U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Elementos de Inteligência Artificial e Ciência de Dados Trabalho Prático 2

Data exploration and enrichment for supervised classification

Grupo 22

Trabalho realizado por:

- Inês Alves up202104656
- Maria Cruz up202104592

O Problema

Neste 2º trabalho vamos explorar Hepatocellular Carcinoma (HCC) Dataset.

O objetivo é analisar os diferentes pacientes (que correspondem a cada linha do DataSet) de modo a que seja possível criar um modelo, através da inteligência artificial, que consiga prever a sobrevivência do paciente após 1 ano do diagnóstico. Os dados reais dessa sobrevivência são dados pela última coluna do conjunto de dados, coluna com uma importância acrescida no trabalho, designada por ‘Class’.

Neste problema temos 165 pacientes e 50 colunas, onde 102 sobrevivem passado 1 ano de diagnóstico e 63 não.

Portanto vamos procurar encontrar algum padrão de forma a conseguir tirar as conclusões mais acertadas acerca do diagnóstico.

Pesquisas relacionadas com o trabalho

- Usamos os materiais do Moodle para o desenvolvimento do trabalho tal como o ChatGPT e os seguintes links:

  https://journals.sagepub.com/doi/10.1177/1460458220984205

  https://www.sciencedirect.com/science/article/pii/S1532046415002063

  https://socgastro.org.br/novo/wp-content/uploads/2021/01/easl-easl-guidelines-management-of-hepatocellular-carcinoma.pdf

- Link do GitHub:

  https://github.com/MariaSCruz/Trabalho-IA---grupo22

Descrição do problema e implementação

- **Data exploration** – Ao começar a explorar o dataset vimos que possuía muitos valores perdidos, representados por '?' e também em certas colunas como 'Encephalopathy' e 'Ascites' continham bastantes valores nulos. No entanto estas últimas eram para considerar os valores como nulo mesmo, enquanto que os outros deveriam ser identificados e substituídos de forma conveniente.

  Também é importante referir que só a coluna 'Age' é que estava a ser identificada como int64, havendo outras colunas com valores numéricos, mas especificamente float64 mas estavam a ser identificadas como object.

- **Data preprocessing** – Portanto fizemos as devidas alterações em relação aos dados, substituindo os valores perdidos, no caso das colunas numéricas pela média e no caso das não numéricas pela moda. Também alteramos a identificação do tipo dos atributos que estavam a ser mal identificados.

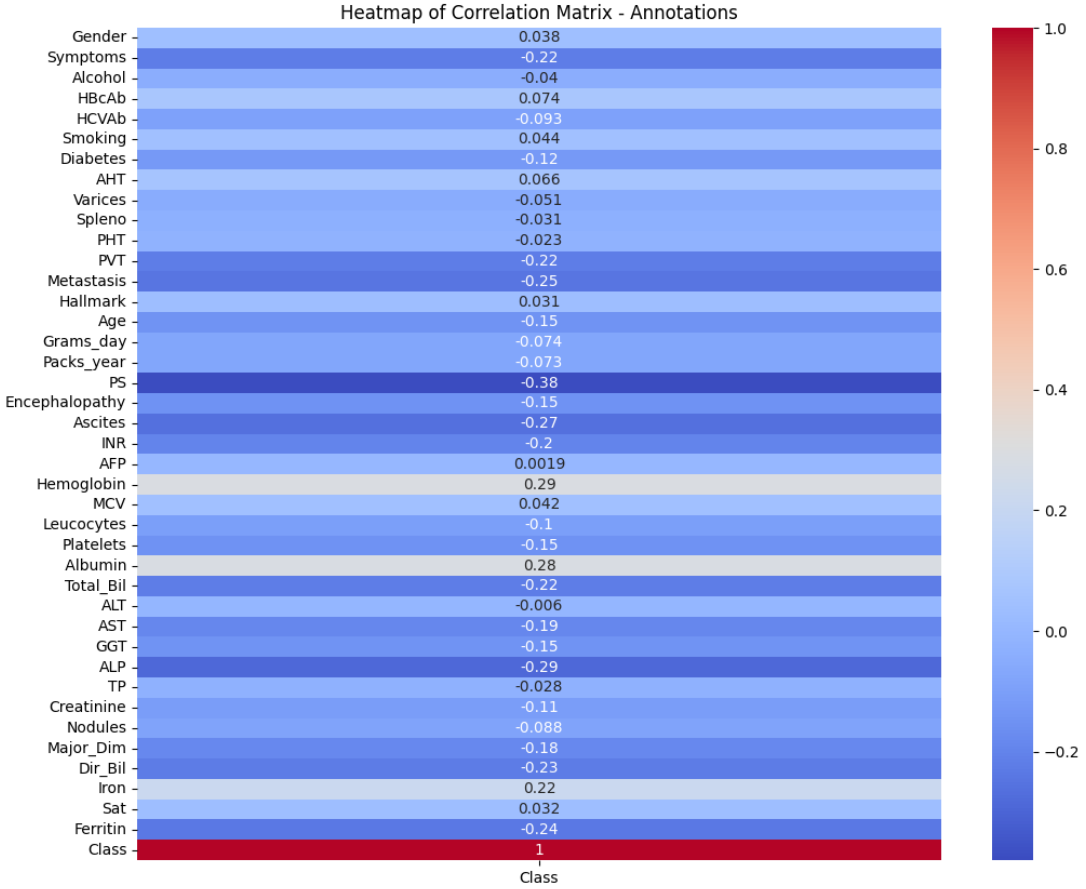
```
for x in dt.columns:
    if x != 'Encephalopathy' and x!= 'Ascites':
        dt[x] = dt[x].replace('?', np.nan)
    elif x=='Encephalopathy':
        dt[x] = dt[x].replace(np.nan, 'None')
        dt[x] = dt[x].replace('?', np.nan)
    else:
        dt[x] = dt[x].replace(np.nan, 'None')
        dt[x] = dt[x].replace('?', np.nan)
```

```
# Replace null values in numeric columns with the average
for col in number_cols:
    mean_value = dt[col].astype(float).mean()
    dt[col] = dt[col].fillna(mean_value)

# Replace null values in non-numeric columns with mode
for col in complement_number_cols:
    mode_value = dt[col].mode()[0]
    dt[col] = dt[col].fillna(mode_value)
```

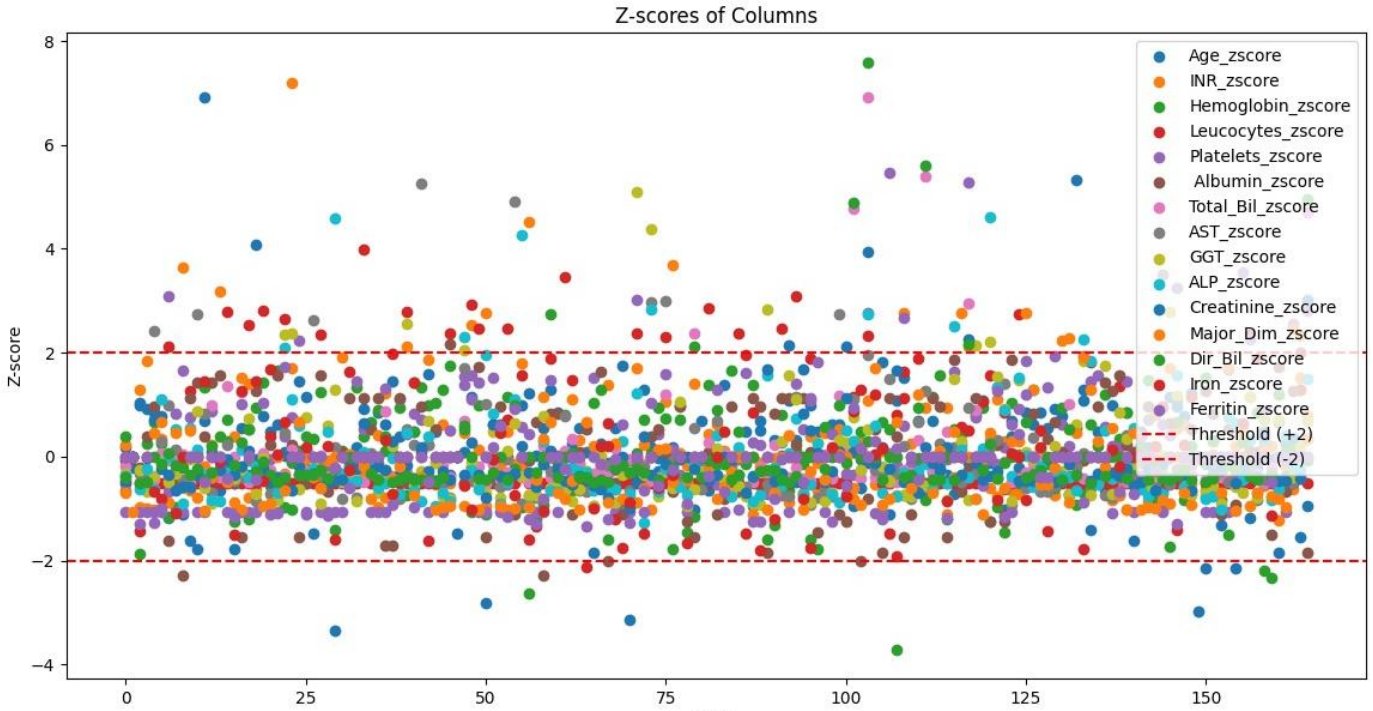
Data preprocessing (cont)

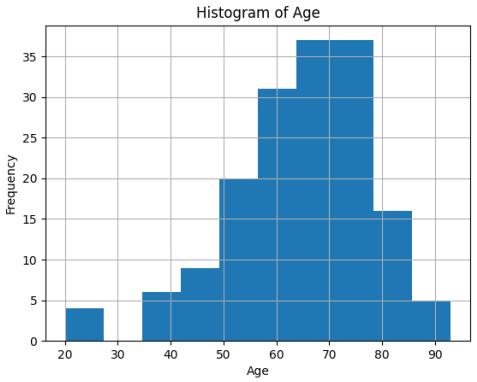
- Fizemos histogramas, boxplots e gráficos de dispersão para analisar os dados.
- Acabamos por remover as colunas com pouca variância, nos casos das colunas com dois tipos de valores, o que tinha menor ocorrência se fosse menor ou igual a 20. Isto porque, tendo em conta o objetivo, achamos que essas colunas não iam ser indicadoras da sobrevivência, uma vez que na 'Class' temos 102 'Lives' e 63 'Dies'.
- Mudamos os valores não numéricos para 0's e 1's para facilitar a visualização dos resultados e para podermos comparar as colunas todas umas com as outras.
- Fizemos também os coeficientes de correlação tanto para criar a matriz de correlação como para fazer gráficos de calor com os valores dos coeficientes de correlação de todas as colunas com a 'Class' que neste problema era o que nos interessava. Optamos por dois caminhos no cálculo dos coeficientes, um em que fizemos diferente para as colunas que só tinham 0's e 1's e outro que calculamos tudo por Pearson. Embora a conclusão que tiramos é que o resultados dos valores são melhores no segundo caso.
- Também deixamos cair as colunas que tinham coeficientes muito baixos (abaixo de 0.1).
- Calculamos o Z-score e os outliers definidos por serem maior que 2 no módulo do Z-score. Verificamos que tinha um valor pequeno de colunas em que apenas 5% dos dados estavam nessas condições pelo que não alteramos o Dataset nesta etapa.
- Normalizamos os dados através do MinMaxScaler() numa tentativa de preparação para o Machine Learning. (Data Scaling)

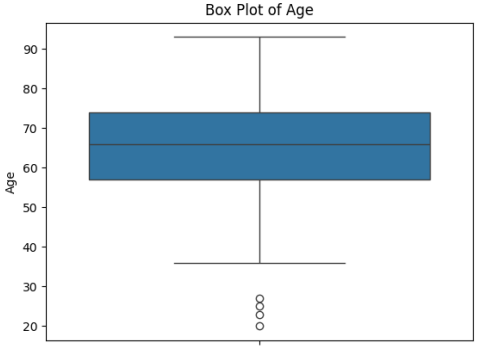


Aqui, concluímos que os coeficientes na sua generalidade são muito baixos daí os tons frios deste mapa. No entanto, verificamos que a coluna ‘PS’ é a que se destaca mais pela negativa, ou seja, está mais relacionada com ‘Dies’.

Já aqui, podemos observar que temos poucos outliers como já foi referido, pois num intervalo de -2 a 2 os Z-scores encontram-se na sua generalidade dentro do mesmo.

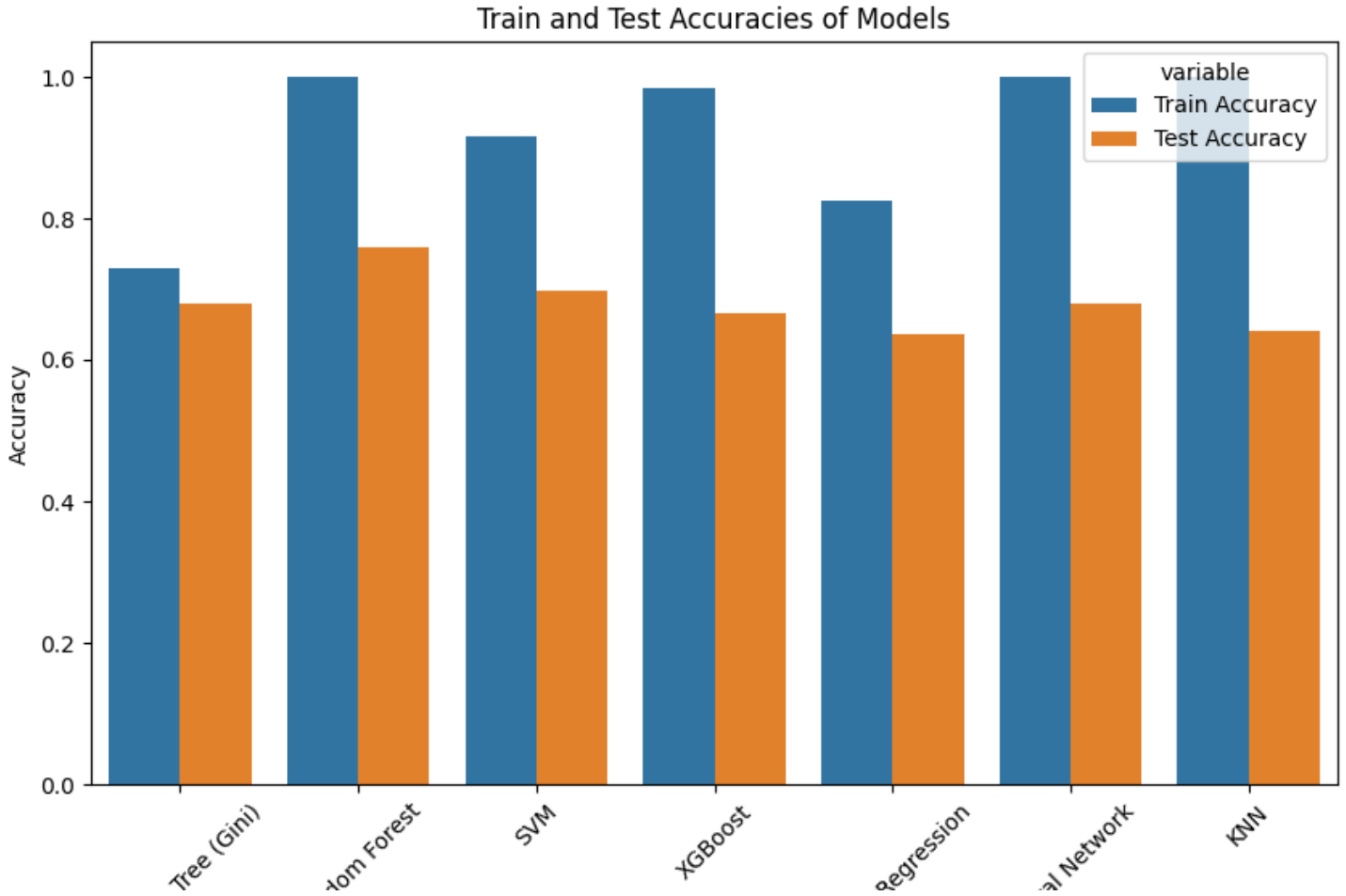






Exemplo para a coluna ‘Age’.

Data Modeling (Supervised Learning)



Usamos estes modelos para a previsão em relação aos dados da 'Class'.

Por exemplo, em relação à árvore de decisão, tem uma boa correspondência entre as acurácias de treino e teste, indicando um modelo que não está muito sobreajustado (overfitting) ou subajustado (underfitting). No entanto, a diferença de desempenho sugere que o modelo pode não ser suficientemente complexo para capturar todos os padrões nos dados de treino. Uma sugestão para a melhoria deste modelo passa talvez por aumentar os nós a serem expandidos.

Os modelos que apresentaram melhores resultados no teste foram o **Random Forest** e a **SVM**.
Modelos como **XGBoost**, **Logistic Regression**, **Neural Network** e **KNN** apresentaram desempenho inferior e indicaram uma tendência ao overfitting.
A **Decision Tree** mostrou-se um modelo intermediário, com acurácia relativamente consistente entre treino e teste.

Data Modeling (Supervised Learning)

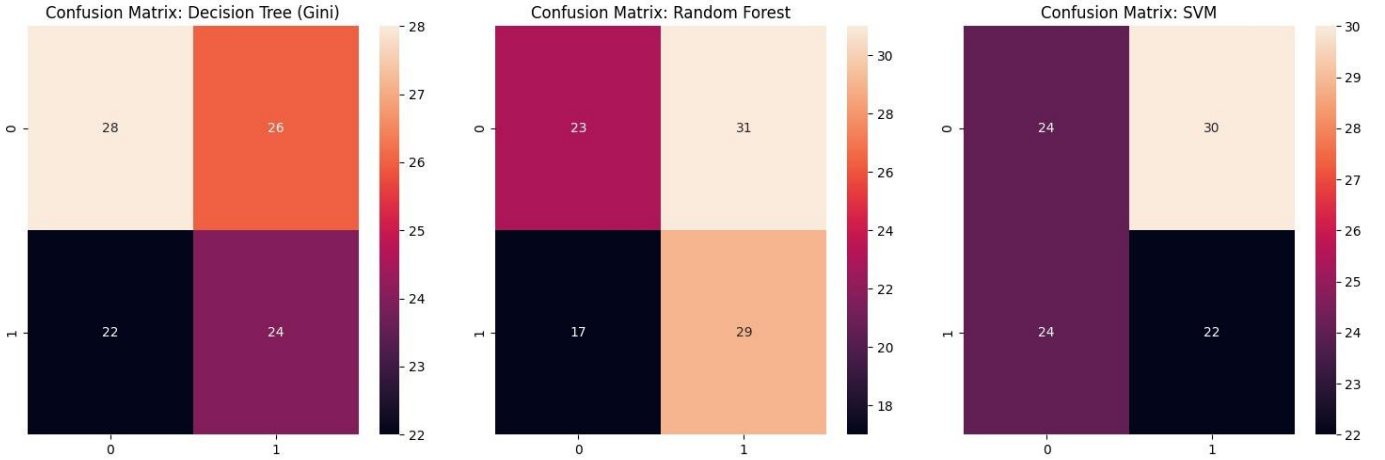
Ajuste de Hiperparâmetros: Realizar uma busca em grade (grid search) ou otimização bayesiana para encontrar os melhores hiperparâmetros para cada modelo.

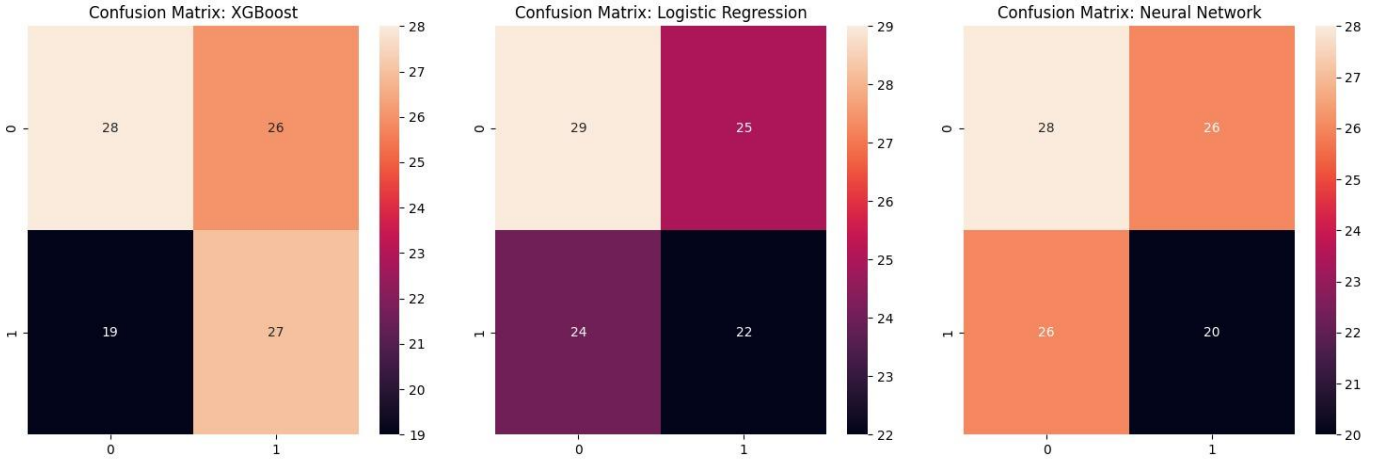
Feature Engineering: Melhorar a seleção e criação de features pode ajudar modelos simples a capturar padrões mais complexos.

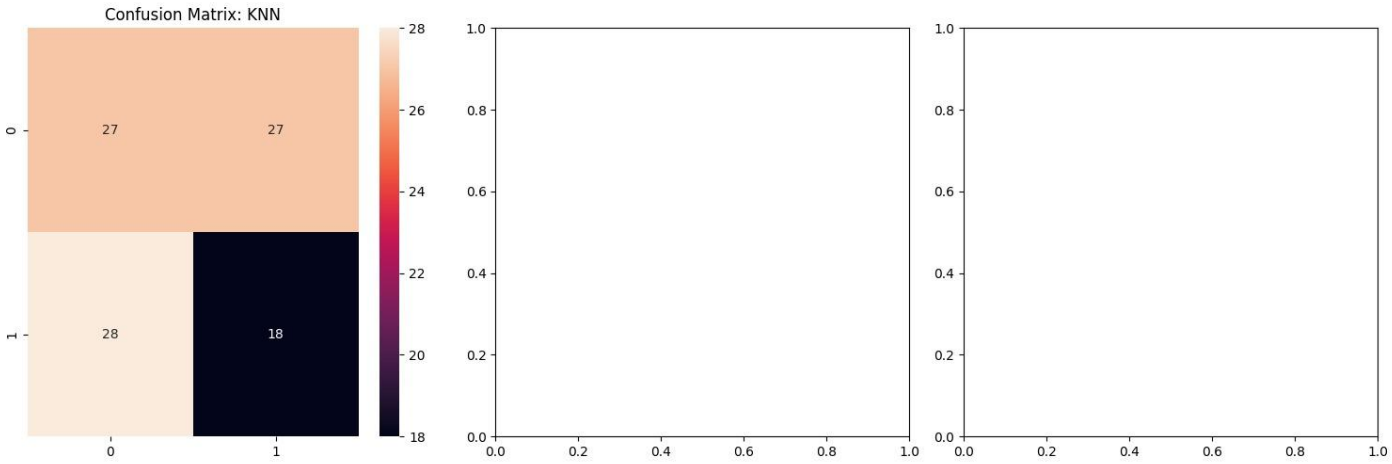
Validação Cruzada: Utilizar k-fold cross-validation para garantir que os resultados não sejam específicos de uma única divisão dos dados.

Regularização: Aplicar técnicas de regularização para modelos que apresentam overfitting, como redes neurais e Random Forest.

Essa análise detalhada pode ajudar a entender o comportamento dos modelos e direcionar esforços para melhorar seu desempenho em futuras iterações.







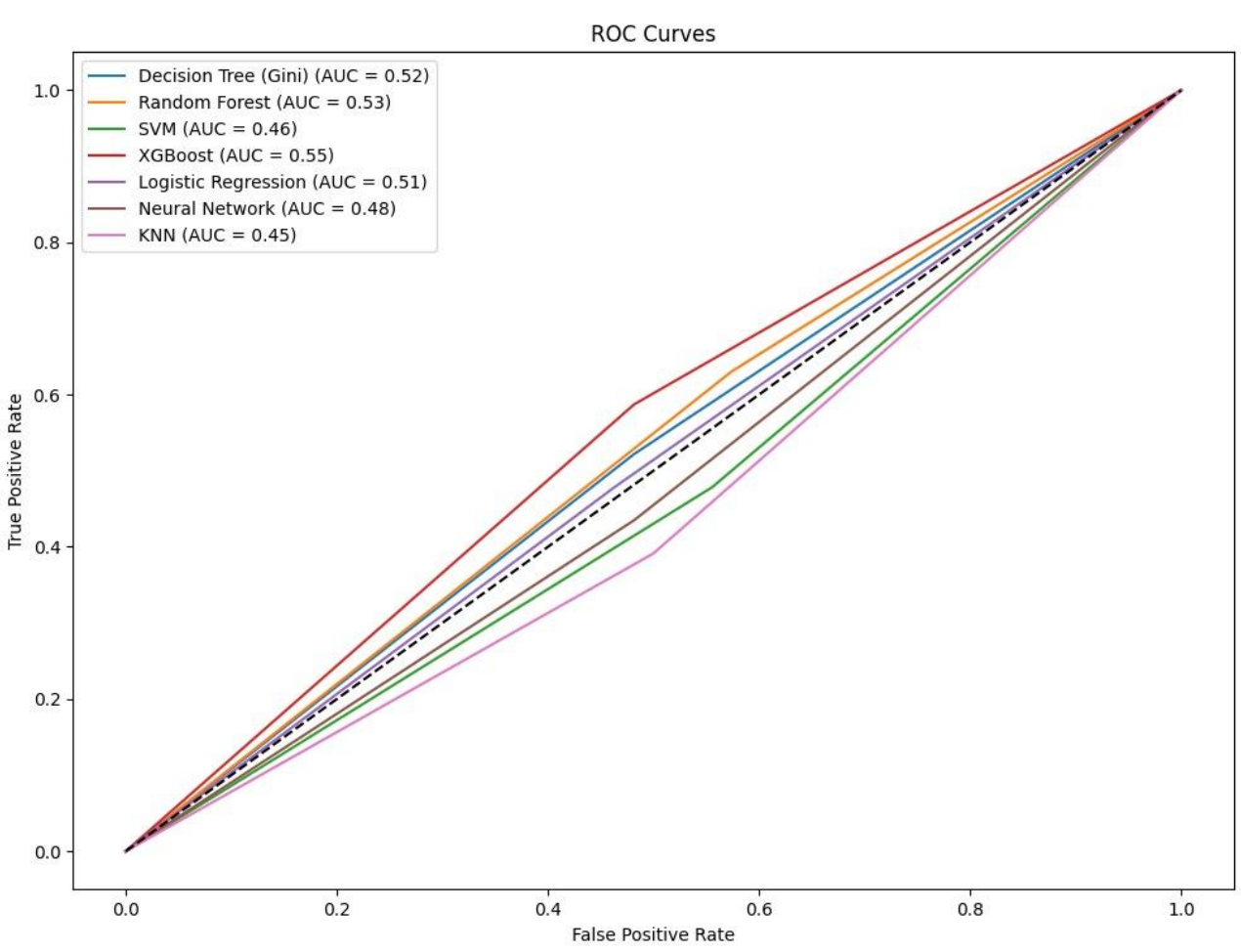
Data Evaluation

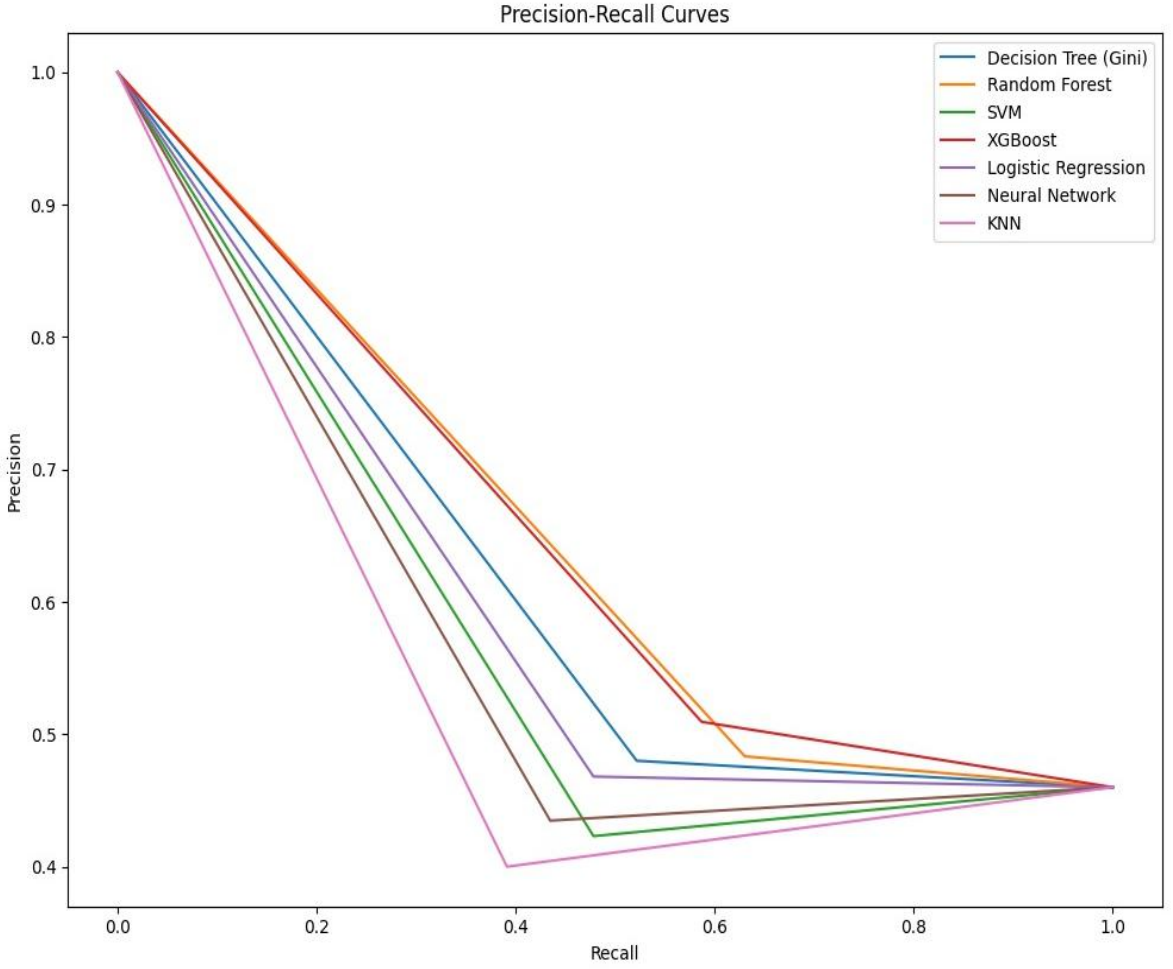
Ao analisar estas matrizes é comum reparar que, em geral, o valor dos True Positive são os mais elevados, porém os valores dos True Negative são os menores, o que é um ponto negativo na precisão do nosso modelo. Também é possível reparar que, em geral os False Positive apresentam valores inferior e os False Negative valores superiores.

É possível concluir que a precisão dos valores positivos é superior à dos valores negativos.

Este padrão não acontece em todas as variáveis, por exemplo em KNN.

ROC and Precision-Recall Curves



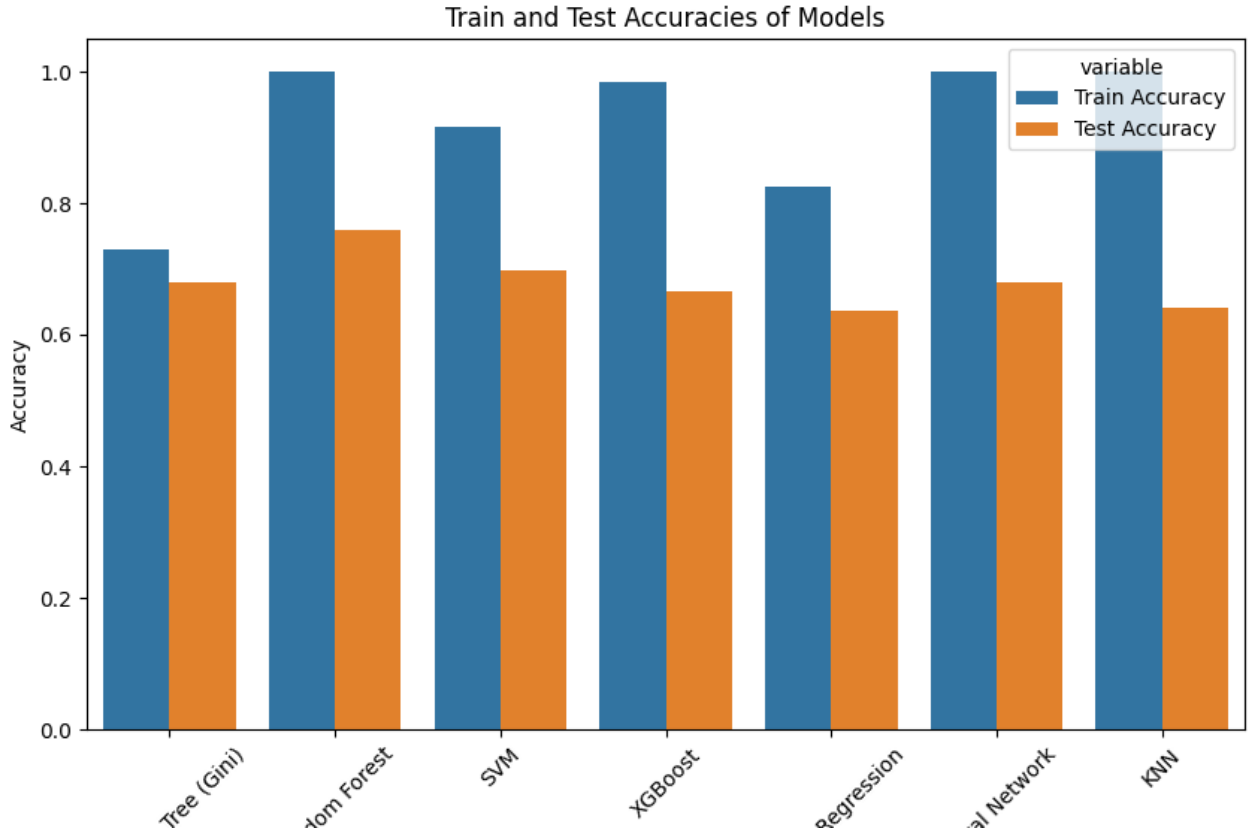


Análise dos gráficos:

ROC: XGBoost tem um desempenho ligeiramente superior (AUC = 0.55), seguido com pouca diferença pelo Random Forest (AUC = 0.53) e Decision Tree (Gini) (AUC=0.52).

PR: Os modelos mostram variações, mas o desempenho é comparável entre eles, sem um modelo que se destaque significativamente.

Nota: Tivemos dificuldade em gerar as matrizes de confusão devido a erros na etapa de previsão necessária para criá-las. Além disso, estamos com problemas ao calcular as probabilidades, o que nos impede de criar os gráficos de curva ROC e recall-precisão. Portanto os nossos dados para a criação das mesmas são aleatórios, pelo que podem diferir um pouco do esperado.



Conclusões

O desenvolvimento deste trabalho de machine learning revelou desafios significativos, particularmente relacionados ao overfitting e à necessidade de modelos mais robustos. A performance geral sugere que enquanto alguns modelos têm potencial, ajustes adicionais são necessários para alcançar uma precisão de previsão mais confiável e generalizável para a sobrevivência de pacientes com HCC.