

NASA Asteroid Classification

Projeto 2 de IA

Group_A2_83

- João Cardoso: 202108732
- Luiz Queiroz: 202102362
- Isabel Moutinho: 202108767

Especificação do trabalho

Problema

Este projeto trata da aplicação de modelos e algoritmos de machine learning relacionados a supervised learning para a classificação de asteroides próximos da Terra (NEOs), através de dados que já foram coletados. A classificação será se um asteroide é potencialmente perigoso ou não.

Pré

Processamento

Análise do set de dados, definição de training e test sets, filtração de atributos irrelevantes ou adição dos que estão em falta.

Objetivo

Através de vários atributos de um asteroide, conseguir dizer se ele poderá provocar perigo para a Terra (hazardous).



Algoritmos

Definição e aplicação de algoritmos de supervised learning.



Avaliação

Análise gráfica e estatística dos algoritmos usados, bem como comparações entre eles e o processo de aprendizagem.



Bibliografia



Asteroides

- **Dataset:**
<https://www.kaggle.com/datasets/lovishbansal123/nasa-asteroids-classification>
- **API:** <http://neo.jpl.nasa.gov>
- **NeoWs** (Near Earth Object Web Service):
<https://www.neowsapp.com/swagger-ui/index.html>



Algoritmos

- <https://www.coursera.org/articles/decision-tree-machine-learning>
- <https://www.solver.com/xlminer/help/neural-networks-classification-intro>
- <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>

Descrição

Ferramentas

Para este projeto será usada a linguagem de programação Python em conjunto com pacotes e bibliotecas para análise de dados como pandas (para manipulação de dados), matplotlib e Seaborn (para visualização dos dados em Python e gráficos estatísticos), Scikit-Learn (para desenvolvimento de algoritmos de machine learning), NumPy e SciPy para lidar com diferentes operações usando grande número de dados. Será usado o Jupyter Notebook para uma computação interativa e exploratória. O ambiente de desenvolvimento será criado usando o Anaconda para gerenciar os diferentes pacotes e bibliotecas.

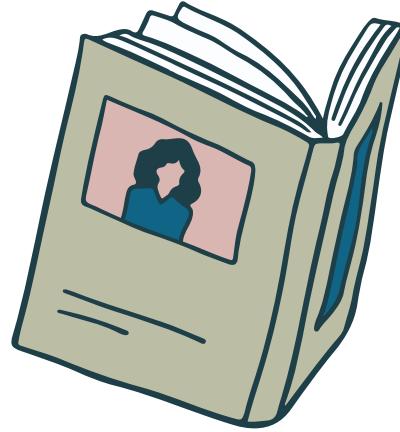


Algoritmos

- Decision Tree: É representado como uma estrutura de árvore, onde cada nó interno representa uma "pergunta" ou "teste" num atributo, cada ramo representa o resultado desse teste e cada folha da árvore representa uma classe.
- Neural Network: Baseado na estrutura neuronal do cérebro, uma rede neuronal processa os dados de entrada através de camadas interconectadas de neurônios que aplicam funções de ativação, ajusta os pesos internos durante o treinamento usando backpropagation para minimizar a função de custo e usa a camada de saída, geralmente com uma função softmax, para prever a categoria dos dados de entrada.
- K-Nearest Neighbors: Classifica uma nova observação com base nas categorias das suas 'k' observações mais próximas no conjunto de treinamento. Ele mede a distância entre os pontos de dados, seleciona os 'k' pontos mais próximos e atribui a classe mais comum entre os vizinhos como a prevista para o input.

Trabalho Implementado

01



Pesquisa de informação sobre asteróides, nomeadamente NEOs e PHOs, bem como atributos usados para derivar se são perigosos ou não e mais informações sobre os algoritmos a serem utilizados.

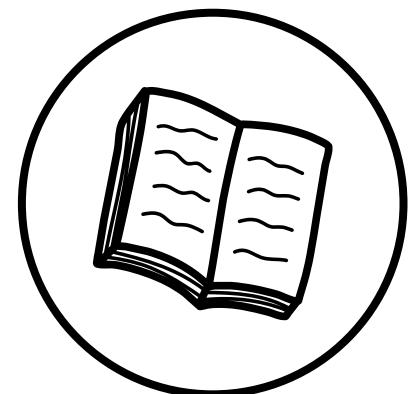
02



Carregamento do dataset e análise dos dados contra possíveis valores duplicados, irrelevantes, como os de natureza informativa, e valores constantes.

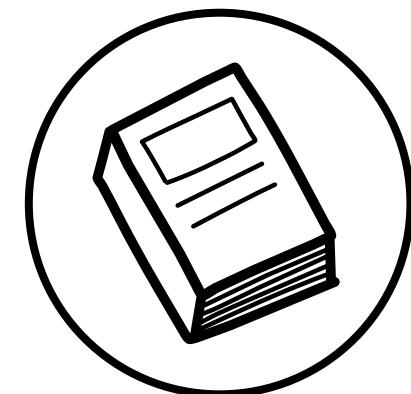
Observação da correlação entre características dos asteroides, e análise da relação características-perigosidade. Simples teste de Decision Tree.

Data Pre-Processing



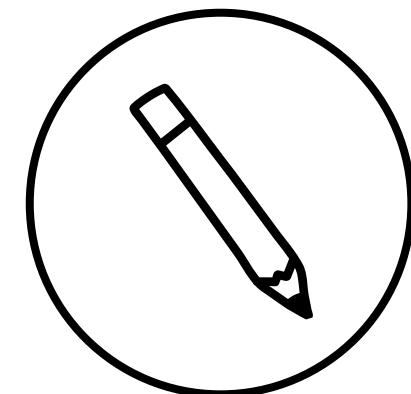
Valores em Falta

Foi feita uma inspeção dos valores de todas as colunas, de forma a verificar que todos tinham informação.



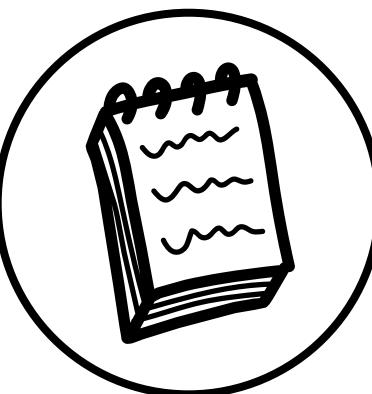
Data Duplicada

Através de uma análise de heatmaps de correlações, detetamos conjuntos de colunas que fornecem a mesma informação.



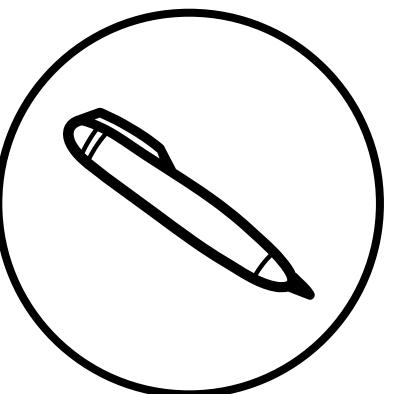
Atributos Informativos

Atributos puramente informativos, como IDs, foram removidos, já que não têm significado no contexto da classificação do asteroide como perigoso.



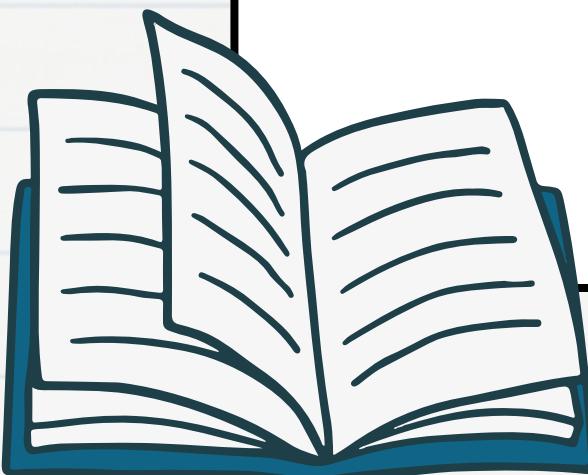
Valores Unicos

Atributos que apenas têm um possível valor, entre todas as entradas do data set, são irrelevantes e portanto foram removidos.

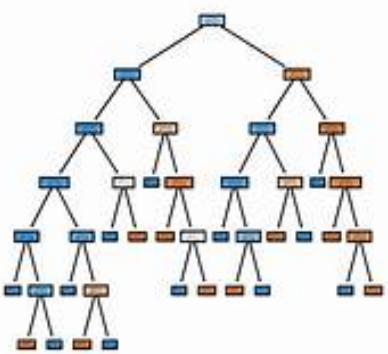


Correlação de dados

Foi feita uma análise dos pares de atributos com alta correlação, de forma a induzir quais devem ser retirados ou combinados.

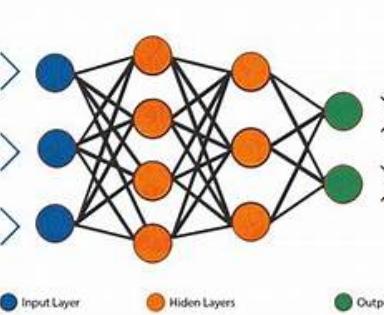


Developed Models



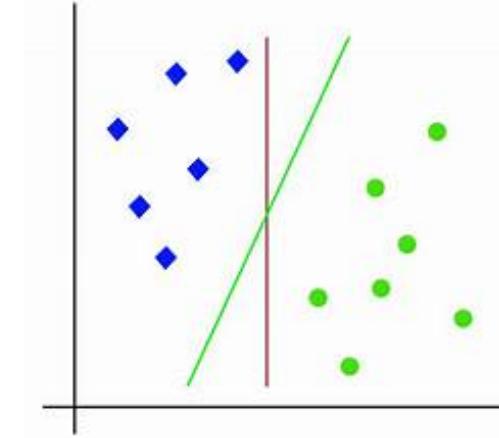
Decision Tree

Nós utilizamos o criterion de gini e o best splitter. Este classificador é o único que não normaliza o dataset.



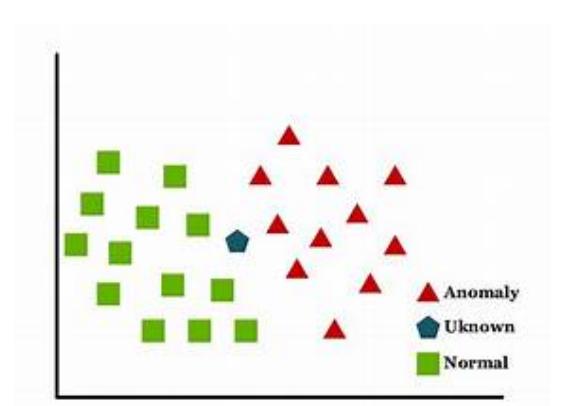
Multi Layer Perceptron

Usamos o solver lbfgs com constant learning rate e com 200 de iterações máximas.



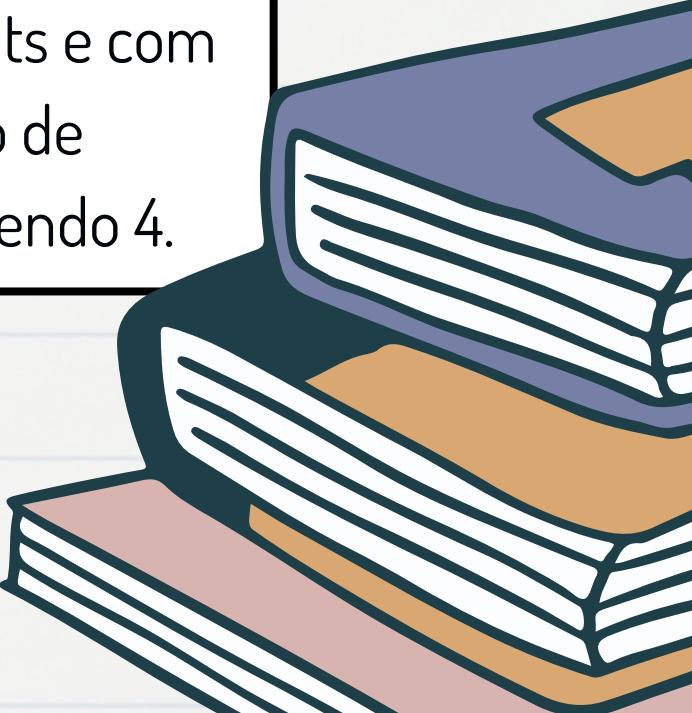
Support Vector Machine

Foi usado o kernel rbf com balanced class weights e com 500 de iterações máximas

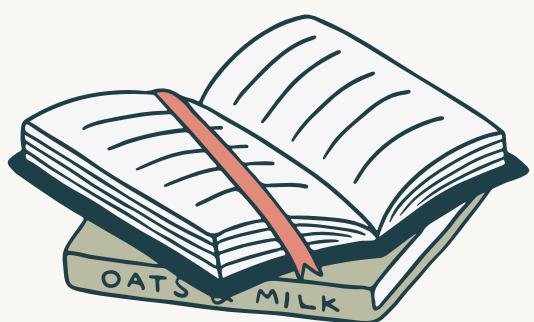
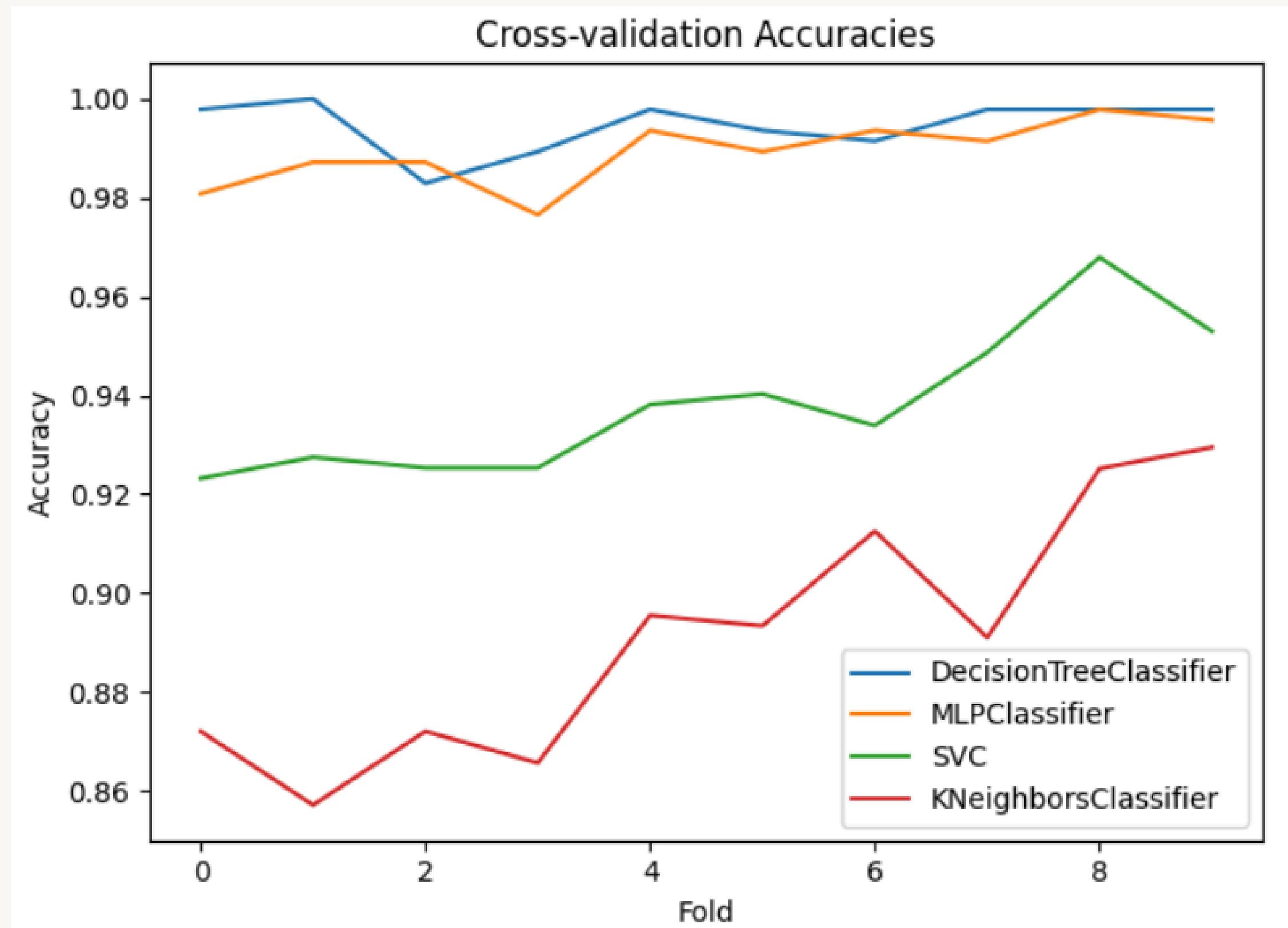


K Nearest Neighbours

Nós usamos o algoritmo auto com distance weights e com o número de neighbours sendo 4.

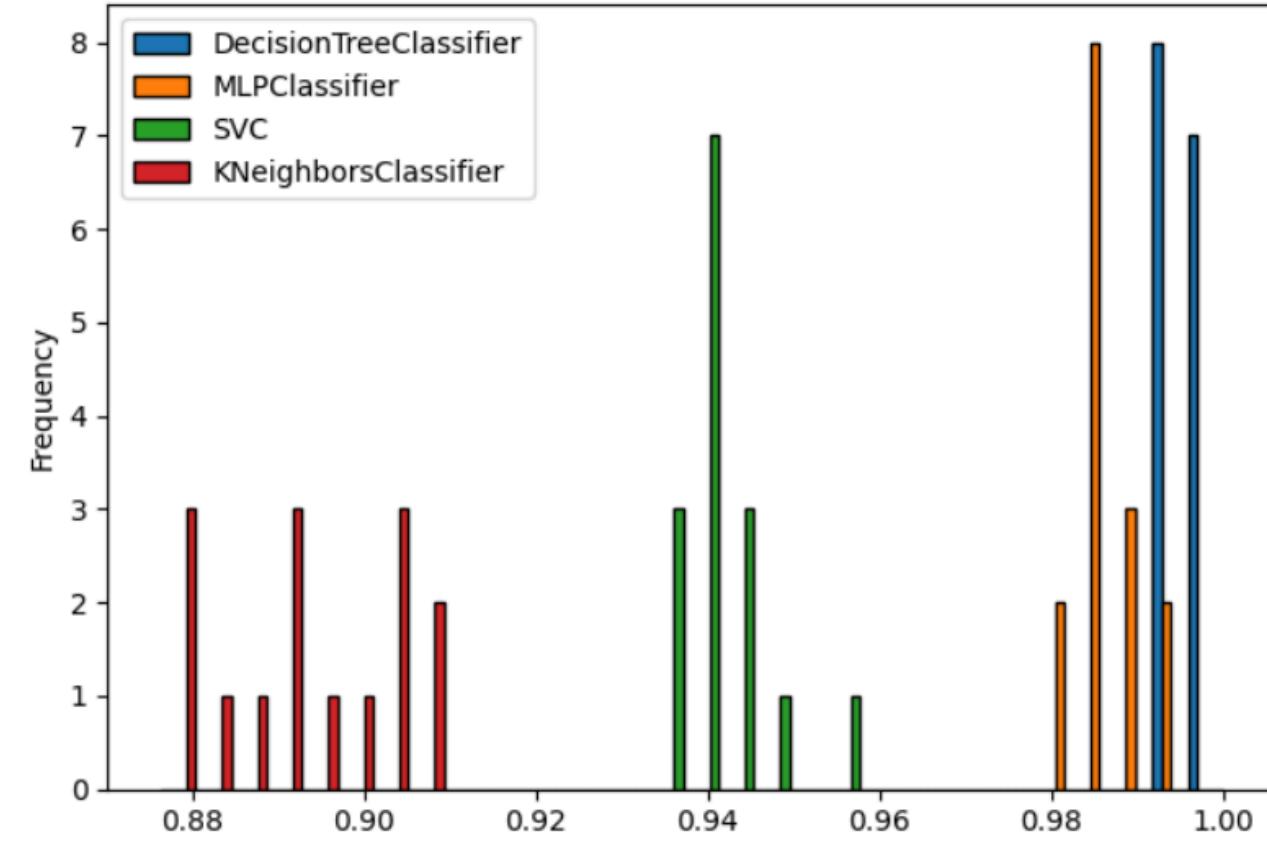


Cross-Validation

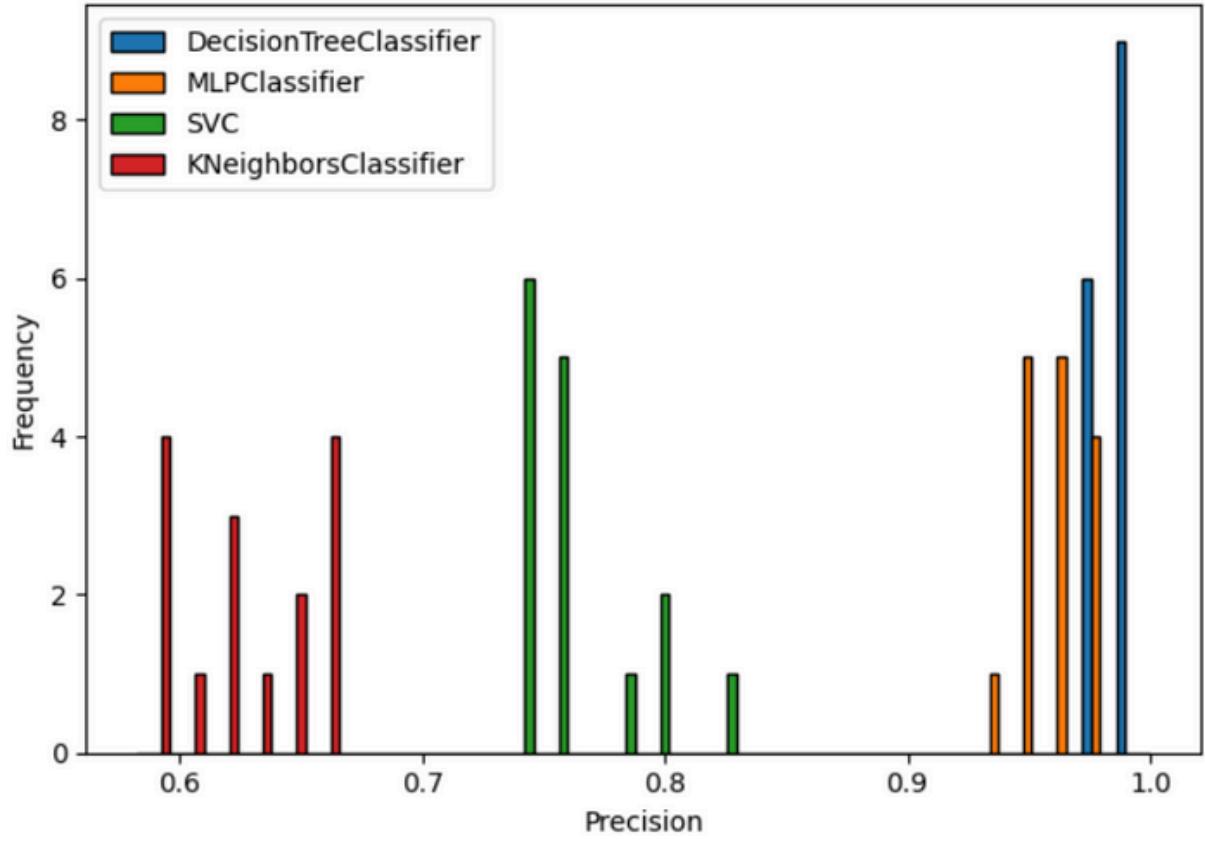


Evaluation Metrics

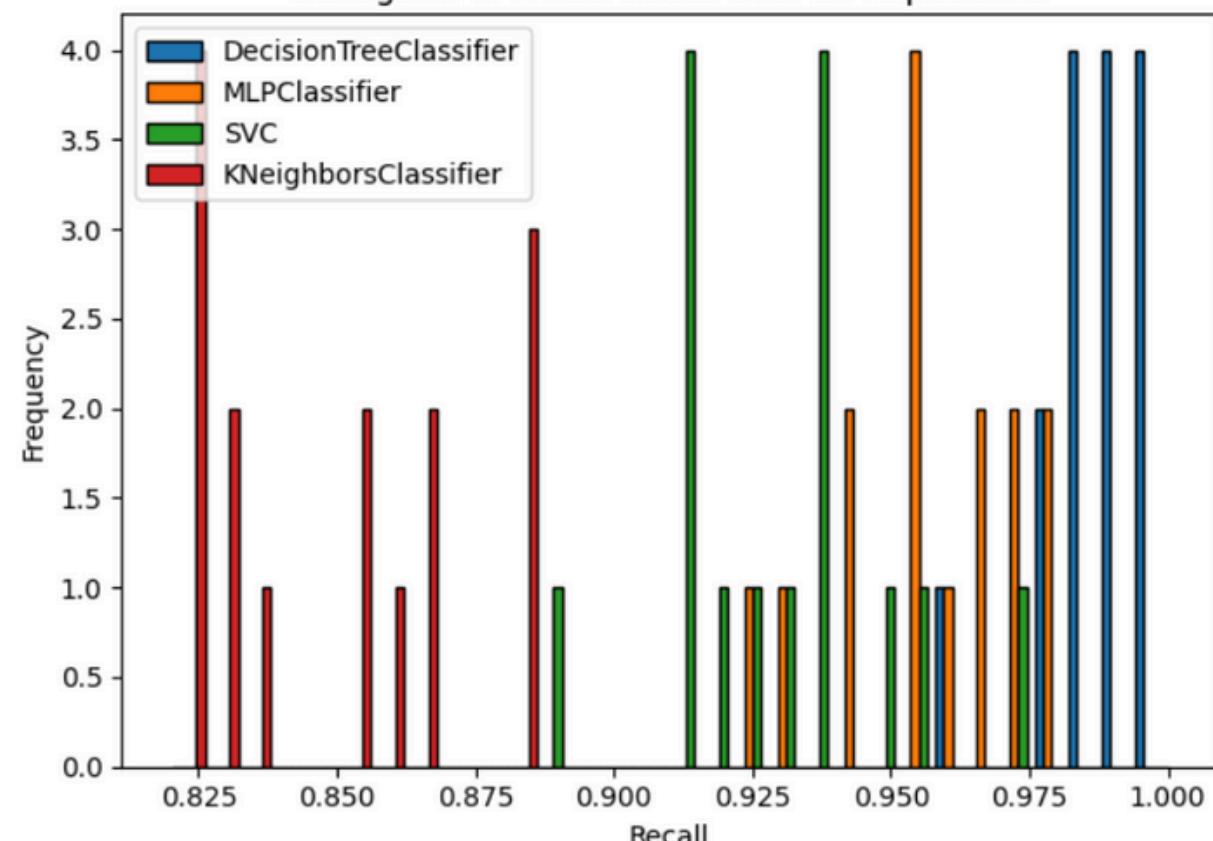
Histogram of Model Accuracy over 15 Repetitions



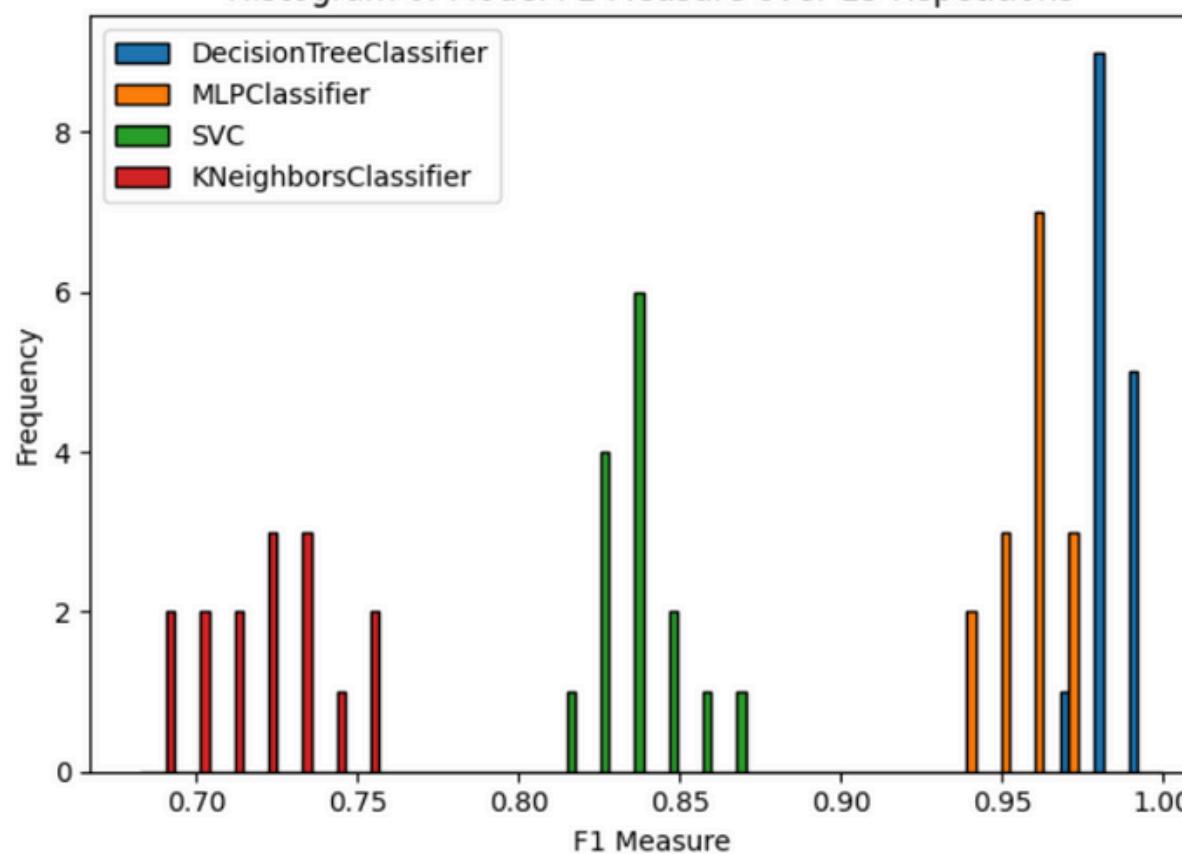
Histogram of Model Precision over 15 Repetitions



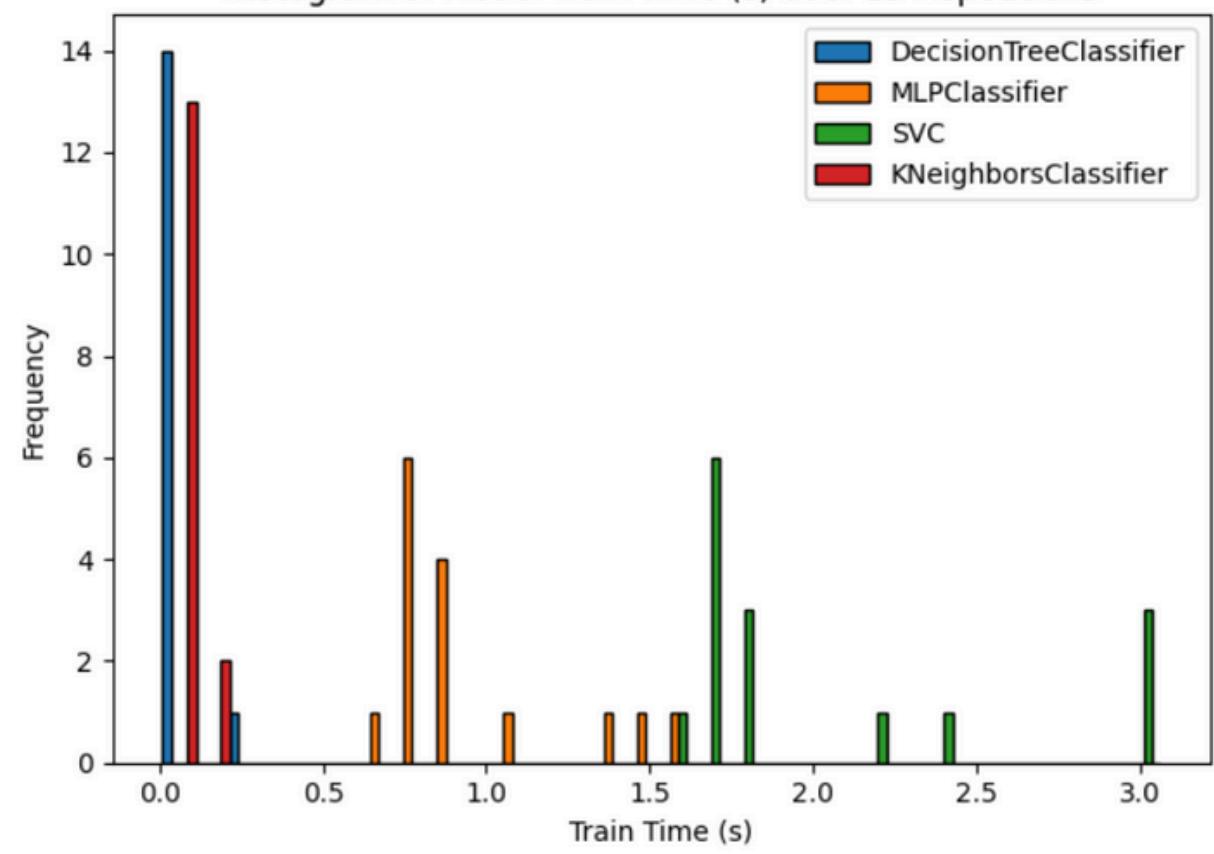
Histogram of Model Recall over 15 Repetitions



Histogram of Model F1 Measure over 15 Repetitions



Histogram of Model Train Time (s) over 15 Repetitions



Learning Curve and ROC

