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

MARIA FAGNA FELIX DE SOUZA

USO DE *MACHINE LEARNING* PARA CLASSIFICAÇÃO DE RISCO DE ÓBITOS POR COVID-19 NO ESTADO DE MATO GROSSO.

CAMPINA GRANDE - PB
2022

MARIA FAGNA FELIX DE SOUZA

USO DE *MACHINE LEARNING* PARA CLASSIFICAÇÃO DE RISCO DE ÓBITOS POR COVID-19 NO ESTADO DE MATO GROSSO.

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
Coorientador: Profa. Dra. Ana Patrícia Bastos Peixoto

CAMPINA GRANDE - PB
2022

S586u Silva, Maria Fagna Felix da.

Uso de *Machine Learning* para classificação de risco de óbitos por Covid-19 no estado de Mato Grosso. [manuscrito] / Maria Fagna Felix da Silva. - 2022.

40 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Estatística) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia , 2022.

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

"Coorientação: Profa. Dra. Ana Patrícia Bastos Peixoto , Coordenação do Curso de Estatística - CCT."

1. Covid-19. 2. LightGBM. 3. Regressão logística. 4. Óbitos. 5. Estatística. I. Título

21. ed. CDD 519.5

Elaborada por Deivid A. M. Neves - CRB - 15/899 BC/UEPB

MARIA FAGNA FELIX DE SOUZA

USO DE *MACHINE LEARNING* PARA CLASSIFICAÇÃO DE RISCO DE ÓBITOS POR COVID-19 NO ESTADO DE MATO GROSSO.

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 01 de DEZEMBRO de 2022.

BANCA EXAMINADORA

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

Prof. Dr. Sílvio Fernando Alves Xavier Júnior
Universidade Estadual da Paraíba (UEPB)

Documento assinado digitalmente
gov.br
CRYSTTIAN ARANTES PAIXAO
Data: 06/12/2022 21:04:43-0300
Verifique em https://verificador.iti.br

Prof. Dr. Crysttian Arantes Paixão
Universidade Federal da Bahia (UFBA)

Dedico, a Deus, aos meus familiares e amigos.

AGRADECIMENTOS

Agradeço a Deus por ter sido minha fortaleza e me amparado em todos os momentos.

Aos meus familiares e amigos, em especial ao meu cônjuge, que sempre me apoiou, sendo o meu principal incentivador em todos os momentos, a quem eu agradeço pela paciência e por acreditar no meu potencial.

A minha mãe que sempre incentivou o estudo e a busca por conhecimento, e que me inspira desde a infância.

Aos meus amigos e colegas de curso, pelos momentos, experiência e conhecimentos compartilhados, em especial a Débora, Lucas, Samuel e Suziane e aos demais que em algum momento estiveram juntos nesta caminhada.

Ao meu orientador Prof. Dr. Tiago Almeida de Oliveira, com quem desenvolvi um projeto de iniciação cientifica, foi através dessa experiência que pude expandir o conhecimento para além da graduação, que contribuiu muito para meu crescimento profissional.

A minha coorientadora Profa. Dra. Ana Patrícia Bastos Peixoto, agradeço por todo apoio, compreensão e incentivo, que por seu intermédio tive a oportunidade de desenvolver um projeto de iniciação cientifica.

A todos os professores do Departamento de Estatística por compartilhar seu conhecimento e experiências ao longo desses anos, e em particular ao Prof. Dr. Sílvio Fernando Alves Xavier Júnior e ao Prof. Dr. Ricardo Alves de Olinda por todas as contribuições durante o curso.

À Universidade Estadual da Paraíba pela oportunidade de alcançar esse objetivo, a graduação.

“Em Deus nós confiamos; todos os outros devem trazer dados.”
(William Edwards Deming)

RESUMO

A crise sanitária global causada pela Covid-19, tornou indispensável o uso de dados provenientes dos prontuários de pacientes para obtenção de modelos preditivos capazes de contribuir com a tomada de decisão. O uso de *Machine Learning* foi introduzido em vários estudos. Neste contexto, o presente trabalho utilizou amostras de pacientes com Covid-19 do estado de Mato Grosso, e através de características biológicas e morbidades dos pacientes se propôs obter um modelo capaz de prever a evolução da doença, fazendo a distinção entre dois possíveis cenários, recuperado ou óbito. Os algoritmos utilizados foram Regressão Logística, *Random Forest*, *XGBoost*, *LightGBM* e *CatBoost*. Os resultados dos modelos se mostraram satisfatórios obtendo a área sob a curva (AUC) ROC superior a 0,80, ou seja, com uma alta taxa de verdadeiros positivos e baixa taxa de falsos positivos. As variáveis que mais contribuíram com o modelo foram a idade, comorbidade, hipertensão, diabetes e obesidade. O intuito de obter um modelo capaz de discriminar entre as duas classes possíveis foi atendido, ressaltando que uma análise aprofundada sobre a estrutura de causa e efeito entre as variáveis previsoras e a variável resposta não foi a premissa do estudo, mas a obtenção de um modelo aceitável capaz de distinguir satisfatoriamente entre as duas classes de interesse, recuperado ou óbito.

Palavras-chaves: Covid-19; LightGBM; Regressão logística; óbitos; Estatística.

ABSTRACT

The global health crisis caused by Covid-19 has made it essential to use data from patient records to obtain predictive models capable of contributing to decision-making. The use of *Machine Learning* has been introduced in several studies. In this context, the present work used samples of patients with Covid-19 in the state of Mato Grosso, and through the biological characteristics and morbidities of the patients, it was proposed to obtain a model capable of predicting the evolution of the disease, distinguishing between two possible scenarios, recovered or died. The algorithms used were *Logistic Regression*, *Random Forest*, *XGBoost*, *LightGBM* and *CatBoost*. The results of the models were satisfactory, obtaining the area under the curve (AUC) ROC greater than 0.80, that is, with a high rate of true positives and low rate of false positives. The results of the models were satisfactory, obtaining the area under the curve (AUC) ROC greater than 0.80, that is, with a high rate of true positives and low rate of false positives. The variables that most contributed to the model were age, comorbidity, hypertension, diabetes and obesity. The aim of obtaining a model capable of discriminating between the two possible classes was met, emphasizing that an in-depth analysis of the cause and effect structure between the predictor variables and the response variable was not the premise of the study, but obtaining a model acceptable able to distinguish satisfactorily between the two classes of interest, recovered or death.

Keywords: Covid-19; LightGBM; Logistic regression; deaths; Statistic.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A crise sanitária da Covid-19 (SARS-Covid-19) originária da província de Wuhan, na China, em dezembro de 2019, teve seu primeiro caso identificado no Brasil em 25 de fevereiro de 2020. Apesar do governo federal ter declarado estado de emergência em 4 de fevereiro de 2020, adotando medidas para conter a chegada do vírus chinês, a primeira morte causada pelo novo coronavírus ocorreu em 12 março de 2020. A Organização Mundial da Saúde (OMS) já havia declarado estado global de emergência em 30 de janeiro do mesmo ano, em 11 de março foi declarado pandemia, quando existiam mais de 100 mil casos em mais de 100 países e cerca de 4,2 mil mortes. No Brasil, 69 casos de Covid-19 haviam sido identificados e confirmados pelo Ministério da Saúde (SAÚDE, 2022).

Devido a velocidade de propagação do vírus, logo o Brasil tornou-se um dos países com mais casos e mortes por número de habitantes, ocupando a posição 14º no ranking mundial, com 3.214 mortes a cada milhão de habitantes no final de setembro de 2022, com aproximadamente 80% da população vacinada até 2º dose da vacina ou dose única. O cenário descrito ocorreu mesmo com evidências de que medidas não farmacológicas são eficazes no controle da transmissão do Covid-19, por diminuírem a transmissão da doença e contribuir para a redução de casos graves, diminuindo o número de pessoas hospitalizadas (GARCIA et al., 2020).

Desde de então, tornou-se indispensável o uso de dados provenientes da Covid-19 para obtenção de modelos preditivos, com capacidade de contribuir para tomada de decisão de forma mais assertiva. A observação de características biológicas, sintomatológicas e morbidades são utilizadas para construção de modelos capazes de prever a evolução da doença. Estudo realizado no primeiro trimestre de 2020, utilizou-se de informações de prontuário de pacientes da região de Wuhan para obter um modelo de *Machine Learning* (ML) capaz de selecionar biomarcadores capazes de predizer a sobrevida de pacientes individuais (COSTA, 2020; YAN et al., 2020).

Diversos estudos científicos consideram que indivíduos portadores de determinadas comorbidades, entre elas, obesidade, diabetes, hipertensão e outras doenças crônicas, assim como fatores biológicos, como idade e sexo, apresentaram maiores riscos de morte por Covid-19. É importante frisar que no Aprendizado de Máquina o fato de algum atributo em estudo ter forte influência sobre a variável resposta não quer dizer necessariamente que ele seja um bom previsor, ou seja, bons preditores não precisam causar o desfecho, apenas predizer (BATISTA; FILHO, 2019; MASCARELLO et al., 2021).

O Aprendizado de Máquina, apesar de ter surgido na década de 1950, apenas em meados da década de 2000 passou a ser amplamente conhecido. A necessidade de que máquinas pudessem ser programadas para realizar tarefas especificas foi o que propiciou o surgimento do *Machine Learning*. Atualmente, diversos algoritmos foram desenvolvidos para realização de determinadas tarefas. No campo do aprendizado supervisionado, destacam-se os algoritmos, *Random Forest*, *eXtreme Gradient boosting*, *Light Gradient Boosting Machine* e *CatBoost*. O ML possui suas bases teóricas porém permite flexibilização, pois diferentemente da estatística não se restringe

aos pressupostos matemáticos (SILVA, 2020).

Devido o crescente volume de dados que são criados diariamente, o uso da inteligência artificial (IA) faz-se cada dia mais necessário, sendo o ML a área da IA mais amplamente utilizada. Diferentemente dos modelos tradicionais de regressão, que modela covariáveis e suas reações, em *Machine Learning* o interesse é combinar um grande número de previsores lidando com números maiores de observações de modo a ter resultados superiores aos métodos tracionais. Diante deste cenário, a necessidade do uso de métodos modernos para reduzir incertezas na tomada de decisão baseada em dados é crucial (BATISTA; FILHO, 2019; SCHLEDER; FAZZIO, 2021).

Neste contexto, o presente trabalho tem por finalidade realizar a previsão da evolução de casos de Covid-19, com base em algumas características dos indivíduos, seja a presença de fatores biológicos, como idade e sexo, ou presença de doenças crônicas. Para isso, utilizou-se algoritmos de *Machine Learning* para obter um modelo capaz de prever o desfecho da doença, ou seja, se óbito ou recuperado. Além de obter um modelo que seja um bom previsor, avalia-se também quais das variáveis estudadas foram mais importantes na construção do modelo. Os dados são de pacientes do Estado de Mato Grosso referente o ano de 2020, obtidos por meio do site da secretaria de saúde do estado.

2 REVISÃO DE LITERATURA

Este capítulo apresenta a base teórica dos algoritmos de *Machine Learning*, são discutidas as estruturas básicas dos principais algoritmos, bem como as principais métricas de avaliação de desempenho de modelos e os possíveis problemas de aprendizagem e previsões. Aborda-se as principais técnicas utilizadas no aprendizado supervisionado para a subárea de classificação.

2.1 Aprendizado de Máquina

Entende-se por aprendizado de máquina, ou do inglês *"Machine Learning"*, projetar algoritmos que possam extrair informações valiosas dos dados. O objetivo é obter modelos eficientes e que generalizem bem para dados futuros e não conhecidos pelo o algoritmo. O aprendizado é entendido como a forma que o algoritmo encontra padrões e estrutura nos dados, e começa a aprender, seja por memorização ou experiência. É um campo de pesquisa que cruza as linhas da estatística, inteligência artificial e ciência da computação (DEISENROTH; FAISAL; ONG, 2020).

Atualmente, tem-se conhecimento de alguns tipos de aprendizado de máquina, em todos eles, o algoritmo aprende com os dados. A diferença do aprendizado supervisionado para o não supervisionado é a existência de rótulos, ou seja, o algoritmo recebe os dados com rótulos e tem como tarefa identificar padrões e em seguida aplicar aos novos dados. Já no aprendizado não supervisionado, o algoritmo aprende a encontrar padrões e separar as observações em grupos de acordo com as suas características comuns, no aprendizado não supervisionado, somente os dados de entrada são conhecidos e nenhum dado de saída conhecido é fornecido ao algoritmo (MUELLER; GUIDO, 2016).

Outro tipo de aprendizado que deve ser mencionado é o aprendizado semi-supervisionado, nesta situação, os dados possuem observações rotuladas e não rotuladas e o processo posterior é o mesmo do aprendizado supervisionado. Já o aprendizado por reforço é o aprendizado pela a experiência, o algoritmo aprende a maneira correta de realizar a tarefa por meio da tentativa e erro, sendo recompensado quando acerta e sofrendo punição quando erra. Em linhas gerais, a escolha de cada tipo de aprendizado depende do problema e da estrutura dos dados que se tem a disposição (IGNACIO, 2021).

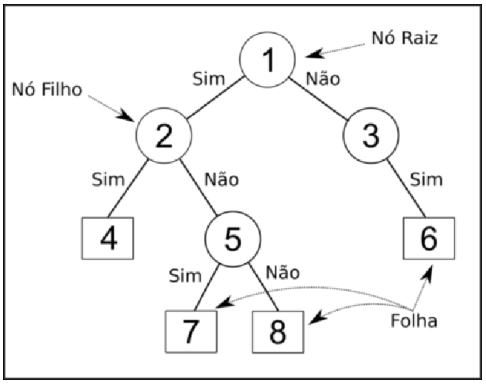
2.2 Algoritmos de Aprendizado de Máquina

Nesta seção, são apresentados os principais algoritmos utilizados em *Machine Learning* e sua respectiva fundamentação teórica, a base matemática existente na construção dos algoritmos é apresentada de maneira sucinta com foco geral na ideia de como cada algoritmo funciona. Aborda-se o algoritmo de Regressão Logística e algoritmos baseados em árvores de decisão, como, *Random Forest*, *eXtreme Gradient Boosting*, *Light Gradient Boosting Machine* e *CatBoost*.

2.2.1 Árvores de Decisão

Árvores de decisão são algoritmos que podem executar tarefas de classificação e regressão, algoritmos como o *Random Forest* utiliza-se de componentes de árvores de decisão em sua estrutura e estão entre os algoritmos de aprendizado de máquina mais poderosos na atualidade (GÉRON, 2022). Considere uma árvore de decisão como sendo um preditor do tipo $h : x \rightarrow y$, que prediz o rótulo associado a uma instância x, que na prática percorre a árvore começando do nó raiz até uma folha. No caso da classificação binária, os rótulos podem assumir valores 0 e 1, ou seja, $y = \{0, 1\}$. Na Figura 1 é exibida um exemplo de uma árvore de decisão. É importante ressaltar que árvores de decisão podem ser utilizadas em outras situações para além de problemas de classificação.

Figura 1 – Ilustração da estrutura de uma árvore de decisão.



Fonte: Sato et al. (2013)

Caso as características do atributo, presentes no nó raiz (Figura 1) sejam atendidas, a árvore percorre o ramo esquerdo até o nó filho, senão percorre o ramo a direita até o nó filho, dessa forma cada observação percorre a árvore de modo que chega até as folhas, onde ocorre a previsão para aquelas observações. É um tipo de abordagem que pode facilmente levar ao sobreajuste de modelos (SHALEV-SHWARTZ; BEN-DAVID, 2014). O atributo de maior importância é colocado no topo da árvore (nó raiz); para medir o grau de heterogeneidade de cada nó da árvore pode se optar pelo cálculo do índice *Gini* ou *Entropia*.

Considere um conjunto de dados S, que contém n amostras, que pode ter c classes distintas. O índice *Gini* para cada nó é calculado por:

$$Gini = 1 - \sum_{i=1}^{c} {p_i}^2, \tag{2.1}$$

em que, p_i é a probabilidade relativa de cada classe em cada nó. O índice *Gini* é puro quando o valor se aproxima de 0 e impuro quando se aproxima de 0,5, quando o valor é próximo de 0,5 aumenta-se o número de classes deste nó.

A *Entropia* de Shannon é um critério também utilizado para o cálculo do ganho de informação e poder ser obtida por:

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i, \tag{2.2}$$

de modo que, p_i é a probabilidade dos dados em S que pertencerem à classe c em cada nó, com i variando de 1 a c. A pureza do coeficiente de *Entropia* está entre 0 e 1. Ambos os coeficientes de *Gini* e *Entropia* levam a resultados semelhantes, sendo o *Gini* mais rápido, porém tende a isolar a classe mais frequente, enquanto a *Entropia* produz árvore mais equilibradas (GÉRON, 2022). O ganho de informação ou *Gain* para determinado atributo, que será denotado por A em um conjunto de dados S, é dado por:

$$Gain(S,A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v), \tag{2.3}$$

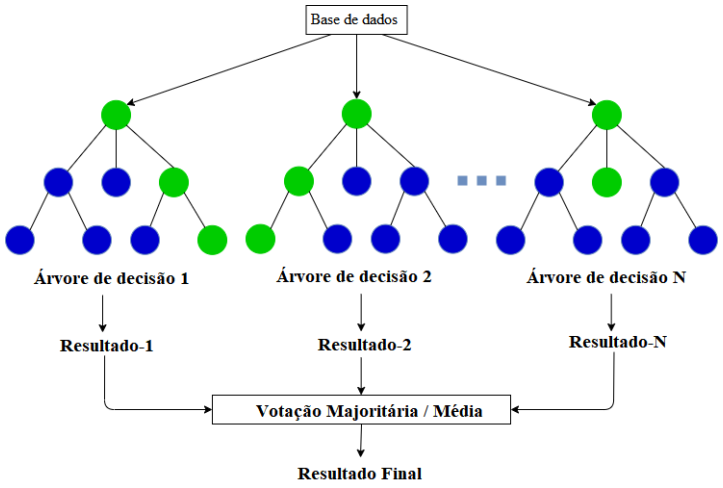
sendo, v um elemento do conjunto A, portanto, S_v é o subconjunto de S formado pelos os dados em que $A = v$. O ganho de informação determina quais atributos ficarão no topo da árvore, quanto maior seu valor, maior o grau de importância. Estruturas de árvores de decisão se adaptam aos dados de treinamento muito de perto, o que provavelmente resultará em modelos sobreajustados, uma maneira de evitar o sobreajuste é restringindo a profundidade máxima da árvore (GÉRON, 2022).

2.2.2 Random Forest

Random Forest é um algoritmo que consiste em um conjunto de árvores de decisão, a ideia por trás do método é criar coleções de árvores que melhore suas previsões e que são ligeiramente diferentes entre si (MUELLER; GUIDO, 2016). Na prática, o algoritmo obtêm conjuntos de dados da amostra utilizando a técnica de amostragem *bootstrap*, e para cada conjunto de dados gera uma árvore de decisão, a combinação das varias árvores de decisão é feita utilizando o método *bagging*, ou seja, combina-se vários modelos de aprendizado e dessa maneira tem-se o aumento do desempenho geral do modelo (FREITAS et al., 2021).

Conforme pode ser observado na Figura 2, existe o conjunto de dados de treinamento, e a partir dele, subconjuntos aleatórios de amostras são obtidos para a construção das árvores de decisão individuais. A previsão do algoritmo *Random Forest* é obtida por votação majoritária sobre as previsões das árvores individuais, para o caso da classificação (SHALEV-SHWARTZ; BEN-DAVID, 2014). Os resultados são mais demorados, porém mais precisos se comparado com uso de apenas uma árvore de decisão.

Figura 2 – Ilustração do funcionamento do algoritmo *Random Forest*.



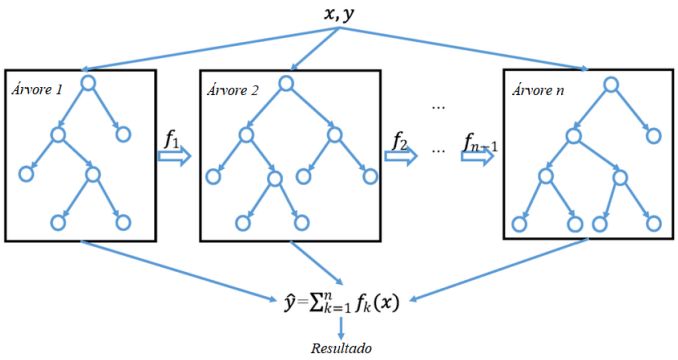
Fonte: Ampadu (2021)

A técnica usada pelo algoritmo *Random Forest* é conhecida por *ensemble*, que é um conjunto de previsões que se utiliza de diferentes estimadores, baseando-se em métodos do tipo *bagging* e *boosting*, métodos estes criados com intuito de evitar sobreajuste em modelos baseado em árvores de decisão. O uso de florestas aleatórias reduz problemas de sobreajuste. Ao construir muitas árvores, todas funcionando razoavelmente bem e superajustadas de maneiras diferentes, pode-se reduzir a quantidade de sobreajuste calculando a média dos seus resultados (MUELLER; GUIDO, 2016).

2.2.3 eXtreme Gradient Boosting

O algoritmo *eXtreme Gradient Boosting* ou apenas *XGBoost* é uma versão melhorada do algoritmo *gradiente boosting*, e diferentemente do *Random Forest* que usa uma abordagem baseada no algoritmo *bagging*, o *XGBoost* usa a abordagem baseada em *boosting* que cria árvores de forma sequencial, minimizando os erros de modelos anteriores, ou seja, é um treinamento aditivo no qual cada nova árvore é construída com base no aprendizado resultante do resíduo da árvore anterior (GALETTO, 2022). A Figura 3 está ilustrando como ocorre a construção de modelos utilizando a abordagem *eXtreme Gradient Boosting*.

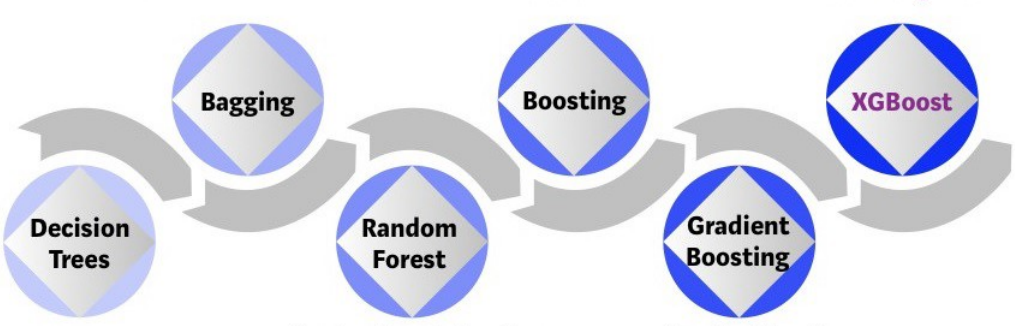
Figura 3 – Ilustração do funcionamento do algoritmo *XGBoost*.



Fonte: Wang et al. (2019)

Na Figura 3, f é uma função do tipo $f(x) = y$, que rotula as novas instâncias de entrada. Para entender as diferenças entre as abordagens utilizadas por cada algoritmo é preciso conhecer a evolução desses algoritmos. Na Figura 4 está sendo representado a linha do tempo dos algoritmos baseados em árvores de decisão. O *bagging* foi desenvolvido para combinar diversas árvores de decisão usando o mecanismo de votação majoritária, a partir desse método criou-se o *Random Forest* que seleciona de forma aleatória subconjuntos das características obtidas pelo método *bagging* e cria coleções de árvores de decisão (GÉRON, 2022).

Figura 4 – Ilustração da evolução dos algoritmos baseados em árvores de decisão.



Fonte: Gomes (2019)

Como descrito anteriormente, o algoritmo *Random Forest* introduz aleatoriedade para desenvolver árvores coletando algumas características ao dividir um nó, em vez de selecionar as melhores características. Partindo deste problema, foram desenvolvidos os modelos sequenciais, ou seja, o método *boosting*. A ideia da maioria dos métodos *boosting* é treinar sequencialmente os previsores, de forma que cada árvore criada tenta corrigir o erro do seu antecessor (GÉRON, 2022).

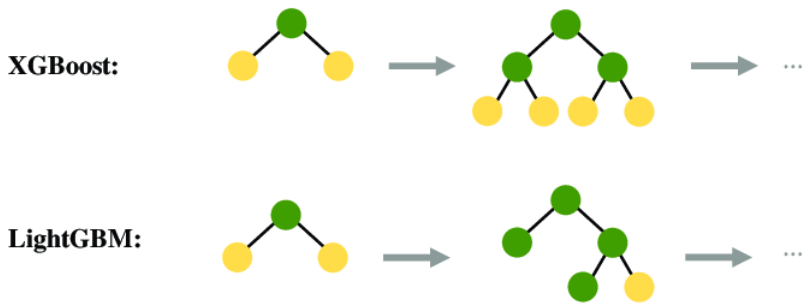
O *Gradient Boosting* é um algoritmo de descida gradiente, método usado para minimizar perdas ao adicionar modelos sequenciais. O *XGBoost* é uma melhoria do *Gradient Boosting* e utiliza o gradiente otimizado para a construção de árvores reforçadas. O método empregado considera que quanto mais uma característica é utilizada para tomar decisões, maior será a sua pontuação (ROCHA; CARMO; VASCONCELOS, 2018). Além disso, o algoritmo *XGBoost* manipula valores ausentes no conjunto de dados e também faz a regularização evitando viés em seus modelos.

2.2.4 Light Gradient Boosting Machine

A estrutura *Light Gradient Boosting Machine (LightGBM)* desenvolvida pela *Microsoft* e posterior ao *XGBoost*, é mais um método baseado em árvores de decisão que usa uma estrutura de aumento de *Gradient Boosting*, o método usado aumenta a eficiência do modelo e reduz uso de memória (VILLANUEVA, 2021).

A ideia por trás da estrutura *LightGBM* é parecida com a utilizada pelo *XGBoost* e possui as mesmas vantagens com a diferença na forma com que as árvores são construídas. O *XGBoost* constrói suas árvores de maneira horizontal focado em níveis, ou seja, na profundidade, enquanto o *LightGBM* as constrói verticalmente em forma de folha, escolhendo a que acredita ter maior redução de perda (PIOVEZAN et al., 2022). A diferença entre os tipos de construções de árvores citada é demostrado na Figura 5.

Figura 5 – Diferença entre construções de árvores de decisões em nível e folha.



Fonte: Rezazadeh (2020)

O *LightGBM* utiliza duas técnicas, são elas *Gradient-based One-Side Sampling (GOSS)* e *Exclusive Feature Bundling (EFB)*. A primeira tem como objetivo fazer a amostragem, enquanto que a segunda faz o agrupamento do número de recursos. A divisão de recursos é baseada em histogramas, usando a amostragem baseada em *GOSS* reduz-se a complexidade por meio de gradientes com o foco nas instâncias com grandes gradientes (ADEBAYO, 2021).

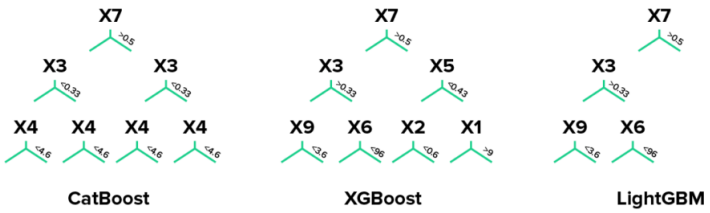
Por padrão, o *LightGBM* suporta dados categóricos, ou seja, possui parâmetro para lidar com recursos categóricos, enquanto que para o *XGBoost* precisa-se fazer a codificação manualmente (ADEBAYO, 2021). A proposta da *LightGBM* foi construída para oferecer uma estrutura com implementações mais rápida e com maior velocidade de execução (NEVES, 2020). A construção de árvores em forma de folha pode levar a problemas de superajuste de modelos, caso não seja utilizado os parâmetros adequados (CORPORATION, 2022).

2.2.5 CatBoost

O *CatBoost* é um algoritmo que utiliza implementação *Gradient Boosting*, é uma biblioteca criada pela empresa de tecnologia *Yandex*. O lançamento do *CatBoost* ocorreu posterior ao *LightGBM*. Algumas diferenças que torna o algoritmo superior aos demais é o quesito velocidade e facilidade de implementação, conforme documentado por (YANDEX, 2022). A Figura 6 ilustra as diferenças entre os métodos de construção de árvores de decisões utilizada pelos três algoritmos de aumento de gradiente.

O *CatBoost* contrói árvores simétricas (balanceadas), ao contrário do *XGBoost* e *LightGBM*, em cada etapa é utilizado a mesma condição para divisão das folhas da árvore anterior, o par de recursos que representa menor perda é usado para os nós de nível (JOHN, 2022). A estrutura diminui o tempo de previsão, torna o algoritmo mais rápido, controla o sobreajuste e converte valores categóricos em números.

Figura 6 – Diferença entre construções de árvores de decisão usando *CatBoost*, *XGBoost* e *LightGBM*.



Fonte: Alal (2019)

A Amostragem de Variação Mínima (MVS) é a técnica utilizada pelo *CatBoost*, que faz com que diminua a quantidade de exemplos amostrados e a qualidade do modelo melhora em comparação com modelos gerados por algoritmos como *XGBoost*, que não utiliza nenhuma técnica de amostragem ponderada, motivo pelo qual o processo de divisão usado pelo *XGBoost* é mais lento quando comparado com *LightGBM* e *CatBoost* (ADEBAYO, 2021).

2.2.6 Regressão Logística

A Regressão Logística é uma técnica estatística usada para tarefas de classificação, ou seja, quando a variável dependente é categórica. A partir de uma função logística é calculado a probabilidade de uma entrada x pertencer a uma determinada classe de y (SHALEV-SHWARTZ; BEN-DAVID, 2014). Esse método pertence a classe dos modelos lineares generalizados, definido pela distribuição binomial com função de ligação canônica (logit). A função logística é definida como:

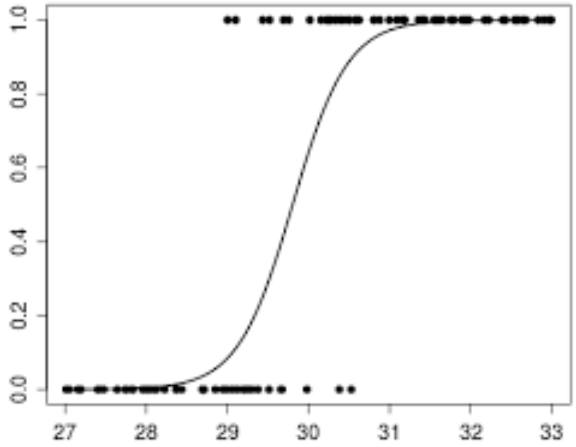
$$logit(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki}, \qquad i = 1, 2, ..., n. \tag{2.4}$$

Em que, o termo p_i é a probabilidade de cada classe, n corresponde ao número de observações na amostras, e k a quantidade de variáveis preditoras. A fração $\frac{p_i}{1-p_i}$ representa a razão de probabilidades entre duas classes, também chamada chance ou *odds* em inglês, e seu logaritmo natural é o logit. Realizando a exponenciação da equação (2.4) em ambos os lados tem-se que a probabilidade p_i é obtida por:

$$p_i = \frac{1}{1 + e^{\beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki}}}. \tag{2.5}$$

De forma similar a regressão linear, a saída y é prevista com base nos valores de entrada x, onde β_0 é o intercepto e os demais β_k são os coeficientes para cada coluna de dados de entrada associados a x_k, que é o valor da variável preditora para cada observação (SILVERIO, 2015).

Figura 7 – Ilustração do gráfico da função logit ou curva sigmóide.



Fonte: Gonzalez (2018)

No gráfico da função logit, representado pela Figura 7, os valores assumem 0 ou 1 no eixo y, que representa a classe dado o preditor representado no eixo x. Este gráfico pode ser também chamado curva sigmóide, que significa “em forma de S”, referindo-se a curva desta

função. Observe que as probabilidade obtidas pela função logística são transformadas em valores binários (0 ou 1) para prever as classes. Isso é feito ao estimar os coeficientes β_k nos dados de treinamento usando a estimativa de máxima verossimilhança (FORTI, 2018).

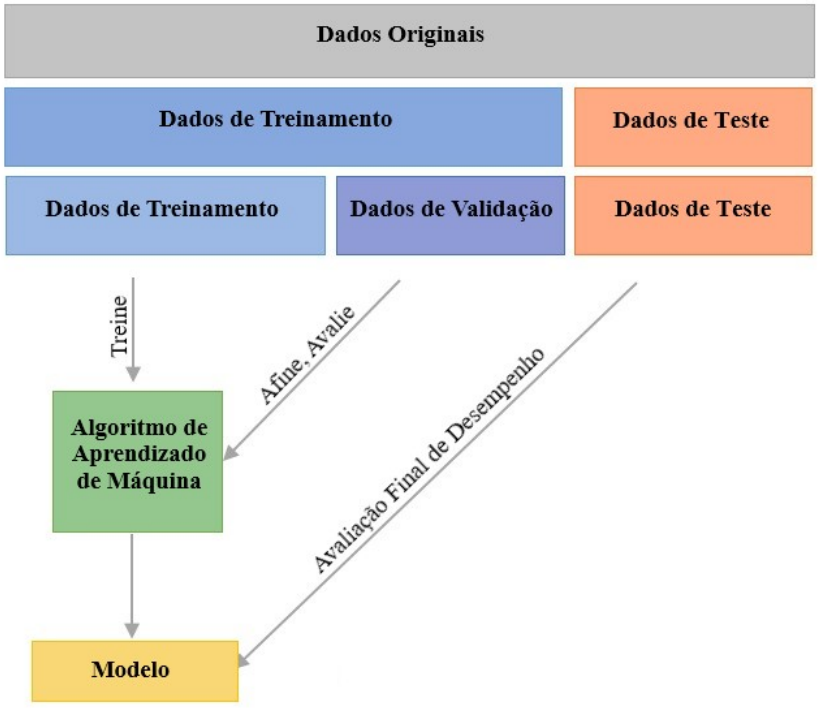
2.3 Seleção e validação do modelo

Alguns cuidados devem ser tomados antes de selecionar o modelo final, a validação é uma maneira de reduzir incertezas sobre qual modelo apresentará melhores resultados. A ideia básica é dividir o conjunto de dados de treinamento em dois conjuntos (SHALEV-SHWARTZ; BEN-DAVID, 2014). Quando a base de dados é grande, a melhor abordagem é dividir os dados aleatoriamente em três conjuntos, treinamento, validação e teste (SANTOS, 2018).

A validação pode ser usada para seleção de modelos, primeiro treina-se algoritmos diferentes, usando o conjunto de treinamento. E para selecionar o modelo que será utilizado na base de teste, amostra-se o conjunto de validação e escolhe o modelo que minimiza o erro sobre o conjunto de validação (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Observa-se na Figura 8, que neste contexto o conjunto de validação é independente do conjunto de treinamento. A validação abordada é usada quando os dados são abundantes, em situações que não é possível amostrar um novo conjunto de dados para validação, utiliza-se outra técnica para validar modelos de treinamento, que chamamos de validação cruzada *k-fold*.

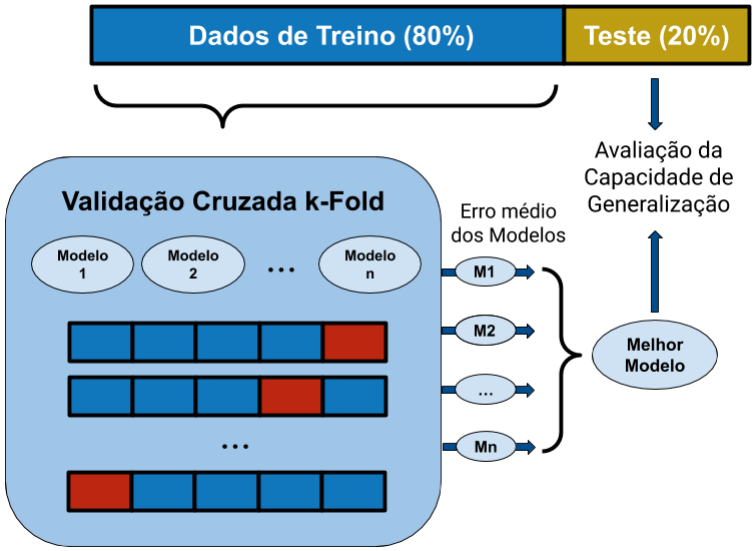
Figura 8 – Ilustração do processo de treinamento, validação e teste de modelos.



Fonte: Freecodecamp (2018)

Na validação cruzada *k-fold*, o conjunto de dados é particionado aleatoriamente em k subconjuntos de tamanhos iguais, na prática iguais a 5 ou 10. Destes dados, $k-1$ é usado para treinamento e o restante para validação (SANTOS, 2018). A Figura 9 ilustra o funcionamento desse tipo de validação.

Figura 9 – Ilustração do processo de treinamento e validação cruzada *k-fold* com $k = 5$.



Fonte: Scaccia (2020)

Conforme demonstrado na Figura 9, o processo de validação cruzada *k-fold* se repete até que todas as partições tenham participado tanto no treinamento quanto da validação. O processo resulta em k estimativas que serão resumidas por meio do cálculo da média e erro padrão (SANTOS, 2018).

2.4 Problemas do Aprendizado de máquina

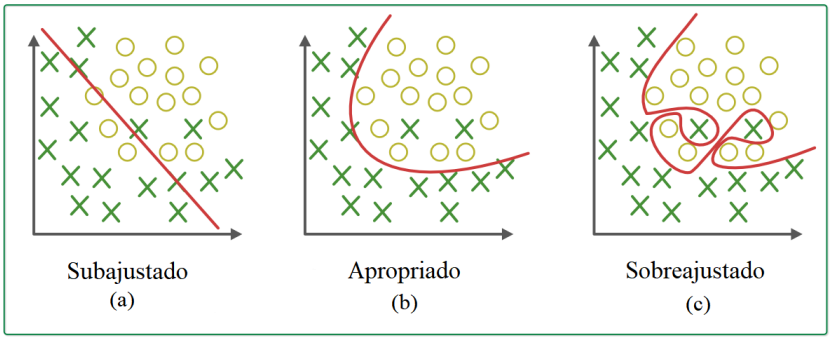
Esta seção aborda os principais problemas que podem surgir ao se trabalhar com aprendizado de máquina. O uso de algoritmos simples e robustos, dependendo tipo de dados utilizados, podem resultar em modelos subajustados e sobreajustados, respectivamente. Bases de dados desbalanceadas podem dificultar o processo de aprendizagem.

2.4.1 Sobreajuste e Subajuste

Alguns cuidados devem ser tomados quando se trabalha com modelos de aprendizado de máquina, suponha que um pequeno grau de algum parâmetro do modelo não se ajuste bem aos dados e gere um grande erro de aproximação. Enquanto que um alto grau em algum parâmetro levará a um grande erro de estimativa, ou seja, um problema de sobreajuste (*overfitting*) (SHALEV-SHWARTZ; BEN-DAVID, 2014).

A Figura 10 ilustra três cenários, a situação (a) ocorre quando existe um problema de subajuste (*underfitting*) em modelos, neste caso o modelo obtido é muito simples para explicar a variância. O *underfitting* leva à um erro elevado tanto na base de treino quanto na base de teste. As duas maneiras de identificar subajuste em modelos, são graficamente e através do erro.

Figura 10 – Ilustração de modelos subajustado, apropriado e sobreajustado.



Fonte: Geeksforgeeks (2022)

Em (b) na Figura 10 temos um exemplo de um modelo adequado. Já em (c) ocorre um problema de *overfitting*, situação em que o modelo decora a estrutura dos dados de treinamento. Quando ocorre sobreajuste o modelo se torna adequado apenas nos dados de treino e não consegue generalizar, tendo um desempenho ruim quando ajustado a novos dados (SANTOS, 2018).

As principais causas de sobreajuste são, complexidade do algoritmo, poucos dados de treinamento e ruído nos dados, como valores extremos ou incorretos. Todavia devemos tomar certos cuidados, por exemplo, algoritmos simples demais, ou com poucos parâmetros pode resultar em subajuste, assim como modelos com muitas restrições e atributos não representativos (GEEKSFORGEEKS, 2022).

2.4.2 Dados desbalanceados

Em problemas de classificação é comum que uma das classes seja mais frequente que a outra, principalmente em classificação binária. Mueller e Guido (2016) abordam essa questão citando um exemplo em que uma das classes de interesse representa 99% dos dados, e que o classificador tenha 99% de precisão, isso não garante que ele seja realmente um classificador bom, pois o modelo pode está apenas prevendo a classe majoritária. O desbalanceio de classes poder ser tratado de três maneiras, uma delas é subamostrando a classe mais frequente, ou seja, excluindo observações da classe majoritária utilizando técnicas de reamostragem, de modo a deixar ambas as classes equilibradas.

O método mais usual é realizar amostragem estratificada para criar reamostras da classe menos frequente. Dessa forma, otimiza-se o modelos de modo a aumentar a sensibilidade da classe menos frequente e balancear as classes do atributo de interesse (SANTOS, 2018). O processo de reamostragem permite que não exista um classe predominante no conjunto de treinamento, criando artificialmente réplicas com reposição da classe menos frequente (LOPES, 2018). Desse modo, reduz-se altos erros de previsão na classe minoritária, algo que precisa ser evitado em problemas de diagnóstico médico, onde a classe minoritária é a que possui presença de uma doença grave (FRIZZARINI; LAURETTO, 2013). Também pode ser utilizado os dois métodos descritos de forma conjunta e avaliar qual é o mais indicado para o tipo de problema.

2.5 Métricas de avaliação em aprendizado de máquina

Em classificação, para avaliar o desempenho de modelos, utiliza-se algumas métricas, entre elas, acurácia, precisão, sensibilidade, especificidade, *F-score*. Essas métricas são obtidas a partir de uma matriz de confusão, que está sendo representada pela Figura 11. Note que a matriz apresenta uma tabulação cruzada entre as classes observadas e preditas. Avaliar a matriz de confusão diretamente não nos permite ter conclusões precisas, mas resumindo as informações com cálculos de métricas especificas pode-se chegar conclusões adequadas (MUELLER; GUIDO, 2016).

Figura 11 – Ilustração de uma matriz de confusão.

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

Fonte: Rodrigues (2019)

Uma forma de resumir a matriz de confusão, calculando a proporção de acertos do modelo em ambas as classes, é através da acurácia. A acurácia não é uma métrica usual para avaliação de modelos, visto que, avalia apenas quanto o modelo está acertando, independente da classe. A precisão avalia dentre as classes positivas a proporção de acerto do modelo. A sensibilidade (*recall* ou revocação) avalia a capacidade do modelo em detectar com sucesso resultados classificados como positivos e a especificidade avalia a capacidade de detectar resultados negativos. A métrica *F-score* é um cálculo baseado na precisão e sensibilidade, a Tabela 1 demonstra as equações utilizadas para o cálculo das principais métricas de avaliação de modelos em *Machine Learning*.

Tabela 1 – Métricas de avaliação de modelos com suas respectivas equações obtidas a partir das taxas de acertos, verdadeiros positivos (VP), falsos positivos (FP), verdadeiros negativos (VN) e falsos negativos (FN).

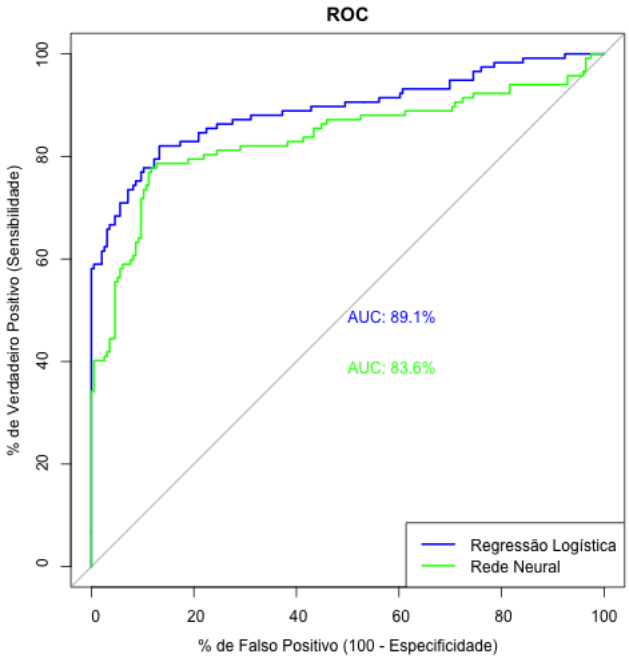
Métrica	Fórmula
Acurácia	$\frac{VP+VN}{VP+FN+VN+FP}$
Precisão	$\frac{VP}{VP+FP}$
Sensibilidade	$\frac{VP}{VP+FN}$
Especificidade	$\frac{VN}{VN+FP}$
F-score	$2 \times \frac{Preciso \times Sensibilidade}{Preciso+Sensibilidade}$

Fonte: Elaborado pelo autor.

A métrica *F-score* considera a precisão e a sensibilidade e é uma medida melhor por considerar a média harmônica entre elas, sendo uma melhor alternativa a acurácia (MUELLER; GUIDO, 2016). A ferramenta comumente usada para avaliar o desempenho de modelos de classificação é a *Receiver operating characteristics Curve* (Curva ROC) que é um gráfico que pode distinguir entre duas classes. A curva ROC considera todas os possíveis limites para um determinado classificador, mostrando a taxa de falsos positivos em relação à taxa de verdadeiros positivos (MUELLER; GUIDO, 2016).

Na Figura 12 é demonstrado uma curva ROC com sua respectiva AUC (*area under the ROC curve*). Para essa ilustração, visualiza-se a curva para dois modelos, Regressão Logística e Rede Neural, respectivamente.

Figura 12 – Ilustração de uma Curva ROC.



Fonte: Rodrigues (2018)

O ideal é que a curva ROC esteja próxima ao canto superior esquerdo, pois o adequado é obter um classificador que produza uma alta sensibilidade enquanto mantém uma baixa taxa de falsos positivos. É altamente recomendável usar AUC para avaliar modelos em dados desequilibrados. A AUC não possui um limite padrão, sendo necessário em algumas situações ajustar um limite de decisão para obter resultados com alta AUC (MUELLER; GUIDO, 2016).

3 METODOLOGIA

Esta seção aborda a descrição dos dados utilizados na análise, o processo de tratamento e preparação dos dados para aplicação dos algoritmos de *Machine learning*, os métodos de reamostragem, pacotes utilizados e hiperparâmentos otimizados.

3.1 Material e métodos

Para aplicação, utilizou-se a base de dados referente pacientes que contraíram Covid-19 no período de março a dezembro de 2020. A base de dados originalmente tinha 460.721 observações, e devido apresentar muitos valores inconsistentes e observações faltantes para oito das onze variáveis preditoras utilizadas no estudo, resultando em dificuldades no tratamento dessas observações, optou-se por realizar a retirada das mesmas do banco de dados que passou a contar com 440.153 observações.

As variáveis preditoras são do tipo numéricas e categóricas conforme dispostas na Tabela 2. A variável resposta utilizada foi Situação, ou seja, o desfecho final do paciente, onde assume o valor 0, para recuperado e 1, para óbito. Todas as variáveis utilizadas neste estudo são categóricas, com exceção da variável Idade.

Tabela 2 – Variáveis consideradas no estudo de casos de Covid-19 no estado do Mato Grosso e suas respectivas codificações.

Variáveis	Categoria\Tipo
Situação	0 - recuperado, 1 - óbito
Idade	Variável Numérica
Sexo	1 - Masculino, 0 - Feminino
ProfissionalSaude	1 - Sim, 0 - Não
Comorbidade	1 - Sim, 0 - Não
Cardiovascular	1 - Sim, 0 - Não
Diabetes	1 - Sim, 0 - Não
Hipertensão	1 - Sim, 0 - Não
Neoplasia	1 - Sim, 0 - Não
Obesidade	1 - Sim, 0 - Não
Pulmonar	1 - Sim, 0 - Não
Gestante	1 - Sim, 0 - Não

Fonte: Elaborado pelo autor.

Todas as variáveis preditoras foram escalonadas, e para treinamento dos algoritmos de aprendizado de máquina utilizou-se 80% dos dados obtidos após o pré-processamento. A divisão deu-se por estratificação da variável resposta, visto que, a classe positiva (óbito) representava menos de 2% do total da base de dados. Além disso, utilizou-se a técnica de reamostragem para equilibrar as classes da variável Situação durante o treinamento dos modelos, aplicou-se a subamostragem da classe majoritária, ou seja, excluiu-se aleatoriamente observações da classe mais frequente.

Para realizar as previsões, os algoritmos utilizados foram, *Random Forest, XGboost, LightGBM, CatBoost* e Regressão Logística. Alguns dos algoritmos baseados em árvores de decisão são altamente propensos a sobreajuste, para evitar *overfitting* alguns hiperparâmentos foram ajustados, a Tabela 3 resume os modelos e hiperparâmetros otimizados.

Tabela 3 – Algoritmos usados na análise com hiperparâmetros otimizados e respectivos pacotes em Python.

Algoritmo (pacote)	Hiperparâmetros otimizados
Regressão Logística *(scikit-learn)*	-
Random Forest (scikit-learn)	*min_samples_split:* O número mínimo de amostras necessárias para dividir um nó interno. *n_estimators:* O número de árvores na floresta aleatória.
XGBoost(xgboost)	*max_depth:* Profundidade máxima de uma árvore, ao aumentar esse valor o modelo torna-se mais complexo e propenso a *overfitting*. Ao treinar uma árvore profunda o *XGBoost* consome memória de forma agressiva.
LightGBM (lightgbm)	*max_depth:* Para limitar explicitamente a profundidade da árvore, diminuindo este parâmetro reduz o tempo de treinamento.
CatBoost (catboost)	*max_depth:* Profundidade da árvore, uma valor alto aumenta o consumo de memória e torna o modelo mais complexo. *eta:* A taxa de aprendizado, usado para reduzir a etapa de gradiente. Quanto menor o valor, mais lento e preciso será o treinamento, é necessário alterar o número de iterações na mesma proporção. *n_estimators:* O número máximo de árvores que podem ser construídas, esse número deve ser alterado em proporção a uma mudança no parâmetro da taxa de aprendizagem.

Fonte: Elaborado pelo autor.

Os ajustes de hiperparâmetros realizados foram necessários na maioria dos casos para evitar o sobreajuste de modelos, o algoritmo de *Random Forest* foi aquele que apresentou maior propensão a sobreajuste durante o treinamento dos modelos, sendo necessário aumentar quantidade de árvores e o número mínimo de amostras para divisão de cada nó. Para identificar o melhores hiperparamentos a serem ajustados, foi usada a função *GridSearchCV* do pacote *scikit-learn* em algumas situações (LEARN, 2022b). As demais bibliotecas utilizadas para implementação dos algoritmos de *Machine Learning* foram, *xgboost*, *lightgbm* e *catboost* (XGBOOST, 2022; CORPORATION, 2022; YANDEX, 2022). As bibliotecas utilizadas para manipulação e análise dos dados foram *pandas*, *numpy* e *matplotlib* (PANDAS, 2022; NUMPY, 2022; MATPLOTLIB, 2022). A biblioteca *imblearn* foi usada para tratar da aprendizagem com dados desbalanceados (LEARN, 2022a).

A validação cruzada *k-fold* com $k = 5$ foi utilizada para avaliar o modelo treinado quanto a possibilidade de sobreajuste nos dados de treinamento. Para cada algoritmo foi calculado o *ranking* de importância dos atributos, considerando as diferenças entre as abordagens de cada algoritmo, como é o caso da Regressão Logística em comparação com o *Random Forest*, nem todas as variáveis apresentarão o mesmo *ranking* de importância. A análise foi realizada usando a linguagem Python 3 na ferramenta Jupyter Notebook, os dados foram baixados em formato CSV e importados brutos para o Python.

A máquina utilizada para análise possui as seguintes especificações: Sistema operacional Windows 10 Home com arquitetura de 64 bits, com processador Intel Core i5 de 3.20 GHz de velocidade e 8,00 GB de RAM. O tempo de processamento para treinamento dos modelos, utilizando a técnica de Subamostragem teve duração de 12 minutos.

4 RESULTADOS E DISCUSSÕES

Os resultados da análise descritiva para as variáveis categóricas estudadas estão dispostos na Tabela 4, onde podem ser observadas as frequências e porcentagem para cada uma das categorias das *features* e da variável resposta.

Tabela 4 – Descrição das variáveis categóricas analisadas e suas respectivas frequências considerando cada categoria.

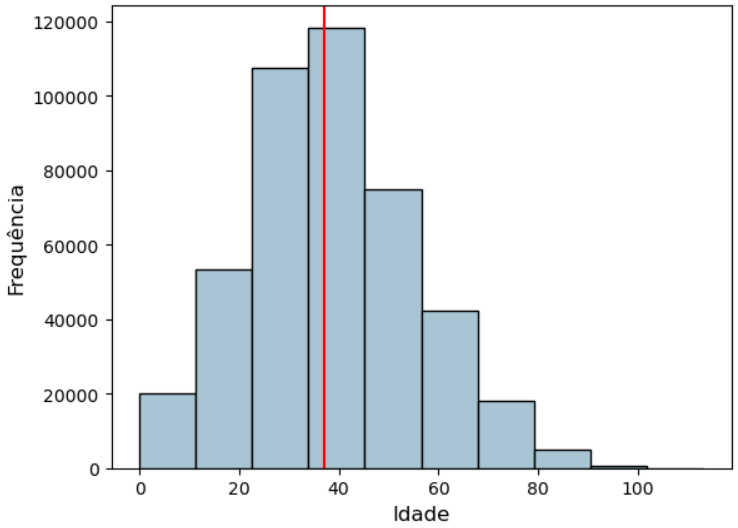
Variáveis	Frequência (n_i)		Porcentagem 100(f_i)	
	Sim	Não	Sim	Não
ProfissionalSaude	12018	428135	2,73	97,27
Comorbidade	71561	368592	16,26	83,74
Cardiovascular	11177	428976	2,54	97,46
Diabetes	17313	422840	3,93	96,07
Hipertensão	37143	403010	8,44	91,56
Neoplasia	762	439391	0,17	99,83
Obesidade	9137	431016	2,08	97,92
Pulmonar	5011	435142	1,14	98,86
Gestante	2355	437798	0,54	99,46
	Masculino	Feminino	Masculino	Feminino
Sexo	207680	232473	47,18	52,82
	Óbito	Recuperado	Óbito	Recuperado
Situação	8688	431465	1,97	98,03

Fonte: Elaborado pelo autor.

Dos pacientes que participaram do estudo, pouco mais de 16% possuíam algum tipo de comorbidade e aproximadamente 8,5% tinha problemas de hipertensão. Gestantes e portadores de algum tipo de neoplasia correspondem a menos de 0,60% dos pacientes que notificaram ter contraído Covid-19 neste período. A proporção de mulheres que participaram do estudo foi superior a de homens e menos de 2% do total de pacientes vieram a óbitos do total de 440.153 indivíduos que compõe a amostra analisada.

A Figura 13 demonstra a distribuição da variável Idade, a linha em vermelho corresponde a mediana das idades dos pacientes, a partir dessa informação, considera-se que a metade dos pacientes tinham aproximadamente idade superior a 37 anos, sendo mais frequente pacientes com idades entre 21 e 44 anos.

Figura 13 – Distribuição das Idades dos pacientes que contraíram Covid-19 no período analisado.



Fonte: Elaborado pelo autor.

A partir da Tabela 4 podemos observar um desbalanceio muito grande da amostra em relação a classe de interesse, óbito. Além disso, algumas variáveis previsoras exibiram um desbalanceio ainda maior que o da variável resposta, o que levantou questões sobre o grau de contribuição para a obtenção de um modelo de previsão. Os resultados iniciais do estudo, após a divisão dos dados entre treinamento e teste usando a estratificação pela classe, com 80% e 20% dos dados respectivamente, mostraram um resultado insatisfatório.

Os resultados iniciais do treinamento dos modelos resultaram em AUC de aproximadamente 0,70, porém, ao testar os modelos obtidos em novos dados, estes não conseguiram realizar boas previsões, obtendo AUC inferior a 0,55 na base de teste. Tendo em vista o problema para obter um modelo capaz de prever a evolução da doença, foi então utilizado algumas técnicas de reamostragem. Inicialmente, usou-se a técnica de Superamostragem da classe minoritária, ou seja, réplicas de observações da classe menos frequente foram criadas artificialmente de maneira que ambas as classes ficaram balanceadas.

Os resultados da técnica de Superamostragem não foram satisfatórios, mesmo ajustando alguns hiperparâmetros dos algoritmos, os modelos apresentaram um alto sobreajuste. A AUC dos modelos treinados chegou a 0,97 porém não conseguiu distinguir bem entre as classes quando aplicando a novos dados, obtendo nos dados de teste uma AUC inferior a 0,60. Por fim, a técnica de Subamostragem da classe majoritária permitiu obter modelos razoavelmente capazes de distinguir entre ambas as classes quando aplicado a dados nunca visto antes, essa técnica excluiu observações de forma aleatória da classe majoritária balanceando ambas as classes, os resultados estão disponíveis nas Tabelas 5 e 6.

A Tabela 5 dispõe dos resultados da base de dados de treinamento, onde é possível observar os valores da AUC, desvio padrão, valores dos hiperparâmetros ajustados e os cinco previsores que mais contribuíram para construção dos modelos em um *Ranking* de importância para cada um dos algoritmos utilizados no estudo. Todos os algoritmos consideraram a variável Idade como o previsor mais importante com exceção do *XGBoost*, que considerou Comorbidade o previsor mais importante, obtendo no treinamento uma AUC de 0,809.

Tabela 5 – Resultados obtidos na base de treinamento com respectivo desvio padrão, parâmetros otimizados e os previsores que mais contribuíram para o modelo obtido.

Algoritmos	Treinamento AUC(dp)	Hiperparâmentros otimizados	Ranking de importância dos previsores
Regressão Logística	0,807 (0,0033)	-	Idade Comorbidade Sexo Obesidade Gestante
Random Forest	0,817 (0,0033)	*min_samples_split*=25 *n_estimators*=1200	Idade Comorbidade Hipertensão Diabetes Obesidade
XGBoost	0,809 (0,0033)	*max_depth* = 2	Comorbidade Idade Obesidade Sexo Diabetes
LightGBM	0,807 (0,0033)	*max_depth* = 2	Idade Diabetes Obesidade Sexo Comorbidade
CatBoost	0,810 (0,0033)	*max_depth* = 2 *n_estimators*=1300 *eta*=0,04	Idade Comorbidade Sexo Obesidade Diabetes

dp: desvio padrão

Fonte: Elaborado pelo autor.

Os modelos com melhores desempenhos nos dados de treinamento foram, *Random Forest* *CatBoost* e *XGBoost*, sendo que todos apresentaram uma AUC superior a 0,80. Os previsores que mais influenciaram nos modelos foram Idade, Comorbidade, Diabetes, Obesidade e Sexo. Os ajustes de hiperparâmetros realizados melhorou o resultado dos modelos de modo a dificultar a possibilidade de sobreajuste e equilibrar o tempo de treinamento para os modelos de *Random Forest* e *CatBoost* que tiveram o número de estimadores aumentados. Os modelos *XGBoost* e

LightGBM tiveram o tempo de treinamento reduzido apenas com intuito de reduzir o consumo de memória, esses ajustes não afetaram o desempenho geral dos modelos.

Dentre todos os algoritmos, aquele que apresentou maior tendência a sobreajuste foi o *Random Forest*, para garantir um bom desempenho dos modelos e evitar que viessem a apresentar um resultado muito inferior no dados de teste optou-se também por aumentar o número mínimo de amostras para dividir os nós internos em cada árvore da floresta aleatória, esse processo tornou o treinamento do modelo mais lento quando comparado com os demais algoritmos, por outro lado garantiu que o modelo não sofresse *overfitting*. O ajuste realizado nos hiperparâmetros *n_estimators* e *eta* do algoritmo *CatBoost* garantiram uma maior precisão do modelo sem aumentar o tempo de treinamento e sem levar o modelo a sobreajuste, isso devido a redução da profundidade da árvore.

Ao submeter o melhor modelo de cada algoritmo a base de teste, os resultados foram similares a base de treinamento, conforme pode ser observado na Tabela 6. Todos os modelos apresentaram redução no valor da AUC na base de teste, o modelo *Random Forest* apresentou a maior diferença em comparação com a base de treinamento. O modelo final que teve o melhor desempenho foi a Regressão Logística com AUC de 0,806, com a capacidade do modelo de detectar corretamente casos positivos de 0,798 e uma taxa de falsos positivos (FP) de 0,186.

Tabela 6 – Resultados do desempenho dos modelos na base de teste com suas respectivas AUC, taxa de verdadeiros positivos e taxa de falsos positivos e acurácia.

Algoritmos	Teste AUC	VP	FP	Acurácia
Regressão Logística	0,806	0,798	0,186	0,813
Random Forest	0,799	0,808	0,209	0,791
XGBoost	0,803	0,816	0,211	0,790
LightGBM	0,804	0,810	0.202	0,798
CatBoost	0,805	0,807	0,197	0,802

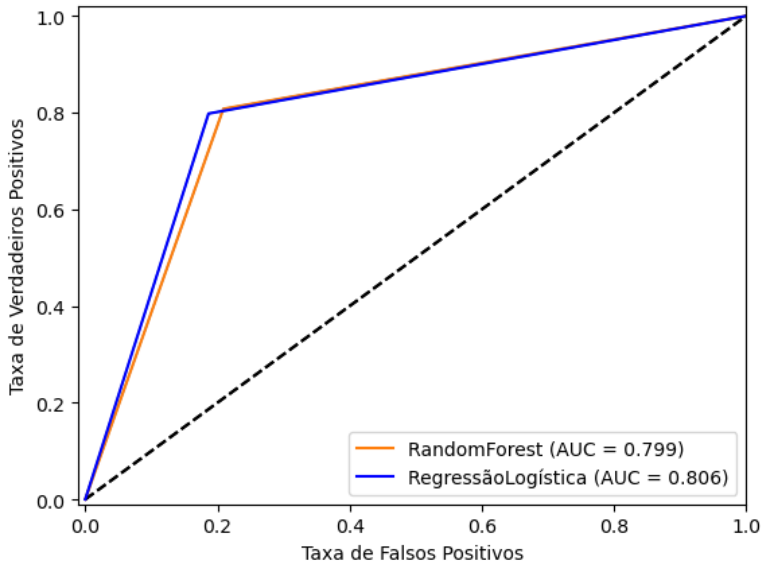
VP: Verdadeiros Positivos, FP: Falsos Negativos

Fonte: Elaborado pelo autor.

A capacidade do modelo de detectar casos negativos, ou seja, a especificidade é complementar a taxa de falsos positivos, que para o caso do modelo de Regressão Logística é de 0,814, sendo a Regressão Logística o algoritmo que apresentou a menor taxa de falsos positivos. O *Random Forest* apresentou a menor AUC, teve uma taxa de verdadeiros positivos maior que a Regressão Logística, mas a taxa de falsos positivos foi superior, resultados similares podem ser observados para os demais modelos. O *CatBoost* foi com modelo com a segunda melhor AUC teve uma capacidade maior de detectar corretamente casos positivos, ou seja, detectou melhor os pacientes que teria o desfecho final como sendo óbito, porém o taxa de falsos positivos foi maior que o modelo de Regressão Logística, o que interferiu no seu poder discriminante, conferindo uma AUC de 0,805, que é um resultado interessante.

Considerando que se deseja evitar ao máximo FP, pois não seria interessante o modelo prever que uma pessoa vai morrer quando na verdade ela não vai, o melhor modelo a utilizar seria aquele com maior especificidade. Por outro lado, do ponto de vista prático deseja-se um modelo capaz de discriminar entre duas classes obtendo um equilíbrio entre a taxa de VP e a taxa de FP, ou seja, com maior AUC. A Figura 14 ilustra a Curva ROC para o modelo de *Random Forest* que teve menor desempenho e para o modelo de Regressão Logística que teve melhor desempenho, simultaneamente.

Figura 14 – Visualização da Curva ROC para os modelos obtidos pelo algoritmos *Random Forest* e Regressão Logística.



Fonte: Elaborado pelo autor.

A comparação feita entre os modelos com menor e maior AUC indica uma pequena diferença na área da Curva ROC, como é evidenciado nos resultados da Tabela 6. Todos os modelos apresentaram um boa performance, sendo os modelos de Regressão Logística, *CatBoost* e *LightGBM* que apresentaram melhores resultados na base de teste, respectivamente. O interessante de se observar nos resultados é que o algoritmo *XGBoost* foi o único a considerar o atributo Idade como o segundo previsor com melhor contribuição para o modelo, ficando atrás do atributo Comorbidades, fato que indica algumas peculiaridades de cada modelo. Ajustes foram realizados retirando as variáveis comuns aos modelos e que apresentaram menores contribuições, na tentativa de tornar o modelo mais parcimonioso, porém pode-se notar pequenas perdas no poder de predição dos modelos, mostrando que mesmo com poucas contribuições todas os previsores utilizados no estudo foram importantes na construção do modelo.

O desempenho de modelos preditivos pode ser influenciados por alguns fatores, como a forma de coleta da amostra, atributos utilizados e balanceamentos dos dados. Rodrigues e Kreutz (2022) realizaram um estudo similar para dados desbalanceados, em que, o número de recuperados correspondia a 96,82% e óbitos a 3,18%, obtendo AUC-ROC de 0,97 para o modelo *Random Forest* usando também o método de subamostragem aleatória *(random under sampling)*. Algumas particulares podem ser observadas no estudo, como a utilização de variáveis sintomáticas, por exemplo, febre, tosse e dor de garganta.

Silva (2021) e Costa (2020) realizaram estudo para criar um modelo de previsão de óbitos baseando-se apenas em dados de pacientes que foram hospitalizados, ou seja, entre os pacientes que foram hospitalizados, desenvolveu-se um modelo capaz de prever a evolução, para óbito ou recuperado. Os modelos com melhores resultados tiveram AUC-ROC de 0,85 e 0,98, respectivamente. Características clínicas dos pacientes como, tosse falta de ar, febre, dor de garganta, dor de cabeça, coriza e diarreia, foram alguns dos atributos utilizados na construção do modelo.

5 CONCLUSÃO

O estudo das variáveis capazes de prever a evolução da Covid-19 em pacientes acometidos pelo o vírus provou-se importante desde os primeiros meses da descoberta da doença. Apesar de alguns estudos iniciais tentarem buscar uma relação entre previsores e a variável resposta, não necessariamente essa relação precisa existir para que o atributo seja um bom previsor, como é caso da variável Sexo que foi um bom previsor da evolução da doença, mas que não permite afirmar com base no Sexo do indivíduo uma relação direta com a evolução da doença.

Em linhas gerais, a performance de modelos preditivos depende muito das informações contidas nas variáveis previsoras e de seu poder discriminatório em relação a resposta de interesse. No contexto estudado, todas as variáveis foram importantes, e outras indispensáveis para construção de um bom modelo. Os resultados obtidos mostraram que o uso de diferentes algoritmos é necessário visto que cada modelo aprende e faz associações de modo diferente, todavia chegaram a resultados semelhantes.

Este estudo identificou fatores que são bons previsores da evolução da Covid-19, utilizando amostras do Estado de Mato Grosso no ano de 2020. O intuito não foi obter uma análise aprofundada em estrutura de causa e efeito, mas obter um modelo de *Machine Learning* capaz de distinguir entre os dois possíveis desfecho com base em algumas informações preliminares dos pacientes. Contudo, obteve-se um modelo com poder discriminatório aceitável (AUC ROC superior a 0,80), capaz de distinguir satisfatoriamente entre as duas classes de interesse.

REFERÊNCIAS

ADEBAYO, S. *HOW CATBOOST ALGORITHM WORKS IN MACHINE LEARNING,*. 2021. Urlhttps://dataaspirant.com/catboost-algorithm/. Citado 2 vezes nas páginas 18 e 19.

ALAL, N. *XGBoost, LightGBM or CatBoost – which boosting algorithm should I use?,*. 2019. Urlhttps://www.riskified.com/resources/article/boosting-comparison/. Citado na página 19.

AMPADU, H. *Random Forests Understanding,*. 2021. Urlhttps://ai-pool.com/a/s/random-forests-understanding. Citado na página 16.

BATISTA, A. F. de M.; FILHO, A. D. P. C. Machine learning aplicado à saúde. *Sociedade Brasileira de Computação*, 2019. Citado 2 vezes nas páginas 11 e 12.

CORPORATION, M. *Welcome to LightGBM's documentation,*. 2022. Urlhttps://lightgbm.readthedocs.io/en/v3.3.2/. Citado 2 vezes nas páginas 19 e 28.

COSTA, L. P. P. d. Características dos casos graves no brasil sobre vítimas da covid-19, com modelo de machine learning para predição de mortes. 2020. Citado 2 vezes nas páginas 11 e 35.

DEISENROTH, M. P.; FAISAL, A. A.; ONG, C. S. *Mathematics for Machine Learning*. [S.l.]: Cambridge University Press, 2020. 371 p. ISBN 9781108455145. Citado na página 13.

FORTI, M. *Técnicas de machine learning aplicadas na recuperação de crédito do mercado brasileiro*. Tese (Doutorado), 2018. Citado na página 21.

FREECODECAMP. *How to Get a Grip on Cross Validation in Machine Learning,*. 2018. Urlhttps://cdn-media-1.freecodecamp.org/images/augTyKVuV5uvIJKNnqUf3oR1K5n7E8DaqirO. Citado na página 21.

FREITAS, A. L. de S. et al. Aprendizado de máquina aplicado à predição de doenças cardiometabólicas com utilização de indicadores metabólicos e comportamentais de risco à saúde. *Anais do Computer on the Beach*, v. 12, p. 301–308, 2021. Citado na página 15.

FRIZZARINI, C.; LAURETTO, M. S. Proposta de um algoritmo para induçao de árvores de classificaçao para dados desbalanceados. In: SBC. *Anais do IX Simpósio Brasileiro de Sistemas de Informação*. [S.l.], 2013. p. 722–733. Citado na página 24.

GALETTO, R. V. Comparação entre regressão logística multinomial e extreme gradient boosting para predição de canais de negociação em cobrança. 2022. Citado na página 16.

GARCIA, L. P. et al. O potencial de propagação da covid-19 e a tomada de decisão governamental: uma análise retrospectiva em florianópolis, brasil. *Revista Brasileira de Epidemiologia*, SciELO Public Health, v. 23, p. e200091, 2020. Citado na página 11.

GEEKSFORGEEKS. *ML | Underfitting and Overfitting,*. 2022. Urlhttps://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/. Citado na página 23.

GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*: Concepts, tools, and techniques to build intelligent systems. [S.l.]: O'Reilly Media, Incorporated, 2022. ISBN 9781098125974. Citado 3 vezes nas páginas 14, 15 e 17.

GOMES, P. C. T. *Conheça o algoritmo XGBoost,*. 2019. Urlhttps://www.datageeks.com.br/xgboost/. Citado na página 17.

GONZALEZ, L. d. A. Regressão logística e suas aplicações. Universidade Federal do Maranhão, 2018. Citado na página 20.

IGNACIO, L. F. F. Aprendizado de máquina: da teoria à aplicação. Volta Redonda, 2021. Citado na página 13.

JOHN, B. *When to Choose CatBoost Over XGBoost or LightGBM [Practical Guide],*. 2022. Urlhttps://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm. Citado na página 19.

LEARN, I. *Imbalanced-learn documentation,*. 2022. Urlhttps://imbalanced-learn.org/stable/index.html. Citado na página 28.

LEARN scikit. *scikit-learn: Machine Learning in Python,*. 2022. Urlhttps://scikit-learn.org/stable/. Citado na página 28.

LOPES, L. P. Poder preditivo de métodos clássicos e de statistical machine learning na classificação de dados desbalanceados em seguros. *Revista de Finanças e Contabilidade da Unimep*, v. 5, n. 2, p. 88–109, 2018. Citado na página 24.

MASCARELLO, K. C. et al. Hospitalização e morte por covid-19 e sua relação com determinantes sociais da saúde e morbidades no espírito santo: um estudo transversal. *Epidemiologia e Serviços de Saúde*, SciELO Brasil, v. 30, 2021. Citado na página 11.

MATPLOTLIB. *Matplotlib 3.6.2 documentation,*. 2022. Urlhttps://matplotlib.org/stable/index.html. Citado na página 28.

MUELLER, A. C.; GUIDO, S. *Introduction to Machine Learning with Python*: A guide for data scientists. [S.l.]: O'Reilly Media, 2016. ISBN 9781449369415. Citado 7 vezes nas páginas 13, 15, 16, 23, 24, 25 e 26.

NEVES, J. M. M. *Otimização de hiperparâmetros em machine learning utilizando uma surrogate e algoritmos evolutivos*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2020. Citado na página 19.

NUMPY. *NumPy documentation,*. 2022. Urlhttps://numpy.org/doc/stable/. Citado na página 28.

PANDAS. *Pandas documentation,*. 2022. Urlhttps://pandas.pydata.org/docs/. Citado na página 28.

PIOVEZAN, R. P. B. et al. Método de aprendizagem de máquina visando prever a direção de retornos de exchange traded funds (etfs) com utilização de modelos de classificação e regressão. Joinville, SC, 2022. Citado na página 18.

REZAZADEH, A. A generalized flow for b2b sales predictive modeling: An azure machine-learning approach. *Forecasting*, MDPI, v. 2, n. 3, p. 267–283, 2020. Citado na página 18.

ROCHA, D. S.; CARMO, A. C. do; VASCONCELOS, J. A. de. Máquina de aprendizagem aplicada ao reconhecimento automático de falhas em motores elétricos. *Anais do Computer on the Beach*, p. 482–491, 2018. Citado na página 18.

RODRIGUES, G.; KREUTZ, D. Modelo preditivo para classificação de risco de óbito de pacientes com covid-19 utilizando dados abertos. In: SBC. *Anais do XXII Simpósio Brasileiro de Computação Aplicada à Saúde*. [S.l.], 2022. p. 144–155. Citado na página 35.

RODRIGUES, V. *Entenda o que é AUC e ROC nos modelos de Machine Learning,*. 2018. Urlhttps://medium.com/bio-data-blog/entenda-o-que-e-auc-e-roc-nos-modelos-de-machine-learning-8191fb4df772. Citado na página 26.

RODRIGUES, V. *Métricas de Avaliação: acurácia, precisão, recall... quais as diferenças?,*. 2019. Urlhttps://vitorborbarodrigues.medium.com/metricas-de-avaliacao-acuracia-precisao-recall-quais-as-diferencas-c8f05e0a513c. Citado na página 24.

SANTOS, H. G. d. *Comparação da performance de algoritmos de machine learning para a análise preditiva em saúde pública e medicina*. Tese (Doutorado) — Universidade de São Paulo, 2018. Citado 4 vezes nas páginas 21, 22, 23 e 24.

SATO, L. Y. et al. Análise comparativa de algoritmos de árvore de decisão do sistema weka para classificação do uso e cobertura da terra. *XVI Simpósio Brasileiro de Sensoriamento Remoto*, p. 2353–2360, 2013. Citado na página 14.

SAÚDE, M. da. *Ministério da saúde,*. 2022. Urlhttps://www.gov.br/saude/pt-br. Citado na página 11.

SCACCIA, K. *Validação Cruzada Aninhada com Scikit-learn,*. 2020. Urlhttps://dataml.com.br/validacao-cruzada-aninhada-com-scikit-learn/. Citado na página 22.

SCHLEDER, G. R.; FAZZIO, A. Machine learning na física, química, e ciência de materiais: Descoberta e design de materiais. *Revista Brasileira de Ensino de Física*, SciELO Brasil, v. 43, 2021. Citado na página 12.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding Machine Learning: From Theory to Algorithms*. [S.l.]: Cambridge University Press, 2014. ISBN 9781107057135. Citado 5 vezes nas páginas 14, 15, 20, 21 e 22.

SILVA, A. O. d. *Uso de machine learning para previsão da evolução de casos de SRAG incluindo casos de COVID-19 considerando variáveis clínicas e demográficas*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2021. Citado na página 35.

SILVA, G. F. d. S. *Fraude de cartão de crédito: como a estatística e o machine learning se conversam*. Tese (Doutorado) — Universidade de São Paulo, 2020. Citado na página 12.

SILVERIO, M. Aplicação de algoritmos de aprendizado de máquina no desenvolvimento de modelos de escore de crédito. 2015. Citado na página 20.

VILLANUEVA, R. M. P. Inteligencia artificial y machine learning para el desarrollo e implementación de softsensors en la predicción del p80 para la molienda sag (sabc-a)-minera las bambas. Universidad Nacional de San Agustín de Arequipa, 2021. Citado na página 18.

WANG, Y. et al. A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science*, Springer, v. 364, n. 8, p. 1–13, 2019. Citado na página 17.

XGBOOST. *XGBoost Documentation,*. 2022. Urlhttps://xgboost.readthedocs.io/en/stable/. Citado na página 28.

YAN, L. et al. A machine learning-based model for survival prediction in patients with severe covid-19 infection. *MedRxiv*, Cold Spring Harbor Laboratory Press, 2020. Citado na página 11.

YANDEX. *CatBoost,*. 2022. Urlhttps://catboost.ai/. Citado 2 vezes nas páginas 19 e 28.