RELATÓRIO FINAL DE INICIAÇÃO CIENTÍFICA

NOME DO ALUNO: Willian Penteado
ORIENTADORA: Sandra Mara Guse Scós Venske
REFERENTE AO PERÍODO: 01/09/2022 a 31/08/2023
TÍTULO: Prognóstico para Pacientes com COVID-19 Usando Técnica de Aprendizado de Máquina
PALAVRAS-CHAVE: aprendizado supervisionado, problema de classificação, COVID-19, AdaBoost.

RESUMO
A doença do coronavírus 2019 (COVID-19) é uma doença infecciosa altamente contagiosa causada pelo coronavírus relacionada à síndrome respiratória aguda grave 2 (*Severe Acute Respiratory Syndrome Coronavirus* - SARS-CoV-2). A doença teve um efeito catastrófico no mundo, resultando em milhões de mortes em todo o mundo. Este estudo teve como objetivo utilizar Aprendizado de Máquina (AM), que é o processo de indução de uma hipótese a partir de experiências anteriores, para o desenvolvimento de modelo para apoiar o prognóstico para pacientes com COVID-19. Como conjunto de dados foram utilizados dados público pertencentes ao Sistema Único de Saúde (SUS) do Ministério da Saúde do Brasil. Os dados do SUS eram altamente incorretos, inconsistentes, duplicados, ausentes e desbalanceados. Técnicas de pré-processamento foram aplicadas resultando em conjuntos de dados que podem ser utilizados em outros estudos.

Sumário

1 INTRODUÇÃO

A Covid-19 é uma grave doença respiratória causada pelo coronavírus SARS-CoV-2, altamente transmissível e globalmente disseminada. Este vírus foi identificado pela primeira vez em pacientes com pneumonia em Wuhan, China, em dezembro de 2019. É o sétimo coronavírus a afetar seres humanos, e sua origem animal ainda não foi confirmada (Saúde, 2021).

Ao longo dos anos, uma quantidade substancial de dados, incluindo informações clínicas e registros de tratamento de doenças, tem sido acumulada e armazenada por governos e instituições hospitalares. Diante desse cenário, surge a necessidade de explorar esses dados, visando contribuir para a melhoria da saúde pública. Nesse contexto, destaca-se o uso do Aprendizado de Máquina, o qual ganhou relevância não apenas na área da saúde, mas também em diversos outros campos do conhecimento, fundamentado na teoria de que computadores podem aprender com os dados disponíveis (Batista and Chiavegatto, 2019).

O campo do Aprendizado de Máquina (AM), tem ganhado destaque à medida que a complexidade e volume dos dados de diferente áreas aumentam. Com isso, surge a necessidade de ferramentas computacionais mais sofisticadas e autônomas, capazes de lidar com essa quantidade de informações sem depender tanto da intervenção humana e de especialistas. O processo de inferir uma hipótese a partir de experiências anteriores é conhecido como Aprendizado de Máquina (Faceli et al., 2011).

Este estudo propôs o estudo do aprendizado supervisionado de máquina, mais especificamente do algoritmo AdaBoost, a ser aplicado para o desenvolvimento de modelo para apoiar o prognóstico para pacientes com COVID-19. O conjunto de dados utilizado é público e pertencente ao Sistema Único de Saúde (SUS) do Ministério da Saúde do Brasil correspondente a vigilância da Síndrome Respiratória Aguda Grave (SRAG) no Paraná no ano de 2022. O conjunto de dados é composto por exames clínicos, sintomas, datas, dados demográficos e geográficos, sendo inicialmente composto por 561.242 instâncias e 166 atributos, com uma grande quantidade de valores incorretos, inconsistentes, duplicados e ausentes, além de desbalanceado. Técnicas de pré-processamento foram aplicadas a fim de padronizar o conjunto de dados com instâncias corretas para que o modelo de predição gerado pela técnica de AM Adaboost fosse confiável.

Este relatório está organizado como segue. Na Seção 2 são apresentados os objetivos do trabalho. A Seção 3 apresenta os conceitos necessário para o entendimento dos métodos e algoritmos utilizados. Na Seção 4 é explicada a abordagem proposta. Os resultados são apresentados e discutidos na Seção 5. Finalmente, na Seção 6 são apresentadas as conclusões e trabalhos futuros.

2 OBJETIVOS

2.1 Objetivo Geral

O objetivo geral deste plano de atividades foi o estudo do aprendizado supervisionado de máquina a ser aplicado para o desenvolvimento de modelo para apoiar o prognóstico para pacientes com COVID-19.

2.2 Objetivos Específicos

Dentre os objetivos específicos, destacam-se:

- Estudar algoritmos de aprendizado de máquina, mais especificamente, o algoritmo AdaBoost;
- Estudar sobre a doença Covid-19;
- Adquirir habilidade de leitura de textos técnicos, especialmente em inglês.

3 MATERIAIS E MÉTODOS

3.1 Aprendizado de Máquina

O conceito de aprendizagem de máquina é a ciência da programação de computadores para que eles possam aprender com base nos dados, sem serem exclusivamente programados para isso (Géron, 2019).

O Aprendizado de Máquina se destaca nos problemas muito complexos para abordagens tradicionais ou que não possuem um algoritmo conhecido. Como exemplos pode-se citar o reconhecimento de fala e o filtro de *spam* (Géron, 2019).

3.1.1 Aprendizado Supervisionado

Existem diferentes tipos de sistemas de Aprendizado de Máquina sendo que pode-se classificar em diferentes categorias com base em (Géron, 2019):

- Serem ou não treinados com supervisão humana, como aprendizado supervisionado, não supervisionado, semi-supervisionado e aprendizado por reforço;
- Se podem ou não aprender rapidamente, de forma incremental, como aprendizado *online* e aprendizado por lotes;
- Se funcionam simplesmente comparando novos pontos de dados com pontos de dados conhecidos, ou se detectam padrões em dados de treinamento e criam um modelo preditivo, como o aprendizado baseado em instâncias e o aprendizado baseado em modelo.

Neste trabalho foi optado pelo Aprendizado de Máquina Supervisionado, mais especificamente o AdaBoost, para implementação de um classificador para apoiar o prognóstico para pacientes com COVID-19. Neste tipo de aprendizado os dados carregados são divididos em dois conjuntos, um para treino do algoritmo e outro para realização de testes. A parte do *dataset* que é composta pelos resultados é chamada de rótulo. Os rótulos são a principal diferença desse tipo de aprendizado dos demais.

3.2 Dataset

O conjunto de dados (*dataset*) utilizado neste trabalho é público pertencente ao Sistema Único de Saúde (SUS) do Ministério da Saúde. Foi retirado da base de dados do OpenDataSUS[1] e corresponde a vigilância da Síndrome Respiratória Aguda Grave (SRAG) no Brasil no ano de 2022. Composto por exames clínicos, sintomas, datas, dados demográficos e geográficos.

Inicialmente o *dataset* era composto por 561.242 linhas (instâncias) e 166 colunas (atributos), com uma grande quantidade de valores faltantes e inconsistentes.

É importante destacar que foi realizada uma ampla análise no dicionário de dados[2].

3.3 Técnica AdaBoost

O AdaBoost é um algoritmo de AM que utiliza uma abordagem iterativa para melhorar o desempenho do classificador. Começa com a criação de um classificador de base, como uma Árvore de Decisão[3], que é usado para fazer previsões no conjunto de treinamento. Em seguida, ele aumenta o peso das instâncias classificadas incorretamente e treina um novo classificador com base nesses pesos atualizados. Esse processo é repetido várias vezes, resultando em um modelo forte que combina vários modelos fracos, cada um focando em corrigir as deficiências do anterior (Géron, 2019).

3.4 Ferramentas utilizadas

Para a realização deste projeto, foram utilizadas as ferramentas:

- *Frameworks* e Bibliotecas: *scikit-learn*[4], pandas[5], *NumPy*[6], *Matplotlib*[7], *imbalanced-learn*[8] e *seaborn*[9];

[1] https://opendatasus.saude.gov.br/dataset/srag-2021-a-2023/resource/62803c57-0b2d-4bcf-b114-380c392fe825

[2] https://s3.sa-east-1.amazonaws.com/ckan.saude.gov.br/SRAG/pdfs/Dicionario_de_Dados_SRAG_Hospitalizado_19.09.2022.pdf

[3] Árvore de Decisão é um método de Aprendizado de Máquina que utiliza nós hierárquicos para decisões binárias ou multiclasse, com base em atributos do *dataset*, onde o nó raiz representa o atributo mais relevante. Os nós-folha indicam resultados que correspondem às classes ou categorias pertinentes ao problema (Géron, 2019).

[4] https://scikit-learn.org/stable/

[5] https://pandas.pydata.org/

[6] https://numpy.org/

[7] https://matplotlib.org/

[8] https://imbalanced-learn.org/stable/index.html

[9] https://seaborn.pydata.org/

- Ambiente de desenvolvimento: *Google Colaboratory*[10];
- Plataforma *GitHub*[11] utilizada para gerenciar versionamento.

3.5 Métricas de Avaliação

Neste trabalho foram utilizadas duas métricas para avaliar o desempenho dos modelos. Essas métricas são acurácia e F_1-*score*. Tais métricas são detalhadas nas Subseções 3.5.1 e 3.5.2.

As métricas consideradas utilizam as seguintes medições:

- VP são as instâncias classificadas como *Verdadeiros Positivos*;
- VN são as instâncias classificadas como *Verdadeiros Negativos*;
- FP são as instâncias classificadas como *Falsos Positivos*; e
- FN são as instâncias classificadas como *Falsos Negativos*.

3.5.1 Métrica Acurácia

A acurácia do diagnóstico é dada pelo número de exemplos classificados corretamente dividido pelo número total de exemplos classificados e é calculada conforme a Equação 1 (Shaikh et al., 2021).

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{1}$$

3.5.2 Métrica F_1-*score*

A métrica F_1-*score* combina as métricas de precisão (Equação 2) e sensibilidade (Equação 3) em uma única métrica. A Equação 4 define a métrica F_1 (Shalev-Shwartz and Ben-David, 2014).

$$Precisão = \frac{VP}{VP + FP} \tag{2}$$

$$Sensibilidade = \frac{VP}{VP + FN} \tag{3}$$

$$F_1 = 2 \times \frac{Precisão \times Sensibilidade}{Precisão + Sensibilidade} = \frac{VP}{VP + \frac{FN+FP}{2}} \tag{4}$$

4 Abordagem proposta

Neste trabalho, a metodologia adotada é apresentada na Figura 1, dividida em três estágios: análise e estudo do *dataset*, pré-processamento (com foco principal) e modelagem. Cada um desses estágios será discutido nesta seção.

No primeiro estágio, denominado Análise e Estudo do *Dataset*, foram realizadas três tarefas:

1. Primeiro Carregamento do *Dataset*: Nesta etapa, o *dataset* foi carregado pela primeira vez, com o objetivo de visualizar e analisar os dados no ambiente de desenvolvimento. Isso inclui a compreensão do tamanho do *dataset*, o tipo de dados presentes e a identificação de possíveis problemas.

2. Delimitação do *Dataset*: Ao observar que o *dataset* possuía um grande número de atributos, o que poderia prejudicar as previsões do modelo devido à alta dimensionalidade dos dados, enfrentando o desafio conhecido como a "Maldição da Dimensionalidade" que será detalhado na Subsubseção 4.1.1, foram tomadas as seguintes ações:

   Primeiro, foi realizada uma exploração inicial, ainda que superficial, do dicionário de dados para identificar os atributos que poderiam ser reduzidos. Em seguida, foi identificado o atributo que classificava as

[10] https://colab.google/
[11] https://github.com/

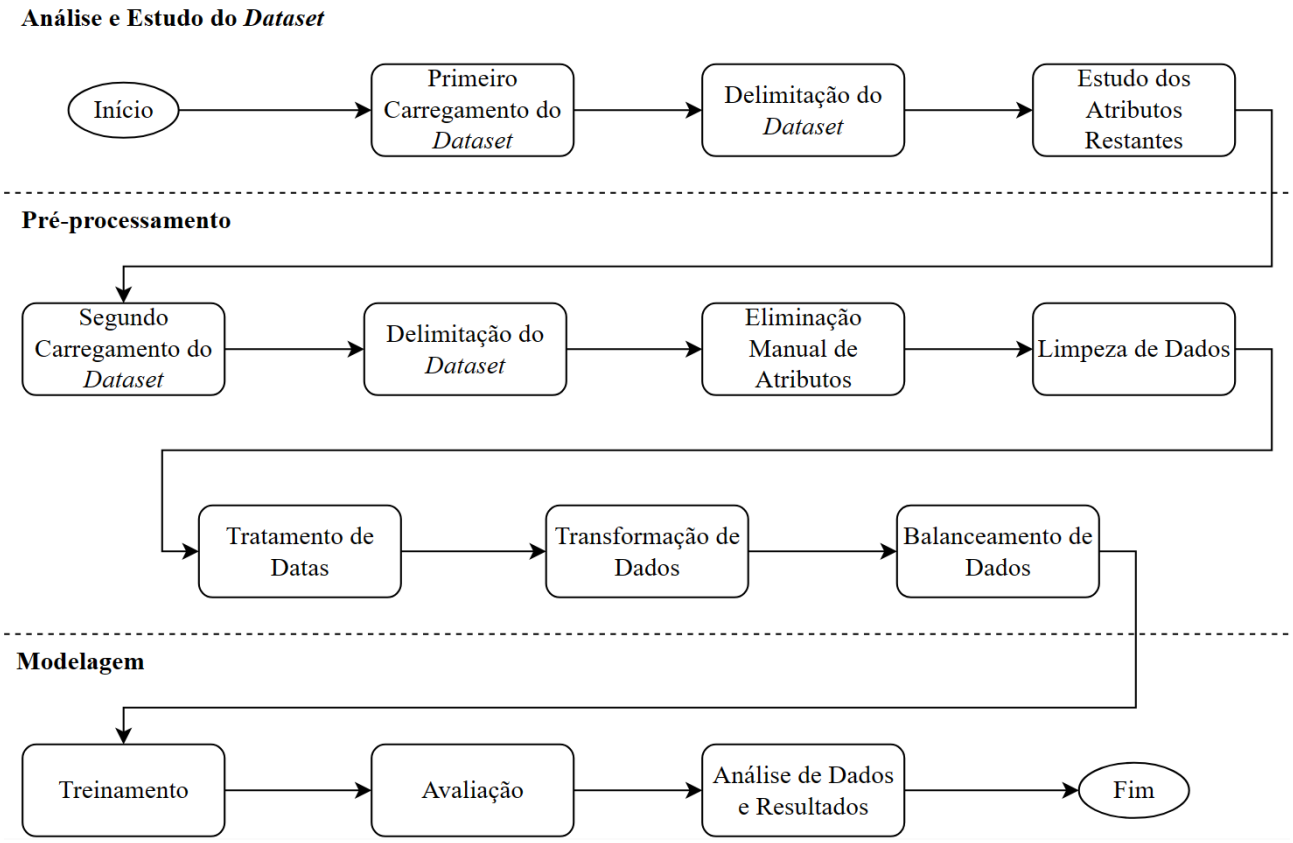

Figura 1: Fluxograma da metodologia. Fonte: o autor.

notificações como relacionados à Covid-19, lembrando que o *dataset* originalmente se referia às SRAGs em geral, não apenas à Covid-19.

Posteriormente, foi decidido restringir o *dataset* apenas às notificações do estado do Paraná, identificando o atributo que representava o estado. Com base nessa delimitação, aplicou-se um filtro para eliminar atributos e instâncias que continham mais de 50% de valores ausentes. Esse processo resultou na remoção de oitenta e um atributos, com o objetivo de reduzir o tamanho do *dataset* antes de prosseguir para uma análise mais detalhada, que representa a próxima etapa deste estágio.

3. Estudo dos Atributos Restantes: Após a redução do número de atributos para oitenta e cinco na etapa anterior, os atributos remanescentes foram submetidos a uma análise mais detalhada com o auxílio do dicionário de dados. Isso permitiu uma compreensão mais aprofundada do significado de cada atributo e orientou as decisões sobre os próximos passos a serem seguidos neste trabalho.

Após a análise realizada nesta etapa, o *dataset* original foi carregado em um novo projeto para ser submetido a um processo de pré-processamento, que é detalhado na Subseção 4.1.

4.1 Pré-processamento de dados

O desempenho dos algoritmos de AM pode ser impactado pelas condições dos dados. Dessa forma, conjuntos de dados podem estar limpos e livre de imperfeições, ou pode conter conter ruídos e problemas, como valores incorretos, inconsistentes, duplicados ou ausentes (Faceli et al., 2021).

Com o objetivo de melhorar a qualidade dos dados e minimizar problemas, técnicas de pré-processamento são aplicadas. Com o auxílio delas, os dados refletem de forma mais precisa os valores reais e os modelos gerados tornam-se mais confiáveis. Também é possível, com o pré-processamento é possível reduzir a complexidade computacional e facilitar o ajuste de parâmetros do modelo, além de adequar os dados, se necessário, para um determinado algoritmo de AM (Faceli et al., 2021).

Nas seções seguintes apresentadas algumas técnicas de pré-processamento utilizadas neste trabalho...

4.1.1 Limitando o *Dataset*

O ano de 2022 foi escolhido por conter mais dados completos, incluindo informações de vacinas e os dados mais recentes.

Como este *dataset* é composto por notificações relacionadas as SRAGs no Brasil no ano de 2022, o primeiro tratamento realizado foi limitar para apenas notificações de Covid-19.

Após isso, uma das estratégias adotadas foi a limitação para dados apenas do estado do Paraná, a fim de obter informações mais específicas e relevantes para o estudo em questão, e também podendo assim eliminar alguns atributos relacionados a localização.

Trabalhar com *datasets* de alta dimensionalidade, ou seja, *datasets* que possuem um grande número de atributos para cada instância de dados, apresenta desafios significativos. Essa complexidade torna o treinamento de modelos mais demorado e a busca por soluções de qualidade consideravelmente mais desafiadora. Esse fenômeno é chamado de"Maldição da Dimensionalidade" (Géron, 2019).

O atributo alvo, possuía quatro possíveis valores: "1 - Recuperado", "2 - Óbito", "3 - Óbitos por outras causas" e "9 - Ignorado". No entanto, para focar no objetivo deste trabalho, as instâncias com valores "3" foram removidas, já que não eram relevantes para o estudo. Além disso, as instâncias com valor "9" foram eliminadas prezando pela qualidade dos dados a fim de gerar um modelo mais preciso.

A Figura 2 resume a delimitação realizada no *dataset*.

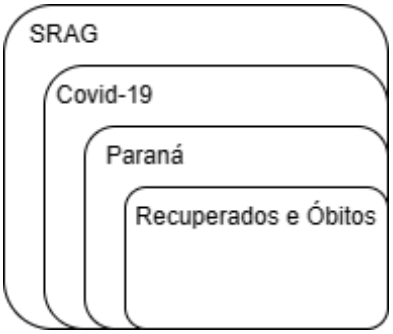

Figura 2: Delimitação do *dataset*. Fonte: o autor.

Após este processamento, o número de instâncias passou de 561.242 para 17.699. O número de atributos não foi alterado.

4.1.2 Eliminação Manual de Atributos

A eliminação manual de atributos é o processo de identificar, avaliar, e remover atributos de um *dataset* que são considerados irrelevantes, redundantes, de baixa qualidade ou que não contribuem significativamente para tarefa de análise ou modelagem de dados em questão (Faceli et al., 2021). Nesta etapa foram considerados os seguintes critérios para remoção:

1. Devido à limitação do *dataset* às notificações de Covid-19 no estado do Paraná, os atributos relacionados a esses valores foram eliminados.

2. Atributos categorizados como "Campos internos" no dicionário de dados foram removidos como parte de uma estratégia para simplificar o *dataset* e concentrar a análise em outros aspectos.

3. Embora se reconheça a importância dos dados geográficos, eles foram omitidos neste estudo. Portanto, os atributos relacionados à localização foram retirados do *dataset*.

4. Os atributos que continham informações sobre lote, fabricante e laboratório da vacina foram excluídos.

5. Foram eliminados os atributos que não estavam documentados no dicionário de dados.

6. Atributos que foram identificados como datas de natureza administrativa foram retirados do *dataset*.

7. Outros atributos foram descartados, pois não se esperava que contribuísse de maneira significativa para o estudo.

Os atributos eliminados foram: “CLASSI_FIN”, “SG_UF_NOT”, “SEM_NOT”, “SEM_PRI”, “ID_REGIONA”, “CO_REGIONA”, “ID_RG_RESI”, “CO_RG_RESI”, “FNT_IN_COV”, “ID_RG_INTE”, “CO_RG_INTE”, “DT_DIGITA”, “ID_MUNICIP”, “CO_MUN_NOT”, “ID_UNIDADE”, “CO_UNI_NOT”, “ID_PAIS”, “CO_PAIS”, “SG_UF”, “ID_MN_RESI”, “CO_MUN_RES”, “CS_ZONA”, “ESTRANG”, “SG_UF_INTE”, “ID_MN_INTE”, “CO_MU_INTE”, “LOTE_1_COV”, “LOTE_2_COV”, “LOTE_REF”, “FAB_COV_1”, “FAB_COV_2”, “FAB_COVREF”, “LAB_PR_COV”, “HISTO_VGM”, “PCR_SARS2”, “DT_ENCERRA”, “DT_COLETA”, “AVE_SUINO”, “CS_RACA”.

Após este processamento, O número de atributos no *dataset* passou de 166 para 127. O número de instâncias não foi alterado.

4.1.3 Limpeza de Dados

Dificuldades em *datasets* podem estar relacionadas à qualidade dos dados. A limpeza dos dados é o processo de identificar, corrigir e/ou remover erros, inconsistências, redundâncias e valores ausentes ou incorretos de um *dataset*. Ela visa melhorar a qualidade e a integridade dos dados, bem como evitar que imperfeições nos dados afetem negativamente as análises ou modelos subsequentes (Faceli et al., 2021).

- **Redundância de dados**: A redundância de dados em instâncias acontece quando uma instância apresenta valores de atributos muito similares aos de outra instância. Além disso, a redundância pode ocorrer nos próprios atributos, caracterizada quando o valor de um atributo pode ser inferido por meio de outro. A redundância de dados é uma preocupação significativa, pois pode levar a uma atribuição incorreta de importância por parte de alguns modelos de AM (Faceli et al., 2021).

  No estudo em questão, foram identificados quatro atributos para representar a idade. Para tratar esse problema, foram realizadas eliminações manuais de três atributos: “DT-NASC”, “TP-IDADE” e “COD-IDADE”. O atributo “NU-IDADE-N” foi mantido como representante da idade, simplificando assim o *dataset* e evitando redundâncias desnecessárias.

- **Dados ausentes:** Valores ausentes referem-se à falta de dados em um *dataset*, ou seja, a ausência de informações em um ou mais atributos para certas instâncias. Isso pode ocorrer por várias razões, como erros na coleta, armazenamento, transmissão de dados ou simplesmente porque algumas informações não estão disponíveis ou não foram registradas (Faceli et al., 2021). É importante lidar adequadamente com dados ausentes, pois eles afetam a qualidade das análises e previsões. Frequentemente pode-se ignorar ou reparar esses dados, preenchendo com outros valores prováveis. Porém, um número excessivo de valores ausentes impactará em previsões mais imprecisas (Mueller and Massaron, 2019).

  Na análise do dicionário de dados, identificou-se que vários atributos do tipo *float* continham o valor “9”, indicando “ignorado”. Esses valores devem ser considerados como dados ausentes, uma vez que não foram preenchidos adequadamente. Para solucionar esse problema, realizou-se um tratamento nos dados, transformando todos os atributos do tipo *float* com valor “9” em NaN (*Not a Number*), um tipo especial de dado utilizado para representar valores ausentes. Essa ação visou garantir que os valores ausentes fossem tratados de forma apropriada durante análises subsequentes, evitando interpretações inadequadas por parte do modelo.

  Neste estudo, foram aplicadas duas abordagens, com foco na qualidade dos dados em vez da quantidade. A primeira estratégia envolveu a remoção das instâncias e atributos que apresentavam mais de 50% de seus dados como faltantes. Posteriormente, para os dados remanescentes, realizou-se o preenchimento dos valores ausentes com a mediana de cada atributo. Essas ações visaram garantir a integridade e confiabilidade dos dados, como parte essencial da análise.

  A escolha da mediana para preenchimento dos dados se deu pelo fato dela ser menos sensível a valores atípicos (*outliers*) do que a média, pois ela se baseia no valor central quando os dados são ordenados em ordem crescente. Isso significa que um valor atípico, que seja muito maior ou muito menor do que os demais valores, terá menos impacto na mediana do que na média (Faceli et al., 2021).

Após este processamento o *dataset* ficou com 17.432 instâncias e 42 atributos.

4.1.4 Tratamento de datas

As datas são fontes ricas de informação que podem ser utilizadas com modelos de AM. No entanto, estas variáveis de data requerem alguma engenharia de características para as transformar em dados numéricos (Faceli et al., 2021).

Foram identificados sete atributos do tipo "object" que originalmente representavam datas nos dados em questão. Para melhorar a capacidade do modelo de AM em extrair informações relevantes desses campos, foram implementados tratamentos desses dados. Os atributos de data originais eram: "DT_NOTIFIC", "DT_SIN_PRI", "DT_INTERNA", "DT_PCR", "DT_EVOLUCA", "DOSE_1_COV", "DOSE_2_COV".

Para otimizar o tratamento desses dados, novos atributos todos do tipo "int" foram criados: "QTD_DIAS", "DIAS_INTERNA", "SINT_ATE_NOTIF", "PCR_EVOLUCAO", "DIAS_DOSE2", "DIAS_DOSE1".

- **QTD_DIAS:** Este atributo foi criado com o propósito de representar a duração do tratamento dos pacientes, que é definida como o período decorrido entre a data de notificação (DT_NOTIFIC) e a data de evolução (DT_EVOLUCA). Foi notado que o atributo DT_EVOLUCA continha algumas instâncias com valores ausentes (NaN), o que resultou na atribuição de NaN a essas instâncias, com a finalidade de tratamento posterior.

  Após a determinação de todos os intervalos de tempo, foi calculado o valor da mediana do atributo QTD_DIAS. Assim, as instâncias que continham valores NaN foram preenchidas com o valor da mediana, assegurando a integridade do atributo. E por último, para refletir a natureza dos dados, o atributo QTD_DIAS foi convertido de um tipo de dado *float* para *int*.

- **DIAS_INTERNA:** Para representar a duração da internação de um paciente, este atributo foi criado. Ele armazena o intervalo entre duas datas: a data de internação (DT_INTERNA) e a data de evolução (DT_EVOLUCA). A aplicação desse cálculo ocorre apenas nas instâncias em que o atributo "HOSPITAL" foi classificado como "1", indicando que o paciente foi internado. No caso das instâncias em que os atributos de data estavam ausentes, foram marcadas com valores NaN, a serem tratadas posteriormente. As instâncias que não atendiam ao critério de "HOSPITAL" igual a "1" receberam o valor 0.

  Uma vez calculados todos os intervalos de tempo, procedeu-se ao cálculo da mediana para o atributo DIAS_INTERNA. Em seguida, as instâncias com valores NaN foram preenchidas com o valor da mediana, garantindo, dessa forma, a consistência do atributo. Para melhor refletir a natureza dos dados, o atributo DIAS_INTERNA foi convertido do tipo de dado *float* para *int*.

- **SINT_ATE_NOTIF:** O atributo SINT_ATE_NOTIF foi introduzido com objetivo de registrar a quantidade de dias que decorrem desde o surgimento dos primeiros sintomas até a data de notificação do paciente. Para calcular essa quantidade de dias, a relação entre os atributos DT_SIN_PRI e DT_NOTIFIC foi estabelecida. Importante ressaltar que ambos os atributos não continham dados ausentes, o que permitiu o cálculo direto da quantidade de dias. Como não se encontravam valores NaN em nenhum momento do processo, não houve a necessidade de conversão do tipo de dados, uma vez que o resultado já estava no formato *int*.

- **PCR_EVOLUCAO:** Este atributo foi gerado para retratar a diferença de dias entre a data em que o teste RT-PCR[12] foi realizado e a data de evolução do paciente. O cálculo dessa diferença utiliza os atributos DT_EVOLUCA e DT_PCR. No entanto, esse cálculo é efetuado apenas para instâncias em que o atributo "PCR_RESUL" difere do valor "4", indicado que o teste foi realizado.

  É importante destacar que ambos os atributos, DT_EVOLUCA e DT_PCR, apresentavam algumas instâncias com valores ausentes. Nessas instâncias, foram atribuídos valores NaN, que posteriormente foram preenchidos pelo valor da mediana do atributo PCR_EVOLUCAO. Finalmente, para melhor refletir a natureza dos dados, o atributo PCR_EVOLUCAO foi convertido de *float* para *int*.

- **DIAS_DOSE2:** O atributo DIAS_DOSE2 foi criado com o propósito de registrar a quantidade de dias decorridos desde a administração da segunda dose da vacina contra a Covid-19 até a data de notificação do paciente. Esse cálculo utiliza os atributos DOSE_2_COV e DT_NOTIFIC. É importante destacar que esse cálculo é efetuado apenas para as instâncias em que o atributo "VACINA_COV" é igual a "1", indicando que o paciente recebeu a vacina. Para as demais instâncias, esse atributo recebe o valor 0, com exceção dos casos de valores ausentes que recebem NaN.

  Vale ressaltar que o atributo DOSE_2_COV continha algumas instâncias com valores ausentes, que posteriormente foram preenchidos com a mediana do atributo DIAS_DOSE2. Além disso, realizou-se a conversão do tipo de dado de *float* para *int*, para melhor representar a natureza dos dados.

[12] O exame PCR é um teste que serve para diagnóstico de diversas doenças além da Covid-19. Fonte: https://www.gov.br/saude/pt-br/assuntos/noticias/2022/fevereiro/entenda-as-diferencas-entre-rt-pcr-antigeno-e-autoteste

- **DIAS_DOSE1:** O atributo DIAS_DOSE1 foi introduzido no *dataset* com o propósito de registrar a quantidade de dias decorridos desde a administração da primeira dose da vacina contra a Covid-19 até a data de notificação do paciente.

  Inicialmente, durante a análise desses dados, foi identificada uma inconsistência em que uma instância continha a data da segunda dose, mas não a da primeira. Esse cenário poderia levar a mais incoerências nos dados, já que poderiam ocorrer instâncias em que os dias entre a segunda dose e a notificação fossem maiores do que os dias entre a primeira dose e a notificação, o que não é possível no contexto da vacinação.

  Para resolver esse problema, foi criado um novo atributo auxiliar denominado "INTERVALO_VACINAS", que guarda a diferença de dias entre a primeira e a segunda dose da vacina. Foi usado a mediana desse atributo para preencher alguns valores ausentes.

  Com o atributo DIAS_DOSE1 em vigor, os cálculos foram baseados nas datas da primeira dose da vacina (DOSE_1_COV) e da data de notificação (DT_NOTIFIC),e em algumas situações, na data da segunda dose da vacina (DOSE_2_COV).

  Essa análise foi limitada às instâncias em que o atributo "VACINA_COV" era igual a "1", indicando que o paciente havia recebido a vacina. Para instâncias com valores diferentes de "1", o atributo DIAS_DOSE1 foi definido como 0.

  O processo começou com a verificação do atributo "VACINA_COV" para determinar se o paciente recebeu a vacina. Em seguida, calculou-se a quantidade de dias entre a data da primeira dose da vacina (DOSE_1_COV) e a data de notificação (DT_NOTIFIC). Caso tanto DOSE_1_COV quanto DOSE_2_COV estivessem ausentes (NaN), o atributo DIAS_DOSE1 recebia NaN, para tratamento posterior. Se apenas DOSE_1_COV estivesse ausente, uma data auxiliar era criada com base na data da segunda dose da vacina, e então subtraída a mediana previamente calculada do atributo auxiliar "INTERVALO_VACINAS". A diferença entre essa data auxiliar e a data de notificação era calculada e atribuída a DIAS_DOSE1.

  Posteriormente, a mediana do atributo DIAS_DOSE1 foi calculada, e os valores NaN nas instâncias foram preenchidos com essa mediana. Por fim, o tipo de dados foi convertido de *float* para *int*, e o atributo auxiliar "INTERVALO_VACINAS" foi removido do *dataset*.

4.1.5 Dados Desbalanceados

Na área de classificação de dados, é comum deparar-se com o desafio dos dados desbalanceados ao trabalhar com *datasets* reais. Isso ocorre quando o número de instâncias em um subconjunto de classes é significativamente maior em comparação com outras classes, resultando em uma disparidade na distribuição dos dados. A presença desse desequilíbrio pode afetar negativamente o desempenho de alguns algoritmos de AM (Faceli et al., 2021). Este problema é visualizado na Figura 3.

Para resolver este problema, foi adotada a técnica de redefinir o tamanho do *dataset*. Desta forma foram testadas três diferentes abordagens, conforme os tópicos abaixo.

- Subamostragem (*under-sampling*): Nesta abordagem, com base no *dataset* original, essa técnica visa reduzir o número de amostras da classe majoritária (Recuperado), a fim de equilibrá-la com a classe minoritária. O algoritmo "*Near-Miss*" da biblioteca "*imbalanced-learn*" foi empregado para realizar a subamostragem. O *Near-Miss* executa a redução com base na distância entre as instâncias das duas classes, priorizando amostras da classe majoritária que se assemelham mais às da classe minoritária (Lemaître et al., 2017).

- Superamostragem (*over-sampling*): Nessa técnica, o foco está em aumentar o número de exemplos na classe minoritária, o oposto da técnica de subamostragem. Foi utilizado o algoritmo "*Synthetic Minority Over-sampling Technique*" (SMOTE), uma técnica da biblioteca do "*imbalanced-learn*", para equilibrar classes desbalanceadas em *datasets*. O SMOTE gera exemplos sintéticos para a classe minoritária, auxiliando algoritmos de aprendizado a lidar de forma mais eficaz com desequilíbrios.

- Outra abordagem adotada envolveu o pré-processamento separado de um segundo *dataset*, limitado ao estado de São Paulo (SP). Nesse contexto, o pré-processamento foi realizado de forma independente, aplicando um filtro para tratar dados faltantes, com uma taxa de 90%, e restringindo as instâncias apenas àquelas classificadas como "Óbito". Em seguida, esse *dataset* secundário foi mesclado ao *dataset* principal, contribuindo para o equilíbrio das classes.

Essa etapa gerou cinco versões diferente de *datasets*:

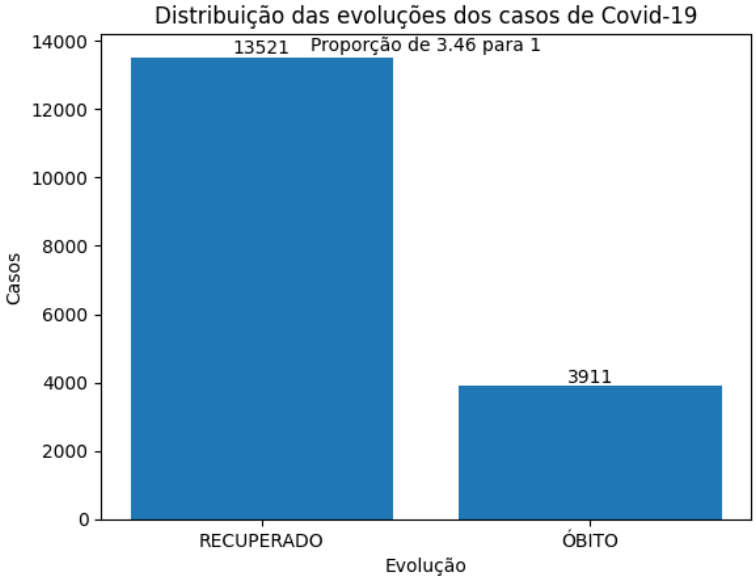

Figura 3: Desbalanceamento dos dados. Fonte: o autor.

1. **Nenhum**: Nessa versão, sem balanceamento, mantendo o *dataset* com 17.432 instâncias com classes desbalanceadas;
2. **NearMiss**: Esta versão, a técnica *Near-Miss* foi aplicada para equilibrar as classes, reduzindo o *dataset* para 7.822 instâncias;
3. **SMOTE100**: Aqui, o SMOTE foi aplicado para igualar a classe minoritária à classe majoritária, resultando em um total de 27.042 instâncias;
4. **SMOTE85**: Nessa abordagem, a técnica de balanceamento SMOTE foi empregada para elevar a classe minoritária a 85% da quantidade da classe majoritária. Resultando em 25.013 instâncias;
5. **DadosSP**: Essa versão envolveu o uso de dados de São Paulo para complementar a classe minoritária, expandindo o *dataset* para 25.210 instâncias.

Essas versões foram criadas para avaliar o desempenho dos modelos sob diferentes técnicas de balanceamento de classes.

4.2 Modelagem

Na fase de modelagem, estão inseridas as etapas de treinamento, avaliação e análise dos dados e resultados. Para a realização dessas etapas, foi utilizada a técnica de *holdout*, que consiste na divisão do *dataset* em dois conjuntos distintos: um conjunto de treinamento e outro de teste. No conjunto de treinamento, os modelos foram treinados, enquanto o conjunto de teste foi reservado para a avaliação de desempenho. Essa divisão ocorreu de forma estratificada, com 67% para o conjunto de treinamento e 33% ao conjunto de teste, garantindo a estratificação dos rótulos a fim de manter a distribuição original das classes.

Essa fase se desenvolve em:

- Treinamento: Nesta etapa inicial, os algoritmos passam por um processo de treinamento. Isso envolve a alimentação dos modelos com dados do conjunto de treinamento, permitindo que eles ajustem seus parâmetros internos. O objetivo principal é capacitar os modelos a aprender com os dados, tornando-os capazes de realizar classificações com base nas informações disponíveis.
- Avaliação: Após o treinamento, segue-se a etapa de avaliação, na qual se mede o desempenho dos modelos. Esta avaliação é realizada utilizando o conjunto de teste, que não foi utilizado durante o treinamento. Os modelos fazem previsões com base neste conjunto de teste, e suas previsões são comparadas com os valores reais para medir sua precisão. Neste trabalho, os modelos foram avaliados em termos de métricas como F_1-*score* e Acurácia.

- Análise dos Dados e Resultados: Após a avaliação, a análise dos dados e resultados é a etapa em que os resultados são examinados em detalhes, auxiliados por tabelas e matrizes de correlação, proporcionando uma compreensão mais profunda do desempenho dos modelos.

Para fins de transparência e replicação deste estudo, todo o código-fonte, dados e recursos relacionados estão disponíveis em um repositório público no *GitHub*. Os leitores interessados podem acessar o repositório: https://github.com/Willian-P/IniciacaoCientifica2-Covid19 para revisar e reproduzir as etapas do trabalho, permitindo uma análise detalhada e a validação dos resultados apresentados neste relatório.

5 RESULTADOS E DISCUSSÃO

Os parâmetros do classificador são apresentados nas Tabelas 1 e 2.

Sem Normalização de Dados	
Balanceamento	**Parâmetros Classificador AdaBoost**
Nenhum	estimator = LogisticRegression(max_iter=1000), learning_rate = 0.5, n_estimators = 150
NearMiss	estimator = LogisticRegression(max_iter=1000), learning_rate = 1.5, n_estimators = 100
SMOTE100	estimator = DecisionTreeClassifier(), learning_rate = 1.5, n_estimators = 50
SMOTE85	estimator = DecisionTreeClassifier(), learning_rate = 0.5, n_estimators = 150
DadosSP	estimator = LogisticRegression(max_iter=1000), learning_rate = 1.5, n_estimators = 150

Tabela 1: Parâmetros utilizados neste trabalho para os dados sem normalização.

StandardScaler	
Balanceamento	**Parâmetros Classificador AdaBoost**
Nenhum	estimator = LogisticRegression(max_iter=1000), learning_rate = 0.5, n_estimators = 100
NearMiss	estimator = LogisticRegression(max_iter=1000), learning_rate = 1.5, n_estimators = 150
SMOTE100	estimator = DecisionTreeClassifier(), learning_rate = 0.5, n_estimators = 100
SMOTE85	estimator = DecisionTreeClassifier(), learning_rate = 1.0, n_estimators = 100
DadosSP	estimator = LogisticRegression(max_iter=1000), learning_rate = 0.5, n_estimators = 150

Tabela 2: Parâmetros utilizados neste trabalho para os dados utilizando *StandardScaler*.

Balanceamento	**Acurácia**		***F_1-score***	
	Sem normalizar	**Normalizado**	**Sem normalizar**	**Normalizado**
Nenhum	0.795	0.795	0.328	0.328
NearMiss	0.788	0.800	0.787	0.787
SMOTE100	0.833	0.854	0.834	0.854
SMOTE85	0.828	0.843	0.814	0.832
DadosSP	0.772	0.768	0.762	0.759

Tabela 3: Tabela de resultados em termos de acurácia e F_1-*score*, com e sem normalização dos dados.

A Tabela 3 mostra os resultados em termos de acurácia e F_1-*score*, com e sem normalização dos dados para as 5 versões consideradas. Pode ser observada uma notável discrepância nas métricas de acurácia e F_1-*score* ao lidar com um *dataset* desbalanceado, com o F_1-*score* demonstrando maior robustez. Em outras configurações, as variações entre essas métricas são mais sutis.

A normalização dos dados não apresentou melhora significativas nos resultados, porém é percebido que nas versões em que instâncias sintéticas foram incorporadas por meio do método SMOTE, houve uma melhora mais notável.

No que diz respeito à acurácia, a versão "SMOTE100 Normalizado" alcançou o desempenho mais notável, com 85,4%, enquanto a versão "DadosSP Normalizado" registrou o desempenho mais baixo, com 76,8%.

Em termos de F_1-*score* a versão que obteve o melhor desempenho também foi a "SMOTE100 com Normalização" que atingiu 85,4% e a pior versão foi "Nenhum" independente de ser normalizado ou não com 32,8%.

Quanto ao F_1-*score*, a versão "SMOTE100 com Normalização" também se destacou, atingindo 85,4%, enquanto a versão "Nenhum", independentemente da normalização, obteve o pior desempenho, com 32,8%.

A análise conjunta das duas métricas revela que a versão "SMOTE100 Normalizado" apresentou o desempenho superior, atingindo 85,4% em ambas as métricas. Em segundo lugar, a versão "SMOTE85 Normalizado" registrou 83,2% de F_1-*score* e, destacando-se principalmente pela sua acurácia de 84,3% em comparação com a terceira colocada. Em terceiro lugar, a versão "SMOTE100 Sem normalizar" apresentou 83,3% e 83,4% de acurácia e F_1-*score*, respectivamente.

6 CONCLUSÕES

Este trabalho teve como objetivo o estudo do aprendizado supervisionado de máquina com a técnica Adaboost a fim de ser aplicado para o desenvolvimento de modelo para apoiar o prognóstico para pacientes com COVID-19.

Para o desenvolvimento, foram aplicadas várias técnicas de pré-processamento nos dados utilizados, já que estes, por serem coletados em ambientes reais, apresentaram diversos problemas. Ruídos em conjuntos de dados podem reduzir o desempenho das técnicas de AM. Este trabalho produziu um conjunto de dados tratado que pode ser utilizado em outros estudos.

Além disso, o trabalho produziu um modelo de AM com Adaboost, testando diferentes técnicas de balanceamento, considerando duas métricas.

Como trabalhos futuros, pode-se utilizar os conjuntos de dados tratados resultado deste projeto, aplicando outras técnicas de AM, incluindo redes neurais artificiais. É possível também aplicar a metodologia aqui desenvolvida para o pré-processamento do conjunto de dados do SUS considerando outros Estados, ou até o Brasil todo. Neste caso, técnicas de aprendizado profundo podem ser mais adequadas.

Referências

Batista, A. F. M. and Chiavegatto, A. D. P. (2019). Machine learning aplicado à saúde. In *Workshop: Machine Learning*. Sociedade Brasileira de Computação. Acesso em: 24 mai 2023.

Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. C. P. d. L. F. d. (2011). *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC, 1st edition.

Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. C. P. d. L. F. d. (2021). *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC, 2nd edition.

Géron, A. (2019). *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. Alta Books.

Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Mueller, J. and Massaron, L. (2019). *Aprendizado de Máquina Para Leigos*. Para Leigos. Alta Books.

Saúde, M. D. (2021). O que é a covid-19? https://www.gov.br/saude/pt-br/coronavirus/o-que-e-o-coronavirus. Accessed: 2023-05-23.

Shaikh, K., Krishnan, S., and Thanki, R. (2021). Breast cancer detection and diagnosis using ai. In *Artificial Intelligence in Breast Cancer Early Detection and Diagnosis*, pages 79–92. Springer.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press.

7 AVALIAÇÃO DO ORIENTADOR SOBRE O DESEMPENHO DO ORIENTADO

O acadêmico apresentou um excelente desempenho na realização das atividades relacionadas ao projeto da IC, desenvolvendo habilidades importantes para a carreira profissional, como iniciativa, poder de síntese, apresentação de ideias de forma clara, entre outras. Além disso, houve um crescimento bastante significativo nos conhecimentos técnicos relacionados.

Guarapuava, 17 de outubro de 2023.

Aluno

Orientadora