UCI Open Data: Previsão da qualidade dos vinhos

1º João Lucas Oliveira Mota
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
jlucasoliveira2002@alu.ufc.br

2º João Lucas Lima Monteiro
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
joaolucaslima@alu.ufc.br

Abstract*— Nesse artigo, propomos uma abordagem abrangente para a análise exploratória de dados utilizando o conjunto de dados da UCI Open Data, focando na previsão da qualidade do vinho. Inicialmente, realizamos uma análise detalhada dos atributos do conjunto de dados, explorando padrões e tendências. Em seguida, implementamos técnicas de pré-processamento para garantir a qualidade dos dados, lidando com valores ausentes e padronizando as variáveis. O principal objetivo é empregar algoritmos de classificação para categorizar os vinhos como "bom" ou "ruim". Para isso, aplicamos métodos avançados de aprendizado de máquina, como regressão logística, classificador k-vizinhos próximos e o classificador multi camadas do perceptron. Avaliamos o desempenho desses modelos por meio de métricas específicas, como precisão, recall e acurácia, visando encontrar a abordagem mais eficaz na classificação da qualidade do vinho. Exploramos a interpretabilidade dos modelos, analisando quais características são mais relevantes para as previsões. Este estudo não apenas contribui para a compreensão da qualidade do vinho, mas também fornece insights valiosos sobre a aplicação de técnicas de ciência de dados na classificação de produtos em diversos setores. Em síntese, este trabalho oferece uma abordagem prática e informativa para a aplicação de algoritmos de predição para previsão da qualidade de vinhos, com implicações significativas para a indústria vinícola e o campo mais amplo da análise de dados.

Keywords—Vinhos, análise estatística, classificação, redes neurais.

I. Introdução

No *dataset Wine Quality*[1], foram realizadas medições de diferentes características físico-químicas e sensoriais de diferentes vinhos. Entre essas medições encontra-se o teor alcóolico desses vinhos.

Neste artigo, realizamos uma análise de diferentes modelos de classificação, tanto lineares quanto não-lineares, além de realizarmos uma comparação entre os resultados e o desempenho desses modelos buscando definir modelos de classificação para a qualidade dos vinhos definidos no *dataset.*

A. Conhecimento necessário

Focamos, neste trabalho, em problemas de classificação, tanto lineares quanto não lineares, e na comparação entre seus resultados. Para isso, precisamos de algumas definições.

1) Problemas de classificação

Ao nos referirmos a problemas de classificação, estamos nos referindo à utilização de algoritmos de aprendizado de máquina para a previsão de uma categoria à qual uma instância de dados pertence. Para tal, utilizaremos de três modelos, resumidamente explicados a seguir:

a) Regressão Logística

A Regressão Logística foca em modelar a probabilidade de uma instância de dados pertencer a uma determinada classe. Esse modelo é melhor utilizado para problemas de classificação binária, e cujas classes são distintas e as relações entre as características e a probabilidade de pertencer a uma classe são assumidas como lineares.

b) MLP Classifier

O método de Classificação com *Perceptrons* Multicamadas, como o próprio nome já revela, utiliza de uma implementação de redes neurais artificiais do tipo *Perceptron* Multicamadas para a modelagem necessária para o problema de classificação.

Por se tratar de um modelo de redes neurais, o *MLP Classifier* é melhor utilizado para problemas em que supõe uma relação não linear entre os preditores e a coluna alvo.

c) KNN

O algoritmo de *K-Nearest Neighbors* utiliza de comparações entre as instâncias de dados, partindo da ideia de que dados com características e saídas semelhantes tendem a pertencer à mesma classe.

Devido a sua metodologia, o *KNN* é geralmente utilizado em problemas cujas fronteiras de decisão não são lineares. Outra vantagem dele é sua menor sensibilidade a *outliers*, por sua decisão ser baseada em uma votação local.

Para todos os métodos explicados, utilizamos as suas implementações em *python,* com o auxílio das funções presentes na biblioteca *scikit-learn.*

2) Métricas de desempenho

Para a avaliação de eficiência e corretude dos modelos classificação, precisamos considerar os seguintes tipos de resultados:

1. **Verdadeiro Positivo (True Positive - TP):**
Instâncias positivas corretamente classificadas como positivas pelo modelo.

2. **Falso Positivo (False Positive - FP):**
Instâncias negativas incorretamente classificadas como positivas pelo modelo.

3. **Verdadeiro Negativo (True Negative - TN):**
Instâncias negativas corretamente classificadas como negativas pelo modelo.

4. **Falso Negativo (False Negative - FN):**
Instâncias positivas incorretamente classificadas como negativas pelo modelo.

Tais resultados podem ser ilustrados na chamada Matriz de Confusão, representada pela Tabela 1

Tabela 1: Matriz de Confusão

	Real Positivo	**Real Negativo**
Previsto Positivo	TP	FP
Previsto Negativo	FN	TN

Utilizando os resultados presentes nessa tabela, podemos calcular diferentes valores, as chamadas Métricas de Desempenho[3], segundo as definições abaixo:

a) Accuracy

A acurácia mede a proporção de predições corretas em relação ao total de predições, segundo a Equação 1:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

b) Precision

A precisão mede a proporção de verdadeiros positivos em relação ao total de instâncias previstas como positivas, segundo a Equação 2:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

É importante ressaltar que um modelo que não produz falsos positivos possui precisão igual a 1.

c) Recall

A sensibilidade mede a proporção de verdadeiros positivos em relação ao total de instâncias positivas, segundo a Equação 3

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

É importante ressaltar também que um modelo que não produz falsos negativos possui recall igual a 1.

II. Métodos

Nesta seção, vamos abordar os dados que estamos trabalhando e explicar os métodos utilizados.

A. Pré-processamento

A priori, o Dataset utilizado tem as seguintes caraterísticas: **N** (Observações) = 6497; **P** (Número de preditores) = 12 e **L** (número de saídas possíveis para a coluna alvo "quality") = 10. Além disso, é importante observar que o dataset não possui valores faltantes nos conjuntos de dados, então não foi necessária a retirada de nenhuma coluna.

A Figura 1 mostra a Matriz de Correlação dos preditores, ilustrando todos os valores (positivos e negativos, segundo a legenda presente na Figura) entre os diferentes preditores. Podemos perceber algumas coisas interessantes, como o nível de correlação alto entre *'quality'* e *'alcohol'*, o que demonstra que quanto maior a quantidade de álcool no vinho melhor a sua qualidade. Além disso, percebemos o contrário em *'quality'* e *'chlorides'*, onde temos uma correlação negativa, indicando possivelmente que quanto menor a quantidade de 'chlorides' no vinho maior sua qualidade.

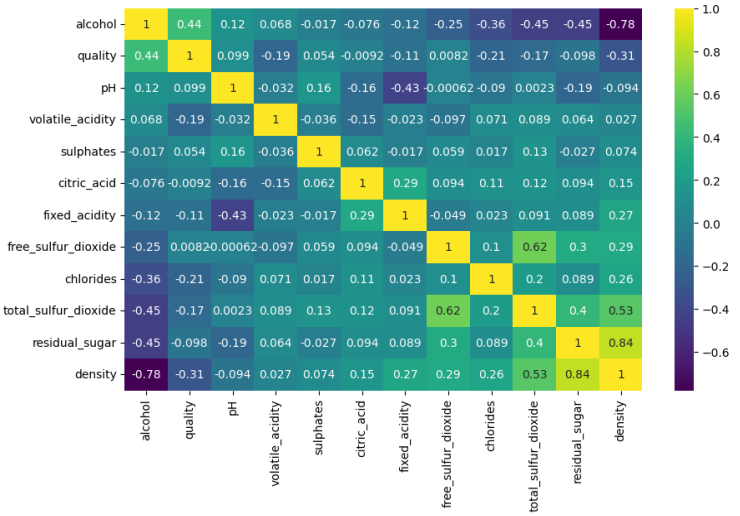

Figura 1: Matriz de correlação dos preditores

Este *dataset* utilizado possui muitos *outliers* em todos os preditores, portanto foi necessário a remoção de tais pontos, pois os *outliers* são observações que apresentam um grande afastamento das demais da série, ou que é inconsistente. Estas observações não refletem o significado real do *dataset*, por isso atrapalha no desempenho dos algoritmos. Ademais, aplicamos outro procedimento para a correção da assimetria dos preditores *'volatile_acidity'*, *'chlorides'*, *'free_sulfur_dioxide'*. A assimetria no *dataset* atrapalha no desempenho dos algoritmos por conta da presença de heterogeneidade de variâncias e dados com distribuição não normal. Para esta correção, aplicamos a transformação de Box-Cox nos três preditores citados anteriormente.

Em seguida, as Figuras 2 a 4 são gráficos de barra que separamos para exemplificar o comportamento de alguns preditores com a variação da qualidade dos vinhos.

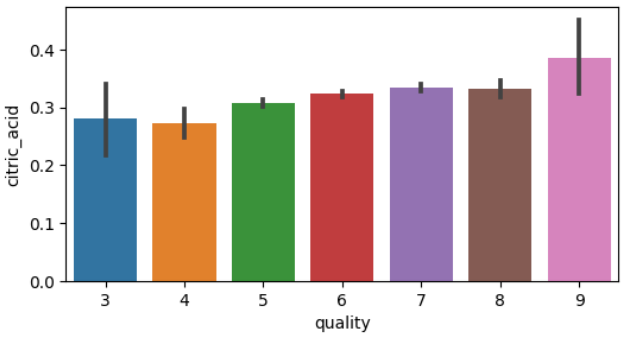

Figura 2: Barplot da quality x citric_acid

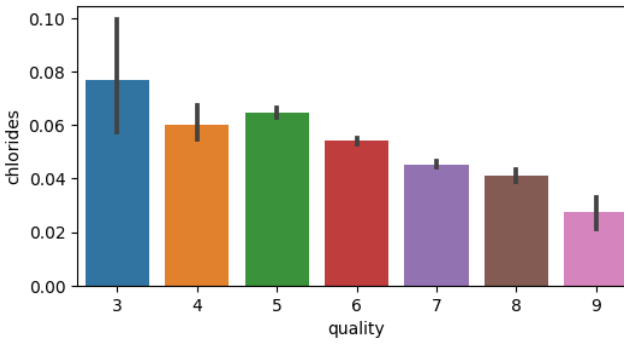

Figura 3: Barplot da quality x chlorides

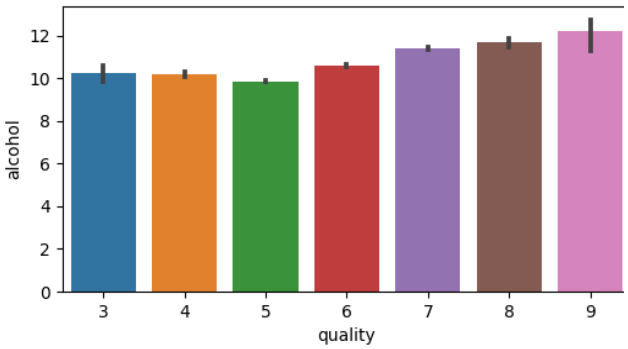

Figura 4: Barplot da quality x alcohol

Para a Figura 2, podemos perceber que quanto maior a qualidade do vinho maior a presença de ácido cítrico na amostra.

Para a Figura 3, podemos perceber que quanto menor a presença de cloretos maior a qualidade do vinho representado.

Por fim, para a Figura 4, podemos perceber uma correlação direta entre o nível de álcool presente e a qualidade do vinho observado, onde percebemos que vinhos com qualidade maior têm também maior índice de álcool.

Tais relações, dentre outras presentes no *dataset*, podem representar uma tendência para um padrão de classificação a ser utilizado em todos os vinhos.

Em seguida, realizamos um processo de transformação nos valores do dataset, visando a utilização de métodos binários de classificação para auxiliar no desempenho do modelo. Para os valores entre 0 e 5 de qualidade, definimos o vinho como "ruim", atribuindo o valor binário 0; enquanto para os valores de 6 a 10 definimos como "bom", atribuindo o valor binário 1.

Após os procedimentos descritos, as características do *dataset* a ser trabalhado na aplicação dos modelos são as seguintes:

N = 4596 observações,

P = 12 preditores.

L = 2 saídas possíveis(0 ou 1).

Por fim, os dados foram divididos entre conjunto de treino e de teste, seguindo uma proporção 70% treino e 30% teste, concluindo o pré-processamento dos dados, e logo após aplicamos os algoritmos de Regressão Logística, MLP Classifier(Classificador Perceptron Multicamadas) e KNN respectivamente.

III. Resultados

Neste tópico, apresentaremos os resultados das manipulações e técnicas apresentadas. Não aplicamos a validação cruzada em nenhum dos algoritmos para ter a mesma base de comparação entre todos.

A. Modelo de Regressão Logística

Aplicando o modelo de Regressão Logística presente na biblioteca *scikit-learn* apresentada anteriormente e utilizando a proporção de conjunto de treino e teste definidas no pré-processamento, chegamos aos seguintes resultados, ilustrados na Figura 5.

Para tal ajuste, utilizamos todos os 12 preditores presentes do *dataset*, conforme foi explicado no pré-processamento.

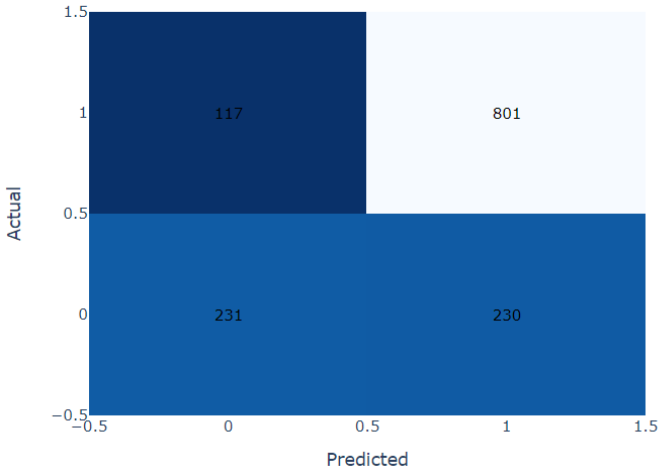

Figura 5: Matriz de Confusão para a Regressão Logística

Percebe-se as seguintes informações sobre os resultados: 117 FN, 801 TP, 231 TN e 230 FP. A partir destas informações podemos usar as fórmulas descritas na introdução para calcular as métricas de desempenho.

Utilizando os valores presentes na Matriz de confusão e as Equações 1, 2 e 3, chegamos nos valores presentes na Tabela 2:

Tabela 2: Métricas de desempenho da Regressão Logística

Accuracy	Precision	Recall
0,7483	0,7769	0,8725

Analisando os valores encontrados, temos que a acurácia é de 74,83%, indicando um desempenho razoável na previsão correta de classificações positivas.

É importante salientar que apenas com os 3 valores é possível tirar conclusões satisfatórias sobre o desempenho do modelo utilizado.

Por exemplo, vamos analisar um modelo hipotético que, por exemplo, tem acurácia 75%. Dessa forma, o modelo pode parecer promissor, mas ao considerar um caso hipotético com 100 vinhos no total, sendo 25 vinhos classificados como ruins (com 4 TN e 21 FP) e 75 vinhos bons (74 TP e 1 FN) podem ser tiradas conclusões errôneas.

Dos 75 vinhos bons deste caso hipotético, o modelo identifica corretamente 74 como bons (TP), um valor satisfatório. No entanto, dos 25 vinhos ruins, o modelo pode identificar apenas, por exemplo, apenas 4 como ruim de fato (TN), um resultado terrível, já que 21 vinhos ruins não são diagnosticados corretamente.

Assim, para garantir que as conclusões acerca do modelo são corretas, faz-se necessária a análise conjunta de todas as métricas de desempenho.

Então, ao analisar os valores de Precisão e Recall do nosso modelo, temos que quando o algoritmo prevê que o vinho é bom (classe positiva), está correto em 77,69% do tempo e que o algoritmo identifica corretamente 87,25% de todos os vinhos classificados positivamente, o que nos leva a concluir que o modelo de Regressão Logística é satisfatório para a classificação do *dataset* utilizado. Mas a seguir vamos analisar com a aplicação feita em modelos não lineares.

B. Modelo MLPClassifier e KNN

Novamente, importamos da biblioteca citada as funções necessárias, dessa vez para os modelos *MLPClassifier* e *KNN* e aplicamos estes modelos. Assim, geramos as figuras 6 e 7, respectivamente.

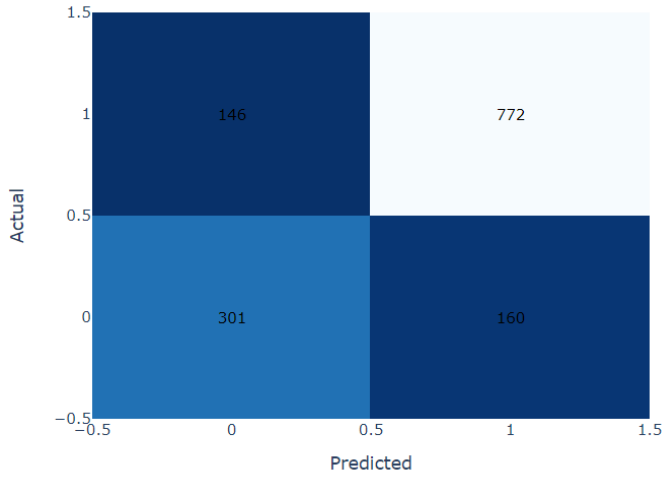

Figura 6: Matriz de Confusão para o *MLP Classifier*

Percebe-se as seguintes informações sobre os resultados: 146 FN, 772 TP, 301 TN e 160 FP. De primeira, podemos analisar que os valores TN aumentaram e os FP diminuíram razoavelmente. Entretanto, o valor de TP diminuiu e o de FN aumentou, ambos levemente, podendo demonstrar um desempenho melhor em comparação à Regressão Logística. A partir destas informações podemos usar as fórmulas descritas na introdução novamente para calcular as métricas de desempenho.

Com os valores da Figura 6, realizamos os mesmos procedimentos feitos anteriormente e encontramos os valores de Métricas de Desempenho representados na Tabela 3.

Tabela 3: Métricas de Desempenho para *MLP Classifier*

Accuracy	Precision	Recall
0,7781	0,8283	0,8409

Assim como no modelo anterior, só é possível observar o desempenho geral do modelo *MLP Classifier* ao considerar todas as métricas de desempenho em conjunto.

Analisando os valores encontrados, temos que a acurácia é de 77,81%, indicando um desempenho melhor que o do algoritmo anterior.

Por fim, ao analisar os valores de Precisão e Recall, temos que o algoritmo prevê que o vinho é bom (classe positiva), está correto em 82,83% do tempo e que o algoritmo identifica corretamente 84,09% de todos os vinhos classificados positivamente,

Assim, temos que o modelo MLP Classifier é levemente superior ao modelo linear de Regressão Logística em duas das três estatísticas.

Em seguida, repetimos os procedimentos uma última vez, dessa vez com o uso dos valores da matriz de confusão para o modelo *K-Nearest Neighbors,* representada na Figura 7.

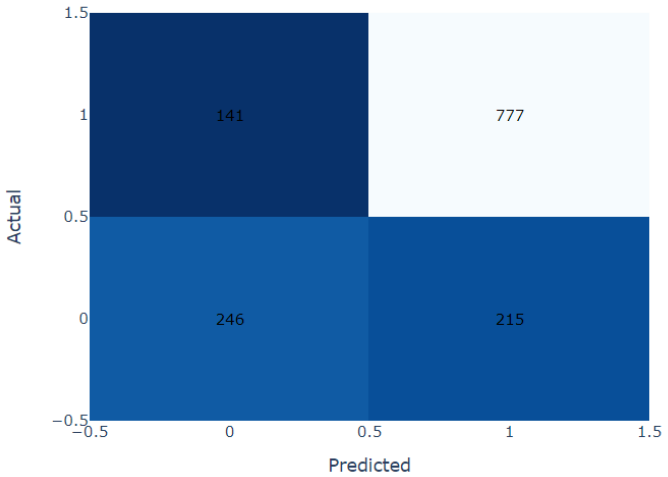

Figura 7: Matriz de Confusão para o KNN

Por último, temos as seguintes informações sobre os resultados: 141 FN, 777 TP, 246 TN e 215 FP. Primeiramente, percebe-se o aumento leve de valores TP e diminuição leve de valores FN, o que pode indicar leve melhora no desempenho. Entretanto, os valores TN diminuíram e FP aumentaram, ambas razoavelmente, por isso o modelo pode indicar que não teve uma melhora ou até mesmo piorou seu desempenho. Para isso precisamos analisar corretamente as métricas. A partir destas informações podemos usar as fórmulas descritas na

introdução novamente para calcular as métricas de desempenho.

Sendo assim, temos as seguintes métricas de desempenho abaixo:

Tabela 4: Métricas de Desempenho para *KNN*

Accuracy	Precision	Recall
0.7418	0.7832	0.8464

Ao observar todos os valores das métricas de desempenho, notamos novamente um leve aumento, especialmente no valor de Precisão, que define a probabilidade de o algoritmo estar correto ao prever que o vinho é bom, em relação a mesma métrica no modelo de Regressão Logística. Porém, a mesma métrica teve uma leve piora em comparação a aplicação do *MLPClassifier*. A acurácia mostrou-se no mesmo nível em relação ao modelo de Regressão Logística, mas teve razoável piora em relação a aplicação do modelo MLPClassifier. A métrica de Recall ficou acima de 80% na aplicação dos três modelos e não variou muito de um modelo para o outro.

C. Modelo linear x Modelo não linear

A escolha entre modelos lineares e não lineares desempenha um papel crucial na construção de sistemas de aprendizado de máquina. No contexto da previsão da qualidade do vinho, a aplicação de modelos lineares e não lineares revela diferentes nuances e implicações para a eficácia preditiva. Nesta seção, exploraremos a distinção fundamental entre modelos lineares e não lineares e analisaremos como essa diferença pode ter influenciado os resultados obtidos ao aplicar Regressão Logística (um modelo linear), KNN (um modelo não linear) e MultiLayer Perceptron Classifier (MLPClassifier, também não linear).

1) Diferenças entre Modelos lineares e não lineares

Os modelos lineares assumem uma relação linear entre as variáveis independentes e a variável dependente. A Regressão Logística é um exemplo clássico de modelo linear que busca traçar uma linha reta para separar as classes de interesse. Por outro lado, modelos não lineares não impõem essa restrição linear, permitindo capturar relações mais complexas e não lineares nos dados. O KNN e o MLPClassifier são exemplos de modelos não lineares, sendo capazes de lidar com fronteiras de decisão mais flexíveis.

2) Desempenho dos Modelos adquiridos no Contexto da Qualidade do Vinho

Ao analisar os resultados, observamos que as métricas de desempenho do KNN superaram levemente as da Regressão Logística. Essa diferença pode ser atribuída à capacidade intrínseca do KNN de capturar padrões complexos e não lineares nos dados, proporcionando uma representação mais flexível da relação entre os atributos e a qualidade do vinho.

Surpreendentemente, o MLPClassifier apresentou um desempenho superior, superando tanto a Regressão Logística quanto o KNN. Essa superioridade pode ser explicada pela capacidade do MLPClassifier em aprender representações hierárquicas e não lineares por meio das suas múltiplas camadas ocultas. Essa arquitetura mais complexa permite ao MLPClassifier modelar relações intrínsecas aos dados, tornando-se mais adaptável a padrões complexos presentes na qualidade do vinho.

3) Conclusão

Em resumo, a escolha entre modelos lineares e não lineares impacta diretamente a capacidade de um algoritmo em modelar a complexidade intrínseca dos dados. No caso da previsão da qualidade do vinho, a flexibilidade dos modelos não lineares, como o MLPClassifier, parece ser benéfica para capturar nuances sutis que podem escapar da capacidade dos modelos lineares tradicionais. Essa compreensão profunda da natureza dos modelos utilizados proporciona insights valiosos para aplicações de aprendizado de máquina na indústria vinícola e em problemas similares de classificação de produtos.

IV. Referências

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. **Modeling wine preferences by data mining from physicochemical properties**. In Decision Support Systems, Elsevier, 47(4):547-553. ISSN: 0167-9236. Disponível em: https://archive.ics.uci.edu/dataset/186/wine+quality

[2] JAMES, Gareth *et al*. **An Introduction to Statistical Learning with Applications in Python**. S.I: Springer, 2023. Disponível em: https://hastie.su.domains/ISLP/ISLP_website.pdf. Acesso em: 10 set. 2023.

[3] Google for Developers.Classificação: acurácia | Machine Learning | Google for Developers. Disponível em: <https://developers.google.com/machine-learning/crash-course/classification/accuracy?hl=pt-br>.. Acesso em: 22 nov. 2023