UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ESTATÍSTICA
CURSO DE GRADUAÇÃO EM BACHARELADO EM ESTATÍSTICA

JOSEFERSON DA SILVA BARRETO

PREDIÇÃO DA SÍNDROME RESPIRATÓRIA AGUDA GRAVE POR MEIO DE MULTICLASSIFICAÇÃO COM ALGORITMOS DE MACHINE LEARNING

CAMPINA GRANDE - PB
2023

JOSEFERSON DA SILVA BARRETO

PREDIÇÃO DA SÍNDROME RESPIRATÓRIA AGUDA GRAVE POR MEIO DE MULTICLASSIFICAÇÃO COM ALGORITMOS DE MACHINE LEARNING

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Estatística do Departamento de Estatística do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Estatística.

Orientador: Prof. Dr. Tiago Almeida de Oliveira

CAMPINA GRANDE - PB
2023

B273p Barreto, Joseferson da Silva.

Predição da Síndrome Respiratória Aguda Grave por meio da multiclassificação com algoritmos de *machine learning* [manuscrito] / Joseferson da Silva Barreto. - 2023.

45 p.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Estatística) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2023.

"Orientação : Prof. Dr. Tiago Almeida de Oliveira, Coordenação do Curso de Estatística - CCT. "

1. Previsão de SRAG. 2. Saúde pública. 3. Modelos de aprendizado de máquina. 4. Classificação multiclasse. I. Título

21. ed. CDD 629.895

Elaborada por Talita M. A. Tavares - CRB - CRB 15/971 BC/UEPB

JOSEFERSON DA SILVA BARRETO

PREDIÇÃO DA SÍNDROME RESPIRATÓRIA AGUDA GRAVE POR MEIO DE MULTICLASSIFICAÇÃO COM ALGORITMOS DE MACHINE LEARNING

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Estatística do Departamento de Estatística do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Estatística.

Trabalho aprovado em 13/07/2023.

BANCA EXAMINADORA

Prof. Dr. Tiago Almeida de Oliveira(Orientador)
Universidade Estadual da Paraíba (UEPB)

Prof. Dr. Ricardo Alves de Olinda
Universidade Estadual da Paraíba (UEPB)

Prof. Me. Cleanderson Romualdo Fidelis
Universidade Estadual da Paraíba (UEPB)

Dedico este trabalho a Deus, minha família, amigos e colegas que estiveram sempre juntos comigo durante toda minha jornada acadêmica.

AGRADECIMENTOS

Gostaria de expressar meu sincero agradecimento a todas as pessoas que contribuíram para a conclusão bem-sucedida deste trabalho de conclusão de curso. Neste momento especial, desejo expressar minha gratidão a Deus, minha família, amigos, colegas e meu orientador, além dos professores que estiveram ao meu lado durante essa jornada. Em primeiro lugar, agradeço a Deus por me conceder a força, a sabedoria e a perseverança necessárias para superar os desafios ao longo deste processo. Sua orientação e apoio constantes foram fundamentais para alcançar esse marco importante em minha vida acadêmica.

À minha família, meu porto seguro, expresso minha profunda gratidão. O amor, o incentivo e o apoio incondicionais que recebi de meus pais, irmãos e outros familiares foram cruciais para me manter motivado e focado em meus objetivos. Vocês sempre acreditaram em mim e me encorajaram a nunca desistir, mesmo nos momentos mais difíceis. Sou imensamente grato por ter vocês ao meu lado. Aos meus amigos e colegas, sou grato por compartilharmos essa jornada acadêmica juntos. Agradeço por todas as trocas de conhecimento, apoio mútuo, estudo em grupo e momentos de descontração que compartilhamos ao longo dos anos. Vocês tornaram essa jornada mais leve e memorável. Sou grato pela amizade e pelo suporte inestimável que me proporcionaram.

Ao meu orientador, expresso minha sincera gratidão pela sua orientação, conhecimento e dedicação ao longo do processo de desenvolvimento deste trabalho. Suas orientações valiosas, sugestões e críticas construtivas foram fundamentais para moldar este projeto. Agradeço por ter compartilhado seu tempo, conhecimento e experiência comigo. Sou imensamente grato pela sua paciência e pela confiança que depositou em meu trabalho.

Também quero estender meu agradecimento a todos os professores que contribuíram para a minha formação acadêmica. Suas aulas, orientações e ensinamentos foram fundamentais para meu crescimento intelectual e profissional. Sou grato por ter tido a oportunidade de aprender com profissionais tão dedicados e experientes.

Por fim, gostaria de expressar minha gratidão a todos aqueles que, de alguma forma, contribuíram para o meu sucesso nesta jornada. Seja através de palavras de encorajamento, apoio emocional ou assistência prática, cada gesto foi importante e significativo.Este trabalho não teria sido possível sem a presença de todas essas pessoas especiais em minha vida. Agradeço do fundo do coração por todo o apoio, incentivo e confiança que recebi ao longo dessa jornada. Sinto-me verdadeiramente abençoado por ter cada um de vocês ao meu lado. O meu mais sincero obrigado a Deus, à minha família, amigos, colegas, orientador e professores. Vocês foram essenciais para a minha conquista acadêmica e sou eternamente grato por tudo o que fizeram por mim.

“O senhor é meu pastor e nada me faltará!”
(Salmo 23)

RESUMO

A Síndrome Respiratória Aguda Grave (SRAG) é uma condição médica grave que afeta o sistema respiratório, causando sintomas como febre alta, tosse, falta de ar e dificuldade respiratória. Sua etiologia pode ser diversa, incluindo o coronavírus SARS-CoV-2, responsável pela COVID-19. A SRAG pode levar a complicações graves, como pneumonia e insuficiência respiratória, demandando cuidados hospitalares intensivos e, em casos mais graves, pode resultar em insuficiência de múltiplos órgãos e morte. Este trabalho teve como objetivo demonstrar algumas das principais técnicas de *Machine Learning* para a classificação multiclasse de SRAG, visando aprimorar a capacidade de previsão da síndrome. Para isso, foram utilizados diferentes modelos de classificação, voltados especificamente para classificação multiclasse. Os resultados obtidos revelaram que o modelo XGBoost se destacou, alcançando uma performance geral de 83%. Esse modelo apresentou excelentes resultados na classificação de SRAG por influenza, SRAG por Covid e outras SRAGs relacionadas a vírus respiratórios. No entanto, identificou-se um desempenho razoável nas classes 3 e 4, que representam SRAG por outros agentes etiológicos e um tipo não especificado de SRAG, respectivamente. Esses resultados reforçam a importância do uso de técnicas de *Machine Learning* na análise da SRAG, contribuindo para melhorar a capacidade de previsão e diagnóstico da síndrome. Futuras pesquisas podem se concentrar em aprimorar o desempenho do modelo nas classes mais desafiadoras, buscando aperfeiçoar a aplicação prática dessas técnicas na saúde pública e na medicina.

Palavras-chave: previsão de SRAG; saúde pública; modelos de aprendizado de máquina; classificação multiclasse.

ABSTRACT

Acute Severe Respiratory Syndrome (ASRS) is a severe medical condition that affects the respiratory system, causing symptoms such as high fever, cough, shortness of breath, and respiratory distress. Its etiology can be diverse, including the SARS-CoV-2 coronavirus, responsible for COVID-19. ASRS can lead to serious complications such as pneumonia and respiratory failure, requiring intensive hospital care and, in severe cases, can result in multiple organ failure and death. This study aimed to demonstrate some of the main machine learning techniques for multiclass classification of ASRS, aiming to improve the syndrome's predictive capability. Different classification models specifically tailored for multiclass classification were used to achieve this goal. The results obtained revealed that the XGBoost model stood out, achieving an overall performance of 83%. This model showed excellent results in classifying ASRS caused by influenza, ASRS caused by Covid, and other ASRS related to respiratory viruses. However, reasonable performance was identified in classes 3 and 4, representing ASRS caused by other etiological agents and an unspecified type of ASRS, respectively. These results reinforce the importance of using machine learning techniques in ASRS analysis, contributing to improving the predictive capability and diagnosis of the syndrome. Future research can focus on enhancing the model's performance in the more challenging classes, seeking to refine the practical application of these techniques in public health and medicine.

Keywords: ASRS prediction; public health; machine learning models; multiclass classification.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A Síndrome Respiratória Aguda Grave (SRAG), é condição clínica caracterizada por sintomas respiratórios graves que exigem hospitalização e pode ser causada por uma variedade de agentes infecciosos, como vírus, bactérias ou fungos. Ela geralmente se manifesta como uma infecção do trato respiratório inferior, afetando os pulmões e causando sintomas como febre alta, tosse persistente, falta de ar, dor no peito e dificuldade respiratória. Em casos mais graves, a SRAG pode levar à insuficiência respiratória e exigir suporte ventilatório. De acordo com Brasil (2021) são sintomas do SARGS: indivíduo com síndrome gripal (SG) que apresente: dispneia/desconforto respiratório ou pressão ou dor persistente no tórax ou saturação de oxigênio (O2) menor que 95% em ar ambiente ou coloração azulada (cianose) dos lábios ou rosto. Vale salientar, no entanto, que nem todos os casos da síndrome são explicados pela infecção por Sars-CoV-2, pois, como mencionado, outras infecções respiratórias podem desencadear o problema (SANTOS, 2023).

Segundo SILVA (2022), nos últimos anos, o surgimento da Covid-19 e o aumento no número de casos de gripe geraram uma atmosfera de medo e incertezas. Essa situação é agravada pela semelhança entre os sintomas de diferentes síndromes respiratórias graves, tornando a classificação e o diagnóstico um desafio para os profissionais de saúde. Muitas dessas síndromes compartilham sintomas comuns, como febre, tosse, falta de ar e dor no peito, resultando em uma sobreposição de características clínicas que pode dificultar a identificação precisa da doença em questão.

A importância do *Machine Learning* vem crescendo nos últimos anos, impulsionada pelo aumento na disponibilidade de dados, pela evolução dos algoritmos de aprendizagem e pelo avanço da capacidade de processamento dos computadores. Com isso, tornou-se possível realizar análises cada vez mais sofisticadas e precisas em áreas como finanças, marketing, saúde, entre outras. O uso de técnicas de *Machine Learning* proporciona uma abordagem inovadora para lidar com a complexidade dos dados relacionados à SRAG, permitindo que padrões sejam identificados. Essas técnicas explorarão os dados existentes, como informações clínicas, resultados de exames laboratoriais e históricos médicos, a fim de construir um modelo preditivo capaz de auxiliar na identificação correta do tipo de SRAG.

Nesta monografia, o objetivo é desenvolver um modelo de classificação eficaz utilizando técnicas de aprendizado de máquina *Machine Learning* para identificar e classificar corretamente os diferentes tipos de Síndrome Respiratória Aguda Grave (SRAG). Para alcançar esse objetivo, será empregado o uso de algoritmos de *Machine Learning* que analisarão dados clínicos e epidemiológicos relevantes. O intuito é criar um modelo preditivo capaz de distinguir entre os tipos específicos de SRAG de forma mais precisa e eficiente.

2 MATERIAL E MÉTODOS

Será apresentada uma descrição detalhada dos materiais utilizados, bem como os procedimentos adotados para a coleta e preparação dos dados. Além disso, serão explicadas as principais técnicas de *Machine Learning* empregadas nas análises do problema apresentado nesta monografia, voltadas a classificação multiclasse, abrangendo algumas de suas aplicações específicas nesse contexto desafiador.

2.1 Material

Para a análise, foi utilizado um banco de dados de acesso livre e gratuito do OpenDataSUS (SARG[1]) está relacionada aos atributos de pacientes com algum tipo de síndrome gripal e refere-se ao período de 2021 a 2023. O conjunto de dados é formado de 468.994 observações e 166 variáveis: sexo, idade, ID do hospital, entre outras. O dicionário completo pode ser acessado no site oficial do OpenDataSUS ou no anexo desta monografia que contém as descrições das variáveis utilizadas. Na Tabela 1 pode-se observar a descrição de algumas variáveis utilizadas no modelo, incluindo a variável resposta:

[1] https://opendatasus.saude.gov.br/sq/dataset/srag-2021-a-2023

Tabela 1 – Descrição das Variáveis do SRAG Open Data SUS

Variável	**Tipo**	**Categoria**	**Descrição**
AN_PARA3	Categórica	1 - marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico. Parainfluenza 3.
AN_ADENO	Categórica	1 - marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico. Adenovírus.
AN_OUTRO	Categórica	1 - marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico. Outro vírus respiratório.
CLASSI_FIN	Categórica	1 - SRAG por influenza 2 - SRAG por outro vírus respiratório 3 - SRAG por outro agente etiológico, qual: 4 - SRAG não especificado 5 - SRAG por covid-19 pelo usuário	Diagnóstico final do caso.

Fonte: Elaborada pelo Autor, 2023

A variável resposta representa a classificação final do caso, como sendo:

- 1-SRAG por influenza;
- 2-SRAG por outro vírus respiratório;
- 3-SRAG por outro agente etiológico, qual;
- 4-SRAG não especificado;
- 5-SRAG por covid-19.

O ambiente computacional estatístico utilizado para a análise e limpeza dos dados foi o *software R* (R Core Team, 2023) e para a geração dos modelos foi utilizado a *linguagem python* (Python Software Foundation, 2023), os scripts e os demais arquivos utilizados nas análises desse projeto pode ser acessado através do seguinte repositório do github: Prediction_of_SARGS_by_multiclassification.

2.2 Métodos

Inicialmente foi feita a limpeza dos dados, visando garantir a qualidade e confiabilidade do conjunto de dados utilizado. Em seguida, é realizada uma análise exploratória detalhada, permitindo uma melhor compreensão do nosso conjunto de dados e a identificação de possíveis padrões relevantes. A partir disso, foram utilizadas várias técnicas de *Machine Learning* e Estatística, tais como: métodos de reamostragem e subamostragem, seleção de atributos e alguns modelos de *Machine Learning* para classificação multiclasse. Por último, foi realizada a comparação de desempenho entre os modelos.

No método de subamostragem, foi utilizado a amostragem aleatória, onde instâncias(amostras) aleatórias são selecionadas da classe majoritária para corresponder ao número de amostras da classe minoritária. Isso reduz o tamanho da classe majoritária, equilibrando as proporções das classes. Por último, foram criados vários modelos de *Machine Learning* para classificação multiclasse, usando as técnicas antes descritas, a fim de comparar seus resultados e verificar qual modelo foi capaz de classificar as classes corretamente com maior exatidão e que será o possível candidato a implementação. Além disso, foram feitas a Curva Roc para avaliação do modelo com melhores métricas e o gráfico de importância de atributos Shapley que mostrará o quanto cada covariável é importante para o modelo.

2.3 O que é Machine Learning

Segundo Macedo e Charles (2017), *Machine Learning* é um subcampo da Inteligência Artificial que utiliza técnicas de Estatística e Matemática associadas à tecnologia para compreender padrões (características) de um conjunto de dados. Por meio de técnicas estatísticas e computacionais, os modelos de *Machine Learning* são capazes de identificar padrões e relações complexas nos dados, permitindo a realização de previsões e tomadas de decisões automáticas. Essa abordagem revolucionária tem impactado diversos setores, proporcionando avanços significativos em áreas como medicina, finanças, transporte e muitas outras, abrindo caminho para o desenvolvimento de sistemas autônomos e inteligentes. Com o crescente volume de dados disponíveis e o contínuo aprimoramento dos algoritmos, o *Machine Learning* tem se mostrado uma ferramenta poderosa para a solução de problemas complexos e a descoberta de informações valiosas, impulsionando a inovação e transformando a maneira como interagimos com a tecnologia e o mundo ao nosso redor.

2.4 CONCEITOS BÁSICOS

2.4.1 Divisão dos Dados

Uma das principais etapas quando se trabalha com *Machine Learning* é a divisão dos dados, isso nos ajudará a avaliar o desempenho do modelo, e a capacidade de generalização do modelo, prevenir *overfitting*, otimizar os hiperparâmetros, comparar diferentes

modelos e simular cenários futuros. Após a criação do modelo, segundo Oliveira (2021), o modelo deve ser aplicado no conjunto de teste para avaliar sua performance em dados nunca observados. Essa divisão nos ajuda a ter uma visão mais precisa do desempenho do modelo e a tomar decisões mais adequadas sobre seu uso e ajuste.

2.4.2 Avaliação do Desempenho

De acordo com Santos (2020), é necessário obter medidas estatísticas que provem a confiabilidade do algoritmo. A avaliação do desempenho refere-se à análise e medição do quão bem um modelo de aprendizado de máquina está realizando suas tarefas ou previsões. É uma etapa crítica no processo de construção de modelos, pois fornece informações sobre a qualidade e eficácia do modelo em lidar com dados desconhecidos. A avaliação do desempenho geralmente envolve a comparação das saídas ou previsões do modelo com os valores reais, ou esperados, Dependendo da tarefa em questão (classificação, regressão, *clustering*), diferentes métricas de desempenho podem ser usadas.

2.4.3 Overfitting

De acordo com Rosa (2019), o *overfitting* ocorre quando o modelo treinado se ajusta muito bem ao conjunto de dados de treinamento, porém é ineficaz ao tentar classificar novos dados. Ao separar um conjunto de teste, podemos detectar se o modelo está superajustado aos dados de treinamento, pois ele terá um desempenho inferior no conjunto de teste. Isso nos ajuda a identificar a necessidade de ajustar o modelo para evitar o *overfitting*. O objetivo de prevenir o *overfitting*, é encontrar um equilíbrio entre a capacidade do modelo de capturar padrões relevantes nos dados e evitar o ajuste excessivo aos dados de treinamento. Isso resultará em um modelo que pode generalizar bem para novos dados e produzir previsões mais precisas e confiáveis.

2.4.4 Hiperparâmetros

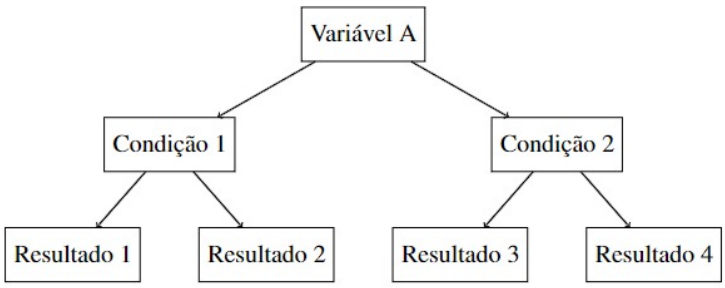
Segundo BARBOSA (2018), alguns algoritmos de classificação possuem parâmetros que precisam ser configurados antes de executá-los para obtenção de um resultado mais satisfatórios. Os hiperparâmetros são parâmetros ajustáveis externamente que influenciam o comportamento e o desempenho de um modelo de aprendizado de máquina, mas não são aprendidos a partir dos dados durante o processo de treinamento. Ao contrário dos parâmetros do modelo, que são ajustados internamente durante o treinamento, os hiperparâmetros são definidos antes do treinamento e afetam como o modelo é treinado e como ele realiza as previsões. Os hiperparâmetros determinam as configurações e as características do modelo, permitindo que você personalize e ajuste o desempenho conforme as necessidades específicas do problema e do conjunto de dados. Esses hiperparâmetros podem variar conforme o algoritmo e o tipo de modelo utilizado.

2.5 Árvore de Decisão

Segundo Lauretto (2010), uma árvore é uma coleção de elementos chamados nós, dentre os quais um é distinguido como uma raiz, juntamente com uma relação de "paternidade" que impõe uma estrutura hierárquica sobre os nós. Trata-se de um modelo que nos ajuda a tomar decisões com base em uma série de condições e critérios definidos. Sua representação gráfica, que se assemelha a uma árvore com nós e ramos, torna o processo de tomada de decisão mais visual e compreensível.

Imagine que estamos diante de um problema complexo que exige uma série de escolhas para chegar a um resultado desejado. A árvore de decisão nos ajuda a dividir esse problema em pequenas partes e nos guia por diferentes caminhos, com base em critérios específicos. Cada nó da árvore representa uma decisão que precisamos tomar, enquanto os ramos representam as possíveis alternativas que podemos seguir.

Figura 1 – Funcionamento da Árvore de Decisão.



Fonte: Elaborada pelo Autor, 2023

Neste exemplo, é apresentada uma estrutura básica de uma árvore de decisão. O nó raiz representa uma variável de interesse, enquanto os nós subsequentes representam diferentes condições ou critérios de decisão. Cada nó de decisão é conectado a dois ou mais nós de resultado, que representam as possíveis ações ou resultados decorrentes daquela condição específica. Nas árvores de decisão temos uma métrica chamada critério de decisão.

2.5.1 Critério de Decisão

O critério em árvores de decisão é uma medida utilizada para determinar como os nós da árvore devem ser divididos durante a construção do modelo. O critério define a função de avaliação que determina qual divisão é considerada a melhor em termos de pureza das classes resultantes.

Existem diferentes critérios utilizados em árvores de decisão, sendo os mais comuns o índice de Gini (Gini impurity) e a entropia. Esses critérios medem a impureza dos dados em um nó da árvore e procuram dividir o nó para reduzir a impureza e aumentar a homogeneidade das classes nós filhos.

2.5.1.1 *Índice de Gini*

O critério de Gini avalia a probabilidade de classificar incorretamente uma amostra aleatória com base na distribuição das classes em um nó. Ele mede a impureza dos dados, sendo que um valor de 0 indica que todas as amostras pertencem a uma única classe e um valor de 1 indica que as amostras estão igualmente distribuídas entre as classes. O índice de Gini pode ser obtido pela seguinte expressão:

$$\text{Gini}(p) = 1 - \sum_{i=1}^{K} p_i^2, \tag{2.1}$$

onde, p representa a distribuição de probabilidade das classes em um nó, e K é o número de classes. Para realizar o cálculo do índice de Gini, é necessário seguir alguns passos. Primeiramente, é preciso calcular o quadrado das probabilidades de cada classe, representadas por p_i^2. Em seguida, esses valores são somados. Por fim, o resultado dessa soma é subtraído de 1.

O índice de Gini é uma medida que varia de 0 a 1. Quando o valor é igual a 0, significa que todas as propriedades pertencem a uma única classe, indicando que o nó é completamente puro. Por outro lado, quando o valor é igual a 1, isso indica que as amostras estão igualmente distribuídas entre as classes, tornando o nó impuro.

2.5.1.2 *Entropia*

A entropia, por sua vez, é uma medida de desordem ou incerteza nos dados. Ela é calculada com base na distribuição das classes em um nó e mede a impureza dos dados. Assim como o critério de Gini, a entropia é maximizada quando as classes estão igualmente distribuídas e minimizada quando todas as amostras pertencem a uma única classe. A entropia é uma medida utilizada em árvores de decisão para quantificar a impureza dos dados em um nó, ela representa a quantidade de informação aleatória ou incerteza presente em um conjunto de dados. A fórmula do índice de entropia calcula a entropia de um conjunto de dados com base na distribuição das classes presentes. De acordo com Lauretto (2010), A medida de entropia busca, então, medir o grau de incerteza presente em cada esquema finito. A entropia pode ser obtida através da seguinte expressão:

$$\text{H(X)} = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i), \tag{2.2}$$

em que a entropia é denotada por $H(X)$, sendo uma medida da incerteza ou desordem presente em um conjunto de dados chamado X. Quanto maior a entropia, maior a incerteza e a falta de informação sobre as classes presentes no conjunto de dados. O valor de n representa o número de classes distintas encontradas em X, ou seja, é a quantidade de categorias diferentes que as amostras podem pertencer. Cada classe é denotada por x_i, onde i é um índice que varia de 1 a n. A probabilidade de ocorrência de uma classe

específica x_i em X é representada por $p(x_i)$, essa probabilidade é calculada dividindo o número de amostras que pertencem à classe x_i pelo número total de amostras presentes em X. Portanto, $p(x_i)$ fornece uma medida de quão frequente ou comum é a classe x_i no conjunto de dados.

A equação calcula a entropia somando as contribuições de cada classe ponderadas pela probabilidade de ocorrência da classe. Quanto mais homogêneo for o conjunto de dados (com todas as amostras pertencendo a uma única classe), menor será a entropia e, consequentemente, maior será a pureza do nó. Por outro lado, se as amostras estiverem igualmente distribuídas entre as classes, a entropia será máxima, indicando alta impureza ou incerteza.

Ao construir uma árvore de decisão, o objetivo é reduzir a entropia dos nós ao máximo possível através da escolha de divisões que resultem em conjuntos de dados mais homogêneos em termos de classes. O critério de entropia é usado para determinar a divisão ótima em cada nó da árvore, buscando maximizar a informação ganha ao realizar a divisão.

2.6 Random Forest

De acordo com Santos (2020), o *Random Forest* é um método de aprendizado conjunto para classificação e regressão que opera construindo várias árvores de decisão no momento do treinamento e produzindo a classe, que é o modo das saídas geradas por árvores individuais. O *Random Forest* é um modelo de aprendizado de máquina que combina a ideia de árvores de decisão com o conceito de conjunto de modelos. O *Random Forest* é composta por um conjunto de árvores de decisão individuais. Cada árvore é treinada independentemente em uma amostra aleatória dos dados de treinamento, e as previsões são feitas pela combinação das previsões de todas as árvores. Essa abordagem de conjunto ajuda a reduzir o viés e a variância do modelo, melhorando seu desempenho e capacidade de generalização.

Uma das características distintivas da *Random Forest* é o uso de duas formas de aleatoriedade: a seleção aleatória dos dados de treinamento para cada árvore e a seleção aleatória das características (ou atributos) considerados em cada divisão da árvore. Essas fontes de aleatoriedade garantem que cada árvore seja treinada de forma independente e diversa, evitando o sobreajuste e aumentando a robustez do modelo.

O processo de construção de uma *Random Forest* ocorre em várias etapas. Primeiro, é feita uma amostragem aleatória com substituição, conhecida como amostragem de *bootstrap*. Assim como no *bagging*, todas as amostras *bootstrap* são identicamente distribuídas. Isto implica que a esperança da média das B árvores é a mesma que a esperança de cada uma delas, ou seja, o viés do modelo de árvores agregadas será equivalente ao observado em cada árvore (MORAIS, 2017). Após a seleção da amostra, uma árvore de decisão é treinada nessa amostra, mas em cada divisão da árvore, apenas um subconjunto aleatório de características é considerado. Esse processo de amostragem e treinamento é

repetido para cada árvore na floresta.

Para fazer previsões por meio da *Random Forest*, cada árvore individual produz uma previsão e a classe ou valor final é determinado por meio de uma combinação dos resultados das árvores. Para classificação, uma votação majoritária é usada para determinar a classe prevista, enquanto para regressão, uma média dos valores previstos é calculada. Essa abordagem de conjunto permite que o modelo aproveite a diversidade das árvores e forneça previsões mais precisas e estáveis.

A *Random Forest* tem uma ampla gama de aplicações em diferentes áreas. É frequentemente usada para problemas de classificação, como detecção de fraudes em cartões de crédito, diagnóstico médico, análise de sentimentos e detecção de spam. Além disso, é aplicada em tarefas de regressão, como previsão de preços imobiliários, previsão de demanda e análise de séries temporais. Sua flexibilidade, robustez e capacidade de lidar com conjuntos de dados complexos tornam-na uma escolha popular em muitos cenários de aprendizado de máquina.

2.6.1 Random Forest Para Classificação

A Random Forest para classificação utiliza um conjunto de árvores de decisão independentes que são treinadas em diferentes amostras aleatórias do conjunto de dados de treinamento. Cada árvore faz uma previsão de classe para uma instância de entrada com base em suas características. A fórmula do Random Forest na classificação é dada por:

$$\hat{y} = \arg\max \left(\sum_{i=1}^{N} \mathbb{I}(h_i(x) = c) \right), \tag{2.3}$$

em que $\hat{y}$ representa a classe prevista para a entrada x, argmax significa que estamos buscando o valor de c que maximiza a expressão que segue, $\sum_{i=1}^{N}$ é a soma sobre todas as N árvores da Random Forest, $h_i(x)$ é a previsão da árvore i para a entrada x, $\mathbb{I}(h_i(x) = c)$ é uma função indicadora que retorna 1 se a previsão da árvore i para x for igual à classe c, e 0 caso contrário.

A fórmula representa o processo de votação majoritária na Random Forest para determinar a classe prevista. Para cada classe possível c, a soma conta quantas vezes a previsão da árvore i para a entrada x é igual a c. A classe c que possui a maior soma é escolhida como a classe prevista para x.

2.7 XGBoost

O *Extreme Gradient Boosting* (XGBoost) é um algoritmo de aprendizado de máquina que se baseia no método de impulso (boosting) para criar modelos preditivos. O XGBoost Segundo Ceccon (2019), foi apresentado em 2016 por Tianqi Chen e Carlos

Guestrin na Conferência SIGKDD e, desde então, tem sido o método de preferência dos profissionais da área.

Esse método busca melhorar o desempenho do *gradient boosting* otimizando a utilização do software (conhecimento técnico dos chefs) e hardware (ferramentas que eles usam para avaliar, como seus sentidos). Seria como criar um ambiente perfeito para que os avaliadores possam fazer o melhor trabalho possível: a nota mais factível no menor tempo. Digamos que os chefs recebem antecipadamente uma relação descritiva dos critérios a serem considerados, e um despalatizante para usarem entre as provas gustativas (CECCON, 2019).

Existem várias razões pelas quais o XGBoost se tornou popular, Ele possui uma eficiência computacional significativa, sendo capaz de lidar com grandes conjuntos de dados e ter um bom desempenho. Além disso, o XGBoost utiliza técnicas avançadas de regularização para evitar o overfitting (sobreajuste) do modelo, o que o torna mais geral e robusto em relação a dados novos.

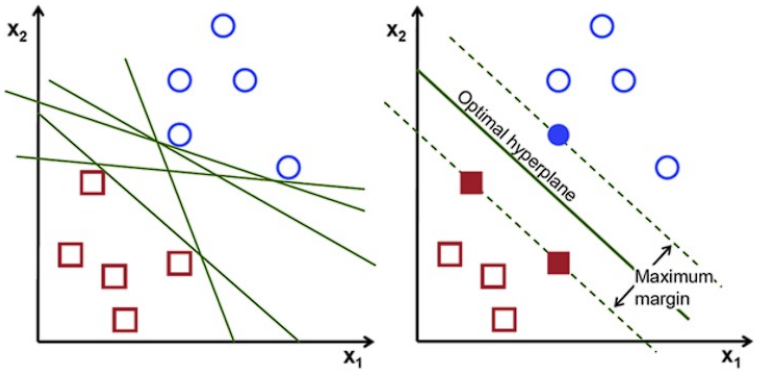
$$\text{Obj}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{2.4}$$

em que a função objetivo a ser otimizada é representada por $\text{Obj}(\theta)$, θ são os parâmetros do modelo em questão, n é o número de amostras de treinamento. Utilizamos a função de perda $l(y_i, \hat{y}_i)$ para medir a diferença entre o valor verdadeiro y_i e o valor predito $\hat{y}_i$, ou seja, basicamente a função de perda é o erro associado ao modelo durante o treinamento e é usada como uma medida de desempenho para ajustar o modelo para obter os melhores resultados possíveis. Por último, $\Omega(f_k)$ é a função de penalidade ou regularização aplicada à árvore f_k. Essa função tem como objetivo evitar overfitting e melhorar a generalização do modelo.

2.8 Suport Vector Machine

O objetivo do algoritmo da máquina de vetores de suporte (SVM –Support Vector Machine) é encontrar um hiperplano em um espaço N-dimensional (N -o número de recursos ou atributos) que classifica distintamente os pontos de dados (Data Science Academy, 2021). Segundo Srinivas (2010), é uma técnica de classificação que procura encontrar um hiperplano que particione os dados por seus rótulos de classe e ao mesmo tempo, evite o ajuste excessivo dos dados maximizando a margem da separação hiperplano. Na Figura 2 podemos visualizar a definição do hiperplano:

Figura 2 – Otimização do Hiperplano



Fonte: Data Science Academy(2022)

O objetivo é encontrar um hiperplano com a margem máxima, ou seja, a distância máxima entre os pontos de dados das duas classes. O modelo SVM pode ser aplicado tanto para dados linearmente separáveis (se tornando um problema de regressão) como também para dados não lineares (se tornando um problema de classificação). Para problemas de classificação, o modelo utilizará o truque de kernel, que envolve mapear os dados do espaço original de características para um espaço de alta dimensionalidade, onde seja possível alcançar uma melhor separação entre as classes. Essa transformação é realizada por meio de uma função kernel, que calcula o produto interno entre dois pontos no espaço de alta dimensionalidade sem a necessidade de calcular explicitamente as coordenadas nesse espaço. Com o uso do truque de kernel, o SVM pode encontrar um hiperplano de separação ótimo mesmo em casos onde as classes não são linearmente separáveis no espaço original.

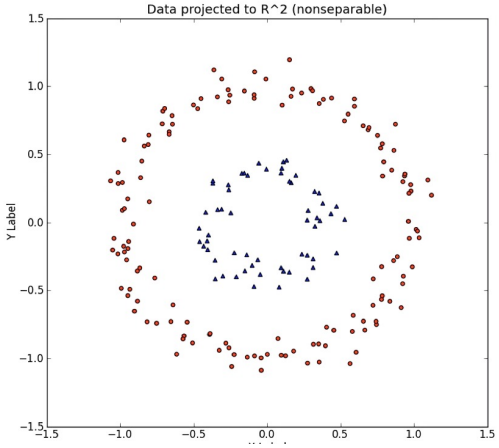
2.8.1 Kernel Linear

O SVM de kernel linear é um tipo de SVM em que se utiliza um kernel linear para mapear os dados de entrada em um espaço de maior dimensionalidade, onde é mais fácil separar as classes por meio de um hiperplano linear. O hiperplano linear é uma fronteira de decisão que separa os exemplos de diferentes classes. Uma das principais vantagens do SVM de kernel linear é sua eficiência computacional, especialmente quando comparado a outros kernels mais complexos. A natureza linear do kernel permite que o SVM seja treinado e usado em grandes conjuntos de dados com menor tempo de processamento. Além disso, o SVM de kernel linear é menos suscetível a overfitting, o que ocorre quando o modelo se ajusta muito aos dados de treinamento e não generaliza bem para dados não vistos.

No entanto, o kernel linear é eficaz apenas quando os dados são linearmente separáveis. Caso contrário, o hiperplano linear não será capaz de separar completamente as classes, resultando em um desempenho inferior. Os dados não lineares nada mais são do que dados que não se consegue dividir por meio de uma reta, em outras palavras,

não é possível traçar uma linha reta que represente de forma precisa a relação entre as variáveis em estudo, é possível perceber essa relação na Figura 3 a seguir:

Figura 3 – Dados Não Lineares



Fonte: Data Science Academy(2022)

Ao contrário dos dados lineares, onde uma relação linear pode ser estabelecida usando uma equação matemática simples (por exemplo, $y = mx + b$), os dados não lineares requerem uma abordagem mais complexa para entender e modelar a relação entre as variáveis. Uma forma de contornar essa situação é usar a função de Kernel polinomial.

2.8.2 Kernel Polinomial

O kernel polinomial é um tipo de kernel utilizado no algoritmo de Support Vector Machine (SVM) que permite mapear os dados de entrada em um espaço de maior dimensionalidade usando funções polinomiais. O kernel polinomial é usado quando os dados não são linearmente separáveis no espaço de entrada original. Ele transforma os dados de entrada em um espaço de maior dimensionalidade, onde pode ser possível encontrar um hiperplano linear que separe as classes. A função polinomial é aplicada aos pares de atributos dos dados, gerando novas características que representam as interações polinomiais entre os atributos.

A principal vantagem do kernel polinomial é sua capacidade de capturar relações não lineares entre os atributos. Ao elevar os atributos a potências mais altas, o kernel polinomial permite que o SVM modele separações mais complexas. A escolha adequada do grau do polinômio é crucial, pois um grau muito baixo pode não ser capaz de separar corretamente as classes, enquanto um grau muito alto pode levar a overfitting e dificuldades computacionais.

Uma desvantagem do kernel polinomial é que, à medida que o grau do polinômio aumenta, o número de características geradas também aumenta exponencialmente. Isso pode levar a problemas de alta dimensionalidade, especialmente quando o número de

atributos originais já é grande. O aumento na dimensionalidade dos dados pode aumentar o tempo de treinamento e tornar o modelo mais suscetível ao overfitting.

O kernel polinomial tem várias aplicações práticas. É comumente usado em tarefas de classificação de imagem, como reconhecimento facial, detecção de objetos e segmentação de imagens. Também é aplicado em problemas de processamento de linguagem natural, como classificação de texto e análise de sentimentos. O kernel polinomial pode ser usado em qualquer problema onde as relações não lineares entre os atributos são importantes para a separação das classes.

O kernel polinomial é definido como

$$K(x,y) = (\langle x,y\rangle + c)^d, \tag{2.5}$$

em que x e y representam os vetores de entrada (dados), $\langle x,y\rangle$ é o produto interno entre os vetores, c é um termo de deslocamento (bias) e d é o grau do polinômio. A função do kernel polinomial consiste em elevar o produto interno dos vetores x e y a potência d, somar o termo de deslocamento c e obter o resultado.

2.9 Seleção de Atributos

A seleção de atributos é um processo importante no campo do aprendizado de máquina. Trata-se de escolher os atributos mais relevantes de um conjunto de dados para melhorar o desempenho dos modelos de machine learning. Existem diferentes abordagens e métodos para realizar essa seleção. Segundo Espinosa, Jiménez e Palma (2023), o processo de seleção de características (FS) permite reduzir a dimensionalidade dos dados, eliminando atributos redundantes e irrelevantes, reduzindo assim a complexidade dos modelos de previsão.

Uma forma de seleção de atributos é utilizando métodos embutidos, nos quais o algoritmo de aprendizado de máquina atribui importância aos atributos durante o treinamento. Por exemplo, o algoritmo Random Forest avalia a importância de cada atributo com base em sua contribuição para a redução da impureza nos nós das árvores de decisão. Além disso, existem técnicas de busca que exploram um espaço de soluções em busca do subconjunto de atributos que otimiza um critério específico. Essas técnicas podem ser exaustivas ou utilizar algoritmos heurísticos para encontrar a melhor combinação de atributos. A seleção de atributos é uma etapa importante no processo de modelagem de machine learning, pois ajuda a reduzir a dimensionalidade dos dados, evitar overfitting e melhorar a interpretabilidade do modelo. A escolha da abordagem e do método depende do contexto e dos requisitos específicos do problema.

2.10 Valor de Shapley

Segundo Bezerra, Grande e Silva (2009), o conceito de valor neste caso não deve ser confundido com o conceito de valor apregoado pela economia, A teoria do

valor de Shapley foi desenvolvida por Shapley (1953), um matemático e economista norte-americano. Os atributos Shapley foram inicialmente propostos por Shapley como uma medida de atribuição justa e equitativa de valores em jogos cooperativos, onde os jogadores contribuem de maneira conjunta para alcançar um resultado coletivo. A teoria ganhou importância em diversos campos, incluindo a economia, a ciência da computação e a análise de dados. Ao calcular os atributos Shapley, é necessário considerar todas as combinações possíveis de variáveis e suas contribuições, a fim de alcançar uma distribuição justa do valor total.

No contexto da análise de dados e aprendizado de máquina, os atributos Shapley podem ser aplicados para explicar as previsões de um modelo. Eles atribuem um valor a cada variável de entrada, indicando sua importância relativa na previsão do modelo. Essa abordagem proporciona uma compreensão mais profunda dos fatores que influenciam as previsões e auxilia na identificação das variáveis que exercem um impacto significativo. Ao atribuir valores aos atributos Shapley, é possível identificar quais variáveis têm um impacto mais significativo nas previsões do modelo. Isso permite compreender quais características ou variáveis têm maior influência na tomada de decisões do modelo e como elas contribuem para o resultado final.

2.11 Undersampling

Segundo Koziarski (2020), problema de desequilíbrio de dados, ocorre na tarefa de classificação sempre que o número de observações pertencentes a uma das classes, a classe majoritária, excede o número de observações pertencentes a uma das outras classes, a classe minoritária. De acordo com Fernandez et al. (2018), *undersampling* consiste em reduzir os dados eliminando exemplos pertencentes à classe majoritária com o objetivo de equalizar o número de exemplos de cada classe Undersampling é uma técnica utilizada no campo do aprendizado de máquina para lidar com conjuntos de dados desbalanceados, nos quais uma classe é significativamente mais representada do que as outras. O objetivo do *undersampling* é reduzir a quantidade de exemplos da classe majoritária, de modo a equilibrar a distribuição das classes. Essa técnica pode ser aplicada de diferentes maneiras. Uma abordagem comum é a remoção aleatória de exemplos da classe majoritária até que a proporção entre as classes seja adequada. Dessa forma, a quantidade de exemplos da classe majoritária é reduzida para que fique mais próxima da quantidade de exemplos da classe minoritária.

Cabe ressaltar que o *undersampling* pode ser realizado de forma simples ou estratificada. No *undersampling* simples, os exemplos são removidos aleatoriamente sem considerar sua relação com os demais exemplos. Já no *undersampling* estratificado, busca-se manter uma distribuição equilibrada nas características dos exemplos remanescentes, preservando assim as informações relevantes da classe majoritária. É importante ressaltar que o *undersampling* pode resultar na perda de informações importantes contidas nos exemplos removidos da classe majoritária. Portanto, sua aplicação deve ser cuidadosa,

considerando o impacto na representatividade e na capacidade do modelo de generalizar os dados.

2.12 Curva de Característica de Operação do Receptor (ROC)

Segundo Braga (2000), A análise ROC (*Receiver Operating Characteristic*) teve origem na teoria de decisão estatística e foi desenvolvida entre 1950 e 1960 para avaliar a detecção de sinais em radar e na psicologia sensorial. A curva de característica de operação do receptor (ROC) é uma ferramenta amplamente utilizada na área de aprendizado de máquina e estatística para avaliar o desempenho de modelos de classificação binária. Ela é construída plotando a taxa de verdadeiros positivos (Sensibilidade) no eixo y versus a taxa de falsos positivos (1 - Especificidade) no eixo x. Um modelo ideal terá uma curva ROC que se aproxima do canto superior esquerdo do gráfico, indicando uma alta sensibilidade (alta taxa de verdadeiros positivos) e uma baixa taxa de falsos positivos. Por outro lado, um modelo aleatório terá uma curva ROC que se assemelha a uma linha diagonal, indicando que sua capacidade de discriminar entre as classes é equivalente à chance aleatória.

A curva ROC também fornece uma medida de desempenho chamada Área Sob a Curva (AUC), que mede a capacidade de discriminação do modelo. O valor da AUC varia de 0 a 1, onde 1 representa um modelo perfeito e 0,5 indica um modelo que é tão bom quanto o acaso. A curva ROC e a AUC são particularmente úteis quando os dados estão desequilibrados, ou seja, quando uma das classes é muito mais comum do que a outra. Elas permitem avaliar a capacidade de um modelo de classificar corretamente as instâncias da classe minoritária, mesmo em condições de desequilíbrio

2.13 Métricas de Avaliação

Para realizar uma análise precisa do desempenho do modelo, é essencial examinar os indicadores de desempenho relevantes. Essas métricas proporcionam uma avaliação quantitativa do desempenho do modelo em relação às tarefas específicas para as quais ele foi desenvolvido. A seguir, serão apresentadas as principais métricas utilizadas para avaliação dos modelos nesta monografia.

- **Precision (Precisão):** é a proporção de exemplos classificados corretamente como positivos em relação a todos os exemplos classificados como positivos (verdadeiros positivos + falsos positivos). Para cada classe, é calculada a precisão.
- **Recall (Revocação):** é a proporção de exemplos classificados corretamente como positivos em relação a todos os exemplos que realmente são positivos (verdadeiros positivos + falsos negativos). Para cada classe, é calculado o recall.
- **F1-score:** é a média harmônica entre precision e recall. É útil quando você deseja ter uma medida única que leve em consideração tanto a precisão quanto o recall. O

F1-score é uma métrica comum para avaliar modelos de classificação.

- **Support (Suporte):** é o número de ocorrências de cada classe no conjunto de dados.
- **Accuracy (Acurácia):** é a proporção de exemplos classificados corretamente em relação a todos os exemplos. É uma medida geral de desempenho do modelo.
- **Macro avg (Média Macro):** é a média das métricas para todas as classes, atribuindo igual peso para cada classe. Essa média dá a mesma importância para todas as classes, independentemente do tamanho.
- **Weighted avg (Média Ponderada):** é a média das métricas para todas as classes, ponderada pelo suporte de cada classe. Essa média leva em consideração a distribuição de classes e é útil quando as classes têm diferentes quantidades de exemplos.

3 APLICAÇÃO E RESULTADOS

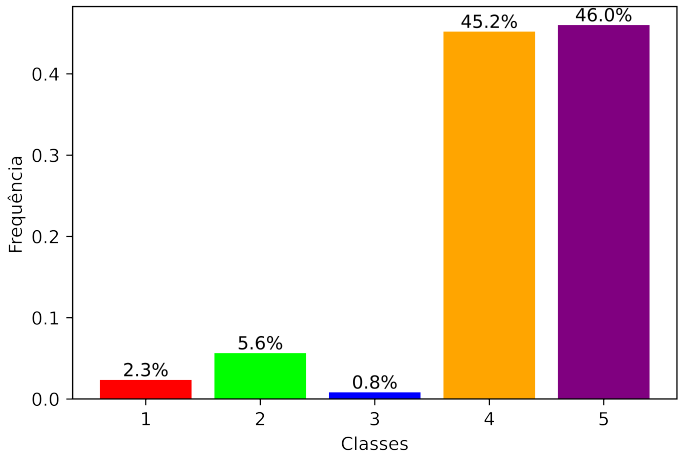
3.1 Resultados

Em uma primeira abordagem, realizou-se a limpeza dos dados, excluindo variáveis que não seriam relevantes para o nosso modelo, como o identificador do hospital, nome da unidade de saúde, estado da unidade de saúde, dentre as variáveis excluídas. Em seguida, foi feita a análise exploratória dos dados para identificar padrões e verificar o balanceamento da variável resposta, também conhecida como variável alvo.

Após o tratamento dos dados, restaram inicialmente 468.994 observações e 66 variáveis, cuja descrição está em anexo nesta monografia, o que evidencia a existência de um conjunto de dados robusto. Para reduzir o tempo de processamento durante o treinamento do modelo, optou-se por dividir o conjunto de dados. Dessa forma, uma parte dos dados foi reservada para treinar o modelo, enquanto a outra parte, à qual o modelo ainda não teve acesso, poderá ser utilizada futuramente em outros procedimentos, que serão discutidos posteriormente. Essa divisão resultou em 50% dos dados destinados à espera (dados a serem utilizados no futuro) e 50% para o treino e teste. Vale ressaltar que a divisão do conjunto de dados foi realizada de forma randomizada, mantendo a proporção do conjunto de dados inicial. Ou seja, utilizamos uma subamostra do conjunto de dados original para reduzir o custo computacional, além de permitir a utilização dos dados de espera no futuro, caso seja necessário.

Na Figura 4, é possível observar a frequência das classes do banco de dados em porcentagem:

Figura 4 – Proporção das Classes do Banco de Dados em Porcentagem



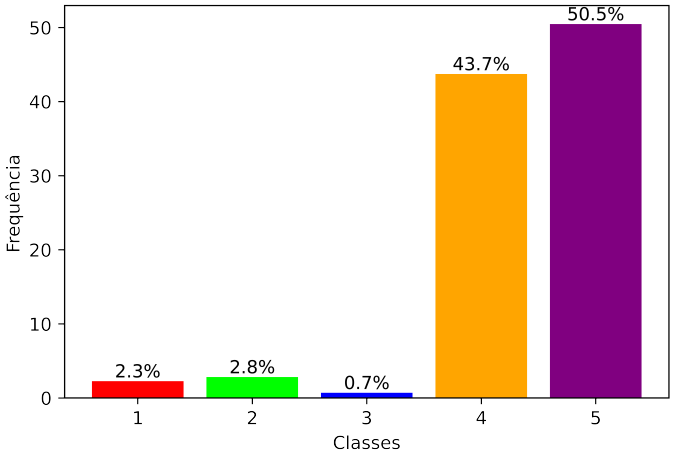
Fonte: Elaborada pelo Autor, 2023

Como pode ser observado, o conjunto de dados está bastante desbalanceado, sendo necessário fazer o balanceamento dos dados. É importante que o banco de dados reduzido mantenha a proporção equivalente ao conjunto de dados original, para que se possa aplicar corretamente os procedimentos.

3.1.1 Após a Divisão dos Dados

Na Figura 5, é possível observar a frequência das classes do dataset reduzido em porcentagem:

Figura 5 – Proporção das Classes em Porcentagem do Banco de Dados Reduzidos



Fonte: Elaborada pelo Autor, 2023

Após a divisão dos dados em dados de espera e dados para treino e teste, no intuito de aumentar as informações sobre a classe minoritária (classe 3), então dividimos mais 50% dos dados de espera, filtramos os valores que são equivalentes a classe 3, uniremos o dataset(banco de dados) que ficou como base reduzida ao dataset filtrando com base na classe 3, após esse procedimento ficamos com a seguinte proporção dos dados:

Tabela 2 – Tabela de Proporções do Dataset Para Treino e Teste

Classe	**Proporção (%)**
1	2,33%
2	5,61%
3	1,43%
4	44,92%
5	45,72%

Fonte: Elaborada pelo Autor, 2023

Embora tenham sido inseridos novos dados referentes à classe 3, pode-se perceber que a proporção referente a essa classe não aumentou significativamente. No entanto, isso já era esperado devido ao desbalanceamento substancial no volume de dados. Agora, é possível prosseguir para o próximo passo, que consiste em executar o *undersampling* nos dados.

3.1.2 Aplicando o Undersampling

Conforme discutido anteriormente, o *undersampling* é uma técnica utilizada no campo de aprendizado de máquina para tratar conjuntos de dados desbalanceados, nos quais uma ou mais classes estão significativamente sub-representadas em relação às outras. Nesse contexto, o *undersampling* envolve a redução do número de amostras da classe majoritária (ou classes majoritárias) para equilibrar a distribuição das classes. Após aplicar a subamostragem, foram obtidas 15.390 observações, sendo 3.078 observações para cada classe. O próximo passo consiste em dividir os dados em conjuntos de treinamento e teste, a fim de prosseguir para a etapa de criação do modelo.

3.1.3 Modelo 1: Árvore de Decisão

O primeiro modelo utilizado será a árvore de decisão, um método amplamente empregado em problemas de classificação. O algoritmo de árvore de decisão utilizado foi o *Classification and Regression Trees*(CART), Na Tabela 3, encontram-se as métricas referentes ao desempenho desse modelo:

Tabela 3 – Métricas do Modelo de Árvore de Decisão

Classe	Precisão	Revocação	F1-Score	Suporte
1	0,91	0,92	0,92	616
2	0,96	0,96	0,96	615
3	0,55	0,57	0,56	616
4	0,60	0,58	0,59	616
5	0,87	0,84	0,85	615
Acurácia			0,78	3078
Média (Macro)	0,78	0,78	0,78	3078
Média Ponderada	0,78	0,78	0,78	3078

Fonte: Elaborada pelo Autor, 2023

Essa é uma matriz de resultados de classificação, conhecida como relatório de classificação ou classification report. Ele fornece várias métricas para avaliar o desempenho de um modelo de classificação. Pode-se notar que o modelo apresenta um desempenho razoável, com uma acurácia de 78%. As classes 1, 2 e 5 apresentam bons resultados, com precision, recall e F1-score acima de 85%. Entretanto, As classes 3 e 4 têm resultados um pouco inferiores, com F1-score em torno de 56% e 59%, respectivamente. Vamos criar outros modelos e verificar se os resultados melhoram.

3.1.4 Modelo 2: Extreme Gradient Boosting (XGBoost)

O segundo modelo será o XGboost, também muito usando em problemas de classificação multiclasse. Na Tabela 4, é possível observar as métricas do modelo XGBoost:

Tabela 4 – Métricas do Modelo XGBoost

Classe	Precisão	Revocação	F1-Score	Suporte
1	0,99	0,93	0,95	616
2	0,96	0,99	0,98	615
3	0,67	0,57	0,62	616
4	0,64	0,75	0,69	616
5	0,91	0,92	0,92	615
Acurácia			0,83	3078
Média (Macro)	0,84	0,83	0,83	3078
Média Ponderada	0,84	0,83	0,83	3078

Fonte: Elaborada pelo Autor, 2023

Conforme observado, com o modelo XGBoost, houve melhorias nas métricas das classes 3 e 4. Além disso, o modelo apresenta métricas excelentes para as classes 1, 2 e 5. Isso indica que o modelo está desempenhando bem na classificação dessas classes específicas.

3.1.5 Modelo 3: Suport Vector Machine

O terceiro modelo será o SVM com kenel polinomial também muito usando em problemas de classificação multiclasse. Na tabela 5, é possível observar as métricas do modelo SVM:

Tabela 5 – Métricas do Modelo SVM Kenel Polinomial

Classe	Precisão	Revocação	F1-Score	Suporte
1	0,98	0,90	0,94	616
2	0,96	0,99	0,97	615
3	0,54	0,54	0,54	616
4	0,55	0,67	0,60	616
5	0,94	0,76	0,84	615
Acurácia			0,77	3078
Média (Macro)	0,79	0,77	0,78	3078
Média Ponderada	0,79	0,77	0,78	3078

Fonte: Elaborada pelo Autor, 2023

Pode-se observar que o modelo SVM teve um desempenho inferior em relação aos modelos anteriores, especialmente nas classes 3 e 4.

3.1.6 Modelo 4: Random Forest

O próximo modelo utilizado foi o Random Forest, Na Tabela 6, é possível observar as métricas do modelo Ramdom Forest:

Tabela 6 – Métricas do Modelo Random Forest

Classe	Precisão	Revocação	F1-Score	Suporte
1	1,00	0,90	0,95	616
2	0,91	0,99	0,95	615
3	0,66	0,45	0,54	616
4	0,58	0,82	0,68	616
5	0,93	0,86	0,89	615
Acurácia			0,80	3078
Média (Macro)	0,82	0,80	0,80	3078
Média Ponderada	0,82	0,80	0,80	3078

Fonte: Elaborada pelo Autor, 2023

Pode-se notar que o modelo apresenta uma melhora em relação ao modelo Svm, entretanto, as métricas ainda estão abaixo do modelo XGBoost.

3.1.7 Modelo 5: XGBoost com Seleção de Atributos

Na Tabela 7 pode-se observar as métricas do XGBoost com seleção de atributos:

Tabela 7 – Métricas do Modelo XGBoost com Seleção de Atibutos

Classe	Precisão	Revocação	F1-Score	Suporte
1	1,00	0,90	0,95	616
2	0,96	0,98	0,97	615
3	0,52	0,49	0,50	616
4	0,56	0,66	0,61	616
5	0,92	0,90	0,91	615
Acurácia			0,78	3078
Média (Macro)	0,79	0,79	0,79	3078
Média Ponderada	0,79	0,78	0,79	3078

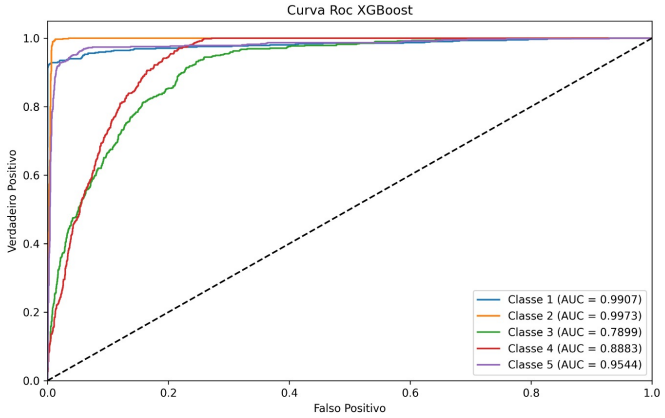
Fonte: Elaborada pelo Autor, 2023

Pode-se notar que o desempenho do modelo utilizando a seleção de atributos teve uma diminuição em sua performance geral, isso indica que possivelmente, mesmos as covariáveis não ter apresentando importância, elas provavelmente contribui de certa forma para a predição do modelo.

3.1.8 Curva Roc

Será feita a observação da curva ROC do melhor modelo, XGBoost(modelo: 2), o qual exibirá a curva ROC para cada classe. O XGBoost é amplamente reconhecido como um algoritmo eficiente e poderoso para problemas de classificação em aprendizado de máquina. A curva ROC é uma ferramenta valiosa para avaliar o desempenho de um modelo de classificação. Na Figura 6 é possível observar os resultados obtidos:

Figura 6 – Curva Característica de Operação do Receptor(ROC)



Fonte: Elaborada pelo Autor, 2023

Embora os resultados obtidos pareçam satisfatórios, ela não é a métrica mais indicada para esse tipo de cenário. A curva ROC é amplamente utilizada em problemas de classificação binária, nos quais existem apenas duas classes a serem previstas. Ela representa a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos em diferentes limiares de classificação. No entanto, ao lidar com problemas de multiclassificação, em que existem mais de duas classes, a curva ROC se torna menos informativa. A principal razão para isso é que a curva ROC é calculada para uma classe versus todas as outras, o que significa que é necessário realizar várias comparações para cada classe individualmente. Esse processo pode levar a uma análise complexa e menos clara do desempenho do modelo. Em vez disso, é preferível utilizar outras métricas mais apropriadas para problemas de multiclassificação como foi apresentado na Tabela 4. Além disso, outra forma de avaliar o modelo é através da matriz de confusão.

3.1.9 Matriz de Confusão do Melhor Modelo

Na sequência, será feita a matriz de confusão para o modelo que obteve as melhores métricas, o modelo Extreme Gradient Boosting (XGBoost). A Tabela 8 a seguir, apresenta as classificações feitas pelo modelo:

Tabela 8 – Matriz de Confusão do melhor modelo: XGBoost

	Classe 1	**Classe 2**	**Classe 3**	**Classe 4**	**Classe 5**
Classe 1	570	1	12	18	15
Classe 2	0	609	1	0	5
Classe 3	3	15	354	219	25
Classe 4	3	0	140	464	9
Classe 5	2	8	19	19	567

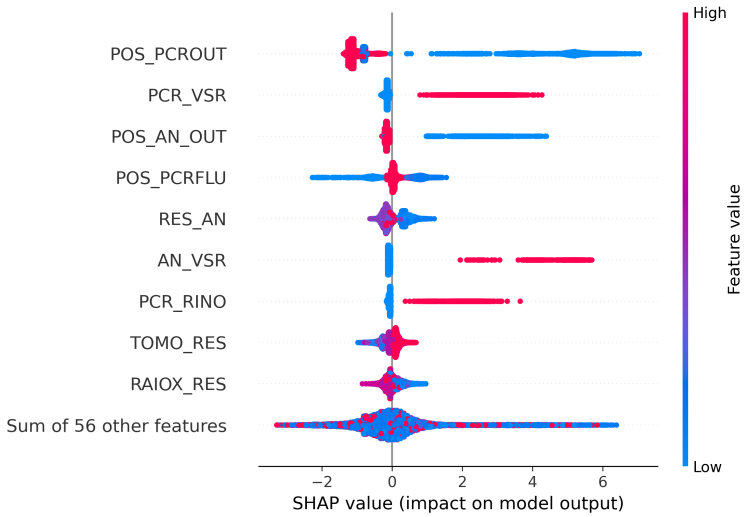
Fonte: Elaborada pelo Autor, 2023

Pode-se observar que as classes 1, 2 e 5 obtiveram boas classificações, enquanto as classes 3 e 4 apresentaram classificações moderadas. É notável, no entanto, que os erros de classificação se concentraram nessas duas últimas classes. Por exemplo, quando o modelo classificou incorretamente um item como pertencente à classe 4, na maioria dos casos, ele o rotulou como pertencente à classe 3, e o mesmo ocorreu para a classe 3, onde a maioria das classificações erradas foi rotulada como classe 4 pelo modelo. Esses resultados sugerem uma proximidade nos atributos dessas duas classes encontrados durante a etapa de treinamento pelo modelo.

3.1.10 Importância dos Atributos Shapley

Por último, será gerado o gráfico de importância dos atributos para o modelo, utilizando o método dos atributos Shapley, que permitirá visualizar e analisar a contribuição individual de cada atributo na tomada de decisões do modelo. Isso fornecerá uma compreensão mais clara e detalhada de como cada variável influencia as previsões realizadas pelo modelo. Na Figura 7 podemos ver a importância das covariáveis para o modelo XGboost:

Figura 7 – Importância dos Atributos Shapley



Fonte: Elaborada pelo Autor, 2023

Pode-se observar na Figura 7 acima, as 9 covariáveis que exercem o maior impacto nas predições do modelo. Entre elas, destacam-se POS_PCROUT(Resultado da

RTPCR foi positivo para outro vírus respiratório), PCR_VSR (Resultado diagnóstico do RTPCR para (VSR)) , POS_AN_OUT(Resultado do Teste Antigênico, que foi positivo para outro vírus respiratório) e POS_PCRFLU(Resultado da RTPCR foi positivo para Influenza), que apresentaram maior relevância para o modelo. É importante ressaltar que, apesar dessas 9 covariáveis serem as mais importantes, a interação entre os outros 56 atributos também desempenha um papel relevante para o modelo.

4 CONCLUSÃO

Neste trabalho foram demonstradas algumas das principais técnicas de machine learning para classificação multiclasse, visando aprimorar a capacidade de previsão de síndromes respiratórias agudas graves (SRAG). Ao longo da pesquisa, foram utilizados diferentes modelos de classificação, sendo o XGBoost o que apresentou melhores resultados em relação às métricas de classificação para as cinco classes analisadas. Essa abordagem permitiu uma performance geral de 83%, destacando-se especialmente na classificação de SRAG por influenza, SRAG por Covid, além das SRAGs por outros vírus respiratórios. No entanto, as classes 3 e 4 demonstraram apenas uma performance razoável, ficando ligeiramente acima do nível aleatório. Essa ocorrência pode ser atribuída a diversos fatores. Embora tenhamos realizado o balanceamento dos dados através do undersampling, equilibrando todas as classes, é possível que as classes 1, 2 e 5 tenham definições mais concretas de seus atributos. Em contraste, a classe 3 abrange a SRAG por outros agentes etiológicos, o que pode significar a existência de múltiplos agentes, impactando a distribuição dos atributos e dificultando sua classificação. De maneira similar, a classe 4 refere-se a um tipo de SRAG que não pôde ser especificado com base em seus atributos. Consequentemente, embora o modelo XGBoost tenha apresentado desempenho geral superior em relação aos demais modelos, é necessário considerar o aprimoramento das métricas nas classes 3 e 4. Para prosseguir com a implementação, é recomendável explorar outras técnicas de processamento de dados, bem como estratégias adicionais de balanceamento e ajustes no modelo. Melhorar o desempenho do modelo é crucial, especialmente considerando o contexto da área de saúde

Após analisar a matriz de confusão do melhor modelo, notou-se que a maioria das predições incorretas das classes 3 e 4 estava concentrada em sua proximidade, indicando uma forte semelhança entre os atributos (características) dessas duas classes. Isso pode ter contribuído para as dificuldades em distingui-las corretamente e resultou em um número maior de classificações equivocadas entre elas. Com tudo, pode-se notar que, mesmo com as ressalvas, o modelo é capaz de prever com excelente performance a classificação de SRAG por influenza, SRAG por Covid, além das SRAGs por outros vírus respiratórios. Vale ressaltar que, quando estamos trabalhando em áreas como a saúde, é importante ter o menor erro possível, pois o erro pode custar vidas.

REFERÊNCIAS

BARBOSA, F. R. M. Otimização de hiperparâmetros em algoritmos de árvore de decisão utilizando computação evolutiva. UNIVERSIDADE FEDERAL DO TOCANTINS, p. 16, 2018. Citado na página 16.

BEZERRA, F. A.; GRANDE, J. F.; SILVA, A. J. d. Análise e caracterização de modelos de custos que utilizam o valor de shapley para alocação de custos entre departamentos. *Gestão Produção*, Universidade Federal de São Carlos, v. 16, n. 1, p. 74–84, Jan 2009. ISSN 0104-530X. Disponível em: <https://doi.org/10.1590/S0104-530X2009000100008>. Citado na página 24.

BRAGA, A. C. da S. Curvas roc: Aspectos funcionais e aplicaÇÕes. Universidade do Minho, p. 32, dez 2000. Citado na página 26.

BRASIL, M. da S. *Saiba como é feita a definição de casos suspeitos de Covid-19 no Brasi*. 2021. <https://brasilescola.uol.com.br/doencas/sindrome-respiratoria-aguda-grave-sars.html>. Citado na página 12.

CECCON, D. Xgboost: A evolução das Árvores de decisão. *IA EXPERT ACADEMY*, abr 2019. Disponível em: <https://iaexpert.academy/2019/04/18/xgboost-a-evolucao-das-arvores-de-decisao/#:~:text=Os%20modelos%20baseados%20em%20%C3%A1rvores,prefer%C3%AAncia%20dos%20profissionais%20da%20%C3%A1rea.> Citado 2 vezes nas páginas 20 e 21.

Data Science Academy. *Curso: Machine Learning*. 2021. <https://www.datascienceacademy.com.br/course/machine-learning-engineer>. Acesso em: 9 jun. 2023. Citado na página 21.

ESPINOSA, R.; JIMéNEZ, F.; PALMA, J. *Multi-surrogate assisted multi-objective evolutionary algorithms for feature selection in regression and classification problems with time series data*. [S.l.: s.n.], 2023. v. 662. 1064-1091 p. Citado na página 24.

FERNANDEZ, A. et al. *Learning from imbalanced data sets*. [S.l.]: Springer, 2018. Citado na página 25.

KOZIARSKI, M. Subamostragem baseada em radial para classificação de dados desequilibrados. *Reconhecimento de padrões*, v. 102, n. 3, 2020. Citado na página 25.

LAURETTO, M. S. Árvores de decisão. USP, p. 02, 2010. Citado 2 vezes nas páginas 17 e 18.

MACEDO, M. D.; CHARLES. O que é machine learning? 2017. Citado na página 15.

MORAIS, R. L. de. Uso de Árvores aleatórias para classificação sensorial de arroz cozido. *Universidade de Brasília*, p. 22–25, 2017. Citado na página 19.

OLIVEIRA, B. B. Estudo dos métodos de classificação supervisionada na identificação de malignidade de tumores mamários. Universidade de São Paulo Instituto de Matemática e Estatística, p. 14, 2021. Citado na página 16.

Python Software Foundation. *Python Language Reference*. [S.l.], 2023. Acesso em: [data de acesso]. Disponível em: <https://www.python.org/doc/>. Citado na página 14.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2023. Disponível em: <https://www.R-project.org/>. Citado na página 14.

ROSA, A. L. da. Classificação de imagens de frutas utilizando aprendizado de máquina. UNIVERSIDADE FEDERAL DE SANTA CATARINA, p. 31, 2019. Citado na página 16.

SANTOS, G. C. Algoritmos de machine learning para previsão de ações da b3. Universidade Federal de Uberlândia Faculdade de Engenharia Elétrica, p. 50, 2020. Citado 2 vezes nas páginas 16 e 19.

SANTOS, V. S. dos. *Síndrome respiratória aguda grave (Sars)*. 2023. <https://brasilescola.uol.com.br/doencas/sindrome-respiratoria-aguda-grave-sars.html>. Acesso em: 9 jun. 2023. Citado na página 12.

SHAPLEY, L. S. A value for n-person games. In: *Contributions to the Theory of Games*. [S.l.]: Princeton University Press, 1953. v. 2, n. 28, p. 307–317. Citado na página 25.

SILVA, R. A. D. AutomedicaÇÃo relacionada À sÍndrome gripal entre adultos jovens em tempos de pandemia da covid – 19. VITÓRIA DE SANTO ANTÃO, p. 14, 2022. Citado na página 12.

SRINIVAS, R. Managing large data sets using support vector machines. *https://digitalcommons.unl.edu*, 2010. Citado na página 21.

5 ANEXO A: DESCRIÇÃO DAS VARIÁVEIS UTILIZADAS

Tabela 1 – Descrição das Variáveis do SRAG OpenDataSUS Utilizadas

Variável	**Tipo**	**Categoria**	**Descrição**
OUTRO_DES	Texto		Sinais e Sintomas/Outros
FLUASU_OUT	Texto		Outro subtipo para Influenza A.
FLUBLI_OUT	Texto		Outra linhagem para Influenza B.
TOMO_OUT	Texto		Informar o resultado da tomografia se selecionado a opção 5-Outro.
NOSOCOMIAL	Categórica	1-Sim 2-Não 9-Ignorado	Caso de SRAG com infecção adquirida após internação.
AVE_SUINO	Categórica	1-Sim 2-Não 9-Ignorado	Caso com contato direto com aves ou suínos.
FEBRE	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou febre?
TOSSE	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou tosse?
GARGANTA	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou dor de garganta?
DISPNEIA	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou dispneia?
DESC_RESP	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou desconforto respiratório?
SATURACAO	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou saturação $O2 < 95\%$?
DIARREIA	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou diarreia?
VOMITO	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou vômito?
OUTRO_SIN	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou outro(s) sintoma(s)?
PUERPERA	Categórica	1-Sim 2-Não 9-Ignorado	Paciente é puérpera ou parturiente (mulher que pariu recentemente – até 45 dias do parto)?
FATOR_RISC	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresenta algum fator de risco
CARDIOPATI	Categórica	1-Sim 2-Não 9-Ignorado	Paciente possui Doença Cardiovascular Crônica?

Variável	Tipo	Categoria	Descrição
DIABETES	Categórica	1-Sim 2-Não 9-Ignorado	Paciente possui Diabetes mellitus?
NEUROLOGIC	Categórica	1-Sim 2-Não 9-Ignorado	Paciente possui Doença Neurológica?
PNEUMOPATI	Categórica	1-Sim 2-Não 9-Ignorado	Paciente possui outra pneumopatia crônica?
IMUNODEPRE	Categórica	1-Sim 2-Não 9-Ignorado	Paciente possui Imunodeficiência ou Imunodepressão (diminuição da função do sistema imunológico)?
RENAL	Categórica	1-Sim 2-Não 9-Ignorado	Paciente possui Doença Renal Crônica?
OBESIDADE	Categórica	1-Sim 2-Não 9-Ignorado	Paciente possui obesidade?
OBES_IMC	Númerico		Valor do IMC (Índice de Massa Corporal) do paciente calculado pelo profissional de saúde.
OUT_MORBI	Categórica	1-Sim 2-Não 9-Ignorado	Paciente possui outros fatores de risco?
SUPORT_VEN	Categórica	1-Sim invasivo 2-Sim, não invasivo 3-Não 9-Ignorado	paciente fez uso de suporte ventilatório?
RAIOX_RES	Categórica	1-Normal 2-Infiltrado intersticial 3-Consolidação 4-Misto 5-Outro 6-Não realizado 9-Ignorado	Informar resultado de Raio X de Tórax
PCR_RESUL	Categórica	1-Detectável 2-Não Detectável 3-Inconclusivo 4-Não Realizado 5-Aguardando Resultado 9-Ignorado	Resultado do teste de RT-PCR/outro método por Biologia Molecula
POS_PCRFLU	Categórica	1-Sim 2-Não 9-Ignorado	Resultado da RTPCR foi positivo para Influenza
TP_FLU_PCR	Categórica	1-Influenza A 2-Influenza B	Resultado diagnóstico do RTPCR para o tipo de Influenza.
PCR_FLUASU	Categórica	1-Influenza A(H1N1)pdm09 2-Influenza A	Subtipo para Influenza A.

Variável	Tipo	Categoria	Descrição
PCR_FLUBLI	Categórica	1-Victoria 2-Yamagatha 3-Não realizado 4-Inconclusivo 5-Outro, especifique:	Linhagem para Influenza B.
POS_PCROUT	Categórica	1-Sim 2-Não 9-Ignorado	Resultado da RTPCR foi positivo para outro vírus respiratório
PCR_VSR	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para (VSR).
PCR_PARA1	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Parainfluenza 1.
PCR_PARA2	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Parainfluenza 2.
PCR_PARA3	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Parainfluenza 3.
PCR_PARA4	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Parainfluenza 4.
PCR_ADENO	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Adenovírus
PCR_METAP	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Metapneumovírus
PCR_BOCA	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Bocavírus.
PCR_RINO	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Rinovírus
PCR_OUTRO	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado diagnóstico do RTPCR para Outro vírus respiratório.

Variável	Tipo	Categoria	Descrição
DOR_ABD	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou dor abdominal?
FADIGA	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou fadiga?
PERD_OLFT	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou perda do olfato?
PERD_PALA	Categórica	1-Sim 2-Não 9-Ignorado	Paciente apresentou perda do paladar?
TOMO_RES	Categórica	1-Tipico covid-19 2- Indeterminado covid-19 3- Atípico covid-19 4- Negativo para Pneumonia 5- Outro 6-Não realizado 9-Ignorado	Informar o resultado da tomografia.
TP_TES_AN	Categórica	1- Imunofluorescência (IF) 2- Teste rápido antigênico	Tipo do teste antigênico que foi realizado
RES_AN	Categórica	1-positivo 2-Negativo 3- Inconclusivo 4-Não realizado 5-Aguardando resultado 9-Ignorado	Resultado do Teste Antigênico
POS_AN_FLU	Categórica	1-Sim 2-Não 9-Ignorado	Resultado do Teste Antigênico que foi positivo para Influenza
TP_FLU_AN	Categórica	1-Influenza A 2-Influenza B	Resultado do Teste Antigênico, para o tipo de Influenza
POS_AN_OUT	Categórica	1-Sim 2-Não 9-Ignorado	Resultado do Teste Antigênico, que foi positivo para outro vírus respiratório.
AN_SARS2	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico, para SARS-CoV-2.
AN_VSR	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico, para VSR.
AN_PARA1	Categórica	1-marcado pelo usuário	Resultado do Teste Antigênico, para Parainfluenza 1.

Variável	**Tipo**	**Categoria**	**Descrição**
AN_PARA3	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico. Parainfluenza 3.
AN_ADENO	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico. Adenovírus.
AN_ADENO	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico. Adenovírus.
AN_OUTRO	Categórica	1-marcado pelo usuário Vazio - não marcado	Resultado do Teste Antigênico. Outro vírus respiratório.
CLASSI_FIN	Categórica	1-SRAG por influenza 2-SRAG por outro vírus respiratório 3-SRAG por outro agente etiológico, qual: 4-SRAG não especificado 5-SRAG por covid-19 pelo usuário	Diagnóstico final do caso.