

# The Hepatocellular Carcinoma Dataset

Elementos de Inteligência Artificial e Ciência de Dados

Assignment No. 2

Data exploration and enrichment for supervised classification



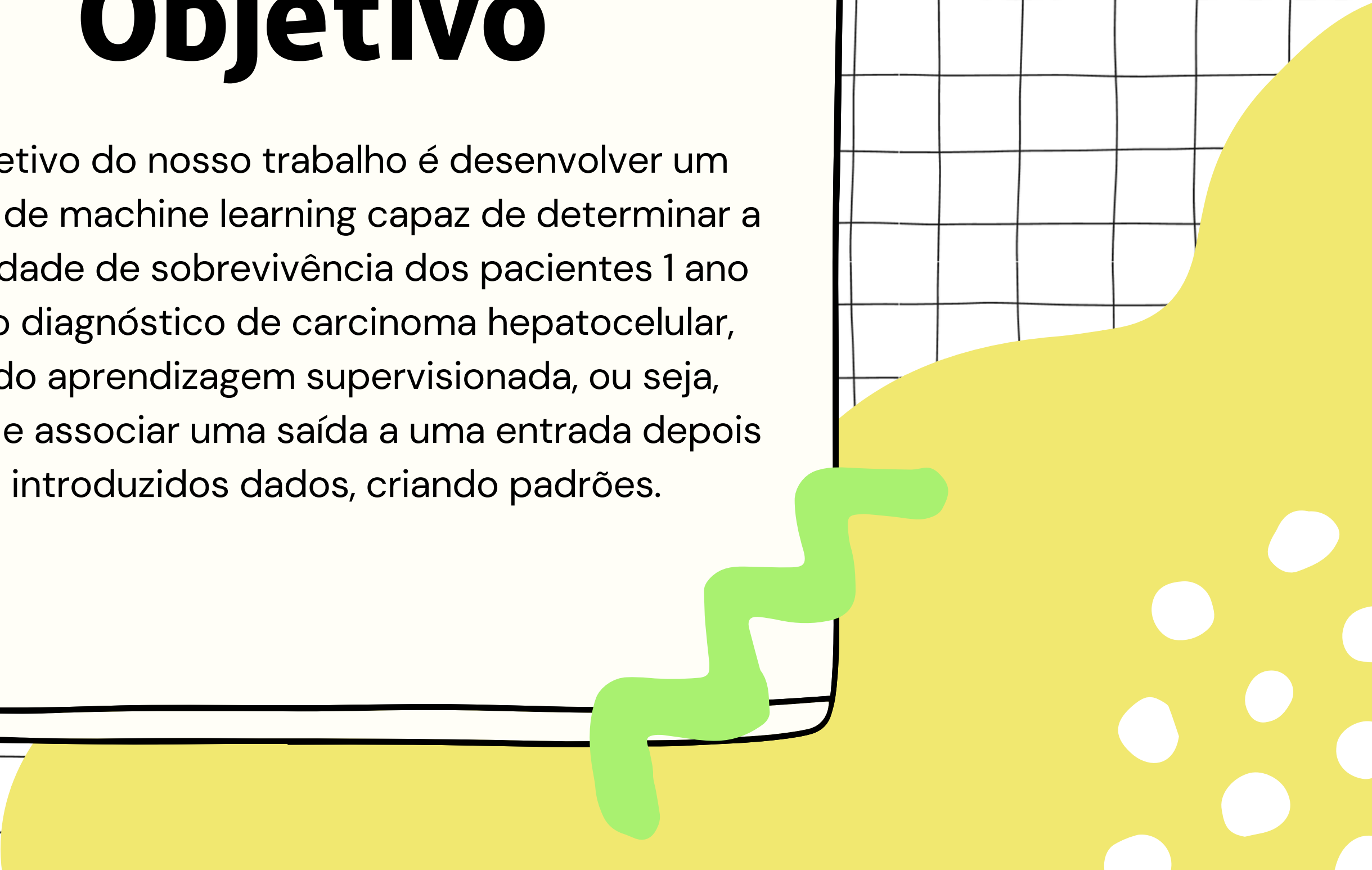
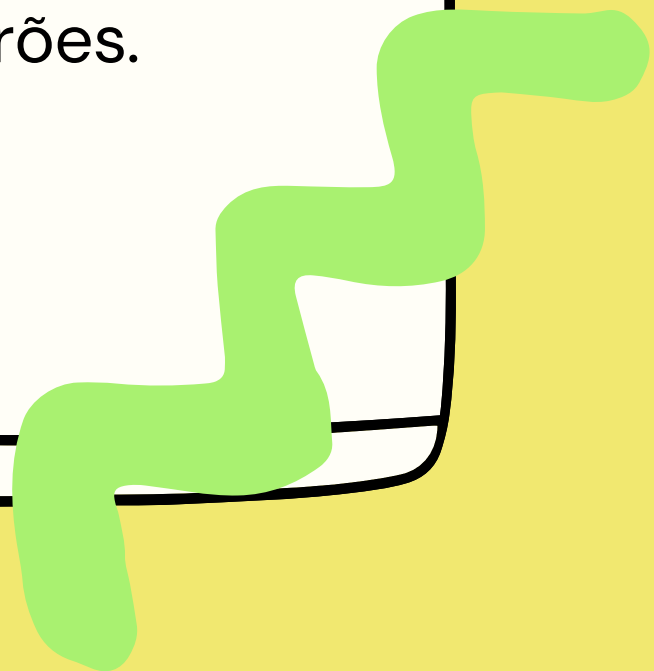
Maria Moreira, Beatriz Marques, Pedro Figueiredo

2023/2024

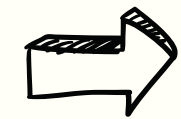
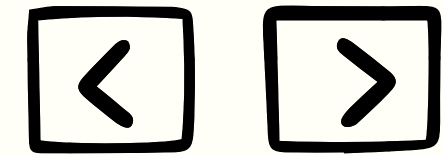


# Objetivo

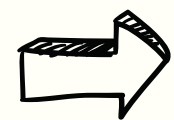
O objetivo do nosso trabalho é desenvolver um projeto de machine learning capaz de determinar a capacidade de sobrevivência dos pacientes 1 ano após o diagnóstico de carcinoma hepatocelular, usando aprendizagem supervisionada, ou seja, capaz de associar uma saída a uma entrada depois de introduzidos dados, criando padrões.



# Sobre os dados



**165 registos e 50 atributos**



**32 mulheres + 133 homens**

## **63 morreram**

idade média: 65 anos  
maioria +60 anos  
mais novo: 27 anos  
sintomas: 79%  
alcool: 76%  
cirrose: 99%  
maioria 1 ou 5 nódulos  
lesões fígado: 100%

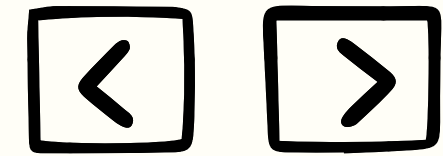
## **102 viveram**

idade média: 63 anos  
mais novo: 20 anos  
mais velho: 93 anos  
sintomas: 43%  
alcool: 73%  
cirrose: 91%  
maioria 1 ou 5 nódulos  
lesões fígado: 63%

## **Divisão do dataframe em atributos específicos**

Será interessante encontrar um padrão e analisar que combinações entre atributos podem tornar mais ou menos provável a sobrevivência do paciente. Contudo, queremos entrar por outras vertentes e também analisar diferenças entre homens e mulheres, grupos etários, obesidade vs. não obesidade

# LIMPEZA DOS DADOS

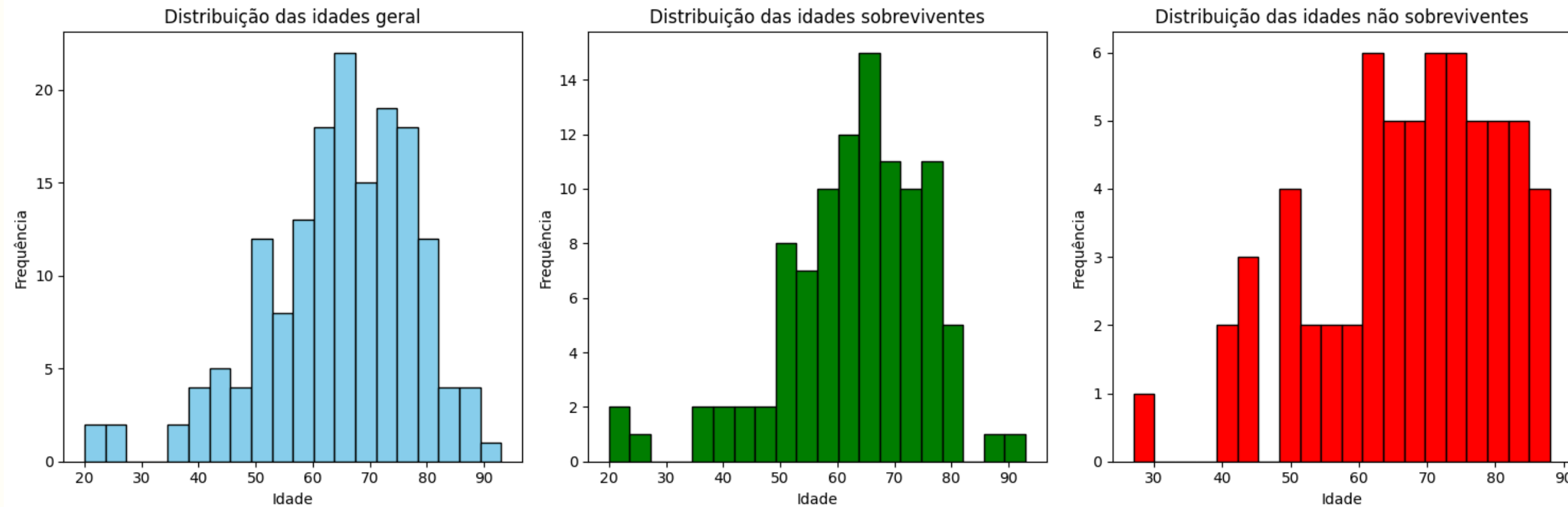


Inicialmente, reparámos na existência de bastantes atributos nulos, o que não permitia a construção do modelo e, por isso aplicamos uma limpeza aos dados para utilizar aqueles que realmente fossem importantes.

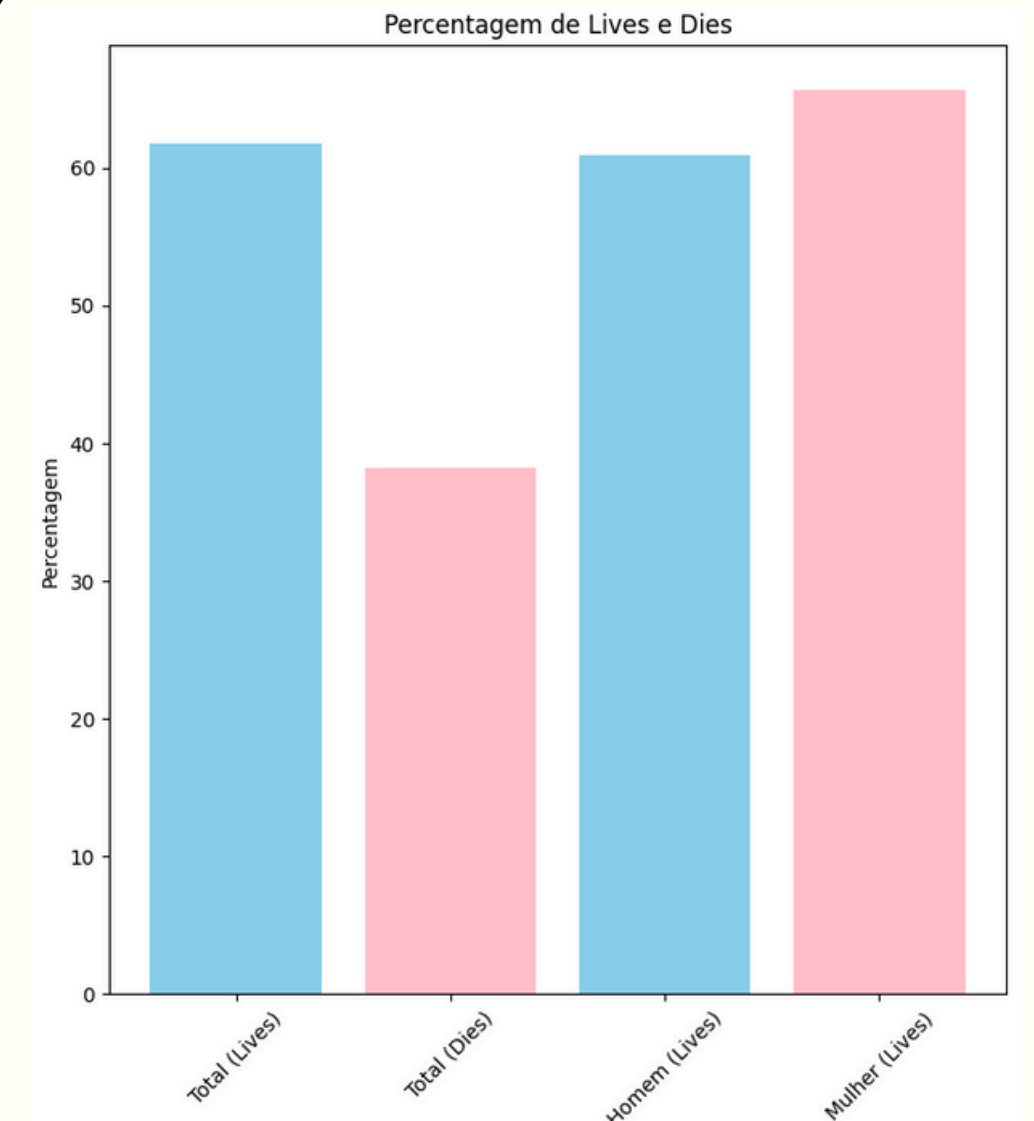
Para isso fizemos os seguintes passos:

1. Listamos os atributos numéricos e categóricos
2. Tornamos os atributos numéricos em float, pois o python estava a reconhecê-los como object
3. Filtramos aqueles que tinham mais de 20% de missing values
4. Transformamos os 'None' para 0
5. Verificamos se ainda haviam missing values, ou 'None', ou duplicados

# Gráficos

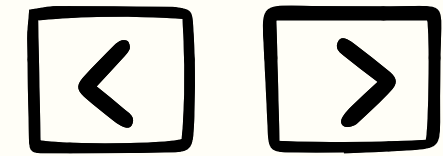


Alguns exemplos de gráficos de apoio de que tiramos partido para analisar os dados são os histogramas contudo, também utilizamos caixas de bigodes e gráficos circulares. Os gráficos dão uma perspectiva visual dos dados permitindo fazer comparações e encontrar padrões.





## Pré-Processamento

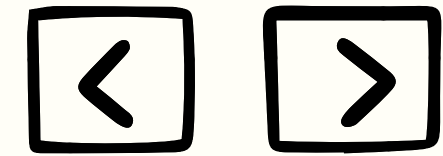


Antes de avançarmos para o modelo é necessário que preparemos os dados, de forma que o modelo seja bem aplicado e possamos assim treiná-lo de forma a ser o mais eficaz possível para, desta forma, tirarmos as melhores conclusões do dados observados.

Neste sentido, seguimos os seguintes passos:

1. Codificamos as variáveis categóricas 'Yes' ou 'No' para 0 ou 1
2. Utilizamos o KNNImputer para substituir os valores vazios pelo "vizinho mais próximo", já que o modelo não os permite
3. Eliminamos atributos sem variância
4. Removemos atributos redundantes (que estavam fortemente correlacionados com outros já utilizados)

# Árvore de decisão



Definimos os features e a classe que era o alvo do estudo ('Class'):

```
X = df[atributos]
```

```
Y = df['Class']
```

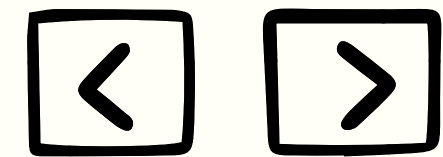
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, stratify=Y)
```

```
clf = DecisionTreeClassifier(max_leaf_nodes=5, criterion='gini', random_state=0)
```

```
clf.fit(X_train, Y_train)
```

Utilizamos a classe **DecisionTreeClassifier** do scikit-learn para construir o modelo de árvore de decisão e especificamos o critério de divisão ('gini') e o número máximo de nós (max\_leaf\_nodes=5) para controlar o crescimento da árvore. Realizamos previsões no conjunto de treino e teste e avaliamos o desempenho do modelo através da matriz de confusão.

## Matrizes de confusão



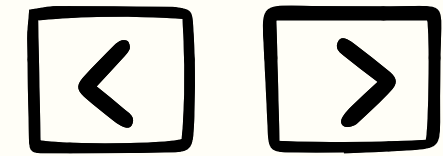
As matrizes de confusão são tabelas que permitem visualizar o desempenho de um modelo de classificação. Foram então utilizadas para avaliar o desempenho do modelo de classificação, tanto nos dados de treino quanto nos dados de teste. Avaliamos assim 'precisian', 'negative\_precision', 'sensitivity' e 'accuracy' que tem os seguintes resultados:

	Vida prevista	Morte prevista
Vida de facto	65	6
Morte de facto	18	26

precisian: 0.7831325301204819  
negative\_precision: 0.8125  
sensitivity: 0.9154929577464789  
specificity: 0.5909090909090909  
accuracy: 0.7913043478260869



# Conclusões



- Qual dos modelos se revelou mais eficiente? Em que situações um modelo é mais eficiente do que outro?
- Que fatores influenciam o desempenho dos modelos?
- Qual a precisão dos modelos?
- No que nos ajudaram as matrizes de confusão?