n_estimators': 200}
```

Fonte: Autora da dissertação

O modelo foi treinado com o aprimoramento dos hiperparâmetros, sendo, *max_depth* igual a 50, *min_sample_split* igual a 4 e *n_estimators* igual a 200. Analisando a tabela 17 pode-se observar que houve pouca mudança em relação a utilização de hiperparâmetros padrão, isso por que o resultado já estava muito bom considerando uma acurácia de 98,60, essa permaneceu a mesma.

Tabela 17 - Random Forest - Comparação das métricas utilizando aprimoramento dos hiperparâmetros

Métrica	**Hiperparâmetro Padrão**	**Aprimoramento dos Hiperparâmetros**
Matriz de confusão	[34954] [288] [226] [229]	[34961] [281] [227] [228]
Acurácia	98,60%	98,60%
Precisão	99,36%	99,35%
Recall	99,18%	99,20%
F1-Score	99,27%	99,27%
Validação Cruzada	99,37%	99,34%

Fonte: Autora da dissertação

O algoritmo *KNeighbors Classifier*, também conhecido como KNN, possui 8 hiperparâmetros, para o aprimoramento dos hiperparâmetros foram utilizados *n_neighbors* e *metric*, estes dois hiperparâmetros são pontos chaves que determinam a métrica de distância e o valor k de vizinhos.

- *n_neighbors*: é o número de vizinhos mais próximos. O número de vizinhos é o principal fator decisivo. K geralmente é um número ímpar se o número de classes for 2, mas o valor mais adequado varia de acordo com a base de dados.

- *metric*: parâmetro utilizado para calcular distâncias, 3 métricas de distância que são frequentemente usadas são Distância Euclidiana , Distância de Manhattan e Distância de Minkowski. A distância euclidiana é a padrão do *sckitlearn* e ela é o comprimento do segmento de linha que liga dois pontos.

Na *GridSearch* utilizou-se 3 valores para os hiperparâmetros, *n_neighbors*, e *metric*, isto gerou 8 combinações possíveis.

Figura 24 - Hiperparâmetros KNN passados para o GridSearchCV

```
#Grid Search
parametros = {'n_neighbors': [3, 5, 7],                    #padrão 5
              'metric': ['euclidean', 'manhattan', 'minkowski']}  #padrão minkowski
```

Fonte: Autora da dissertação

O atributo *best_params_* foi utilizado para identificar os melhores hiperparâmetros, pode-se observar como resultado *n_neighbors* igual a 3 e *metric* igual a manhattan.

Figura 25 - KNN - Melhores resultados usando o atributo best_score

```
# Imprime os parâmetros que produziram o ".best_score_".
grid.best_params_
```

```
{'metric': 'manhattan', 'n_neighbors': 3}
```

Fonte: Autora da dissertação

No primeiro treinamento, o modelo foi treinado com os hiperparâmetro padrão (*n_neighbors* = 5 e *metric* = minkowski), no segundo, o modelo foi treinado com os hiperparâmetros aprimorados (*n_neighbors* = 3 e *metric* = manhattan), o resultado é apresentado na tabela 18. Observa-se que houve melhora em praticamente todas as métricas, exceto na precisão onde houve uma perda 0,07%, a métrica que obteve o melhor ganho foi a *recall* com ganho de 2,02%.

Tabela 18 - KNeighbors - Comparação das métricas utilizando aprimoramento dos hiperparâmetros

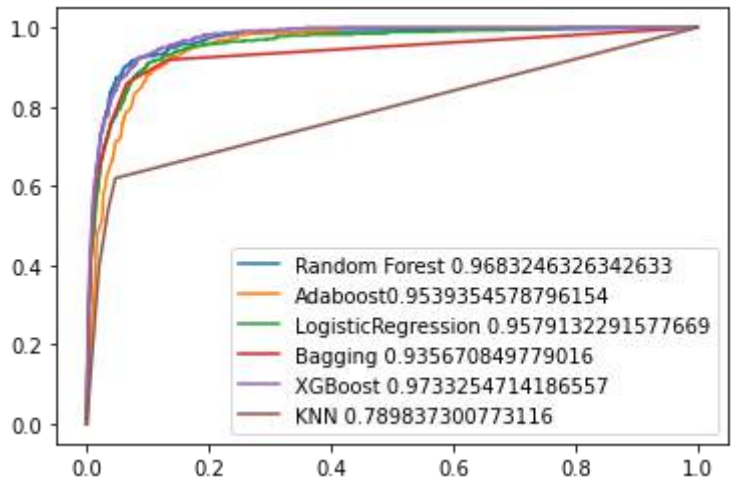
Métrica	Hiperparâmetro Padrão	Aprimoramento dos Hiperparâmetros
Matriz de confusão	[33401] [1841] [208] [247]	[34112 [1130] [238] [217]
Acurácia	94,30%	96,20%
Precisão	99,38%	99,31%
Recall	94,78%	96,79%
F1-Score	97,03%	98,03%
Validação Cruzada (média)	97,02%	98,10%

Fonte: Autora da dissertação

Por fim, foi utilizada a métrica AUROC para todos os algoritmos com o aprimoramento dos hiperparâmetros, apenas os algoritmos *Adaboost* e KNN tiveram uma piora em relação a essa métrica, sendo uma diferença de -1,43% e -1,78% respectivamente.

O algoritmo que apresentou uma maior diferença positiva foi o *Random Forest*, inicialmente *Random Forest* apresentou um resultado de 93,36% e com o aprimoramento dos hiperparâmetros foi para 96,83%, um aumento de 3,47%. Contudo, o algoritmo que apresentou o melhor resultado na métrica AUROC foi o XGBoost com 97,33%. O gráfico 17 apresenta a curva AUROC para os seis algoritmos deste estudo após o treinamento dos mesmos com o aprimoramento dos parâmetros.

Gráfico 177 - Curva AUROC com aprimoramento dos parâmetros



Fonte: Autora da dissertação

4.8 Análise dos resultados

Para avaliar o modelo preditivo de infecção hospitalar, foram utilizados sete métricas de desempenho: matriz de confusão, acurácia, precisão, *recall (*sensibilidade), F1-score, validação cruzada e AUROC.

Os dados foram particionados em 80% para treinamento e 20% para teste. O conjunto de dados de treinamento inclui 142.784 pontos de dados (1.819 com IH e 140.965 sem IH). O conjunto de dados de teste inclui 35.697 pontos de dados (455 com IH e 35.242 sem IH).

No treinamento do modelo que foi realizado inicialmente com o hiperparâmetro padrão de cada algoritmo, pôde-se observar que em termos de acurácia, *random forest* e *bagging* tiveram os melhores resultados, 98,60% e 98,40% respectivamente, já na métrica AUROC, o XGBoost e *adaboost* apresentaram os melhores resultados, com 97,27% e 96,82%, respectivamente. Entretanto, quando utilizado o aprimoramento dos hiperparâmetros, o cenário muda com ganhos significativos em cada modelo, o modelo que apresenta a maior acurácia é o XGBoost (98,80%) seguida pelo *random forest* (98,60%). A curva AUROC também apresentou bons resultados, os melhores resultados foram para o XGBoost (97,33%) e *random forest* (96,83%). Este fato ocorre para 6 das 7 métricas apresentadas neste trabalho, ou seja, em 6 métricas o XGBoost apresenta o melhor resultado e o *radom forest* é o segundo que apresenta o

melhor resultado nessas mesmas 6 métricas, somente na métrica de *recall* o XGBoost e o *random forest* não apresentam os melhores resultados.

4.9 Explicabilidade do modelo

O aprendizado de máquina está revolucionando a saúde e a medicina, os avanços na tecnologia trazem muitas responsabilidades e restrições, com isto, um dos principais desafios em relação ao AM é a falta de explicação suficiente compreensível pelas partes interessadas em relação ao seu diagnóstico e terapêutica, além disto, existe uma preocupação com as questões éticas e de transparência.

Ao contrário dos modelos analíticos clássicos construídos com análise estatística tradicional, os modelos complexos construídos usando métodos de AM podem ser mais difíceis de explicar e justificar para os seres humanos. Como tal, há uma área crescente de pesquisa sobre a explicabilidade dos sistemas de AM, conhecida como Inteligência Artificial Explicável (XAI). O XAI visa fornecer justificativa, transparência e rastreabilidade do aprendizado de máquina, bem como testabilidade de suposições causais. O mecanismo de caixa preta nos métodos de AM muitas vezes torna difícil para pessoas de fora entenderem completamente o algoritmo e identificar possíveis vieses (SHABAN-NEJAD; MICHALOWSKI; BUCKERIDGE, 2021).

A qualidade de um sistema XAI pode ser avaliada usando várias métricas para examinar a qualidade das explicações, satisfação dos usuários, compreensibilidade, confiabilidade e examinar o desempenho do sistema (SHABAN-NEJAD; MICHALOWSKI; BUCKERIDGE, 2021).

O XAI ajuda a fornecer confiança através das seguintes propriedades:

- Maior transparência: como os métodos XAI explicam porque um sistema de AM chegou a uma decisão específica, ele aumenta a transparência na maneira como os sistemas de AM operam e pode levar a níveis maiores de confiança.
- Rastreamento de resultados: As explicações geradas pelos métodos XAI podem ser usadas para rastrear os fatores que afetaram o sistema de AM para prever um resultado.

- Melhoria do modelo: os sistemas de AM aprendem com os dados para fazer uma predição. Às vezes, as regras aprendidas são errôneas e podem levar a previsões errôneas. As explicações geradas a partir dos métodos XAI podem auxiliar no entendimento das regras aprendidas para que os erros possam ser identificados nelas e os modelos possam ser melhorados.

4.9.1 Ferramentas para explicabilidade do modelo

As duas ferramentas mais conhecidas e utilizadas para a explicabilidade dos modelos de aprendizado de máquina são o LIME (*Locally Interpretable Modelagnostic Explanations*) e o SHAP (*Shapley Additive Explanations*). Neste trabalho foi escolhido utilizar a biblioteca SHAP, pois ele garante algumas propriedades desejadas para este trabalho.

A biblioteca SHAP (*Shapley Additive Explanations*) é uma abordagem unificada baseada em Python para explicar a saída de qualquer modelo de aprendizado de máquina. A biblioteca SHAP é baseada na teoria dos jogos com explicações locais. A abordagem da teoria dos jogos é uma maneira de obter previsões se um fator estiver presente ou ausente. Se houver uma mudança significativa no resultado esperado, o fator é muito importante para a variável de destino. Este método une vários métodos anteriores para explicar a saída gerada pelos modelos de aprendizado de máquina. A estrutura SHAP pode ser usada para diferentes tipos de modelos, exceto para modelos baseados em séries temporais. (MISHRA, 2022)

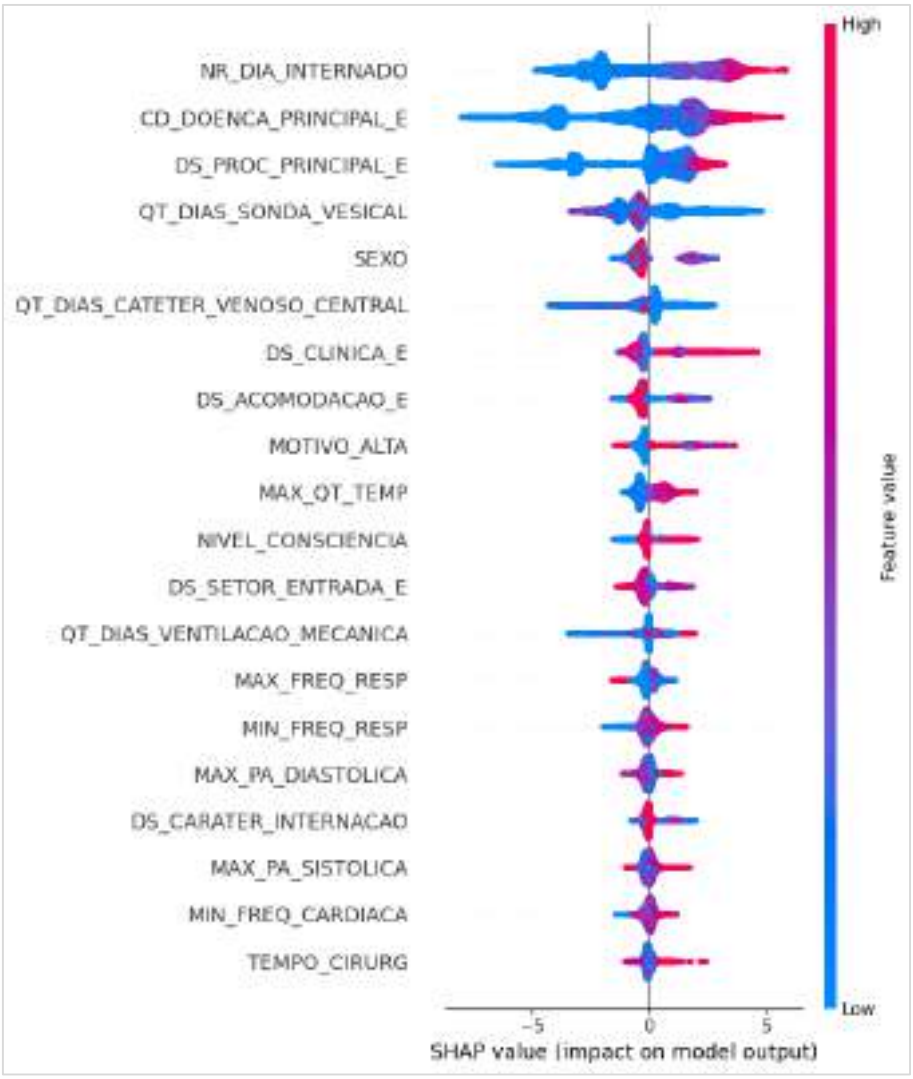
A biblioteca SHAP tem como vantagem ser agnóstica ao tipo de modelo, porém é possível obter ainda maior eficiência no cálculo quando a explicação é restrita para certos tipos de modelo. Isso foi feito por exemplo com o *Tree SHAP*, que é específico para modelos baseados em árvore de decisão, o *Deep SHAP*, que é específico para modelos de *Deep Learning*, e o Linear SHAP, que é específico para modelos lineares.

4.9.2 Avaliação do modelo utilizando SHAP

Utilizando o conjunto de dados de treinamento, foi utilizado SHAP para avaliar como o valor de cada variável influenciou no resultado alcançado pelo modelo preditivo XGBoost, que foi o modelo que obteve as melhores métricas.

Abaixo é apresentado o gráfico de resumo que tem como objetivo mostrar os recursos mais importantes e o impacto de cada recurso no modelo. De acordo com o gráfico, o atributo mais importante é o NR_DIA_INTERNADO que indica o tempo que o paciente ficou internado, a variável tem um impacto alto e positivo na classificação de qualidade. O “alto” vem da cor vermelha e o impacto “positivo” é mostrado no eixo X.

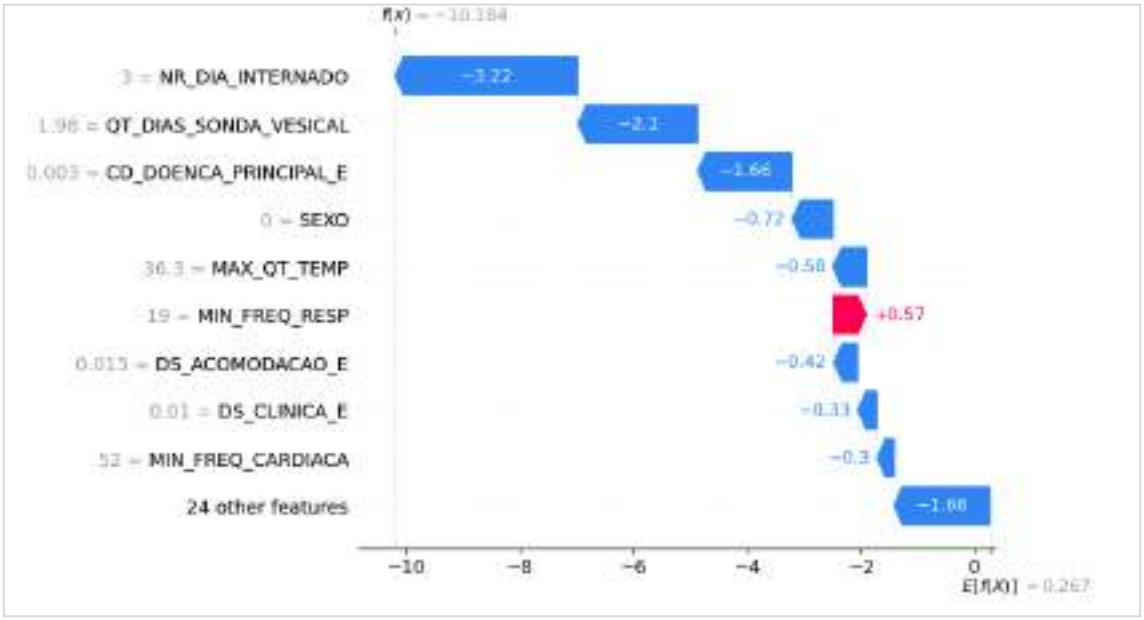
Gráfico 188 - Gráfico de resumo XGBoost



Fonte: Autora da dissertação

O gráfico em cascata mostra de forma individual o motivo de certo registro receber aquela predição de acordo com os valores das variáveis. A parte inferior do gráfico em cascata começa como o valor esperado da saída do modelo e, em seguida, cada linha mostra como a contribuição positiva (vermelha) ou negativa (azul) de cada variável move o valor da saída esperada do modelo sobre o conjunto de dados de fundo para a saída do modelo para esta predição. O texto cinza antes dos nomes dos recursos mostra o valor de cada recurso para esta amostra. Na última linha pode-se observar que os 24 recursos menos impactantes foram reunidos em um único termo para não mostrar mais de 10 linhas no gráfico.

Gráfico 199 - Gráfico de cascata

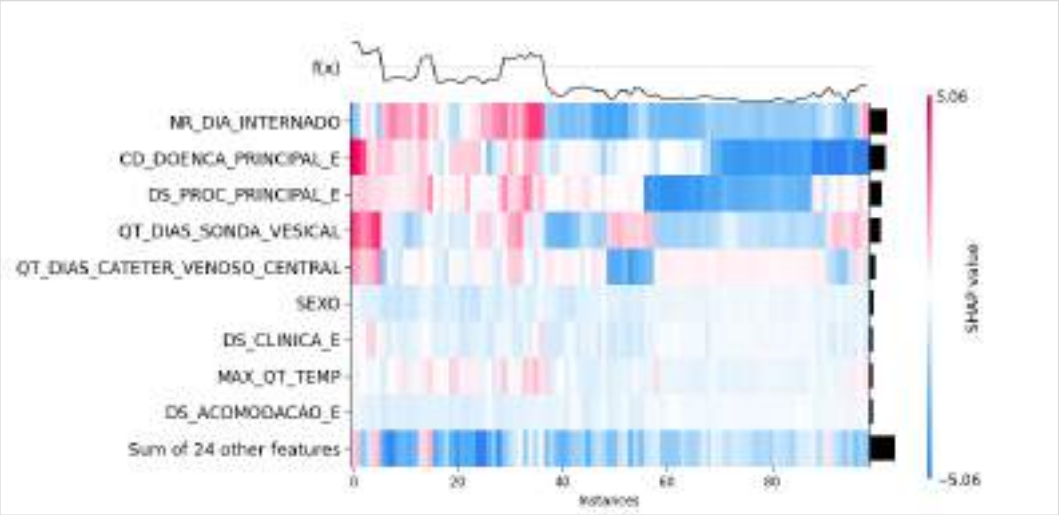


Fonte: Autora da dissertação

O gráfico de calor utiliza uma matriz de valores SHAP que cria um gráfico com as instâncias no eixo x, as entradas do modelo no eixo y e os valores SHAP codificados em uma escala de cores. Por padrão, as amostras são ordenadas com base em um agrupamento hierárquico por sua similaridade de explicação. Isso resulta em amostras que têm o mesmo resultado do modelo pelo mesmo motivo sendo agrupadas

A saída do modelo é mostrada acima da matriz do mapa de calor, e a importância global de cada entrada do modelo é mostrada como um gráfico de barras no lado direito do gráfico.

Gráfico 20 - Gráfico de calor

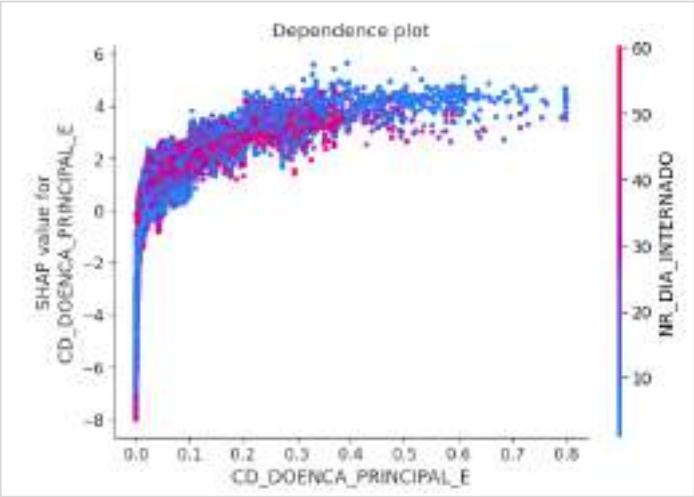


Fonte: Autora da dissertação

Outro gráfico utilizado neste trabalho foi o gráfico de dependência, ele é importante nos resultados do aprendizado de máquina, mostra o efeito marginal que uma ou duas variáveis têm no resultado previsto. Ele informa se a relação entre o alvo e a variável é linear, uniforme ou mais complexa.

O gráfico 21 mostra a relação entre a variável “CD_DOENCA_PRINCIPAL_E” e a variável alvo, e “CD_DOENCA_PRINCIPAL_E” interage com “NR_DIA_ATENDIMENTO”. Podemos observar que a medida que o número de dias de internação aumenta, aumenta também o risco de contrair infecção hospitalar.

Gráfico 21 - Gráfico de dependência



Fonte: Autora da dissertação

O gráfico de força, gráfico 22, tem como objetivo mostrar como os recursos contribuíram para a predição do modelo para uma observação específica. As características que foram importantes para fazer a predição para esta observação são mostradas em vermelho e azul, com o vermelho representando as características que aumentaram a pontuação do modelo e o azul representando recursos que empurraram a pontuação para baixo. Os recursos que tiveram mais impacto na pontuação estão localizados mais perto da fronteira divisória entre vermelho e azul, e o tamanho desse impacto é representado pelo tamanho da barra.

Gráfico 22 - Gráfico de força

Fonte: Autora da dissertação

5 CONCLUSÕES

Este trabalho utilizou métodos de aprendizado de máquina em dados estruturados de uma base de dados hospitalar. Com informações hospitalares de pacientes internados no período de 2015 a 2019, os métodos demonstraram qualidade em definir quais pacientes tem o risco de contrair infecção hospitalar, apesar da base desbalanceada, técnicas de sobreamostragem foram utilizadas para superar este problema.

Os dados foram extraídos a partir da base de dados do hospital, contendo informações de sinais vitais, dados de infecção hospitalar, dispositivos invasivos, e dados demográficos do paciente. Com transformações e a criação de novos atributos no conjunto de dados, a metodologia aplicada visou seguir conceitos clínicos e metodologias da estatística para elaboração de modelos que pudessem ser processados por diversos métodos de Aprendizado de Máquina.

A partir da avaliação dos resultados, foi possível concluir que os modelos foram consistentes e eficazes. O aprimoramento dos hiperparâmetros contribuiu para elevar os resultados e definir o modelo mais adequado para essa estrutura de dados. O modelo XGBoost obteve as melhores métricas de Acurácia (98,80%) e AUROC (97,33%) enquanto o modelo *Random Forest* ficou em segundo lugar com Acurácia (98,60%) e AUROC (96,83%), nas métricas de F1-score e Validação Cruzada os melhores resultados permaneceram com o XGBoost, seguido pelo *Random Forest,* a métrica de Precisão foi exceção, em relação a Precisão o algoritmo Regressão Logística obteve o melhor resultado com 99,83%, seguida pelo *Random Forest* com 99,35%, depois XGBoost com 99,27%. E por último temos a métrica *recall*, que para a área da saúde é de suma importância, já que, uma taxa de falso positivo (classificar pacientes doentes como saudáveis) é muito prejudicial, nesta métrica tivemos como melhor resultado o modelo *XGBoost* com 99,50%, em seguida *Random Forest* com 99,20%.

Apesar dos resultados consistentes, este estudo teve limitações no decorrer do desenvolvimento, dentre elas destacam-se as limitações relacionadas aos dados, os dados coletados não são de domínio público, com isto, foi necessário utilizar os mesmos de acordo com a Lei Geral de Proteção de Dados (LGPD), além disto, dados ausentes, informações de baixa relevância

e dados sem qualquer padrão impactam na performance do algoritmo, o tratamento destas informações foram descritas nas seções coleta de dados e preparação de dados.

Outra limitação deste trabalho foi a aplicação de algoritmos de AM na área da saúde. Uma característica inerente aos dados de saúde refere-se à distribuição desbalanceada das classes de respostas categóricas, ou seja, maior frequência para a classe de indivíduos saudáveis e, por outro lado, uma classe minoritária, representativa de indivíduos doentes. Geralmente, a classe minoritária é a mais importante e também a que mais sofre com classificações erradas, pois os algoritmos, para alcançar melhor desempenho (mais acertos), tendem a priorizar a especificidade em vez do *recall*, o que pode implicar redução do desempenho de modelos preditivos quando aplicado a novos dados.

Para o desfecho deste trabalho seria importante aplicar o modelo a dados que não foram utilizados nas etapas anteriores, nesta etapa o modelo seria efetivamente utilizado para responder as perguntas para as quais foi treinado, porém, devido ao prazo de entrega do trabalho e questões burocráticas para liberação de nova coleta de dados, não foi possível realizar esta última etapa.

Por fim, os algoritmos treinados atingiram resultados satisfatórios, os quais, a partir de melhorias e com validação clínica, estatística e ética, poderão prover informações importantes para adoção de aprendizado de máquina nas rotinas da pesquisa em saúde, acredita-se que este estudo pode servir de alicerce para trabalhos futuros, que poderão complementar e viabilizar a adoção pelo hospital Márcio Cunha ou quaisquer outras instituições de saúde no Brasil.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AL-AHMARI, S.; NADEEM, F. **Machine Learning-Based Predictive Model for Surgical Site Infections: A Framework**. 2021 National Computing Colleges Conference (NCCC). **Anais**... Em: 2021 NATIONAL COMPUTING COLLEGES CONFERENCE (NCCC). mar. 2021.

BASGALL, M. J. et al. **An Analysis of Local and Global Solutions to Address Big Data Imbalanced Classification: A Case Study with SMOTE Preprocessing**. (M. Naiouf, F. Chichizola, E. Rucci, Eds.)Cloud Computing and Big Data. **Anais**...: Communications in Computer and Information Science.Cham: Springer International Publishing, 2019.

BRASIL. Portaria n° 2616 de 13 de maio de 1998. Regulamenta as ações de controle de infecção hospitalar no país. Diário Oficial da República Federativa do Brasil, 15 maio 1998. Seção I.

BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, 1 ago. 1996.

BREIMAN, L. Random Forests. **Machine Learning**, v. 45, n. 1, p. 5–32, 1 out. 2001.

BROWNLEE, J. **Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python**. [s.l.] Machine Learning Mastery, 2020.

CHALLEN, R. et al. Artificial intelligence, bias and clinical safety. **BMJ Quality & Safety**, v. 28, n. 3, p. 231–237, mar. 2019.

CHEN, D. Y. **Análise de dados com Python e Pandas**. [s.l.] Novatec Editora, 2018.

CHEN, T.; GUESTRIN, C. **XGBoost: A Scalable Tree Boosting System**. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. **Anais**... In: THE 22ND ACM SIGKDD INTERNATIONAL CONFERENCE. San Francisco, California, USA: ACM Press, 2016. Disponível em: <http://dl.acm.org/citation.cfm?doid=2939672.2939785>. Acesso em: 25 jun. 2019

CHEN, T.; HE, T. xgboost: eXtreme Gradient Boosting. p. 4, [s.d.].

ESCOVEDO, T.; KOSHIYAMA, A. **Introdução a Data Science: Algoritmos de Machine Learning e métodos de análise**. [s.l.] Casa do Código, 2020.

FILHO, C.; PORTO, A. D. Uso de big data em saúde no Brasil: perspectivas para um futuro próximo. **Epidemiologia e Serviços de Saúde**, v. 24, p. 325–332, jun. 2015.

GÓMEZ-GONZÁLEZ ET AL. **Artificial intelligence in medicine and healthcare: applications, availability and societal impact.** LU: Publications Office, 2020.

HACKELING, G. **Mastering Machine Learning with scikit-learn**. [s.l.] Packt Publishing Ltd, 2017.

HANCOCK, J. T.; KHOSHGOFTAAR, T. M. Survey on categorical data for neural networks. **Journal of Big Data**, v. 7, n. 1, p. 28, 10 abr. 2020.

HO, T. K. **Random decision forests**. Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1. **Anais**...: ICDAR '95.USA: IEEE Computer Society, 14 ago. 1995. . Acesso em: 12 out. 2022

JOHNSON, J. M.; KHOSHGOFTAAR, T. M. **Encoding Techniques for High-Cardinality Features and Ensemble Learners**. 2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI). **Anais**... Em: 2021 IEEE 22ND INTERNATIONAL CONFERENCE ON INFORMATION REUSE AND INTEGRATION FOR DATA SCIENCE (IRI). ago. 2021.

KRAMER, O. K-Nearest Neighbors. Em: KRAMER, O. (Ed.). **Dimensionality Reduction with Unsupervised Nearest Neighbors**. Intelligent Systems Reference Library. Berlin, Heidelberg: Springer, 2013. p. 13–23.

KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. New York, NY: Springer New York, 2013.

LIMA, T. P. F. et al. Death risk and the importance of clinical features in elderly people with COVID-19 using the Random Forest Algorithm. **Revista Brasileira de Saúde Materno Infantil**, v. 21, n. suppl 2, p. 445–451, 2021.

LUZ, C. F. et al. Machine learning in infection management using routine electronic health records: tools, techniques, and reporting of future technologies. **Clinical Microbiology and Infection**, v. 26, n. 10, p. 1291–1299, 1 out. 2020.

MENARD. **Applied Logistic Regression Analysis - Scott Menard**. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=EAI1QmUUsbUC&oi=fnd&pg=PP7&dq=.+%22Applied+logistic+regression+analysis.%22++Menard,+S.+(2002)&ots=4VJPH2nOIW&sig=-1385j2KWiZIt1U_PN4UA18Ws_g#v=onepage&q=.%20%22Applied%20logistic%20regression%20analysis.%22%20%20Menard%2C%20S.%20(2002)&f=false>. Acesso em: 27 jan. 2023.

MISHRA, P. **Practical Explainable AI Using Python - Artificial Intelligence Model Explanations Using Python-based Libraries, Extensions, and Frameworks - Pradeepta Mishra - Apress (2022).pdf**. Disponível em: <https://drive.google.com/file/d/10MbIqUDXW-DKS6FNpfDI1u5hZTcoN1kI/view?usp=drivesdk&usp=embed_facebook>. Acesso em: 13 set. 2022.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre Aprendizado de M´aquina. p. 18, [s.d.].

MY CHAU TU; DONGIL SHIN; DONGKYOO SHIN. **A Comparative Study of Medical Data Classification Methods Based on Decision Tree and Bagging Algorithms | IEEE Conference Publication | IEEE Xplore**. Disponível em: <https://ieeexplore.ieee.org/abstract/document/5380325>. Acesso em: 27 jan. 2023.

PARGENT, F. A Benchmark Experiment on How to Encode Categorical Features in Predictive Modeling. 2019.

PINA, E. et al. Infecções associadas aos cuidados de saúde e segurança do doente. **Revista Portuguesa de Saúde Pública**, v. Tematico, n. 10, p. 27–39, 1 nov. 2010.

RABBY, A. K. M. S. A. et al. **Machine Learning Applied to Kidney Disease Prediction: Comparison Study**. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). **Anais**... Em: 2019 10TH INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION AND NETWORKING TECHNOLOGIES (ICCCNT). jul. 2019.

RASCHKA, S.; MIRJALILI, V. **Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2**. [s.l.] Packt Publishing Ltd, 2019.

SAHA, B.; SRIVASTAVA, D. **Data quality: The other face of Big Data**. 2014 IEEE 30th International Conference on Data Engineering. **Anais**... Em: 2014 IEEE 30TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE). Chicago, IL, USA: IEEE, mar. 2014. Disponível em: <http://ieeexplore.ieee.org/document/6816764/>. Acesso em: 15 nov. 2021

SCHAPIRE, R. E.; FREUND, Y. Boosting: Foundations and Algorithms. **Kybernetes**, v. 42, n. 1, p. 164–166, 1 jan. 2013.

SHABAN-NEJAD, A.; MICHALOWSKI, M.; BUCKERIDGE, D. L. Explainability and Interpretability: Keys to Deep Medicine. Em: SHABAN-NEJAD, A.; MICHALOWSKI, M.; BUCKERIDGE, D. L. (Eds.). **Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability**. Estudos em Inteligência Computacional. Cham: Springer International Publishing, 2021. p. 1–10.

SHUAI, Y.; ZHENG, Y.; HUANG, H. **Hybrid Software Obsolescence Evaluation Model Based on PCA-SVM-GridSearchCV**. 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS). **Anais**... Em: 2018 IEEE 9TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND SERVICE SCIENCE (ICSESS). nov. 2018.

TAYLOR, R. A. et al. Predicting urinary tract infections in the emergency department with machine learning. **PLOS ONE**, v. 13, n. 3, p. e0194085, 7 mar. 2018.

TUKEY, J. W. **Exploratory data analysis**. [s.l.] Reading, MA, 1977. v. 2

WANG, C. et al. **Exploratory study on classification of lung cancer subtypes through a combined K-nearest neighbor classifier in breathomics | Scientific Reports**. Disponível em: <https://www.nature.com/articles/s41598-020-62803-4>. Acesso em: 8 jun. 2023.

WU, T. et al. Logistic regression technique is comparable to complex machine learning algorithms in predicting cognitive impairment related to post intensive care syndrome. **Scientific Reports**, v. 13, n. 1, p. 2485, 11 fev. 2023.

7 APÊNDICES E ANEXOS

Apêndice A – Dicionário De Dados

Nome da variável	Descrição
NR_ATENDIMENTO	Identifica o número do atendimento, cada internação do paciente gera um número de atendimento
CD_PESSOA_FISICA	Código único que identifica o paciente
NR_ANOS	É a idade do paciente no momento da internação, a informação foi calculada subtraindo a data da alta da internação com a data de nascimento
SEXO	Sexo do paciente
DS_TIPO_ATENDIMENTO	Origem do atendimento (Internação ou Pronto-scorro)
DS_CARATER_INTERNACAO	Descrição do carater de internação (1 Eletivo ou 2 Urgência 3 Acidente no Trabalho 4 Acidente trajeto do trabalho)
NR_DIA_INTERNADO	Quantidade de dias que o paciente ficou internado, calculo realizado subtraindo a data de alta pela data de entrada na internação
MOTIVO_ALTA	Descrição do motivo da alta (Alta ou Óbito)
TEMPO_CIRURG	Tempo que o paciente permaneceu em cirurgia
INFEC_CONFIRMADA	Identifica se o paciente contraiu infecção hospitalar durante o período de internação
MIN_PA_SISTOLICA	Valor mínimo de pressão arterial sistólica durante o período de internação
MAX_PA_SISTOLICA	Valor máximo de pressão arterial sistólica durante o período de internação
MIN_PA_DIASTOLICA	Valor mínimo de pressão arterial diastólica durante o período de internação
MAX_PA_DIASTOLICA	Valor máximo de pressão arterial diastólica durante o período de internação
MIN_FREQ_CARDIACA	Valor mínimo de frequência cardíaca durante o período de internação
MAX_FREQ_CARDIACA	Valor máximo de frequência cardíaca durante o período de internação
MIN_FREQ_RESP	Valor mínimo de frequência respiratória durante o período de internação
MAX_FREQ_RESP	Valor máximo de frequência respiratória durante o período de internação
MIN_QT_TEMP	Valor mínimo de temperatura durante o período de internação
MAX_QT_TEMP	Valor máximo de temperatura durante o período de internação
NIVEL_CONSCIENCIA	Nivel de consciência (Alerta, Confuso, Inconsciente, Resposta à dor, Resposta ao chamado verbal)
QT_DIAS_BALAO_INTRA_AORTICO	Quantidade de dias que o paciente ficou com o dispositivo invasivo balão inta aortico
QT_DIAS_CATETER_CENTRAL_PERIF	Quantidade de dias que o paciente ficou com o dispositivo invasivo cateter centra periférico
QT_DIAS_CATETER_DIALISE	Quantidade de dias que o paciente ficou com o dispositivo invasivo cateter de diálise
QT_DIAS_CATETER_TOT_IMPLANTADO	Quantidade de dias que o paciente ficou com o dispositivo invasivo cateter totalmente implantado
QT_DIAS_CATETER_UMBILICAL	Quantidade de dias que o paciente ficou com o dispositivo invasivo cateter umbilical
QT_DIAS_CATETER_VENOSO_CENTRAL	Quantidade de dias que o paciente ficou com o dispositivo invasivo cateter venoso central
QT_DIAS_DVE_DVP	Quantidade de dias que o paciente ficou com o dispositivo invasivo DVE / DVP
QT_DIAS_SONDA_VESICAL	Quantidade de dias que o paciente ficou com o dispositivo invasivo sonda vesical
QT_DIAS_TUBO_TRAQUEAL	Quantidade de dias que o paciente ficou com o dispositivo invasivo tubo traqueal
QT_DIAS_VENTILACAO_MECANICA	Quantidade de dias que o paciente ficou com o dispositivo invasivo ventilação mecânica
DS_CLINICA_E	Nome da clínica de internação (1 Médica, 2 Cirúrgica, 3 Obstétrica, 4 Pediátrica, 5 Oncológica, 6 Ortopédica, 7 Hemodinâmica, 8 Oftalmológica, 10 Quimioterapia, 11 Cirurgia Oncológica, 12 Psiquiátrica)
DS_ACOMODACAO_E	Descrição Acomodação (Apartamento ou Enfermaria)
DS_PROC_PRINCIPAL_E	Procedimento principal indicado pelo médico como motivo da internação
DS_SETOR_ENTRADA_E	Código do setor de internação (UTI (Unidade de Terapia Intensiva), Internação, ECI (Enfermaria de Cuidados Intensivos), Centro Cirúrgico)
CD_DOENCA_PRINCIPAL_E	Também conhecida como CID-10 é a classificação internacional de doenças

Observação: As variáveis que possuem "E" no final indica que a variável passou pela codificação Target Encoder

Anexo A - Parecer consubstanciado do comitê de ética

HOSPITAL MÁRCIO CUNHA - FUNDAÇÃO SÃO FRANCISCO XAVIER

PARECER CONSUBSTANCIADO DO CEP

DADOS DO PROJETO DE PESQUISA

Título da Pesquisa: Modelo preditivo de infecção hospitalar utilizando aprendizado de máquina
Pesquisador: PATRICIA PEDROSA MOREIRA MENDES
Área Temática:
Versão: 2
CAAE: 17987119.3.0000.8147
Instituição Proponente: FUNDAÇÃO SAO FRANCISCO XAVIER
Patrocinador Principal: Financiamento Próprio

DADOS DO PARECER

Número do Parecer: 3.797.742

Apresentação do Projeto:
Aprendizado de Máquina (AM) é uma área da Inteligência Artificial (IA) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Apesar dos grandes avanços científicos e tecnológicos que ocorreram, a infecção hospitalar continua a se constituir em séria ameaça à segurança dos pacientes hospitalizados, contribuindo para elevar as taxas de mortalidade, aumento dos custos de hospitalização e gastos com procedimentos diagnósticos. Com isto, a utilização de técnicas computacionais voltado a infecção hospitalar é de grande relevância nos dias atuais. Este estudo busca compreender a trajetória dos pacientes que contraíram infecção hospitalar ao longo de sua permanência hospitalar. Para tanto, tem como objetivo principal a criação de um modelo preditivo capaz de projetar os perfis dos pacientes que adquiriam infecção hospitalar. Baseado na coleta de dados, técnicas de aprendizado de máquina (machine learning) serão aplicadas para desenvolvimento do modelo preditivo

Objetivo da Pesquisa:
Objetivo Primário:
Compreender a trajetória dos pacientes que contraíram infecção hospitalar ao longo de sua permanência hospitalar. Para tanto, tem como objetivoprincipal a criação de um modelo preditivo capaz de projetar os perfis dos pacientes que adquiriam infecção
hospitalar.Baseado na coleta de dados, técnicas de aprendizado de máquina serão aplicadas para

Endereço: Av. Kiyoshi Tsunawaki, 41 Bairro das Águas Cidade: Ipatinga
Bairro: DAS AGUAS **CEP:** 35.160-158
UF: MG **Município:** IPATINGA
Telefone: (31)3830-5014 **E-mail:** cep@fsfx.com.br

desenvolvimento do modelo preditivo.

Objetivo Secundário:
Construir modelo preditivo para compreender a trajetória dos pacientes que contraíram infecção hospitalar, destacando-se como objetivo secundário a capacidade de explorar as possibilidades de uso de tal modelo.

Avaliação dos Riscos e Benefícios:
Riscos:
Existe um risco mínimo com a manipulação dos dados do prontuário eletrônico dos participantes envolvidos com a pesquisa, pois os dados serão anonimizados de forma que nenhum paciente será identificado.

Benefícios:
Identificar a trajetória dos pacientes que contraíram infecção hospitalar ao longo de sua permanência hospitalar, a partir disto, poderá ser realizada uma análise dos dados de forma a prevenir tal incidência. O aprendizado de máquina é um campo de ciência relativamente novo, focado na construção e estudo de sistemas que podem aprender automaticamente a partir de dados, gerando modelos preditivos de alta precisão.

Comentários e Considerações sobre a Pesquisa:
A metodologia contempla as fases: de coleta de dados, preparação dos dados (separados em duas amostras: uma a ser utilizada no treinamento e outra para a avaliação de performance do modelo),treinamento,avaliação, aprimoramento dos parâmetros e produção. Os dados serão coletados através do banco de dados do hospital no período de 2015 a 2016, totalizando uma média 90 mil pacientes durantes estes 4 anos e sendo assim não terá intervenção direta ou indireta nos pacientes, os dados serão anonimizados de forma a não identifica-los. Todas as despesas serão de total responsabilidade da pesquisadora, para desenvolvimento do trabalho serão utilizados softwares gratuitos.

Considerações sobre os Termos de apresentação obrigatória:
Todos os termos foram apresentados corretamente. Propõe dispensa de TCLE.Trata-se de pesquisa retrospectiva com uso das informações do banco de dados utilizando-se aprendizado de máquina(machine learning). para este estudo é necessário um volume dados muito grande para a que a máquina possa ser treinada e realize predições, diante deste volume de dados fica inviável

aplicar o termo de consentimento para todos os pacientes, além disto, a identificação do paciente não é importante para este estudo de forma que os dados pessoais serão anonimizados. Não haverá retenção de dados e armazenamento.

Recomendações:

Sem recomendações.

Conclusões ou Pendências e Lista de Inadequações:

O parecer nº 3496979 emitido pelo CEP emitido em agosto 2019 apontou como pendência o esclarecimento acerca do projeto, se visa a criação de produto, sendo necessário a definição de propriedades intelectuais e patentes. Em resposta a este parecer o pesquisador anexou um email com devolutiva da CONEP esclarecendo que o desenvolvimento de programas baseados em modelos disponibilizados na literatura científica não são considerados "plágios" e não violam os direitos autorais ou de patente, cabendo à pesquisadora a divulgação do código fonte do programa ou não após o desfecho do projeto. De modo complementar, o pesquisador apresentou uma carta esclarecendo que no decorrer da construção do projeto, caso haja alguma concretude sobre a criação de um produto, será estabelecido um termo aditivo de acordo junto ao contrato entre a Fundação São Francisco Xavier (FSFX) e o Instituto de Pesquisas Energéticas e Nucleares Aplicadas (IPEN). O contrato entre FSFX e IPEN também foi arquivado junto aos documentos na PB. Diante do exposto, o projeto atendeu as pendências exigidas sendo considerado aprovado.

Considerações Finais a critério do CEP:

Este parecer foi elaborado baseado nos documentos abaixo relacionados:

Tipo Documento	Arquivo	Postagem	Autor	Situação
Informações Básicas do Projeto	PB_INFORMAÇÕES_BÁSICAS_DO_PROJETO_1402471.pdf	21/12/2019 17:59:07		Aceito
Declaração de Pesquisadores	Consulta_CONEP_Plagio.pdf	21/12/2019 17:58:13	PATRICIA PEDROSA	Aceito
Declaração de Pesquisadores	CONTRATO_FSFX_IPEN.pdf	21/12/2019 16:31:43	PATRICIA PEDROSA	Aceito
Declaração de Pesquisadores	Carta_Resposta_Parecer.pdf	21/12/2019 16:30:57	PATRICIA PEDROSA	Aceito
Declaração de Pesquisadores	Termo_Compromisso_Propriedade_Intelectual.pdf	21/12/2019 16:30:33	PATRICIA PEDROSA	Aceito
Folha de Rosto	Folha_de_Rosto.pdf	26/07/2019 15:14:28	PATRICIA PEDROSA	Aceito
Outros	Termo_Utilizacao_Dados.pdf	26/07/2019	PATRICIA	Aceito

HOSPITAL MÁRCIO CUNHA - FUNDAÇÃO SÃO FRANCISCO XAVIER

Continuação do Parecer: 3.707.742

Outros	Termo_Utilizacao_Dados.pdf	15:13:17	MOREIRA MENDES	Aceito
Outros	CheckList.pdf	26/07/2019 15:10:55	PATRICIA PEDROSA	Aceito
Projeto Detalhado / Brochura Investigador	Plano_de_Trabalho.pdf	26/07/2019 15:09:56	PATRICIA PEDROSA MOREIRA MENDES	Aceito
Declaração de Pesquisadores	Termo_Responsabilidade_Pesquisador_Principal.pdf	26/07/2019 15:08:48	PATRICIA PEDROSA	Aceito
Declaração de Pesquisadores	Termo_Responsabilidade_Equipe_Pesquisa_1.pdf	26/07/2019 15:08:06	PATRICIA PEDROSA	Aceito
Declaração de Instituição e Infraestrutura	Declaracao_de_Infra_Estrutura.pdf	26/07/2019 15:06:43	PATRICIA PEDROSA MOREIRA MENDES	Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	Termo_Dispensa_TCLE.pdf	26/07/2019 15:04:23	PATRICIA PEDROSA MOREIRA MENDES	Aceito

Situação do Parecer:
Aprovado

Necessita Apreciação da CONEP:
Não

IPATINGA, 10 de Janeiro de 2020

Assinado por:
Luciano de Souza Viana
(Coordenador(a))

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Diretoria de Pesquisa, Desenvolvimento e Ensino
Av. Prof. Lineu Prestes, 2242 – Cidade Universitária CEP: 05508-000
Fone/Fax(0XX11) 3133-8908
SÃO PAULO – São Paulo – Brasil
http://www.ipen.br

O IPEN é uma Autarquia vinculada à Secretaria de Desenvolvimento, associada à Universidade de São Paulo e gerida técnica e administrativamente pela Comissão Nacional de Energia Nuclear, órgão do Ministério da Ciência, Tecnologia, Inovações e Comunicações.