Predicting Customer Churn - Report

Students : Jiana Bdarneh, Ayal Swaid, Maxim Katz, Yuval Levi.

Section 1 - Background & Business model

Customer retention is a critical aspect of business strategy for financial institutions, customer churn, which refers to the loss of clients or customers, poses a significant challenge, as acquiring new customers is often more costly than retaining existing ones.

The bank's profitability depends significantly on retaining existing customers, as loyal customers are more likely to utilize additional services and products. Using machine learning techniques, the bank aims to predict which customers are at risk of churning. This predictive capability allows the bank to proactively engage with at-risk customers through personalized offers, improved service quality, and targeted marketing campaigns.

In this project, we use a dataset containing information about bank customers to develop a binary classification model. The goal is to accurately predict whether a customer will churn based on various features such as account balance. This model serves as a valuable tool for the bank's retention strategies, enabling data-driven decision-making and ultimately enhancing customer loyalty and profitability.

Section 2 - Dataset Overview

The dataset used for this project can be found here kaggle.The data includes various features about the bank's customers.
Below is a general description of the dataset:

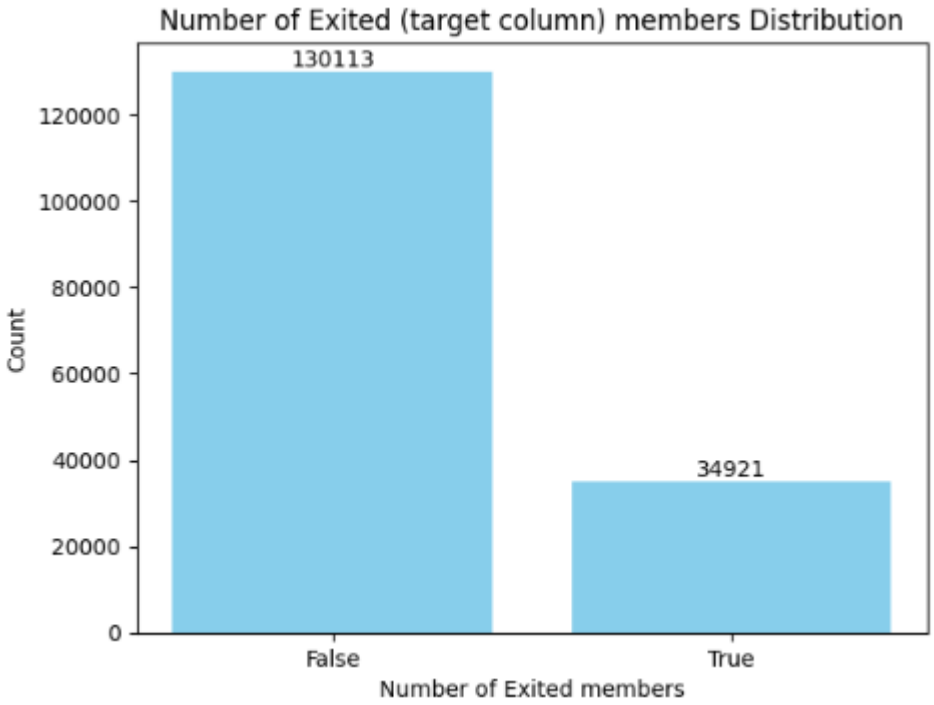
The dataset was generated from a deep learning model trained on the Bank Customer Churn Prediction dataset. Feature distributions are close to, but not exactly the same, as the original.

The dataset contains 14 columns, each representing a different attribute of the customer. The features include:

1. **ID**:Unique identifier for each row (Numeric).
2. **CustomerId**: Unique identifier for each customer (Numeric).
3. **Surname**: Customer's last name (Categorical).
4. **CreditScore**: Customer's credit score (Numeric).
5. **Geography**: Customer's location (Categorical).
6. **Gender**: Customer's gender (Categorical).

7. **Age**: Customer's age (Numeric).
8. **Tenure**: Number of years the customer has been with the bank (Categorical).
9. **Balance**: Customer's account balance (Numeric).
10. **NumOfProducts**: Number of products the customer has with the bank (Categorical).
11. **HasCrCard**: Whether the customer has a credit card (Binary: 1 for Yes, 0 for No).
12. **IsActiveMember**: Whether the customer is an active member (Binary: 1 for Yes, 0 for No).
13. **EstimatedSalary**: Customer's estimated annual salary (Numeric).
14. **Exited**: Whether the customer has exited the bank (Binary: 1 for Yes, 0 for No, this is the target variable).

Exited (Target feature)Distribution:



The distribution of the target variable (Exited) is imbalanced, with a smaller proportion of customers exiting compared to those who remain.

Example of a Typical Data Record:

ID	Customer ID	Surname	Credit Score	GeoGraphy	Gender	Age	Tenure	Balance	NumOfProducts
0	15674932	Okwudilic	668	France	Male	33.0	3	0.0	2

HasCrCard	IsActiveMember	EstimatedSalary	Exited (Target)
1.0	0.0	181449.97	0

Section 3 - Pipeline

We begin by examining the dataset to understand its structure, the types of features it contains, and to identify any missing values or outliers. Next, we generate summary statistics for numerical features and frequency counts for categorical features to get an overview of the data distribution.

If any missing values are found, they are appropriately handled. After that, we convert string categorical values into numerical values. For instance, converting the 'Gender' column to binary values (0 for Female, 1 for Male), then applying one-hot encoding on the 'Geography' column to create binary columns for each geographic location. We also standardize numerical features to have a mean of 0 and a standard deviation of 1, which helps in improving the performance of certain algorithms that are sensitive to feature scaling.

Additionally, we create various visualizations to understand the distribution of features and identify potential relationships between the features and the target variable (Exited).

The data is then split into 70% for training and 30% for testing. We also use KFold validation to ensure the robustness of our models.

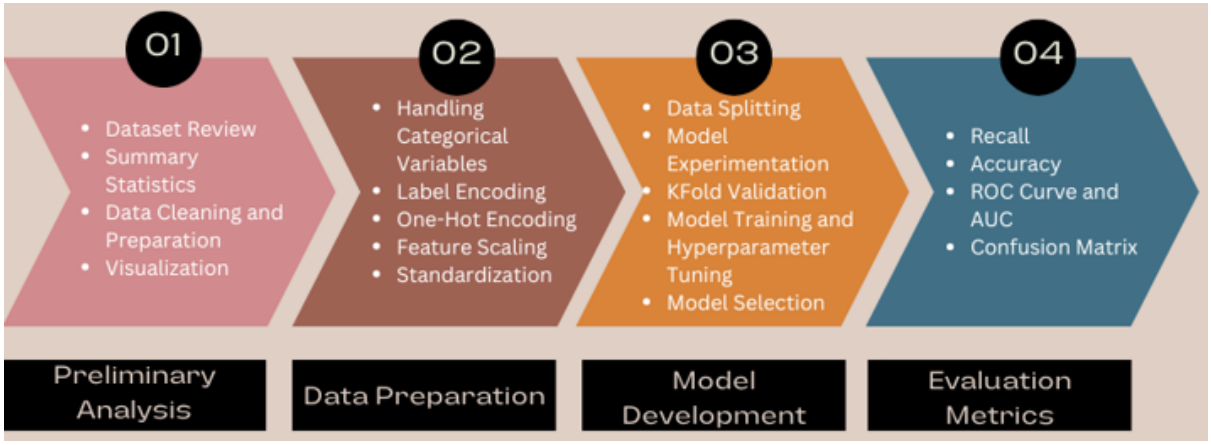
We employ several machine learning models such as XGBoost and Random Forest, as well as deep learning models to build our predictive models. After trying several models, we select the one that yields the best results.

As for the evaluation metrics, we choose the following:

- **Recall**: Given the business importance of identifying all potential churners, recall is prioritized. It measures the proportion of actual positives (customers who churn) that are correctly identified by the model.
- **Accuracy**: Checking the overall accuracy to understand the proportion of correct predictions.
- **ROC Curve and AUC**: Plotting the ROC curve and calculating the Area Under the Curve (AUC) to evaluate the model's ability to distinguish between the classes.

- **Confusion Matrix**: Creating a confusion matrix to visualize the performance of the classification model and understand the distribution of true positives, true negatives, false positives, and false negatives.

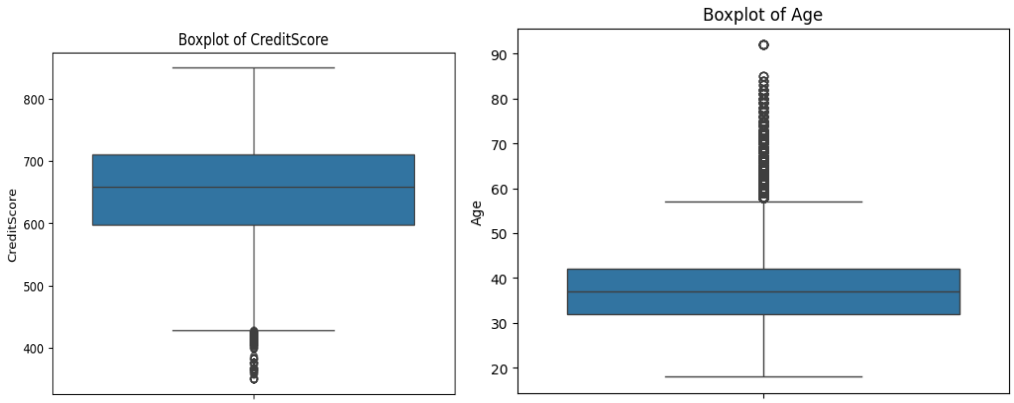
Pipeline diagram:



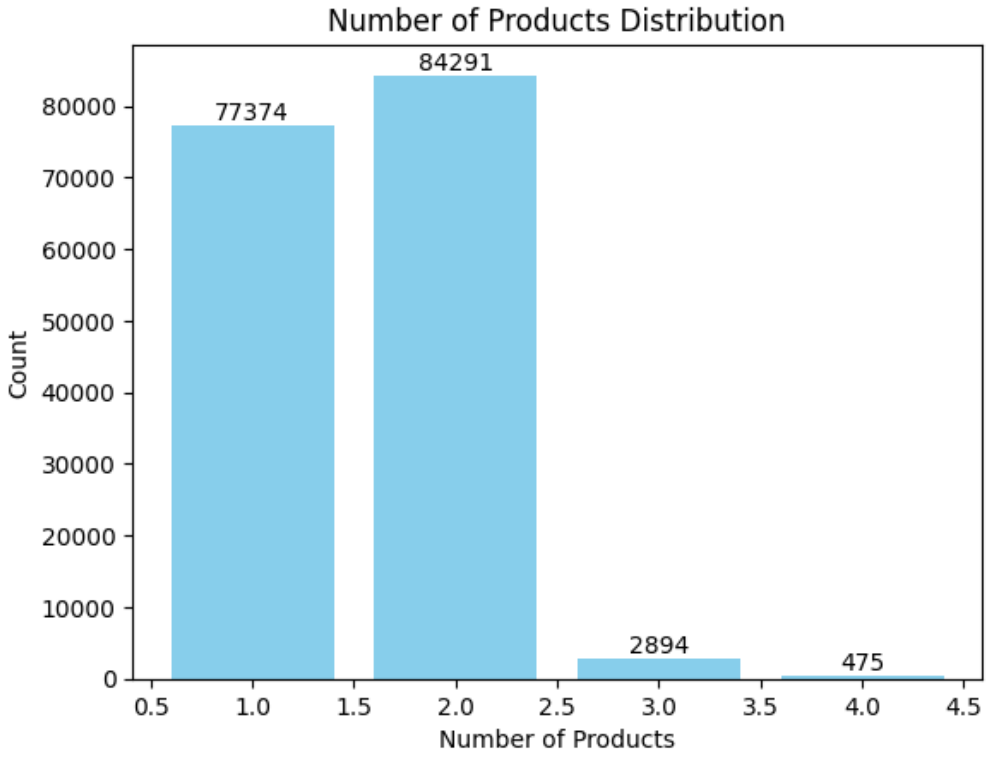
Section 4 - EDA Results

For this section we represent only important, abnormal and interesting insights.

4.1 Outliers

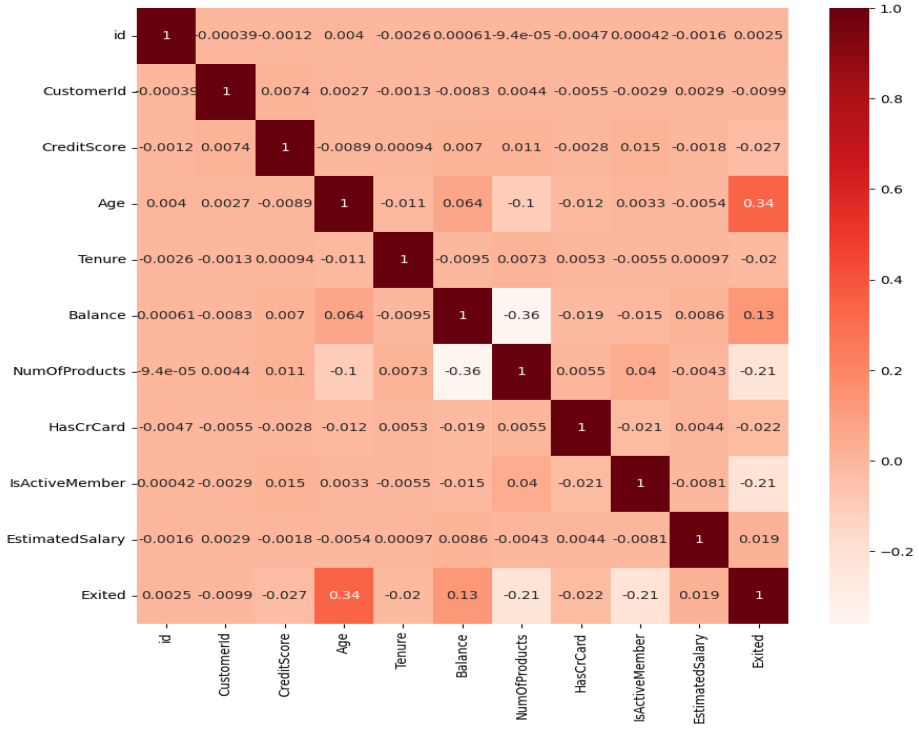


The only columns containing a significant number of outliers are Age and CreditScore. For the Age column, we decided to retain the outliers as they represent older customers (60 years and above) of the company. Similarly, for the CreditScore column, while most customers have a credit score above 500, the outliers reflect a real segment of the customer base and should not be removed.



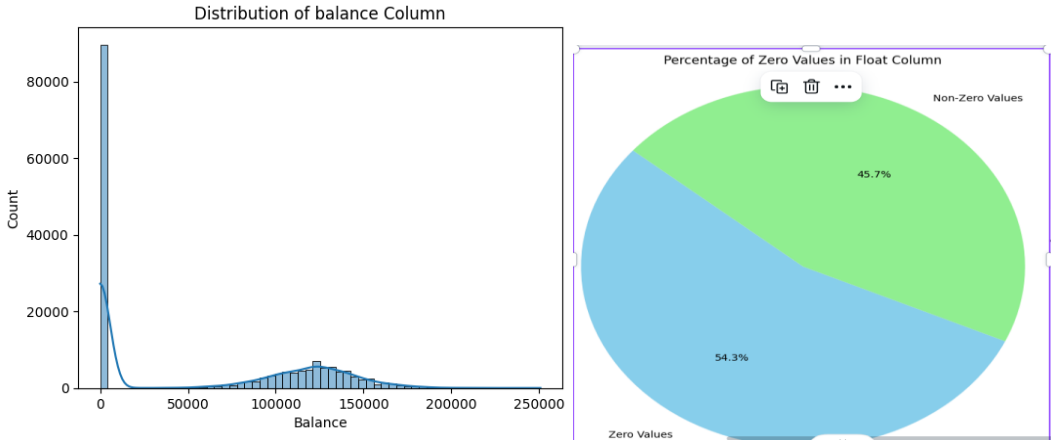
The above figure shows the distribution of the number of products owned by each customer. It is clear that most customers own up to two products.

4.2 Correlation Matrix:



We calculated the Pearson correlation between every feature pair, including the target variable. High correlation can be particularly useful, especially with the target variable. Unfortunately, the highest correlation observed is 0.36 between Balance and Number of Products. However, we do notice a correlation of 0.34 between Exited (the target variable) and Age, which is a positive indication that the model will perform reasonably well.

4.3 Data distribution

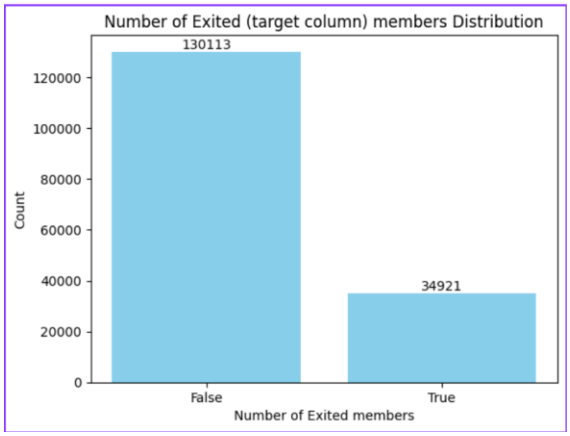


Here is the distribution of the Balance column, it shows two distinct groups: one with balances around zero, comprising a significant number of customers, and another with balances around 120k, following a normal distribution.

The right plot illustrates the percentage of each group, revealing that nearly half of the customers have a zero balance.

To address this distribution, we propose normalizing the data to a range of 0-1 using min-max normalization. This approach can be particularly beneficial for neural networks.

4.4 Target Variable Distribution



We have unbalanced data, with most samples having Exited=False, which poses a challenge since our primary interest is in the customers who exited the bank (Exited=True). There are two approaches to address this imbalance:

1. **Over Sampling**: Generate additional synthetic samples for the True class.
2. **Class Weight**: Assign more weight to the True class in the model parameters, affecting the loss calculation.

For our models, the second option yielded better results and performance.

EDA summary:

We got outliers in the columns Age, CreditScore, and Num of Products. but we decided to keep them in the training data since they are real cases and not noisy data. The main problem is the imbalance in data, columns “Exited” and “Balance”, specifically in the “Exited” column since it is the target variable in our task. We solved the imbalance data by applying standardization and class weights.

Section 5- Preprocessing Step

The first step was checking for missing values:

```
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   id               165034 non-null  int64
 1   CustomerId       165034 non-null  int64
 2   Surname          165034 non-null  object
 3   CreditScore      165034 non-null  int64
 4   Geography        165034 non-null  object
 5   Gender           165034 non-null  object
 6   Age              165034 non-null  float64
 7   Tenure           165034 non-null  int64
 8   Balance          165034 non-null  float64
 9   NumOfProducts    165034 non-null  int64
 10  HasCrCard        165034 non-null  float64
 11  IsActiveMember   165034 non-null  float64
 12  EstimatedSalary  165034 non-null  float64
 13  Exited           165034 non-null  int64
dtypes: float64(5), int64(6), object(3)
```

The dataset did not contain any missing values. First, we dropped the useless columns like CustomerId, surname, id.

We then converted the 'Gender' column to binary values (0 for Female, 1 for Male).

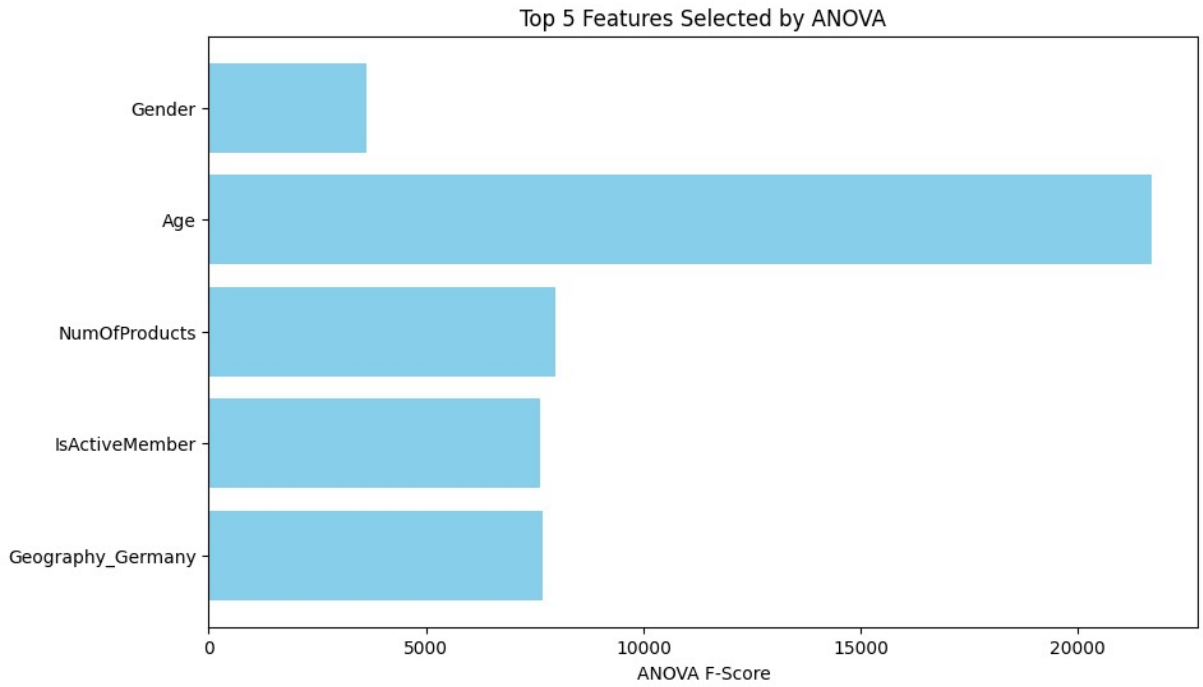
Next, we applied one-hot encoding to the 'Geography' column, creating binary columns for each geographic location.

Then, for ANN we standardized the numerical features to have a mean of 0 and a standard deviation of 1 (which was the game changer for our neural network model).

Section 6- Features Selection

For this section, we used ANOVA for feature selection.
ANOVA (Analysis of Variance) is used for feature selection because it helps determine the statistical significance of each feature in relation to the target variable. By calculating the F-score for each feature, ANOVA assesses how much a feature contributes to differentiating between classes. Features with higher F-scores are considered more important as they have a stronger association with the target, making them more valuable for predictive modeling. This method is particularly useful for selecting the most relevant features in classification tasks.

The following bar plot shows the 5 most important features selected by ANOVA:



As we can see, the most important feature according to the ANOVA is the “Age” feature. After we have the “NumOfProducts”, “IsActiveMember”, and “Geography_Germany” features. And the fifth most important feature is “Gender”.

Section 7- Methods

In this projected we tested 8 classification methods:

- Decision Tree
- Random Forest
- Gradient Boosting
- XGBoost
- CatBoost
- KNN
- AdaBoost
- Naive Bayse
- Fully Connected Neural Network

We also used one Logisitic Regression model

First, we did grid search on the models and defined the following parameters:

	Logistic Regression	Decision Tree	Random Forest	Gradient Boosting	XGBoost	CatBoost	KNN	AdaBoost	Naive Bayse
Parameters	C: [0.01, 0.1, 1,10,100]	max_depth: [None, 10, 20, 30] min_samples_split: [2, 5, 10]	n_estimators: [50,100] max_depth: [None, 10,20]	n_estimators: [50,100] learning rate: [0.01, 0.1] max_depth: [3,4]	n_estimators: [50,100] learning rate: [0.01, 0.2, 0.1] max_depth: [3,4,5]	iterations: [100,200] learning rate: [0.01, 0.1] depth: [3,4]	n_neighbors: [3,5,7,9] weights: ['uniform', 'distance']	n_estimators: [50,100] learning rate: [0.01, 0.2, 0.1]	Default Parameters

In order to check which model performs better, we used and did 10-fold cross validation on the models with their hyper-parameters from the previous grid search. Then we calculated the Accuracy, Precision, Recall, ROC and AUC for each model and found that XGBoost had the best accuracy and Random Forest had the best recall in our task.

Grid search results:

	Logistic Regression	Decision Tree	Random Forest	Gradient Boosting	XGBoost	CatBoost	KNN	AdaBoost	Naïve Bayse
Recall	0.681	0.776	**0.782**	0.55	0.561	0.541	0.194	0.5	0.191
Precision	0.369	0.534	0.546	0.746	0.741	0.753	0.286	0.749	0.592
AUC	0.74	0.87	0.89	0.89	0.89	0.89	0.57	0.88	0.77
Accuracy	0.686	0.809	0.816	0.8653	**0.8657**	0.8656	0.727	0.8587	0.793

Moreover, we used Fully Connected Neural Network with 4 layers with sizes of 12, 9, 27 and 1. We used batch size of 32 and 25 epochs.

Section 8 - Evaluation Results

Then, we did 10-fold CV Grid Search for XGBoost to find the hyper-parameters that will make the model perform best in accuracy. The hyper parameters we defined in the Grid Search were:

- Learning_rate: [0.1, 0.01, 0.05, 0.2]
- max_depth: [3,5,7]
- n_estimators: [100, 200]
- subsample: [0.8, 0.6]
- colsample_bytree: [0.6, 1.0]

- gamma: [0, 0.1, 0.2]

We found that the hyper-parameters that gets the best accuracy are:

- Learning_rate = 0.05
- max_depth = 5
- n_estimators = 200
- subsample = 0.6
- colsample_bytree = 1.0
- gamma = 0.2

we achieved accuracy of 86.61%
precision: 74.28%
recall: 56.12%

Next, we did 10-fold CV Grid Search for Random Forest to find the hyper-parameters that we will make the model perform best in Recall. The hyper parameters we defined in the Grid Search were:

- max_depth: [10, 20]
- n_estimators: [50, 100, 200]
- min_samples_split: [2, 5]

We found that the hyper-parameters that gets the best Recall are:

- max_depth = 10
- n_estimators = 200
- min_samples_split = 5

we achieved recall of 78.33%

We splitted the data for 70% train and 30% test for the final evaluation.

We used the models with their best hyper parameters for the evaluation on the train set.
For the XGBoost model, we achieved:

- Accuracy: 86.9%
- Precision: 75.36%
- Recall: 56.96%
- F1 Score: 74.88%
- AUC: 89.61%

For the Random Forest model, we achieved:

- Accuracy: 83.19%
- Precision: 57.44%
- Recall: 80.5%
- AUC: 90.33%

For the FCNN architecture, we achieved:

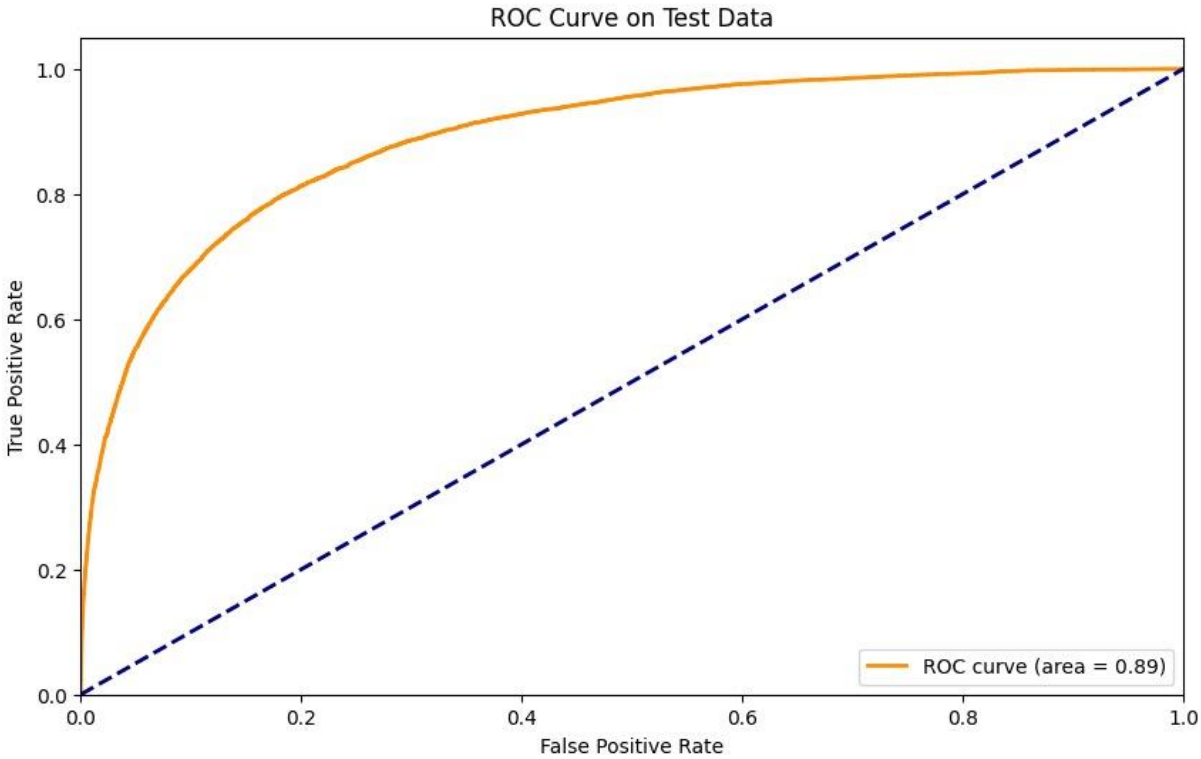
- Accuracy: 0.86
- Recall: 0.75
- Precision: 0.81
- F1 Score: 0.77

The next section shows the evaluation on the test set and we focused on three main methods Random Forest that maximize the Recall (find more people that want to leave), FCNN and XGBoost that maximize the Accuracy.

Section 9 - Selected Models

XGBoost:

ROC Curve:

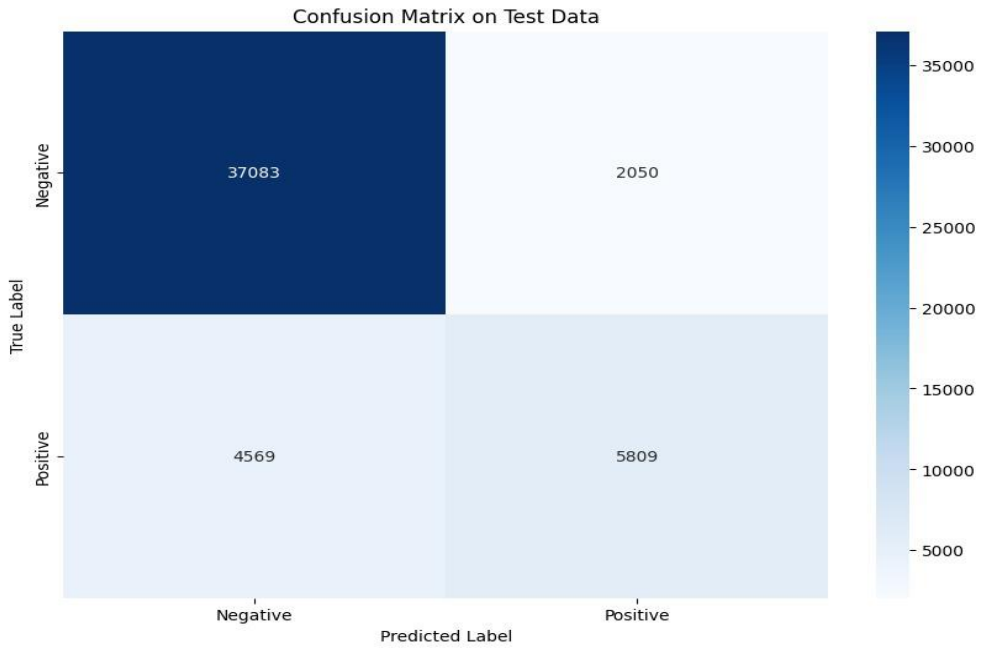


The ROC (Receiver Operating Characteristic) curve shown here plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) across different threshold values. The area under the curve (AUC) is 0.89, indicating that the model has a high discriminative ability, meaning it effectively distinguishes between the positive and negative classes. The closer the curve is to the top left corner, the better the model's performance. An AUC of 0.89 suggests strong performance, with a good balance between sensitivity and specificity.

Metrics:

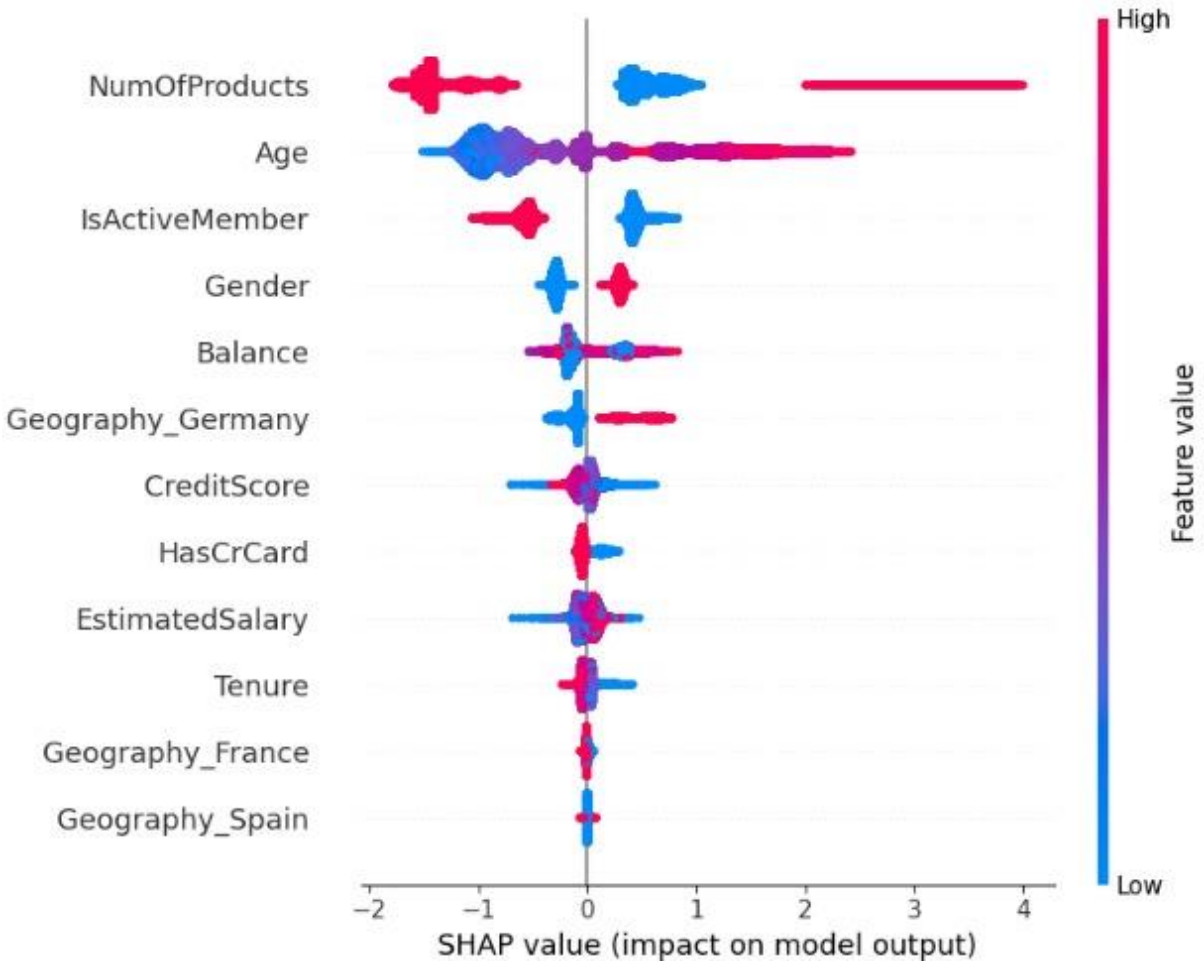
```
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.95      0.92     39133
           1       0.74      0.56      0.64     10378

    accuracy                           0.87     49511
   macro avg       0.81      0.75      0.78     49511
weighted avg       0.86      0.87      0.86     49511

Test Precision: 0.7392
Test Accuracy: 0.8663
Test Recall: 0.5597
Test F1 Score: 0.6371
Test ROC AUC Score: 0.8900
```

Confusion Matrix:

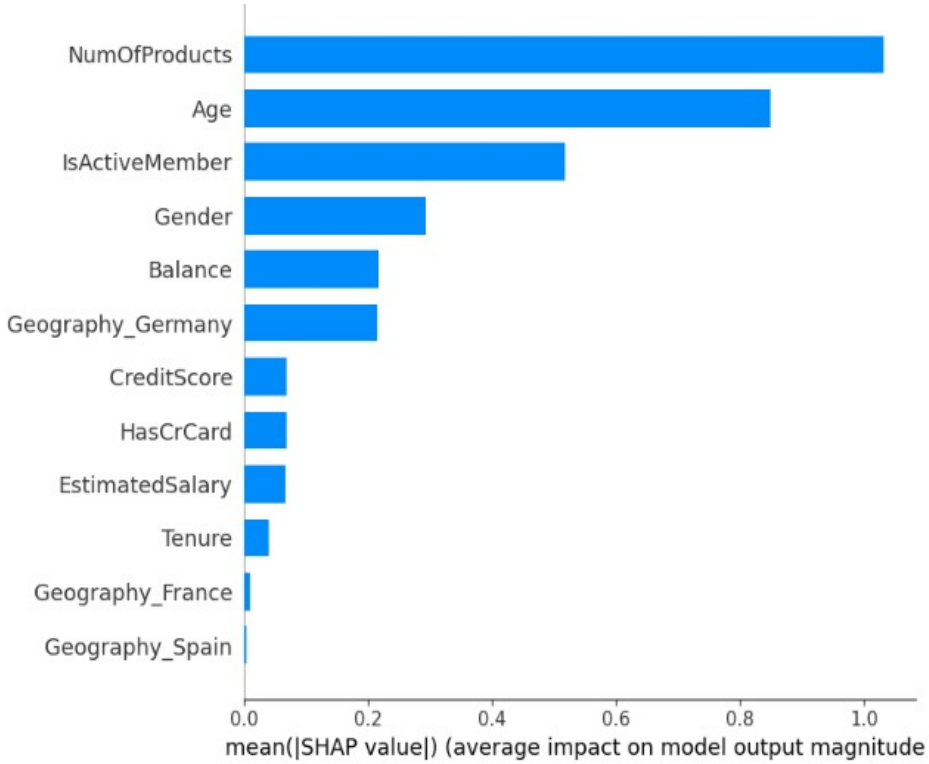


This confusion matrix illustrates the performance of a binary classification model on test data. The model correctly identified 37,083 negative cases and 5,809 positive cases. However, it incorrectly labeled 2,050 negative cases as positive (false positives) and 4,569 positive cases as negative (false negatives). The high number of true negatives suggests the model is effective at recognizing negative cases, but the false positives and false negatives indicate room for improvement in distinguishing between the two classes, particularly in identifying true positives.

SHAP:



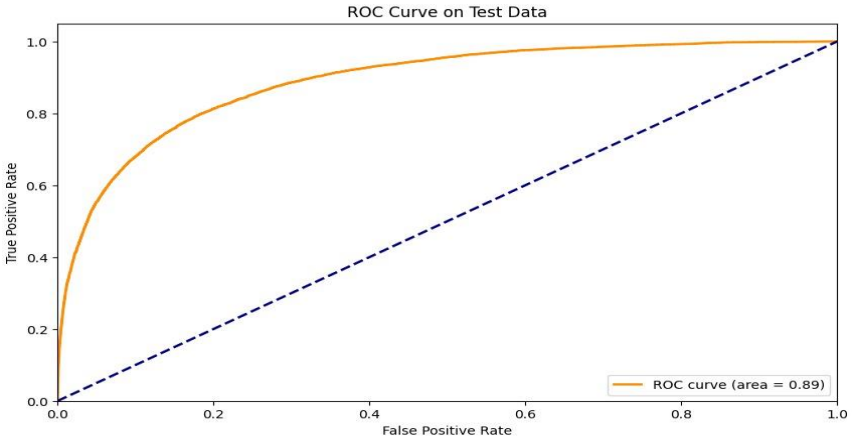
This SHAP summary plot visualizes the impact of features on the model's output for a binary classification task. Each point represents a SHAP value for a specific feature, showing how much that feature contributes to pushing the prediction toward either class. The color gradient from blue to red indicates the feature's value from low(EXITED=0) to high(EXITED=1). For example, "NumOfProducts" and "Age" have the most significant influence on the model's predictions, with higher values generally increasing the likelihood of exited clients. We can see in “NumOfProducts” that lowest values of SHAP values also increase the likehood of exited clients. Finally, “IsActiveMember” and “Gender’ have opposite influence on the likehood of exited or non-exited clients.



This bar plot shows the average impact of different features on the output of a machine learning model, measured by mean SHAP values. "NumOfProducts" and "Age" are the most influential features, indicating they have the highest impact on the model's predictions. Other significant features include "IsActiveMember," "Gender," and "Balance." Features such as "Geography_France" and "Geography_Spain" have minimal impact, suggesting they are less important for the model's decision-making process.

Random Forest:

ROC Curve:



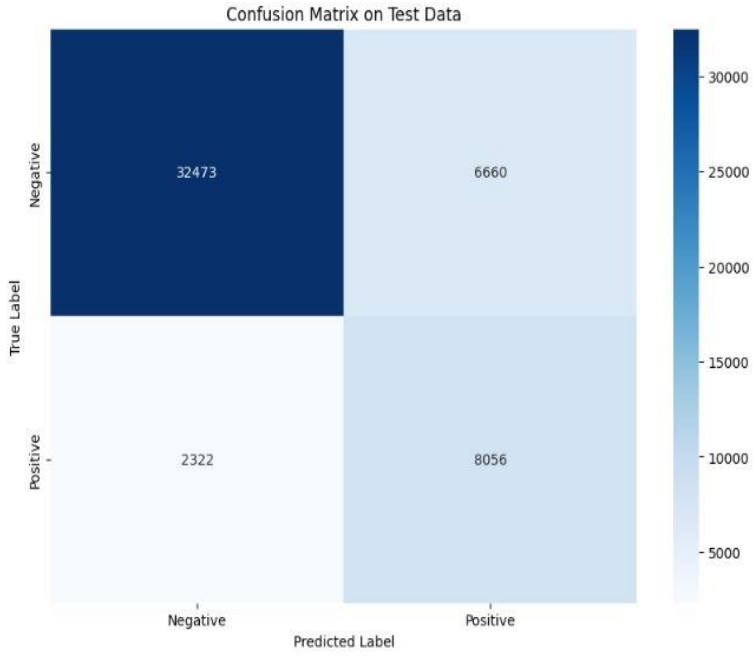
The ROC (Receiver Operating Characteristic) curve shown here plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) across different threshold values. The area under the curve (AUC) is 0.89, indicating that

the model has a high discriminative ability, meaning it effectively distinguishes between the positive and negative classes. The closer the curve is to the top left corner, the better the model's performance. An AUC of 0.89 suggests strong performance, with a good balance between sensitivity and specificity.

Metrics:

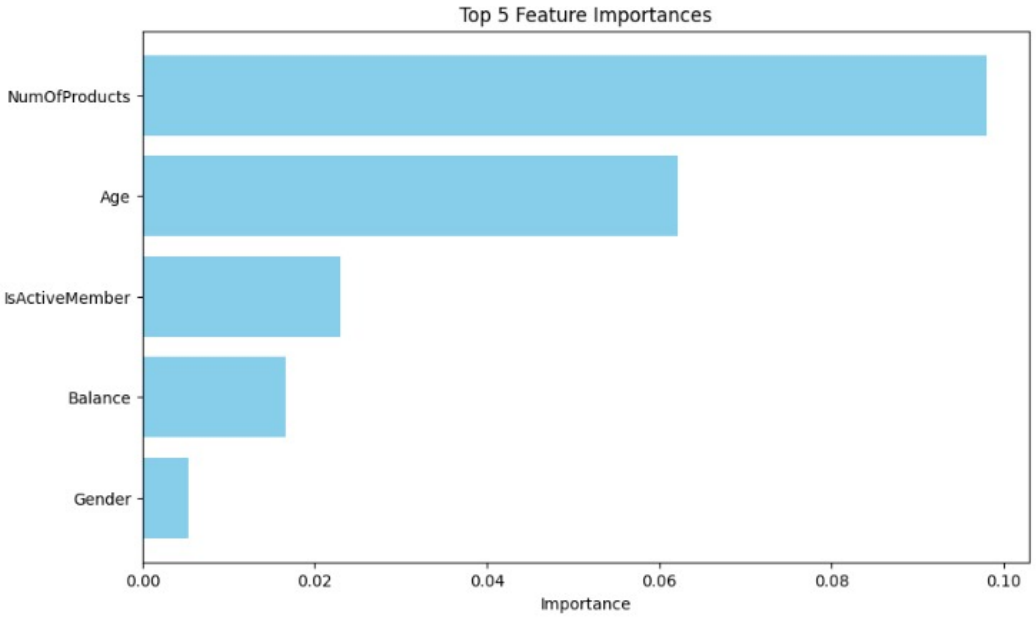
```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.83      0.88     39133
           1       0.55      0.78      0.64     10378

    accuracy                           0.82     49511
   macro avg       0.74      0.80      0.76     49511
weighted avg       0.85      0.82      0.83     49511

Test Precision: 0.5474
Test Accuracy: 0.8186
Test Recall: 0.7763
Test F1 Score: 0.6421
Test ROC AUC Score: 0.8864
```

Confusion Matrix:



This confusion matrix illustrates the performance of a binary classification model on test data. The model correctly identified 32,473 negative cases and 8,056 positive cases. However, it incorrectly labeled 6,660 negative cases as positive (false positives) and 2,322 positive cases as negative (false negatives). The high number of true negatives suggests the model is effective at recognizing negative cases, but the false positives and false negatives indicate room for improvement in distinguishing between the two classes, particularly in identifying true positives.

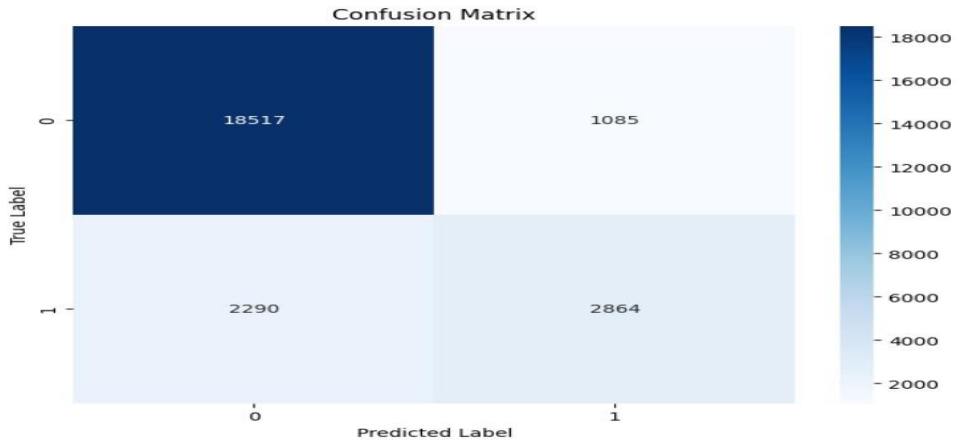
PFI:



This bar plot shows the top 5 feature importances based on Permutation Feature Importance (PFI) for a machine learning model, ranking features by their contribution to model performance. "NumOfProducts" emerges as the most significant predictor, followed closely by "Age," indicating these two features are crucial for the model's accuracy. "IsActiveMember" and "Balance" also play important roles but to a lesser extent, while "Gender" has the least impact among the top five, suggesting it contributes the least to the model's predictions within this subset of features.

FCNN:

Confusion Matrix:



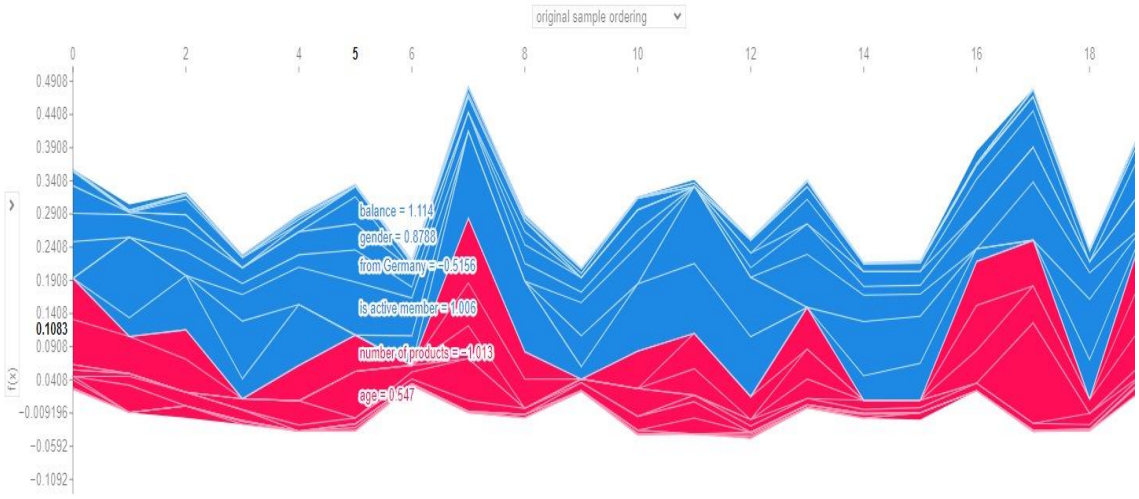
This confusion matrix shows that the model correctly classified 18,517 instances of class 0 and 2,864 instances of class 1. It misclassified 1,085 instances of class 0 as class 1, and 2,290 instances of class 1 as class 0.

Metrics:

	precision	recall	f1-score	support
0	0.89	0.94	0.92	19602
1	0.73	0.56	0.63	5154
accuracy			0.86	24756
macro avg	0.81	0.75	0.77	24756
ighted avg	0.86	0.86	0.86	24756

Kernel SHAP:

This SHAP plot visualizes the impact of different features on the model's output for individual instances. Each column represents a sample from the dataset, with blue lines indicating a negative SHAP value and red lines indicating a positive SHAP value. Key features such as balance, gender, geography (from Germany), isActiveMember, number of products, and age are highlighted with their respective SHAP values. The plot shows how each feature contributes to pushing the prediction higher or lower. For instance, a higher balance or being an active member typically pushes the prediction higher (red), while being from Germany or having a higher number of products can push it lower (blue). This visualization helps in understanding the influence and direction of each feature on the model's predictions for individual samples. Finally, we can find from this SHAP correlation between each feature in each sample towards other features or individual features



Results Summary:

	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	**86.63%**	73.92%	55.97%	63.71%	**89%**
Random Forest	81.86%	54.74%	**77.63%**	64.21%	88.64%
FCNN	86%	**81%**	75%	**77%**	-

Section 10 - Future Work

To improve the performance of the customer churn prediction model, more work can be done in the future. For example:

1. **Adding Demographic Information**: More granular demographic data, such as education level and employment status, could help identify at-risk customers.

2. **Feature Engineering**: Create new features by combining existing ones, to capture more complex behaviors.

3. **Model Ensembling**: Combine predictions from multiple models (e.g., gradient boosting, neural networks) to improve overall accuracy and robustness.

4. **Customers Embedding:** Create embedding of the customers data can help to find similarity between customers.

5. **Anomaly Detection**: Implement techniques to identify outliers and unusual customer behaviors that may signal potential churn.

Section 11 - Summary

This report presents a comprehensive study on predicting customer churn using machine learning and deep learning techniques. Customer churn, the loss of clients or customers, is a significant challenge for financial institutions. The bank's profitability relies heavily on retaining existing customers, as loyal customers are more likely to use additional services and products. By predicting which customers are at risk of churning, the bank can proactively engage with them through personalized offers and targeted marketing campaigns.

The project used a dataset containing various features of bank customers to develop a binary classification model that predicts customer churn. The dataset included 14 columns, such as customer ID, credit score, geography, gender, age, tenure, balance, number of products, credit card status, active member status, estimated salary, and the target variable, 'Excited', indicating whether a customer has exited the bank.

The pipeline involved examining the dataset, handling missing values, converting categorical values to numerical values, standardizing numerical features, and visualizing feature distributions and relationships with the target variable. The data was split into training and testing sets, and several machines learning models, including XGBoost and Random Forest, as well as a deep learning model, were employed to build predictive models.

The evaluation metrics included recall, accuracy, ROC curve, AUC, and confusion matrix. The XGBoost model achieved an accuracy of 86.63%, precision of 73.92%, recall of 55.97%, F1 score of 63.71%, and AUC of 89%. The Random Forest model achieved a recall of 77.63% and an AUC of 88.64%. The Fully Connected Neural Network achieved an accuracy of 86%, recall of 75%, precision of 81%, and F1 score of 77%.

The project on predicting customer churn yielded several significant insights.

The main insights from EDA are that our predictions in the training set are unbalanced. Therefore, we need to do augmentation to smaller prediction or oversample smaller prediction because we can't drop bigger prediction because of the small dataset in order to keep the predictions balanced. Moreover, we don't see any correlations between the independent features and all the features are balanced in categorical features and with normal distribution in continuous features.

Feature selection using ANOVA identified 'Age', 'NumOfProducts', 'IsActiveMember', 'Geography_Germany', and 'Gender' as the most critical features for predicting churn before modeling.

We also had some key insights into the models from the work. First, was that the XGBoost model is a model that returned the best Accuracy and the best AUC, while the Random Forest model returned the best results in terms of Recall on all classic ML models. Second, the ANN model returned less good results than XGBoost in the Accuracy meaure, therefore if we wanted to choose between a deep learning model and classic ML, we would choose XGBoost. Third insight, the ANN excelled in the Precision and F1 Score indices more than the ML models.

Because our main goal is Recall, that is to maximize the finding of customers who are going to leave the bank, we chose the Random Forest model as our classification model.

The insights we were able to understand from SHAP are that after modeling NumOfProducts has the strongest influence on whether a customer will leave or stay with the bank. After it, Age has the strongest influence on prediction and the features after it influence much less, for example Balance, IsActiveMember, Gender and the features that do not influence at all are Geography_France, Geography_Spain. Hence, the geographic location does not affect the model at all and the number of products the customer has in the bank has the greatest influence on whether the customer will stay or leave the bank.

Looking ahead, the project suggests several avenues for future work. These include integrating more detailed demographic data, employing advanced feature engineering techniques, exploring model ensembling to combine the strengths of multiple models, using customer embeddings for better representation, and implementing anomaly detection methods to further enhance the model's predictive capabilities. These recommendations aim to build upon the current findings and continue improving the accuracy and reliability of customer churn predictions.