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 trabalho vamos explorar Hepatocellular Carcinoma (HCC) dataset.

O objetivo é analisar os diferentes pacientes de modo a que seja possível criar um modelo que consiga prever a sobrevivência do paciente após 1 ano do diagnóstico.

Neste problema temos 165 pacientes e 55 atributos, onde 102 sobrevive passado 1 ano de diagnóstico e 63 não.

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

Pesquisas relacionadas com o trabalho

Para já usamos os materiais do moodle para o desenvolvimento desta parte do trabalho.

Também usamos 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

Descrição do problema e implementação

1. **Data exploration** - a nossa database é bastante extensa então primeiro vamos ter que ao analisar criar separações, tanto com o nº de objetos, neste caso os pacientes, assim como nos atributos, tendo atenção que os atributos estão classificados de diferentes maneiras, assim como procurar qualquer tipo de valores não existentes ou possivelmente errados.

2. **Data processing** - A nossa primeira abordagem foi explorar que tipo de classificação tem cada atributo, seguida com a substituição dos valores perdidos, pela moda no caso dos valores não numéricos e pela média nos númericos.

```
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)
```

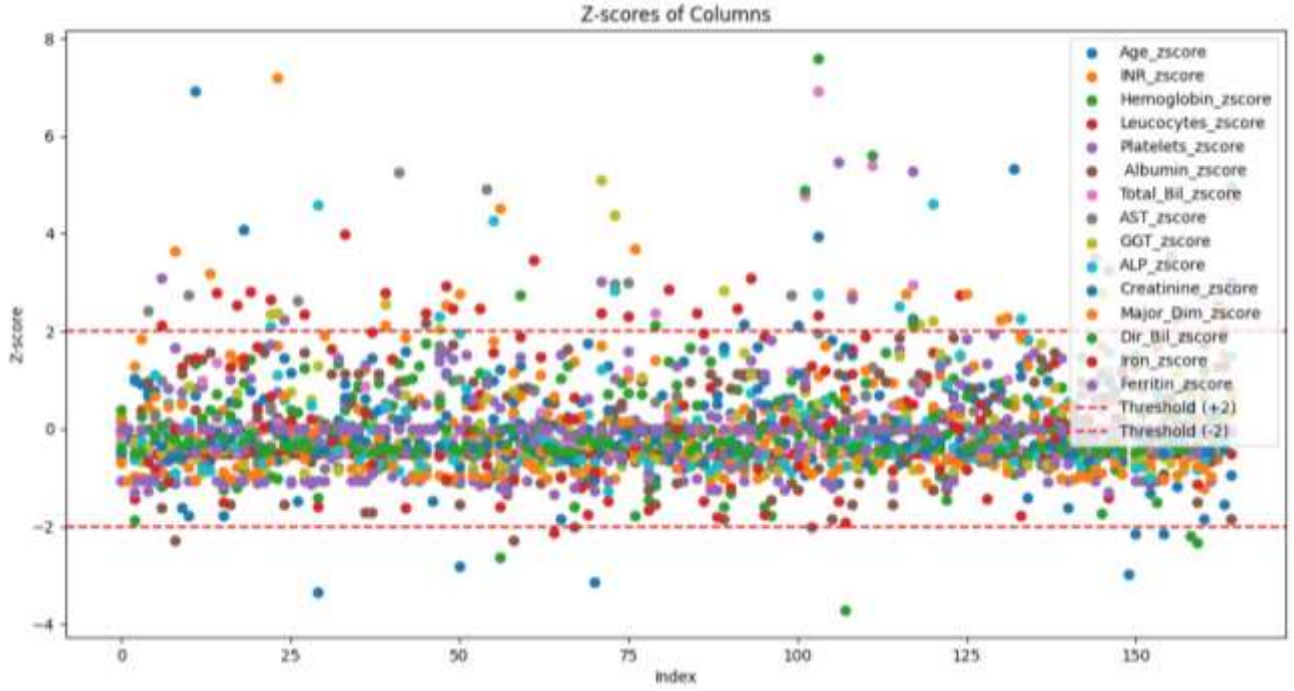
```
# Replace null values in numeric columns with the average
for col in number_cols:
    mean_value = dt[col].astype(float).mean()
    dt[col] = dt[col].fillna(mean_value)

# 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)
```

2.Data processing

- Acabamos por remover as colunas com pouca variância, nos casos das colunas com dois tipos de valores e o que tinha menor ocorrência ser menor ou igual a 20.
- Mudamos os valores não numéricos para 0’s e 1’s para facilitar a visualização dos resultados.

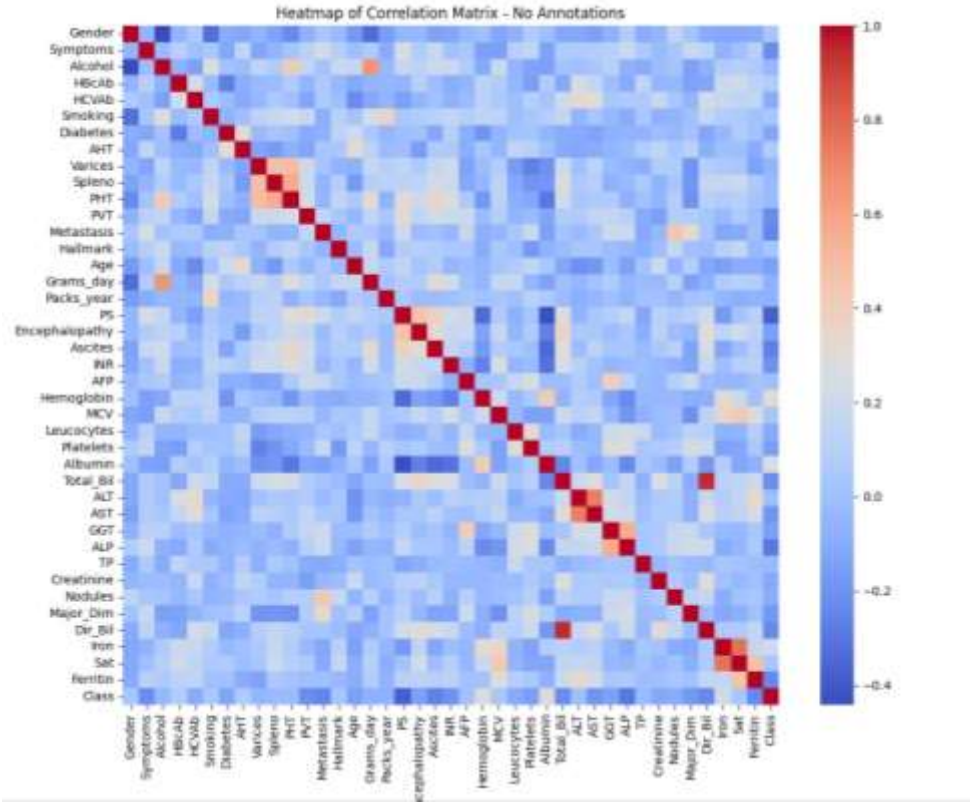
3. Data Modeling (Supervised Learning)

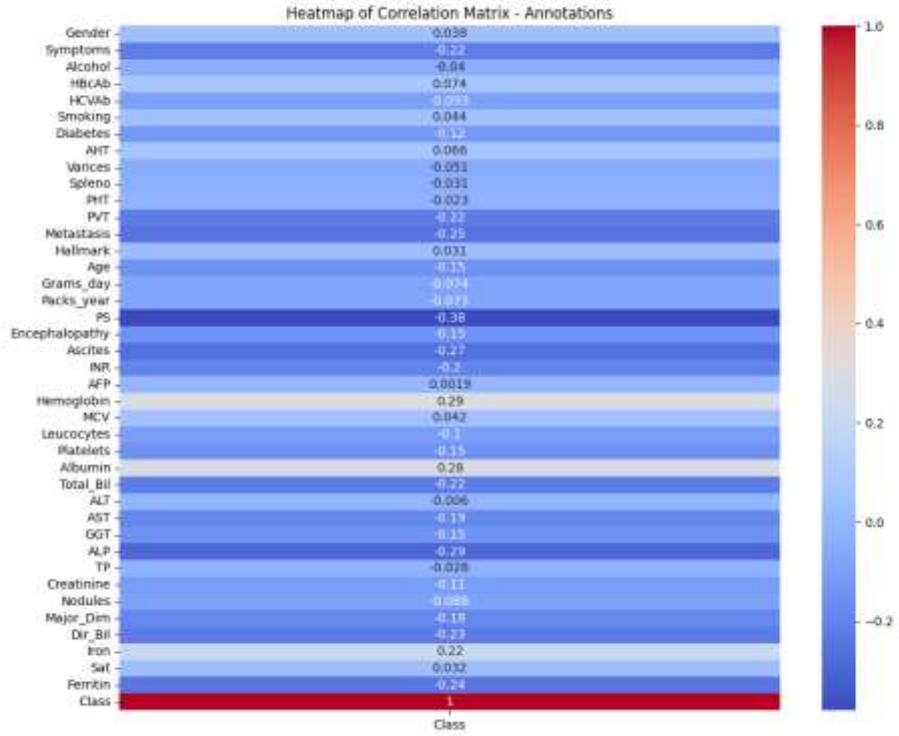
- Primeiramente fizemos a transformação dos atributos não numéricos, para binário, para uma interpretação mais simples;
- Analisámos a variância e o coeficiente de correlação, onde definimos que só iam ser utilizados para o modelo os atributos com o coeficiente de correlação superior a 0.1, o resto foi removido da análise;
- Então ficamos com 22 atributos, 15 numéricos e 7 não numéricos;



- Calculamos os Z-score dos atributos numéricos, para aumentar a precisão do nosso modelo;
- No código também se encontram os histogramas de cada atributo e outros gráficos, para uma análise mais cuidada.

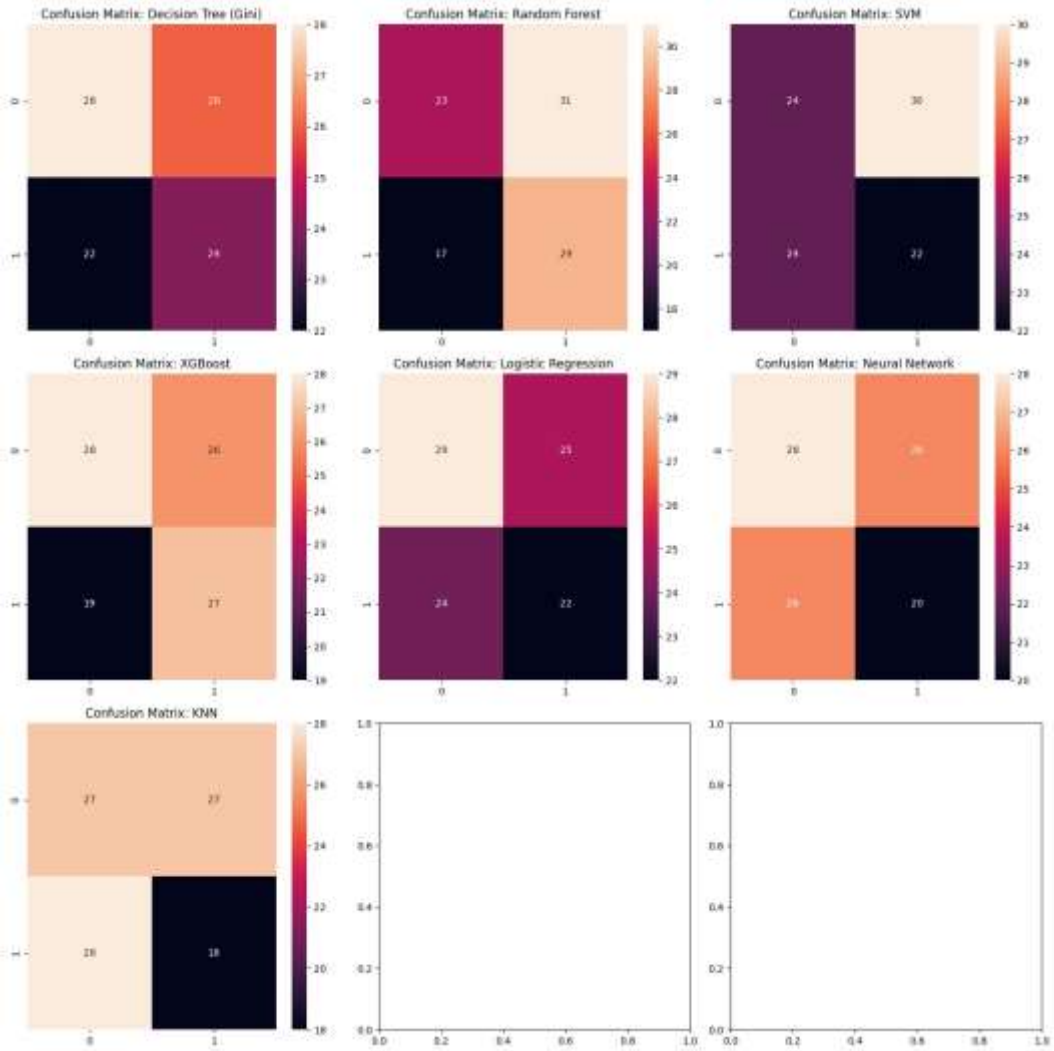
4. Data Evaluation





Ao avaliar os mapas de calor da matriz da correlação é possível reparar, pelo esquema de cores, quais os atributos que foram escolhidos, tendo restado apenas 22 atributos para o nosso modelo.

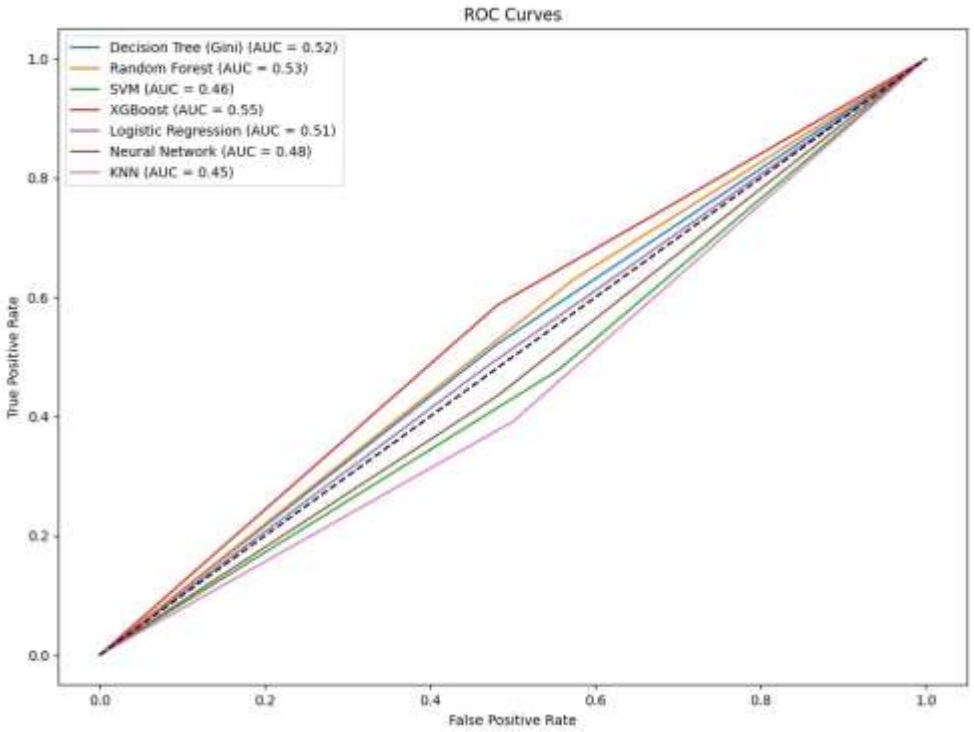
5. Interpretation of Results

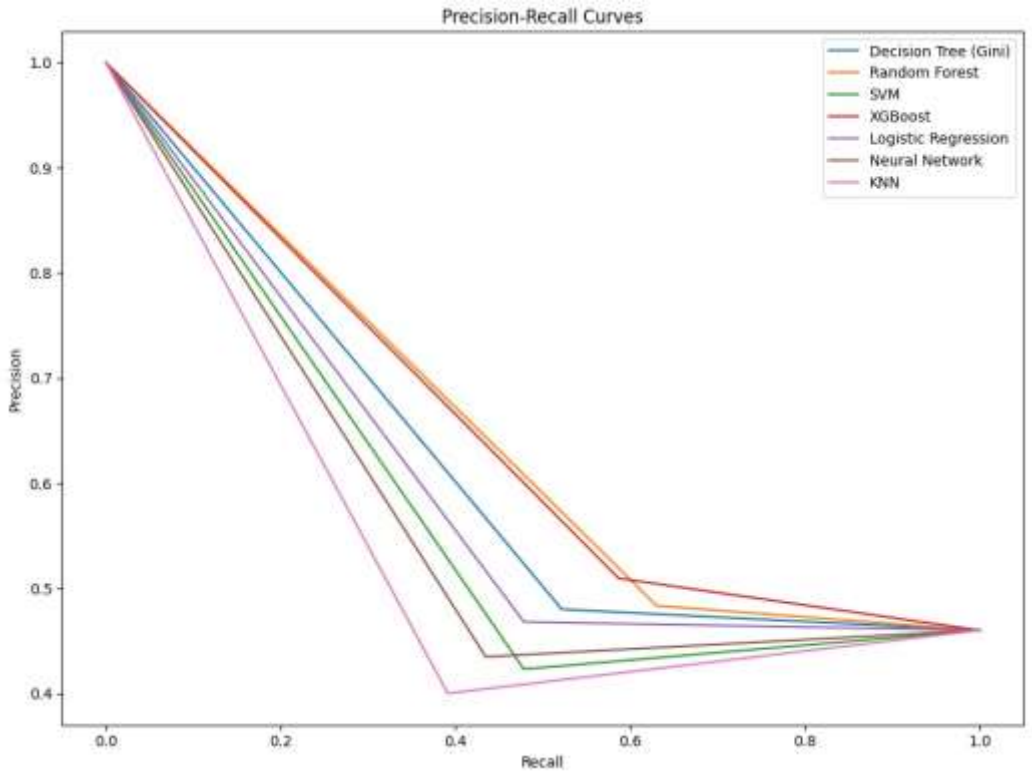


Ao analisar estas matrizes é comum reparar que, em geral, o valor dos True Positive são os mais elevados, porém os valores dos True Negative são os menores, o que é um ponto negativo na precisão do nosso modelo. Também é possível reparar que, em geral os False Positive apresentam valores inferior e os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

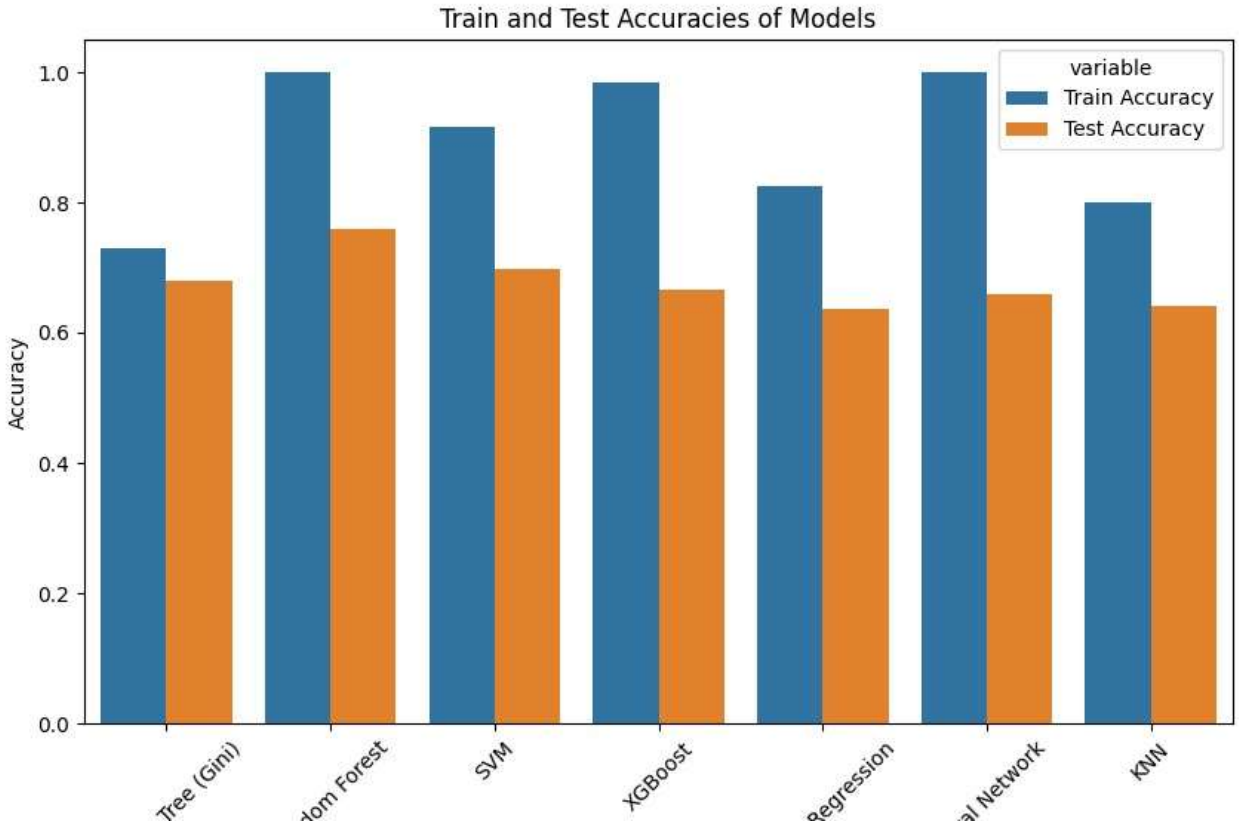
É possível concluir que a precisão dos valores positivos é superior à dos valores negativos.

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





Aqui temos, dois gráficos a comparar os diferentes parâmetros do nosso modelo de precisão, como visto no slide anterior, onde no primeiro é possível analisar qual parâmetro tem uma maior precisão sobre os seus resultados, sendo possível perceber que é o XGBoost, e o segundo onde compara a precisão com a sensibilidade, onde se destaca o XGBoost.



Ao avaliar a precisão do nosso teste, é possível reparar que a precisão dos valores de treino são superiores aos do valor de teste, porém as diferentes variáveis estão mais idênticas entre si nos valores de teste.

É também notável que a precisão é superior nos métodos XGBoost, o Random Forest e Neural Network.