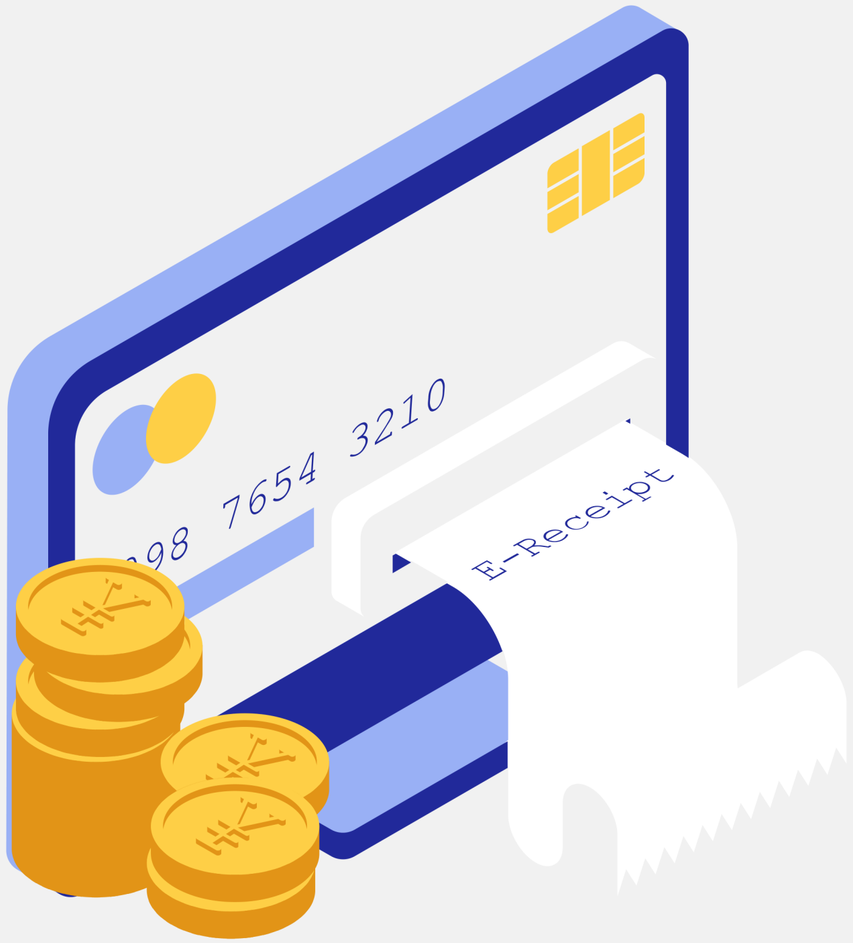


Banking Customer Churn Prediction

Artificial Intelligence Assignment 2

Guilherme Santos, up202105090
Lara Bastos, up202108740
Miguel Barros, up202108678

Specification of the work to be performed

Customer churn refers to **the loss of customers.**
The goal of the project is to implement **Machine Learning Models** using **Supervised Learning Algorithms** to analyse factors influencing customer churn in banking institutions and predict customers at risk of churning based on a given dataset.
The dataset contains a total of **14 independent attributes for each client**, including:

- Surname
- Credit Score
- Geography
- Gender
- Age
- Tenure
- Balance
- Num of Products
- Has Credit Card
- Is an Active Member
- Estimated Salary

The **target attribute** is **Exited**, representing if the customer has exited the bank or not (1 or 0).

Tools used

For the development of the project it was used **Python** in **Jupyter Notebook.**

Additionally the following libraries were used:

- **Pandas** - read and manipulate data.
- **Matplotlib** - plot relevant graphs.
- **Seaborn** - plot relevant graphs.
- **Scikit Learn** - implement classifiers and use metrics.
- **Imblearn** - SMOTE and under sampling.

As the data set is imbalanced we intend to take advantage of **Synthetic Minority Oversampling Technique (SMOTE) and UnderSampling.**

Additionally we tried to remove the feature with less correlation with target to see if it improved any classifier.

For the classification of the data, we plan using the following Supervised Learning Algorithms:

01 **Decision Trees**

02 **K-Nearest Neighbours**

03 **Support Vector Machines**

04 **Bayes Naive**

05 **Neural Network**

For each of these, we used **Grid Search** to optimize the parameterization of classifiers.

Implementation Work Carried Out

Data PreProcessing:

- Removal of unnecessary columns.
- Removal of missing or duplicate values.
- Removal of redundant attributes.
- Encoding of categorical attributes.
- Replacing of outliers

Division in training and testing set

- Divided in a training/testing with a proportion of 30/70.
- Applied SMOTE and Undersampling to tackle dataset imbalance.

Evaluation of Classifiers

- For each classifier:
  - metrics: accuracy, recall, precision and F1.
  - confusion matrix.
  - ROC curve.

Data Analysis:

- Analysis of dataset balance.
- Analyis of correlated attributes.

Machine Learning Models:

- Parameterisation of Models with a Grid Search.
- Decision Trees.
- K-Nearest Neighbour.
- Support Vector Machine.
- Bayes Naive
- Neural Networks

Comparison of Classifiers

- Comparison of Metrics and Time.
- Evaluation of Feature Importance.
- Comparison with Baselines.
- Conclusions.

Data Pre Processing

During this phase we observed:

- There were no missing values.
- There were no duplicated values.
- There were two irrelevant columns for the classification: **RowNumber** and **CustomerId**.
- There was the value "H?" multiple times in the Surname, but we assumed it was a mistaked made in the sam surname multiple times so we left it.
- There were three categorical values left that had to be encoded: **Geography**, **Gender** and **Surname**.
- There were many outliers, that add to be analysis to decide if worth keeping.

Outlier Analysis

For the analysis of outliers we plotted a boxplot per attribute:

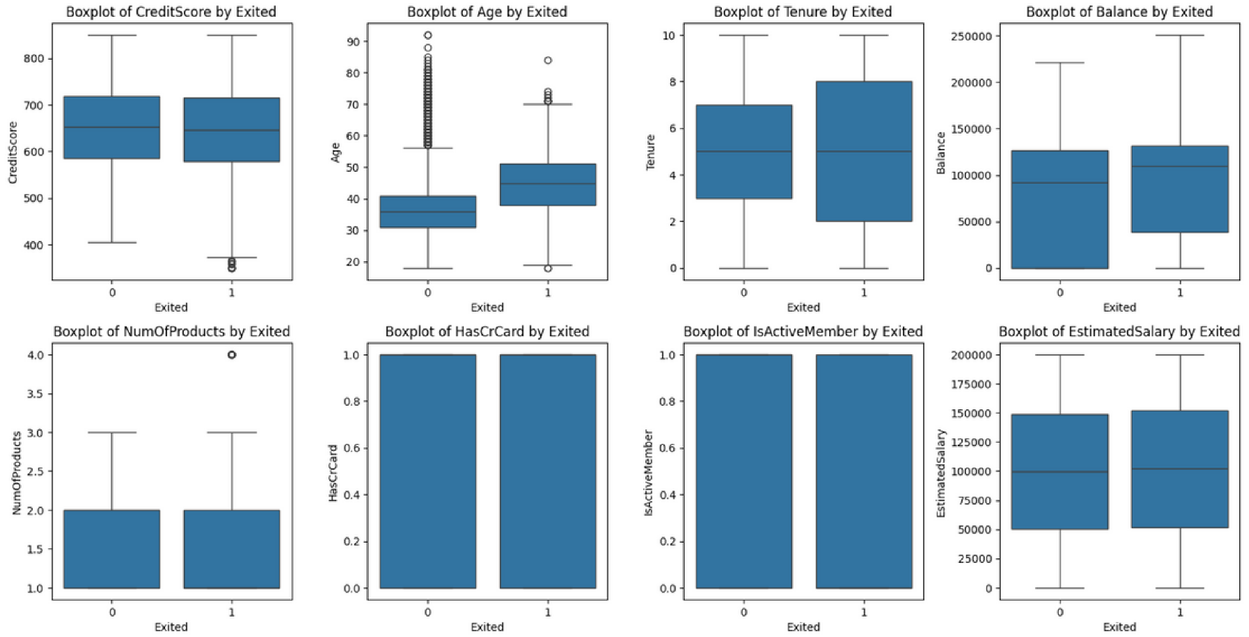


Figure: Boxplot

Multiple outliers were identified in the Age and Credit Score category however, they do not appear to be an error, as they are positioned close to the quartil.

The higher outliers in NumOfProducts and Age, were replaced by the arithmetic average

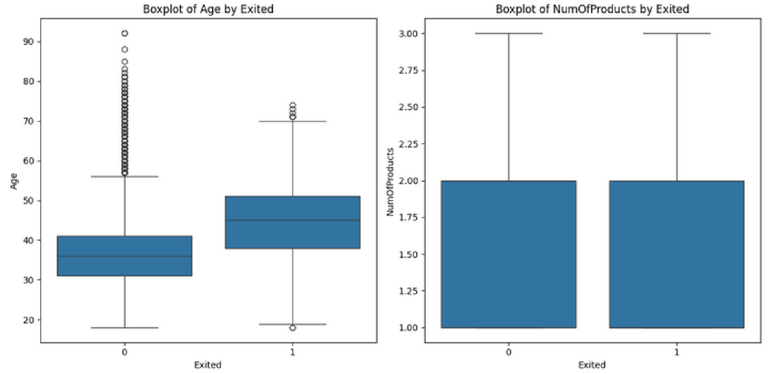


Figure: Boxplot of altered classes

Data Analysis

Evaluating balance

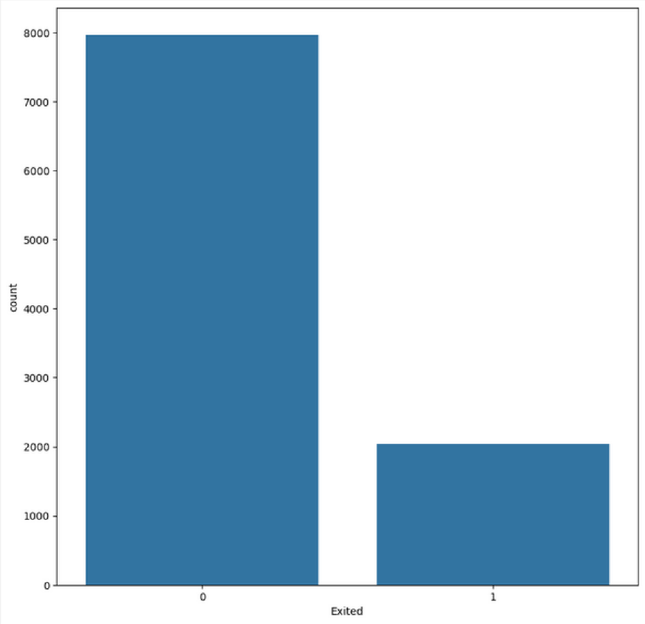


Figure: Class Imbalance

We concluded the datset is **imbalanced**. With this is mind, we will be experimenting with both **SMOTE** and **UnderSampling**.

ScatterPlot

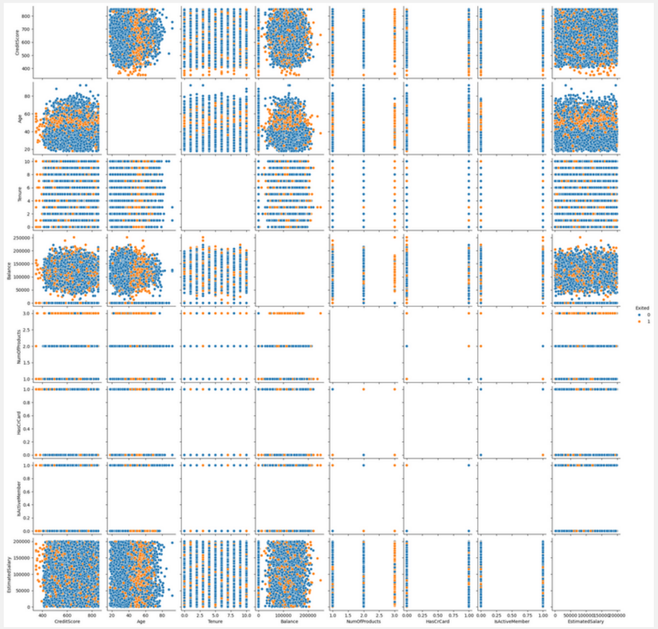


Figure: ScatterPlot

We observed that a significant portion of clients who have exited the bank fall within the age range of 40 to 60. Beyond these observations, no other significant conclusions can be drawn.

Correlation Matrix

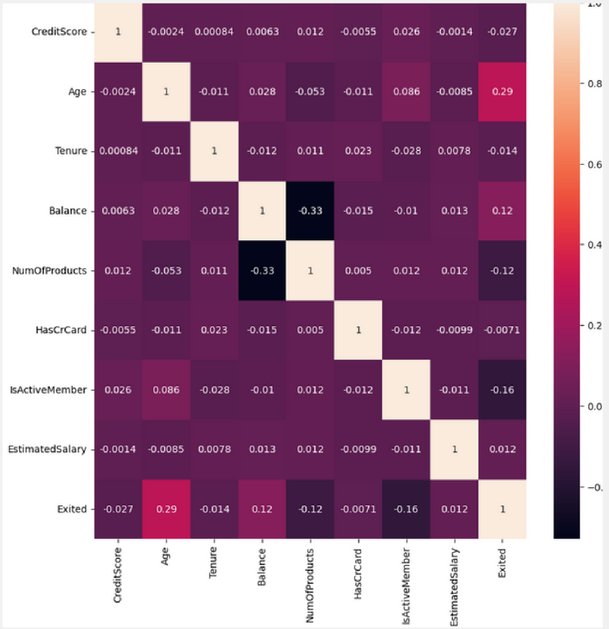


Figure: Correlation Matrix

Only the "Age" appears to influence the "Exited" attribute.

No other particular attributes exhibit a strong correlation with each other.

For each classifier, we will be attempting to remove the attributes with < -0.05 of correlation with the target.

Machine Learning Models

As the Data Set was imbalaced we decided to experiment both **SMOTE** and **Undersampling** for each, as well as experimenting with and without the columns that don't have any correlation with the target value.
For each of the versions of the algorithms, we implemented a **Grid Search** to decide the optimal parameters for training and evaluation.
The implemented classifiers were the following, with the stated parameters:

Decision Tree Classifier

Best Accuracy: 84% for no Sampling Technique
Chosen parameters: {'criterion': 'entropy', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 2}

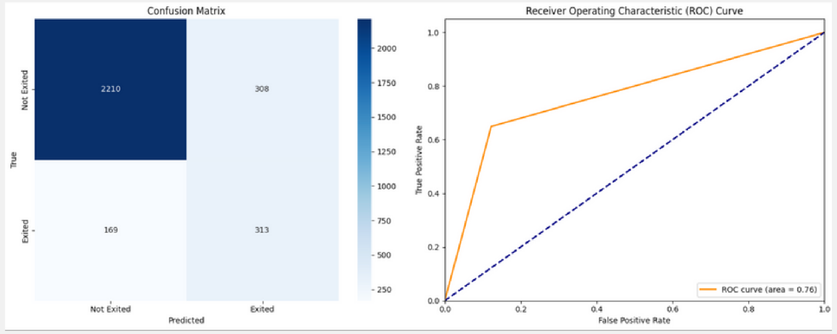


Figure: DT Confusion Matrix and ROC Curve

Support Vector Machines

Best Accuracy: 79% for no Sampling Technique
Chosen parameters: {'C': 1, 'dual': False, 'penalty': 'l1'}

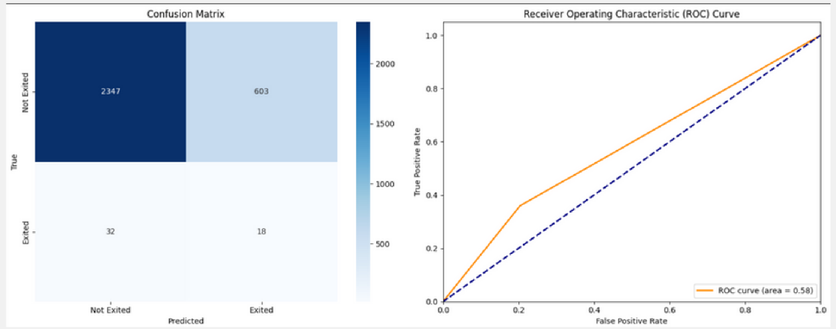


Figure: SVM Confusion Matrix and ROC Curve

K-Nearest Neighbour

Best Accuracy: 77% for no Sampling Technique
Chosen parameters: {'algorithm': 'auto', 'leaf_size': 10, 'n_neighbors': 7, 'p': 2, 'weights': 'uniform'}

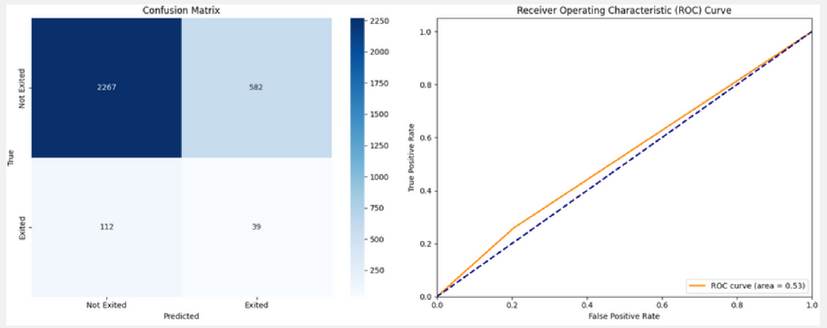
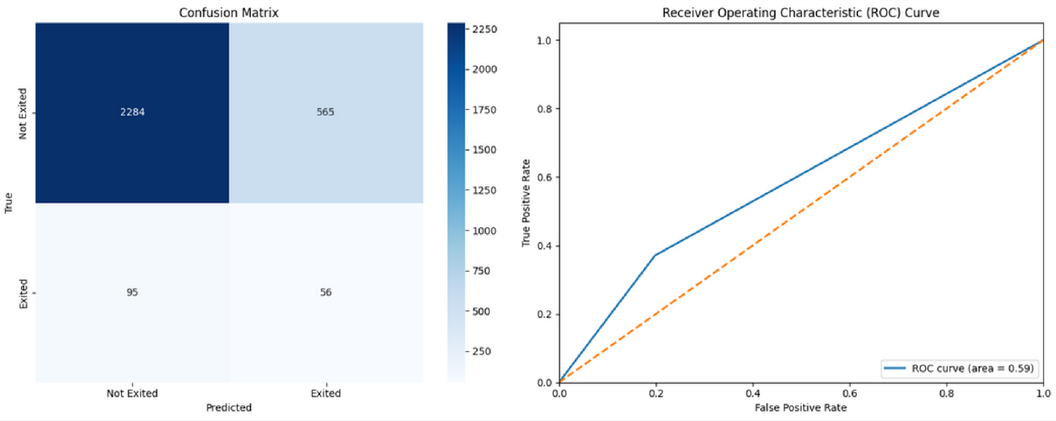


Figure: KNN Confusion Matrix and ROC Curve

NOTE: We ended up using Linear SVM, as the SVC model from skilearn was taking over 40 minutes to run.

Naive Bayes

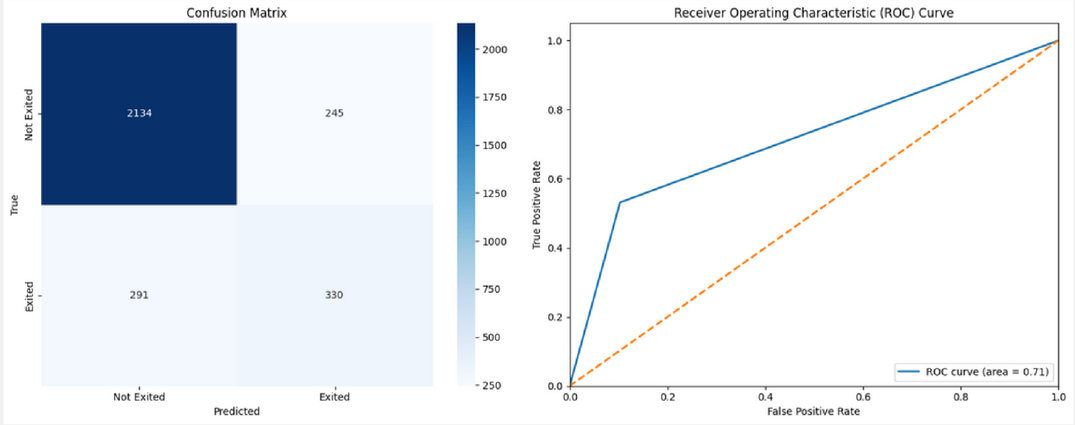
Best Accuracy: 78% for no Sampling Technique



Neural Network

Best Accuracy: 78% for no Sampling Technique and Adam Solver

Chosen Parameters: {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (100, 50), 'learning_rate': 'constant'}



Evaluation of Classifiers

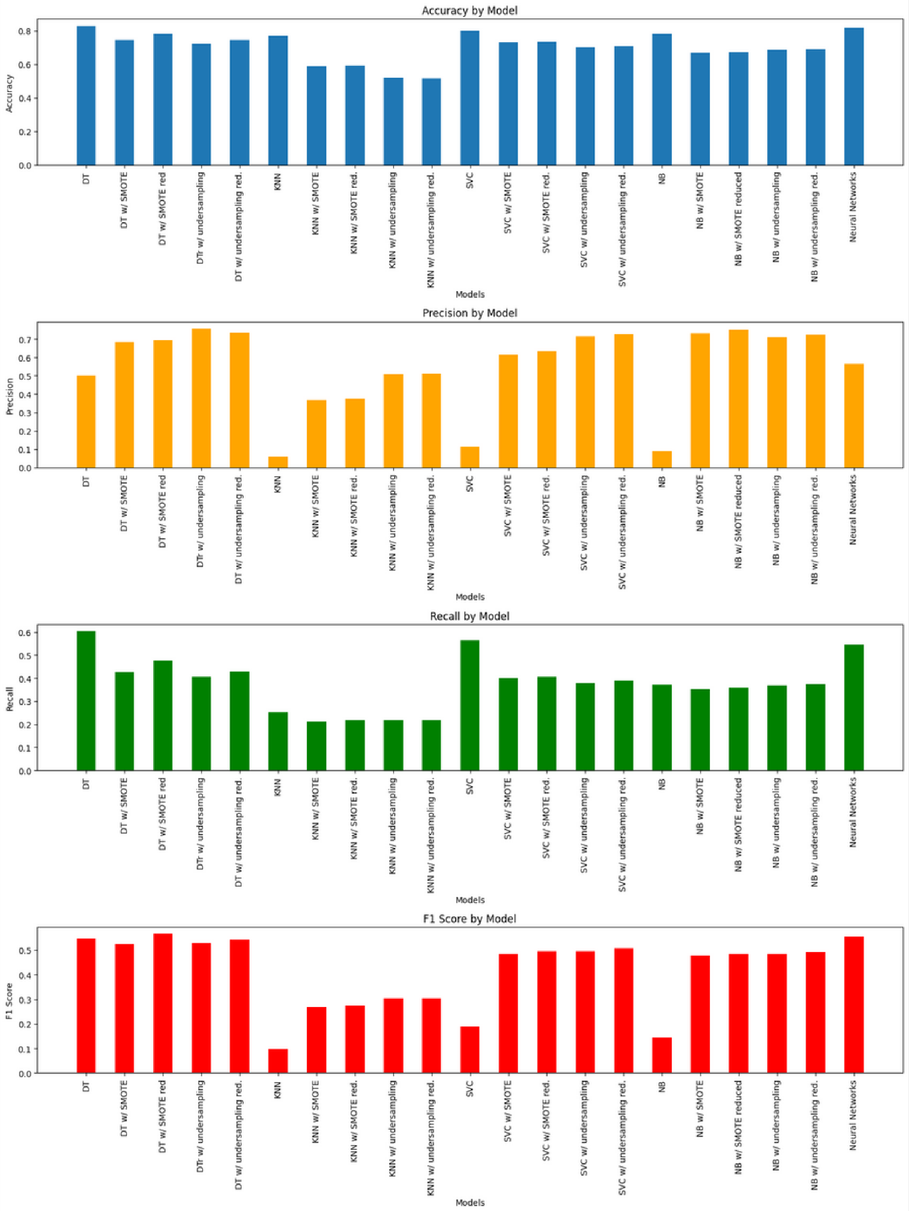
Lastly, we will be comparing the different metrics yielded by each of the Classifiers to understand which one is the most appropriate and efficient for the problem in question.

We used:

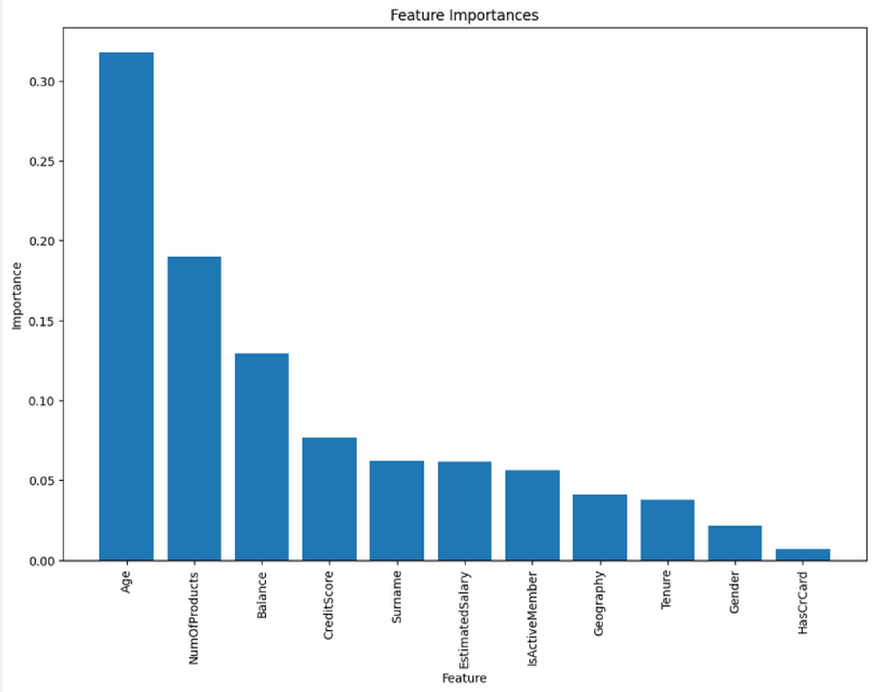
- Comparison of Metrics
- Evaluating Feature Importance
- Comparison with Baselines

	Model	Accuracy	Precision	Recall	F1 Score
0	DT	0.828667	0.502415	0.603482	0.548330
1	DT w/ SMOTE	0.744333	0.684380	0.426707	0.525665
2	DT w/ SMOTE red	0.780000	0.692432	0.478309	0.565789
3	DTr w/ undersampling	0.721000	0.756844	0.406574	0.528981
4	DT w/ undersampling red.	0.743667	0.732689	0.430057	0.541989
5	KNN	0.768333	0.061192	0.253333	0.098573
6	KNN w/ SMOTE	0.586667	0.368760	0.212628	0.269729
7	KNN w/ SMOTE red.	0.590333	0.375201	0.216946	0.274926
8	KNN w/ undersampling	0.517667	0.507246	0.216346	0.303322
9	KNN w/ undersampling red.	0.516333	0.510467	0.216530	0.304077
10	SVC	0.798333	0.114332	0.563492	0.190094
11	SVC w/ SMOTE	0.728667	0.615137	0.399164	0.484157
12	SVC w/ SMOTE red.	0.732667	0.634461	0.406605	0.495597
13	SVC w/ undersampling	0.698667	0.713366	0.378956	0.494972
14	SVC w/ undersampling red.	0.707333	0.726248	0.389129	0.506742
15	NB	0.780000	0.090177	0.370861	0.145078
16	NB w/ SMOTE	0.668000	0.731079	0.353858	0.476891
17	NB w/ SMOTE reduced	0.670667	0.750403	0.358737	0.485417
18	NB w/ undersampling	0.687333	0.711755	0.368027	0.485181
19	NB w/ undersampling red.	0.690333	0.724638	0.372517	0.492072
20	Neural Networks	0.818333	0.563123	0.545894	0.554374

Comparison of Metrics



Evaluating Feature Importance



Comparison with Baselines

The solution found with the best performance achieved an **87% accuracy** with Gradient Boosting Classifier, compared to our best solution, which achieved an 84% accuracy with a Decision Tree Classifier. The solution also consistently **obtained slightly higher results across most of the same algorithms** that we implemented.

We attribute this discrepancy primarily to the **scaling of the data in the data preprocessing phase**, which we did not perform.

Conclusion & Recommendations

- The Decision Tree Classifier without any sampling techniques performed the best across all metrics except precision, where the reduced version outperformed it.
- Neural Networks had a similar performance but the training and testing time was significantly bigger.
- Applying sampling techniques generally improved all metrics except accuracy. However, reducing the dataset did not lead to significant improvements.
- For this particular problem, recall and precision are the most critical metrics. We recommend using models that optimize these two metrics.
- Age is the most significant attribute influencing the risk of churning. Future models should prioritize age as a key factor when predicting churn.
- Performing data scaling before implementing classifiers can potentially improve their performance. This step is recommended for future models.
- The final results were mediocre at best. As the dataset's attributes had a low correlation with the target value, models had a hard time accurately predicting the risk of churning.

References

- Lecture slides
- Seaborn Documentation
- Linear SVC Documentation
- Geeks for Geeks - Decision Trees
- Neural Networks
- Kaggle Resolutions:
  - https://www.kaggle.com/code/chandansingh98/customer-churn-prediction-87-accuracy/notebook
  - https://www.kaggle.com/code/eduardoded/choosing-the-best-classifier-model