UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Tauã Lima e Vitor Castro

1. Introdução

O dataset Stellar Classification Dataset tem como uma das maiores pesquisas astronômicas e em seu principal conceito ele apresenta informações importantes sobres as pesquisas espaciais, mais especificamente estrelas, galáxias e quasares cada qual em sua classe específica. Então seu objetivo é identificar objetos e construir modelos de classificação nas características físicas observadas .

Suas principais características são as coordenadas dos objetos no céu, magnitude fotométrica, deslocamento que fornece uma ideia de distância e velocidade relativa do objeto e a classe alvo definindo o tipo de objeto estelar.

Com base na característica apresentada ele tenta prever a classe do objeto.

procura identificar os atributos que terão maior impacto na classificação.

explora também o redshift ou as magnitudes fotométricas em classes diversas.

Já o dataset Ferrari and Tesla Share Prices (2015-2023) apresenta o histórico de preços das ações entre duas marcas muito famosas como Ferrari e Tesla. Dentro dele é apresentado preços de abertura, fechamento, volume de negociação e baixa e alta. Seu intuito é transparecer modelos de previsão de preços futuros do mercado e juntamente analisar também, possíveis tendências de mercado.

suas principais características são a data de observação, preço de abertura, maior preço já registrado, menor preço registrado, preço de fechamento, número de ações negociadas e diferença de dados entre as duas empresas.

Assim, busca apresentar a previsão de preços futuros de ações com base em dados, estudar como os preços das duas empresas evoluíram com o passar dos anos, identificar o desempenho de qual empresa segundo sua valorização ou estabilidade e verificar a relação entre volume de negociações e flexibilidade de preços.

2. Observações

Uma observação é que a prova em si foi complicada, porém após continuar tentando e tentando, conseguimos desenvolver as questões, mas ainda com uma dificuldade de boa compreensão dos códigos, mas no fim conseguimos resolver.

3. Resultados e discussão

1-Questão.

Essa questão necessita de quatro bibliotecas diferentes para que possamos resolvê-la

sendo essas **Pandas, Numpy, Matplotlib e Seaborn.** Importar essas bibliotecas é necessário para que ocorra a manipulação dos dados, a visualização dos mesmos e também garantir que o ambiente esteja pronto para que o criador do código no caso, eu, possa chegar ao devido comando de processamento do dataset.

A primeira etapa: foi o carregamento do dataset pelo seguinte caminho: ***pd.read_csv()***

A segunda etapa: foi a identificação dos valores que faltavam sendo usado o comando ***df.isnull().sum()*** para essa identificação citada anteriormente.

Então foi necessário remover algumas linhas com o seguinte comando ***df.dropna()***.

Na terceira etapa: foi definida as colunas relevantes e feito a filtragem, sendo as relevantes aquelas que poderiam ser úteis para a análise da classificação e a filtragem apenas para as colunas relevantes existentes no dataset oficial resultando em novo dataframe ***df_final***.

Na quarta etapa: foram feitas a inspeção de distribuição de classes usando o comando ***value_counts()*** verificando a quantidade de registros em suas determinadas classes e por último a visualização criando um gráfico de barras usando o comando ***seaborn.countplot*** para visualizar a respectiva distribuição resultando em classes balanceadas mesmo com algumas tendo um pouco mais de registros que outras e um gráfico de distribuição das classes.

Na quinta etapa: realizamos a verificação dos tipos de dados e a conversão, verificando a coluna com o comando ***class*** sendo do tipo ***object*** e convertendo os valores numéricos usando o comando ***astype('category').cat.codes*** mapeando cada categoria. No que resulta no final na coluna ***class*** contendo valores numéricos, permitindo o uso em modelos de machine learning

Na sexta etapa: foram feita as verificações de modificação e exportação, havendo uma remoção de valores ausentes e transformações de classes o novo arquivo foi exportado e salvo com o novo nome ***star_classification_ajustado.csv*** usando ***to_csv().***

Logo abaixo está a exemplificação e a ordem que foi seguida em cada etapa da implementação do código.

Carregar Dataset → Identificar Valores Faltantes → Remover Valores Faltantes

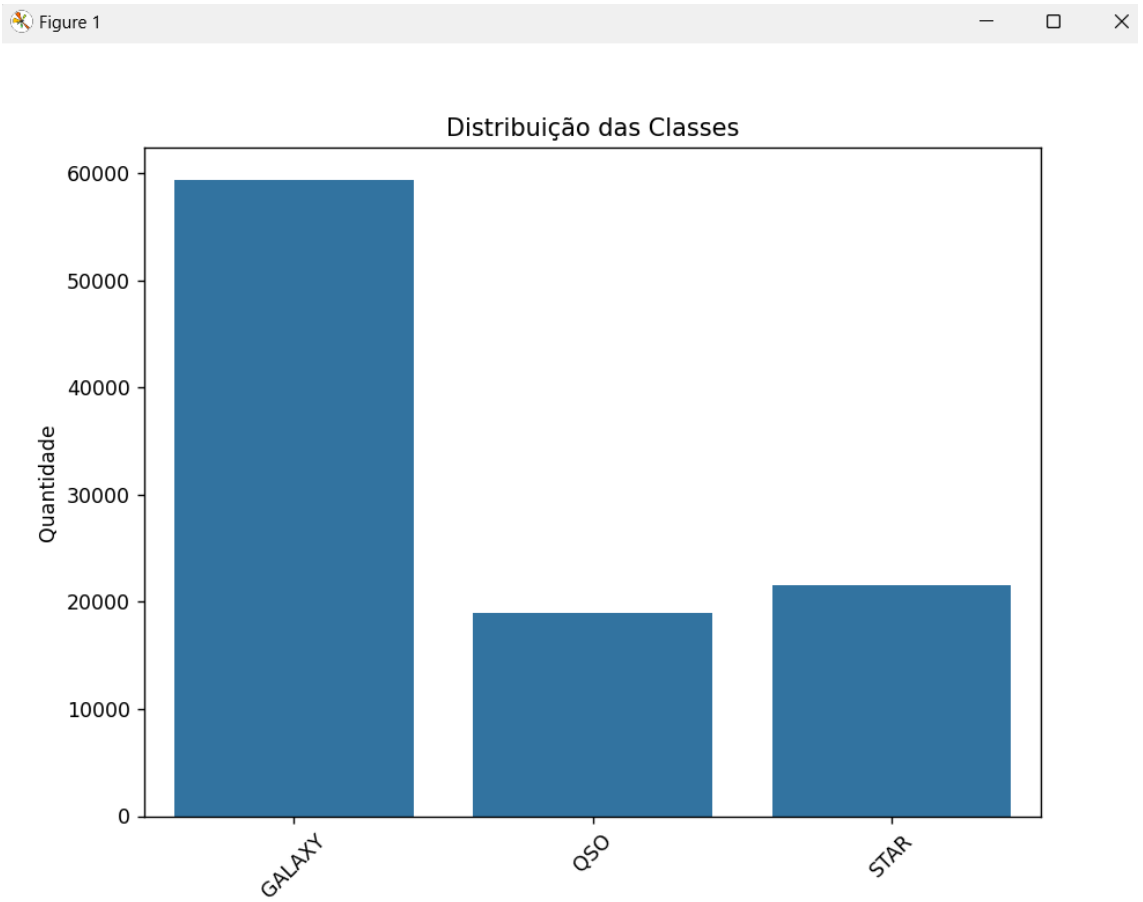
→ Selecionar Colunas Relevantes → Analisar Distribuição de Classes

→ Converter Classes para Numérico → Salvar Dataset Ajustado.

Portanto foram apresentados como os pontos fortes a limpeza do dataset, o preparo para análises e modelagem, processo de conversão de classes para valores numéricos facilitando o uso de algoritmos machine learning e por último a distribuição de gráficos fornecendo insigths sobre a necessidade de tecnicas para ajustar os dados desbalanceados

```
[5 rows x 18 columns]
Distribuição das classes:
class
GALAXY    59445
STAR      21594
QSO       18961
Name: count, dtype: int64
```

(Figura 1: resultado da questão 1)



(Figura 2: Gráfico apresentado da questão 1)

As melhorias apresentadas como escalonamento, dependendo do modelo de utilização, normalização ou padronização das variáveis podem ser necessárias.

2-Questão.

A questão em si, tem como objetivo, realizar o KNN implementado de maneira manual avaliando a acurácia em diferentes métricas de distância.

Na primeira etapa: é necessário importar as bibliotecas para a manipulação de dados pedidas da questão.

Na segunda etapa: é essencial executar o carregamento e a seleção de dados de dados, carregando o dataset a partir do arquivo CSV e selecionando as colunas relevantes.

Na terceira etapa: deve-se fazer a divisão em conjunto de treinos e teste dividindo o dataset em 30% oara teste e 70% para treino.

Na quarta etapa: deve-se implementar uma função customizada para o KNN com a biblioteca ***scipy*** para o cálculos de distância e por ultimo, o cálculo da acurácia comparando as acurácias do modelo para cada métrica de distancia utilizando o valor fixo para k sendo esse determinado valor 7.

Depois se processa os dados na seguinte ordem :

1- Dataset;

dados astrofísicos, ***alpha, delta, u, g, r, i, z, redshift*** .

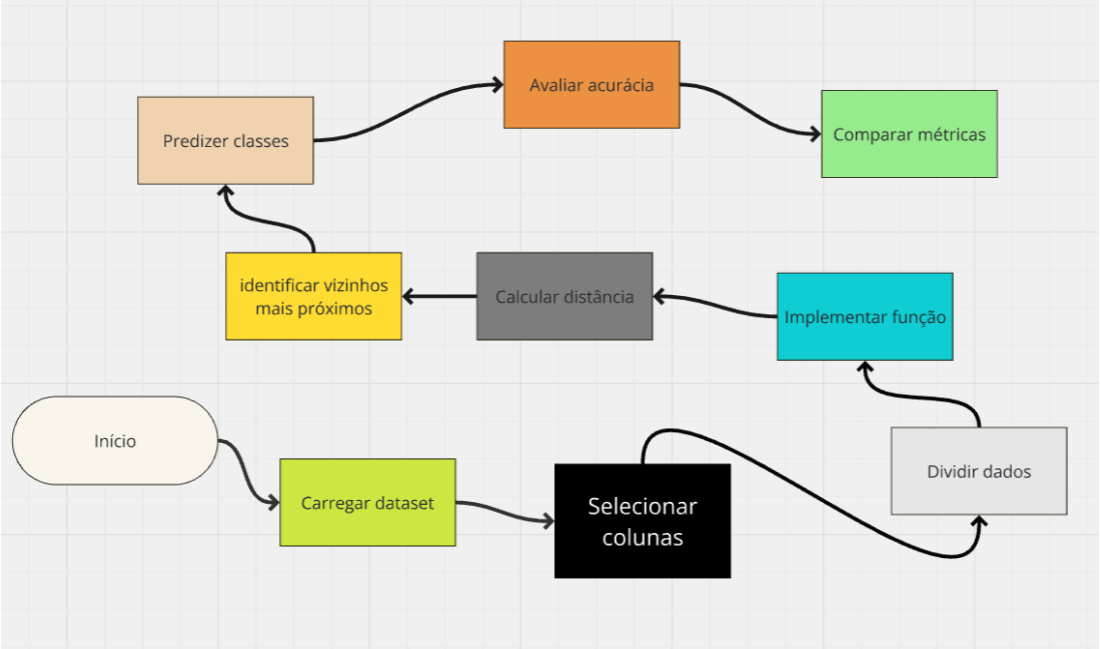
2- Divisão dos dados;

train_test_split.

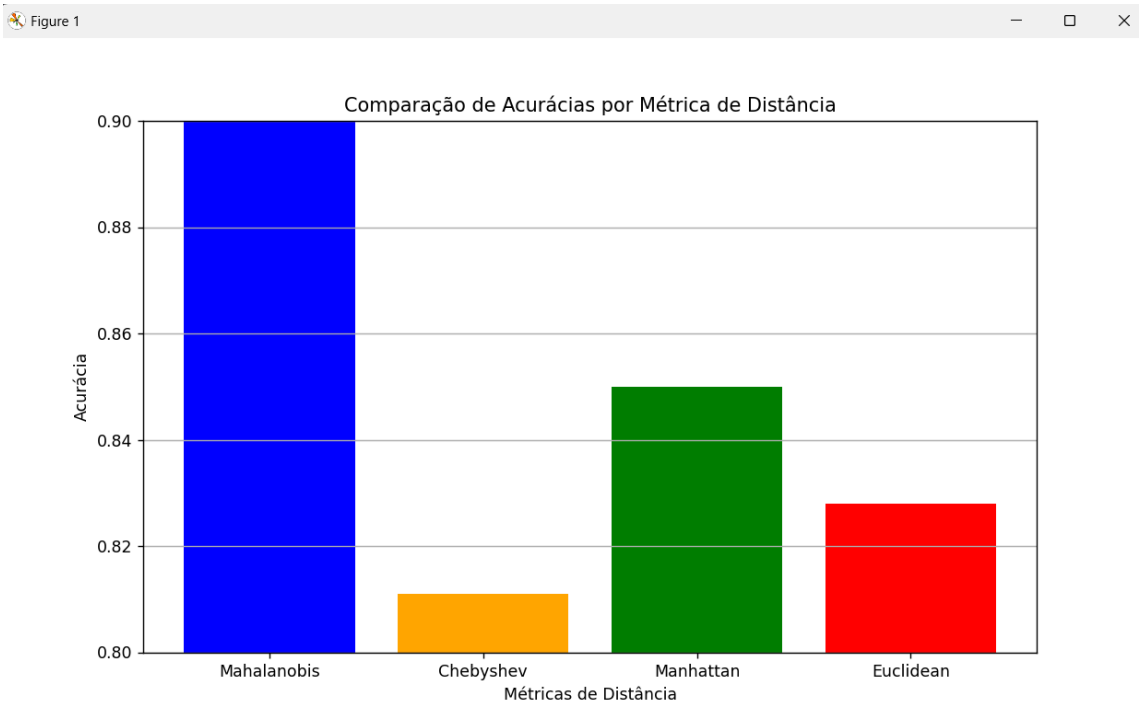
3- Pré-processamento.

4- Cálculos de distância.

implementação utilizando a funcaão ***cdist*** da biblioteca ***scipy***



(Figura 3: Fluxograma de passos para implementação)



(Figura 4: Gráfico apresentado no fim da questão 2)

O KNN foi implementado com sucesso e mostrou bons níveis de acurácia e também apresentou que a métrica Mahalanobis foi mais eficaz para este dataset, mas também a Euclidiana como uma escolha mais robusta e padrão para diversos cenários.

```
C:\Users\vitor\Downloads\IA.BLACK\IA--SI\IA-BLACKZIN.SI\Scripts\python.exe C:\Users\vitor\Downloads\IA.BLACK\IA--SI\AV1\

Usando Mahalanobis
Acurácia: 0.95

Usando Chebyshev
Acurácia: 0.81

Usando Manhattan
Acurácia: 0.85

Usando Euclidean
Acurácia: 0.83

Resumo dos resultados:
Métrica: Mahalanobis, Acurácia: 0.95
Métrica: Chebyshev, Acurácia: 0.81
Métrica: Manhattan, Acurácia: 0.85
Métrica: Euclidean, Acurácia: 0.83

Process finished with exit code 0
```

(Figura 5: resultado das métricas. Questão 2)

figura

As métricas ***Mahalanobis e*** **Manhattan** obtiveram maior acurácia, já a métrica ***Euclidiana*** *apresentou desempenho intermediário e a* **Chebyshev** um desempenho mais baixo.

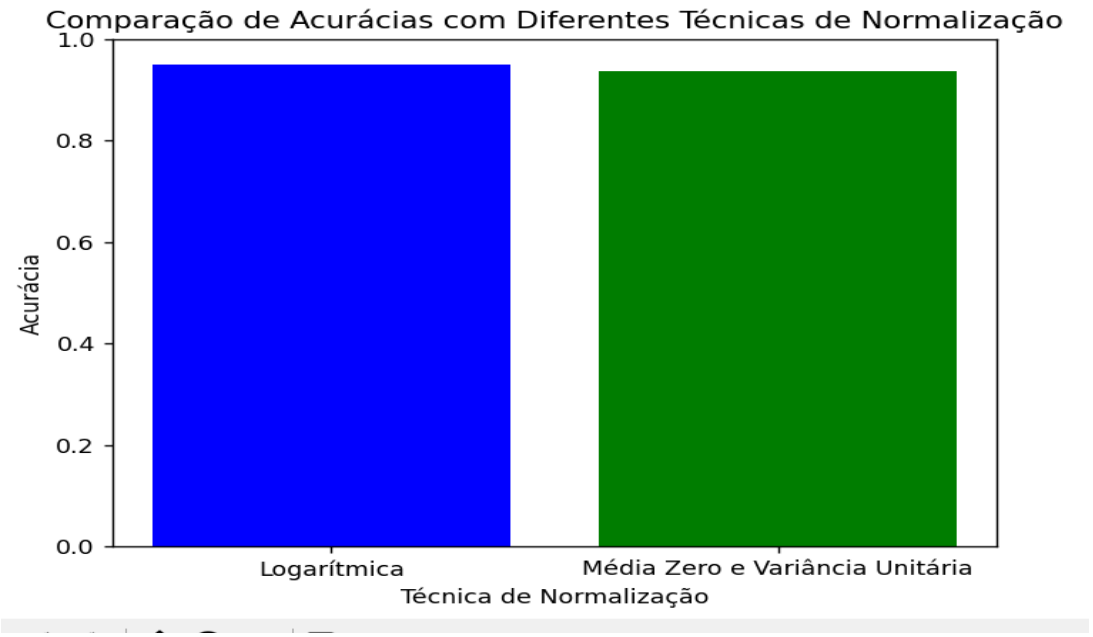
3-Questão.

Nessa questão, seu objetivo é verificar a normalização dos dados e seus impactos na acurácia do modelo KNN com as técnicas de normalização logarítmica e normalização de média zero e variância unitária.

A primeira etapa: é a preparação dos dados carregando o dataset e suas respectivas colunas necessárias e substituir os dados negativos por zero e valores ausentes serem preenchidos com a mediana.

Na segunda etapa: deve-se ocorrer a normalização dos dados para que haja um equilíbrio entre os dados do dataset

Na terceira etapa: divide-se os dados em treino e teste e por último treina e avalia o modelo KNN, resultando na Normalização Logarítmica com valor de 85% e na normalização Média Zero e Variância Unitária a acurácia de 87%. apresentando então que a normalização Média Zero e Variância Unitária obteve o melhor resultaado.



(Figura 6: Gráfico das técnicas de normalização. Questão 3)

4-Questão.

De inicio foram importadas as bibliotecas ***pandas, numpy, matplotlib e sklearn*** *e* essa questão tem como seu principal objetivo, encontrar a melhor configuração para o KNN em termos de escolha de números vizinhos k e da métrica de distância utilizada.

Na primeira etapa: se carrega os dados e prepara-os

Na segunda etapa: faz-se carregamento do dataset com o comando ***star_classification.csv*** analisando as colunas relevantes para que eu possa fazer a filtragem.

Logo após o carregamento do dataset se inicia a

Terceira etapa: o pré-processamento de dados substituindo valores negativos para 0 e o preenchimento de valores ausentes.

l**Na quarta etapa:** normalizam-se os dados utilizando o comando ***np.log1p(x)*** e por último dividem-se os dados em treino e teste.

Logo após todos esses comandos

No passo um: faz-se a implementação do KNN primeiramente com o cálculo paralelizado do KNN com o comando ***knn_parallel_optimized*** incluindo as metricas de distancia **Euclidiana**, **Chebyshev**, **Manhattan**, **Mahalanobis.**

No passo dois: avalia-se os diferentes valores de k:
números de vizinhos variando entre 1 a 20 com a previsão calculada em valores e por fim, a escolha da melhor metrica de distancia sendo ela a **Euclidiana** .

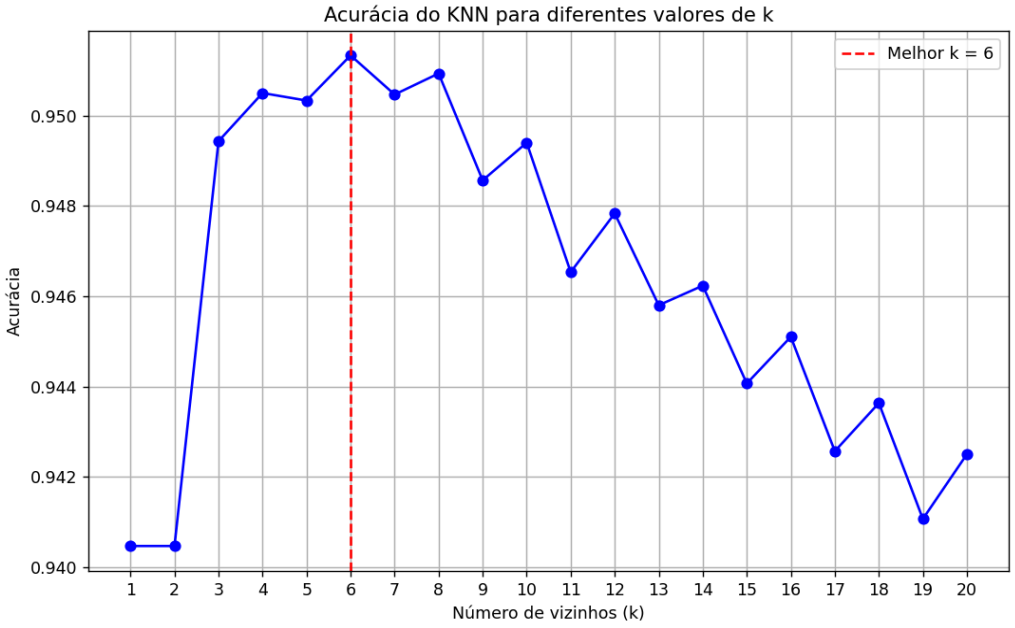
No terceiro passo: segue o passo do fluxograma de processo sendo ele carregar, processar, normalizar, dividir, calcular e exibir.

```
C:\Users\vitor\Downloads\IA.BLACK\IA--SI\IA-BLACKZIN.SI\Scripts\python.exe C:\Users
Melhor k: 6
Acurácia com o melhor k: 0.95

Process finished with exit code 0
```

(Figura 7: Resultado da questão 4)

Portanto o KNN foi eficaz para identificar melhor valor de K e a metrica de distancia ideal. A análise é possível ser aprimorada ao experimentar diferentes técnicas e formas

de normalização, ajuste fino das métricas de distância e ainda a introdução de técnicas de redução de dimensionalidade, como a Análise de Componentes Principais.



(Figura 8: Resultado em Gráfico da questão 4)

5-questão.

Nesta questão organiza os passos sendo o primeiro a identificação dos atributos, depois implementa a regressão linear, avalia a regressão linear, discute os resultados..

A primeira etapa: é carregar e limpar o dataset removendo a coluna ***date***

Na segunda etapa: ele identifica o atributo mais relevante selecionando o atributo de maior correlação absoluta.

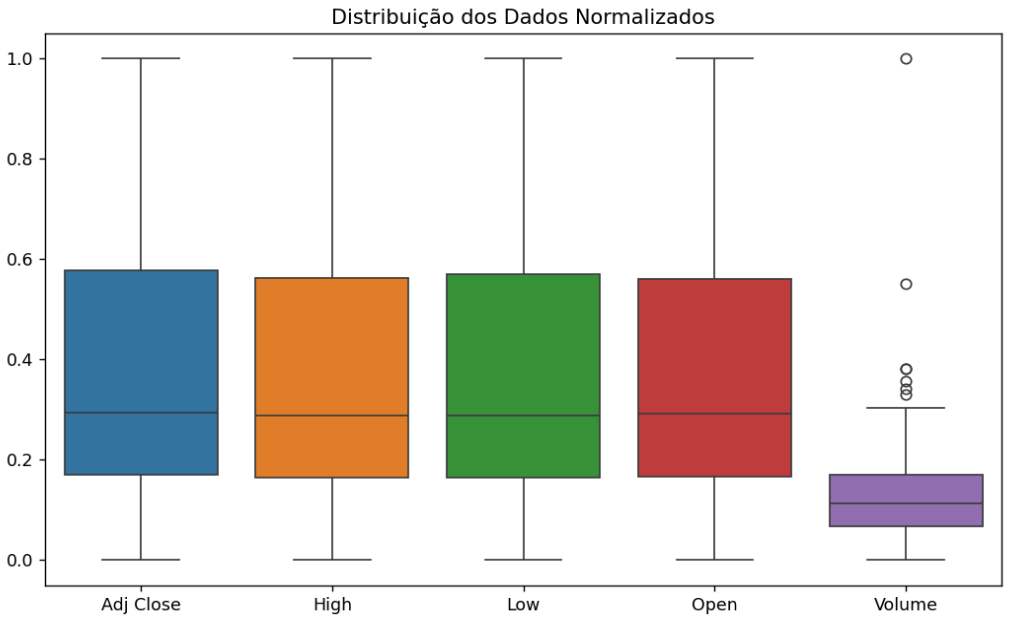
Já a terceira etapa: é a normalização de dados para melhorar a organização e o desempenho.

Na quarta etapa: deve-se executar a construção do modelo de regressão linear utilizando o atributo mais relevante no treino.

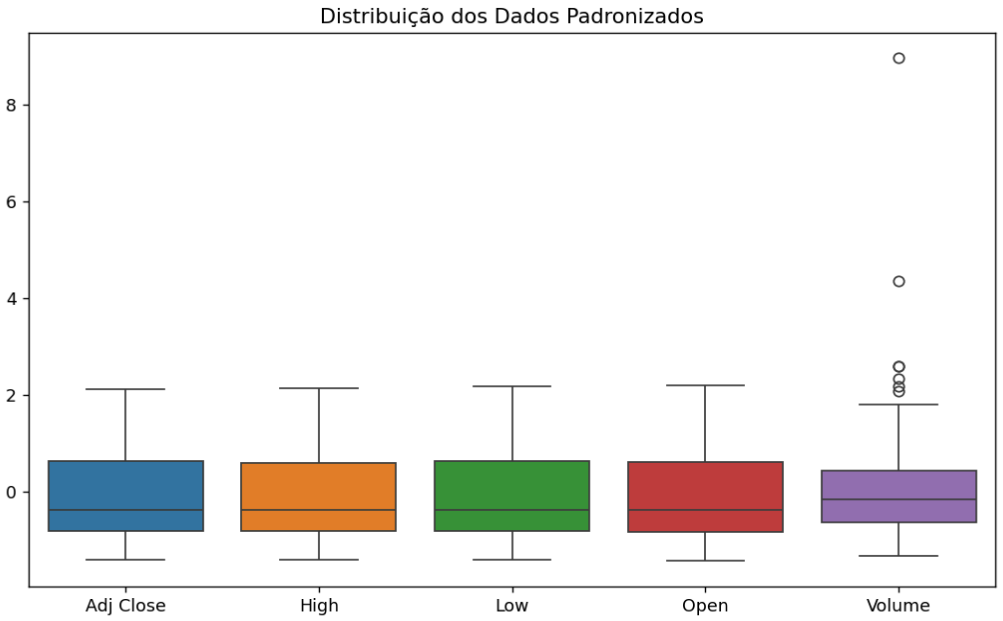
```
Volume      -0.147574
Name: Close, dtype: float64

Atributos relevantes selecionados (correlação > 0.1):
['Adj Close', 'High', 'Low', 'Open', 'Volume']

Dados normalizados e padronizados com sucesso.

Dataset limpo salvo em: C:\Users\vitor\Downloads\IA.BLACK\IA--SI\AV1\Ferrari (20.04.23 - 01.05.24).csv

Gerando gráficos de correlação e distribuições...
```

(Figura 9: Resultado de execução da questão 5)

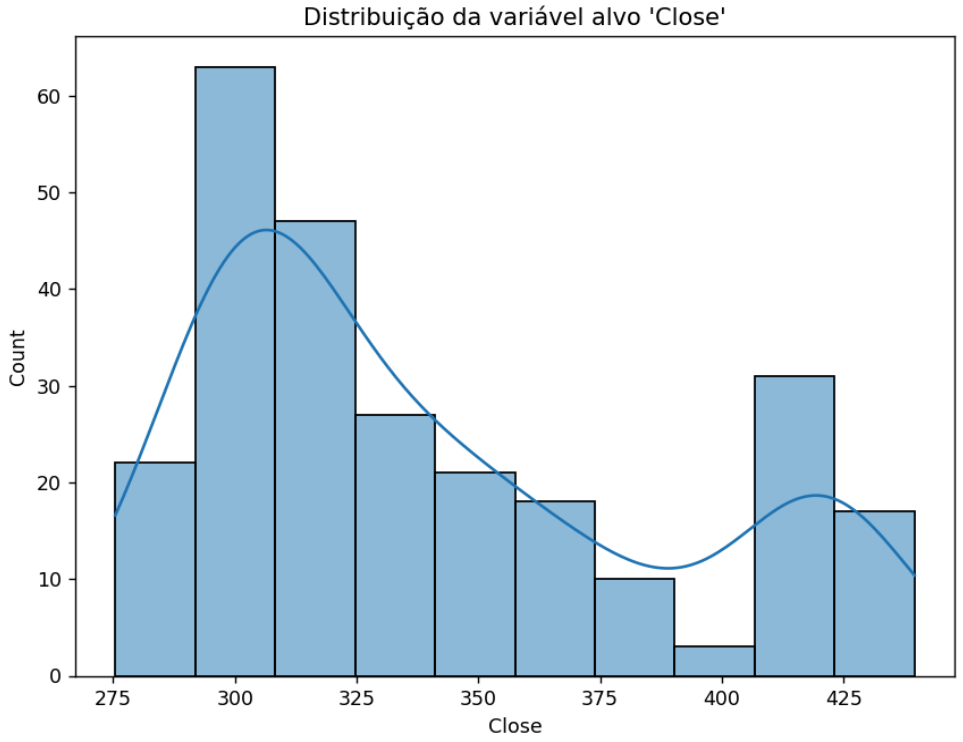
E na quinta etapa: deve-se avaliar o modelo calculando as métricas para avaliar a qualidade do ajuste e por último visualizar o gráfico da reta. então temos como resultado uma visão clara e eficaz do modelo dentro dos atributos relevantes, com possibilidade de ajustar o fluxo conforme a necessidade.



(Figura 10: resultado em gráficos da distribuição normalizada da questão 5)



(Figura 11: Resultado em gráfico da distribuição padronizada da questão 5)



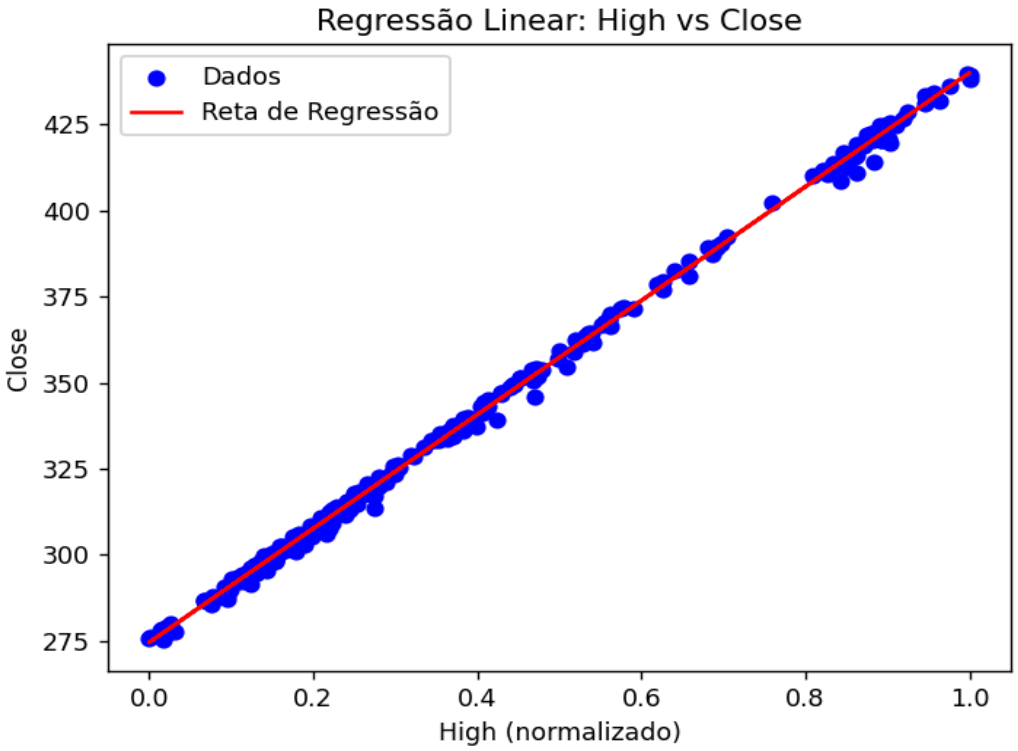
(Figura 12: Resultado em gráfico da distribuição da variável alvo da questão 5)

6-Questão.

Essa questão apresenta como seu objetivo crucial a utilização do atributo mais relevante identificado na analise, para prever o tributo alvo pela regressão linear simples.

Primeira etapa: é normalizar escalando o atributo ***high*** através do comando ***MinMaxScaler*** entre 0 e 1 melhorando a performance do modelo.
Segunda etapa: é treinar o modelo de regressão linear com os valores originais e normalizados.
Terceira etapa: é avaliar as métricas de avaliação ***RSS, MSE, RMSE e R^2.***
Quarta etapa: vamos visualizar plotando a reta de regressão juntamente com os pontos.

Resultados:
Coeficiente com inclinação reta, intercepto irá apresentar o valor de ***close*** quando ***high*** for 0, ***RSS*** simaria os resíduos dos quadrados, ***MSE e RMSE*** irão fazer a medição do erro médio, indicando qual modelo mais próximo das observações reais e o ***R^2*** avaliará a proporção da variância.

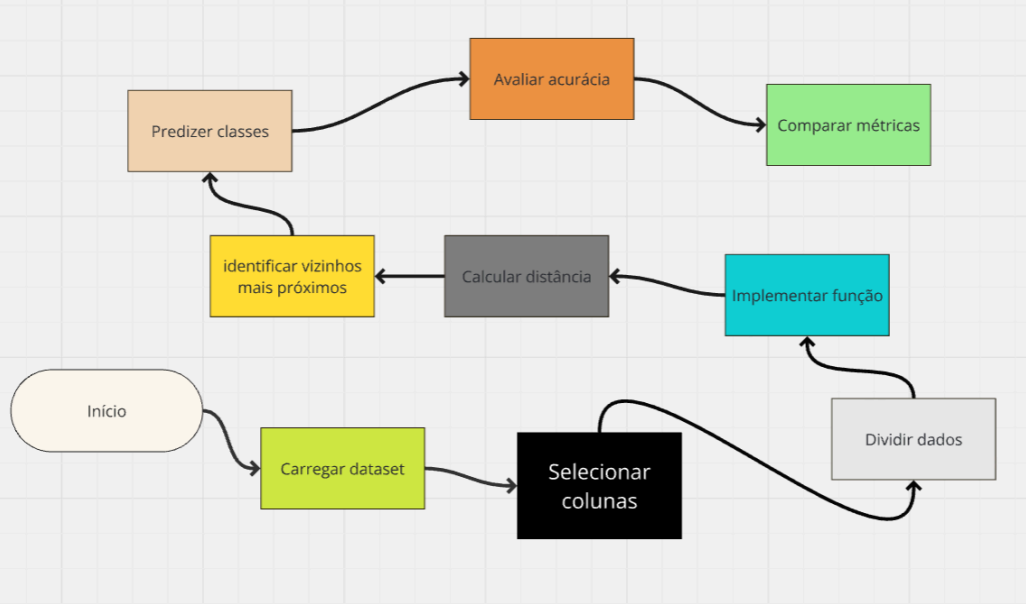


(Figura 13: resultado em gráfico da regressão linear)

7-Questão.

O objetivo dessa questão tem como base a aplicação de técnicas de modelagem preditiva mediante a regressão linear e avaliar o seu referente desempenho em suas variantes em um conjunto de dados reais.

Seguindo esse fluxograma;



(Figura 14: fluxograma de processos)

Na Primeira etapa: é preciso carregar os dados referentes ao dataset do arquivo CSV.
Na segunda etapa: transparece a definição dos modelos sendo eles **Regressão Linear Clássica, Ridge e Lasso.**
Na terceira etapa: Se dará lugar a validação cruzada conhecida como K-fold, dividindo os dados em 5 segmentos folds para uma melhor avaliação,treinando cada modelo, fazendo as predições nos dados de testes e calculando as metricas **RSS, MSE, RMSE e R^2**.
Na quarta etapa: é calculada as métricas em cada modelo folds.
Na quinta etapa: é identificado o melhor modelo e por ultimo a analise visual é realizada com o intuito de comparar as metricas de cada modelo facilitando a interpretação dos resultados. Portanto, apresentado como resultados, a **Regressão Linear** apresentou desempenho consistente e sem regularização, o **Ridge** apresentou um melhor controle de overfitting por conta da regularização L2 e por ultimo o **Lasso** promoveu uma seleção de características ao zerar os coeficientes sendo bastante eficiente em modelos com muitas variaveis.

```
Média das Métricas por Modelo (k-fold):

Linear Regression:
  RSS: 9.7797
  MSE: 0.1893
  RMSE: 0.4178
  R²: 0.9999

Ridge:
  RSS: 9.7814
  MSE: 0.1893
  RMSE: 0.4180
  R²: 0.9999

Lasso:
  RSS: 10.5509
  MSE: 0.2041
  RMSE: 0.4293
  R²: 0.9999

O melhor modelo é: Ridge com R² médio de 0.9999
```

(Figura 15: Resultado da execução do código da questão 7)

4. Conclusões

Sim, mesmo diante da dificuldade do trabalho, conforme as pesquisas e o desenvolver do mesmo, foi possível a absorção de algumas partes do conteúdo, por mais que não possa ter tido a compreensão de tudo o que foi repassado, acredito que tenha sido essencial a elaboração desse trabalho para apresentar os erros e dificuldades com a visão de melhora-los.

5. Próximos passos

Eu sugiro na verdade não um passo em si, mas sim uam correção mais compassada e tranquila dessas questões para que não só eu, mas toda a sala consiga desenvolver um pouco mais de programação ja que o nosso curso tem essa carência nessa area.