C) Banking Customer Churn Prediction Dataset Supervision Learning

Entrega Final Projeto 2 – Inteligência Artificial

Grupo 54

André dos Santos Faria Relva - up202108695

Pedro Guilherme Pinto Magalhães - up202108756

Rafael Azevedo Alves - up202004476

Faculdade de Engenharia da Universidade do Porto

27 de maio de 2024

Definição do Problema de Machine Learning

Neste projeto é nos fornecido um dataset que contém informações sobre clientes de um banco e os seus estados de "churn" (isto é, se saíram ou não do banco).

O objetivo deste projeto é desenvolver um modelo de Machine Learning baseado em algoritmos de classificação de apredizagem supervisionada capaz de determinar, com base em alguns dados, quais os clientes susceptíveis de abandonar o banco.

Para chegar ao modelo ideal, temos de experimentar, pelo menos, três algoritmos de classificação diferentes.

Informações sobre Dados do Dataset

Os dados presentes no dataset fornecido são, os seguintes:

- **RowNumber -** número sequencial atribuído a cada linha do conjunto de dados (1 a 10000)
- **CustomerId -** identificador único para cada cliente
- **Surname -** apelido do cliente
- **CreditScore -** crédito do cliente
- **Geography -** localização geográfica do cliente (país ou região)
- **Gender -** género do cliente
- **Age -** idade do cliente
- **Tenure -** número de anos que o cliente está no banco
- **Balance -** saldo da conta do cliente
- **NumOfProducts -** número de produtos bancários que o cliente possui
- **HasCrCard -** indica se o cliente tem um cartão de crédito (binário: sim(1)/não(0))
- **IsActiveMember -** indica se o cliente é um membro ativo (binário: sim(1)/não(0))
- **EstimatedSalary -** salário estimado do cliente
- **Exited -** indica se o cliente saiu do banco (binário: sim(1)/não(0))

```
RowNumber: int64
CustomerId: int64
Surname: object
CreditScore: int64
Geography: object
Gender: object
Age: int64
Tenure: int64
Balance: float64
NumOfProducts: int64
HasCrCard: int64
IsActiveMember: int64
EstimatedSalary: float64
Exited: int64
```

Descrição das Ferramentas e Algoritmos Utilizados

Utilizamos **Python** como linguagem de programação, programando num ambiente **Jupyter Notebook**.

Para os Algoritmos de Machine Learning recorremos às bibliotecas **Scikit-Learn, Tensorflow, CatBoost**, bem como ao **Pandas** para ler e tratar os dados e ao **Seaborn** e ao **Matplotlib** para a visualização

Os **Algoritmos de Classificação** que usamos são, os seguintes:

- Decision Trees - fácil interpretação para classificação.
- Logistic Regression - simples e eficaz para classificação binária.
- Random Forest – funciona bem com datasets desbalanceados.
- K-NN (K-Nearest Neighbors) - simples mas exige poder computacional em datasets grandes.
- SVM (Support Vector Machine) - robusto que exige grande poder computacional.
- Naive Bayes - rápido e eficiente.
- CatBoost - funciona bem com datasets desbalanceados.
- Neural Networks – em geral, são extremamente poderosas.

Implementação

- **Linguagem de Programação:** Python
- **Ambiente de Desenvolvimento:** VSCode usando JupyterNotebooks e GitHub
- **Estruturas de Dados:**

  O dataset é representado como um "pandas.DataFrame".

```
import pandas as pd
dataset = pd.read_csv('data/Churn_Modelling.csv')
```

Pré-Processamento dos Dados

O nosso dataset estava pronto a ser utilizado, pois:

- Não havia valores em falta (*missing values*) no nosso dataset;
- Havia alguns *outliers*, mas estes não representavam quaisquer erros de medição, pois, conforme se pode averiguar na Figura 1, os valores das colunas apresentadas nos boxplots não apresentam valores negativos, e, conforme se pode verificar na Figura 2, os valores das colunas que deveriam representar valores binários só apresentam os valores 1 e 0 (valores binários), como era de esperar.

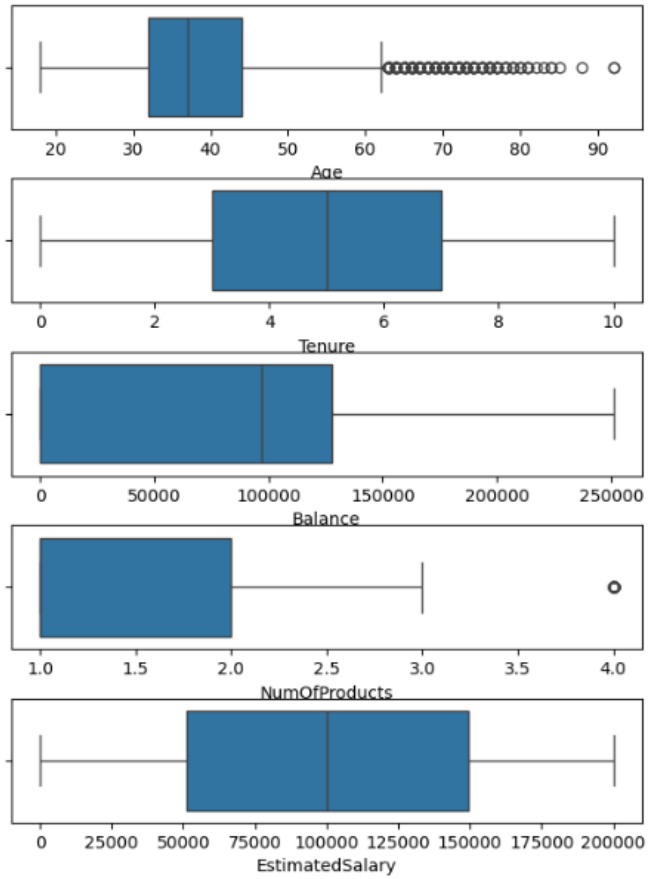


Figura 1 - Boxplots

```
HasCrCard                IsActiveMember           Exited
1    7055                1    5151                0    7963
0    2945                0    4849                1    2037
Name: count, dtype: int64  Name: count, dtype: int64  Name: count, dtype: int64
```

Figura 2 – Contagem de Valores Binários

Análise dos Dados

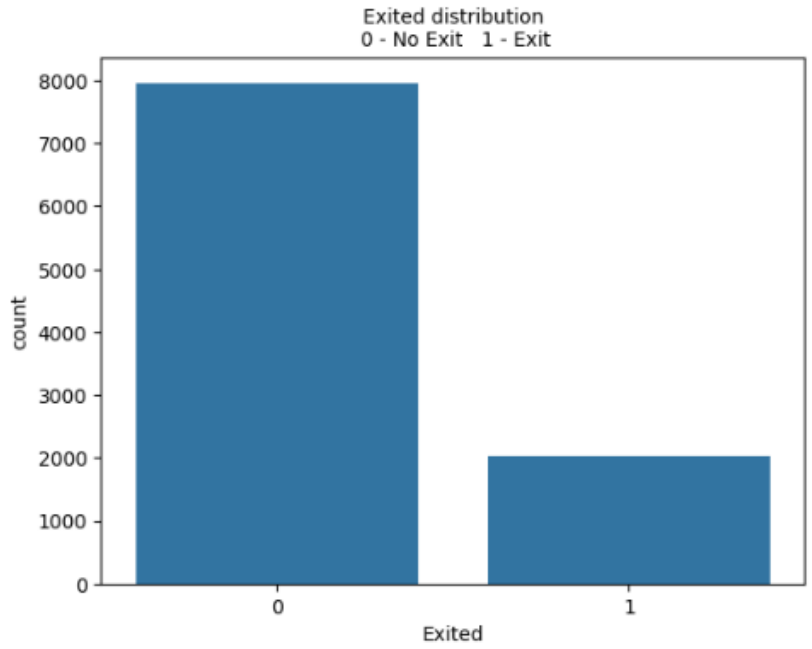


Figura 1 - Distribuição de Valores da variável "Exited"

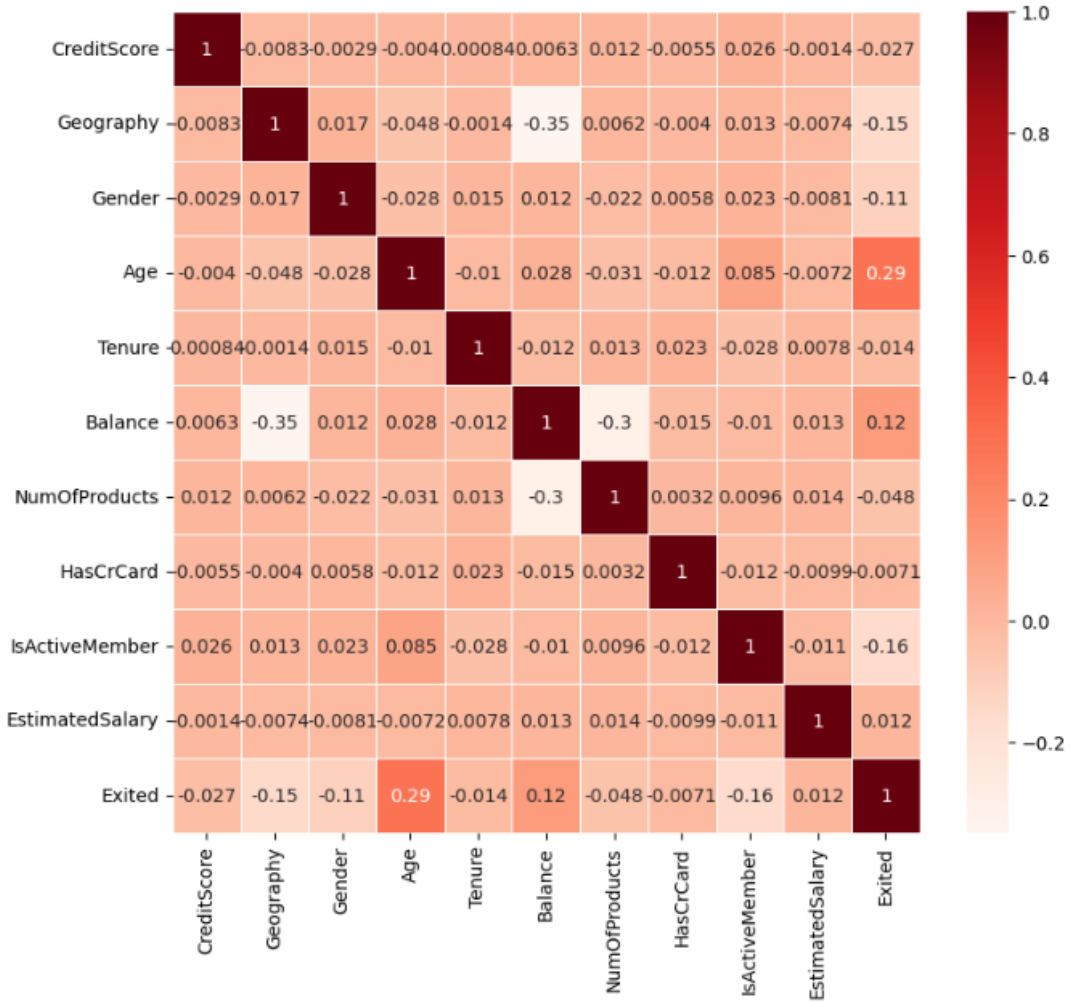
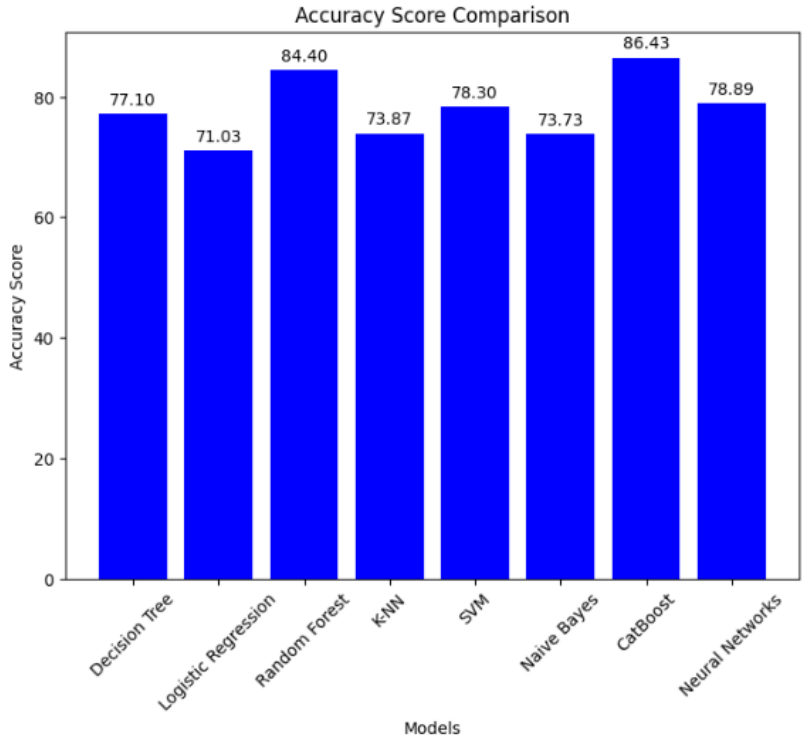


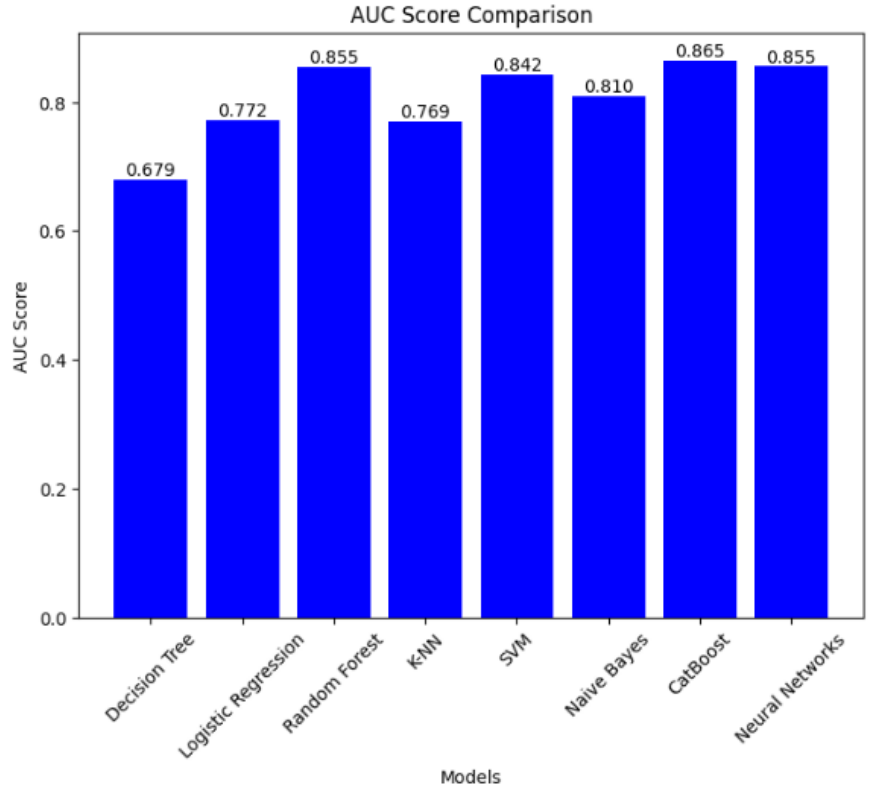
Figura 2 – Matriz de Correlação

Avaliação e Comparação dos Algoritmos Desenvolvidos

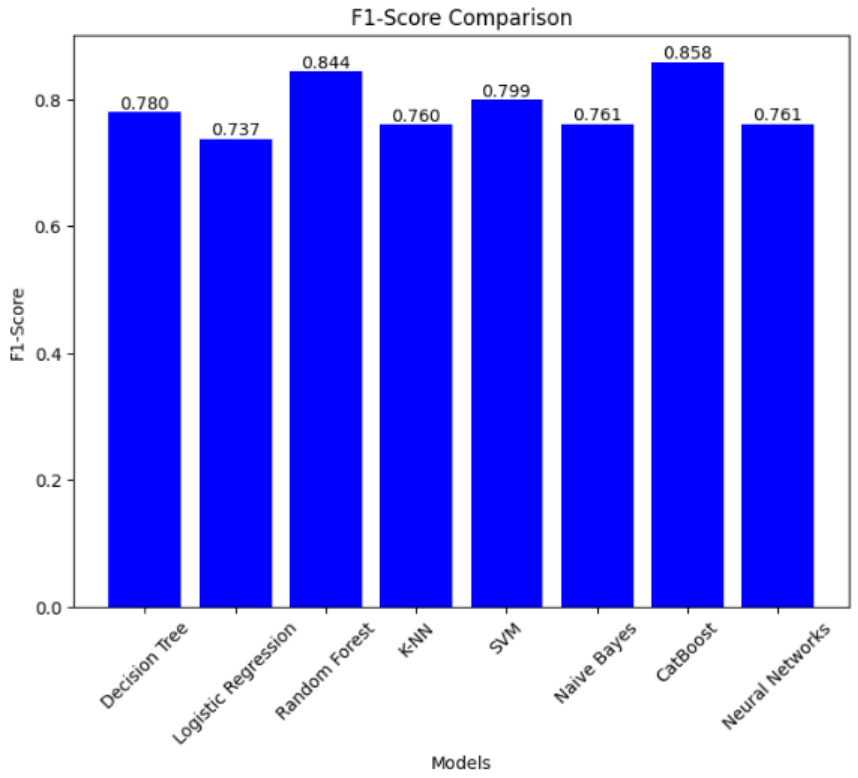
Para os algoritmos dividimos o dataset em um set de treino (70%) e um set de teste (30%), e aplicamos as técnicas de ***Scalling*** e ***SMOTE*** aos mesmos.

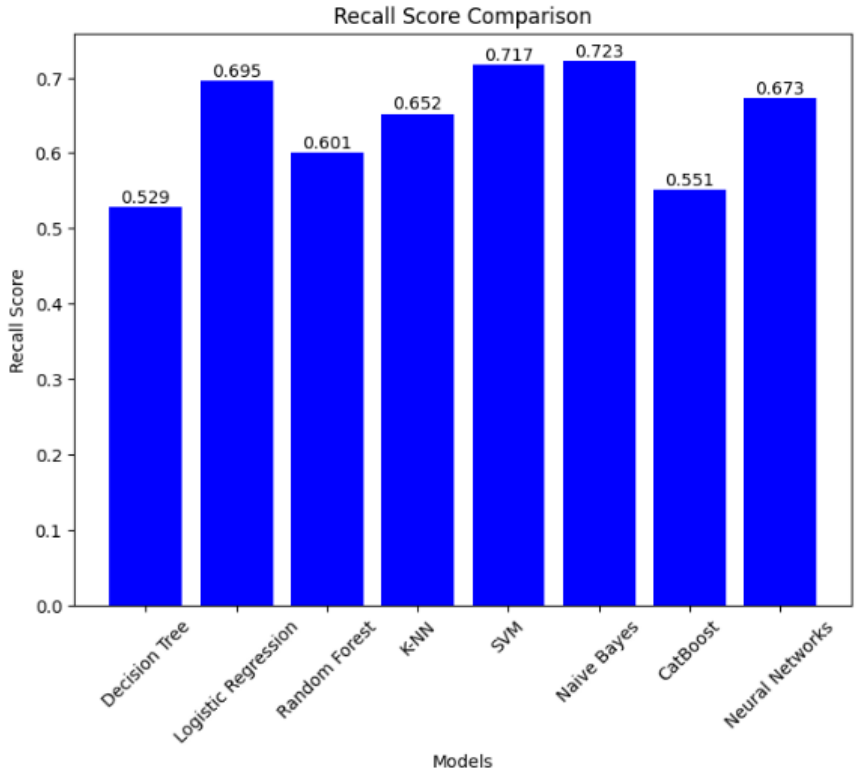
Os resultados das avaliações feitas aos algoritmos desenvolvidos foram, os seguintes:

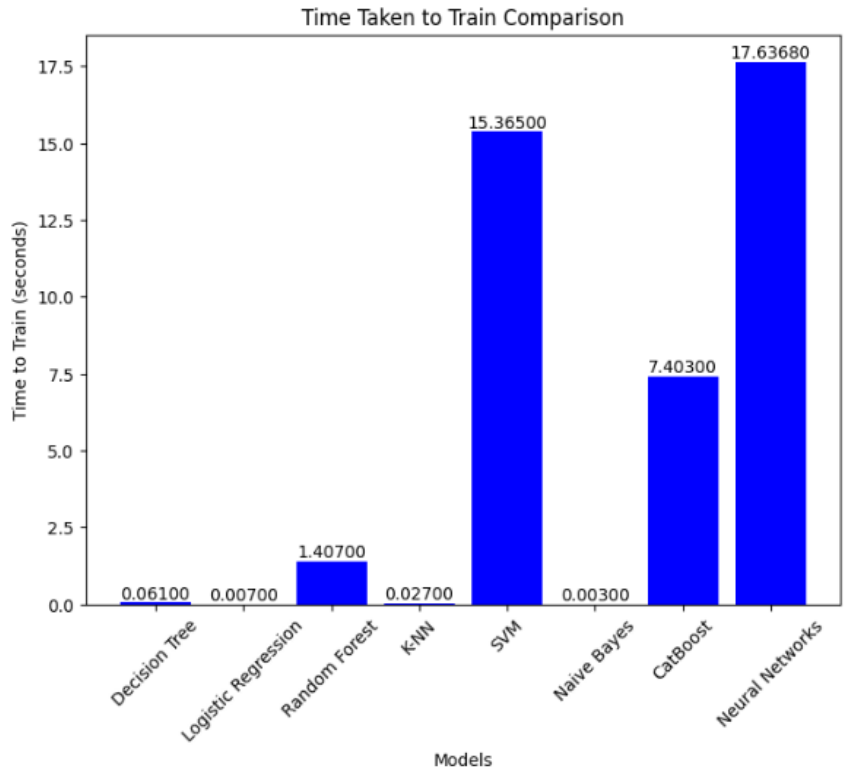




Avaliação e Comparação dos Algoritmos Desenvolvidos







Conclusão

Após a observação dos gráficos é possível concluir que:

- O melhor algoritmo foi o **CatBoost** pois é o que apresenta melhores resultados em termos de AUC e F1-Score.
- Os piores algoritmos foram **Decision Tree** e **Logistic Regression**. Este resultado pode dever-se ao facto do nosso dataset ser desequilibrado.
- Esperava-se que **Neural Networks** tive-se melhores resultados pois, dos algoritmos escolhidos, é considerado o mais "poderoso" de todos.

Em suma, neste trabalho conseguimos utilizar todos os **8 algoritmos** que pretendiamos usar. Com este trabalho foi possível aprofundar o nosso conhecimento na área de Supervision Learning usando algoritmos de classificação.

Referências

- **Websites utilizados para pesquisa:**
  - Scikit-learn (https://scikit-learn.org/stable/)
  - Seaborn (https://seaborn.pydata.org/)
  - Matplotlib (https://matplotlib.org/)
  - CatBoost (https://catboost.ai/en/docs/)
  - Tensorflow (https://www.tensorflow.org/)