

BA 723 – Business Analytics Capstone

Comprehensive Report

Prediction of Customer Churn at Sapphire Bank

Ishika Bhutani

301323447

Table of Contents

1.0 Executive Summary	4
1.1 Executive Introduction	4
1.2 Executive Objective.....	5
1.3 Executive Model Description:	6
1.4 Executive Recommendations	8
2.0 Introduction	9
2.1 Background	9
2.2 Problem Statement	10
2.3 Objectives and Measurement	11
2.4 Assumptions and Limitations	11
3.0 Data Sources.....	13
3.1 Data Set Introduction	13
3.2 Exclusions	14
3.3 Data Dictionary	16
4.0 Data Exploration	17
4.1 Data Exploration Techniques	17
4.1.1 Descriptive Statistics	17
4.1.2 Outlier Detection	19
4.1.3 Visualization.....	27
4.1.4 Class Imbalance	37
4.2 Summary.....	39
5.0 Data Preparation and Feature Engineering	40
5.1 Excluding Irrelevant column	40
5.2 Feature Engineering	40
.....	41
5.3 Transformations	41
6.0 Model Exploration.....	43
6.1 Modelling Approach	43
6.2 Model Technique #1: Full Logistic Regression	43
6.3 Model Technique #2: Forward Logistic Regression	51
6.4 Model Technique #3: Backward Logistic Regression	54
6.5 Model Technique #4: Stepwise Logistic Regression.....	57

6.6 Model Technique #5: Decision Tree model	61
6.7 Model Technique #6: Random Forest model.....	65
6.8 Model Technique #7: Gradient Boosting Model.....	72
6.9 Model Comparison	78
7.0 Model Recommendations.....	80
7.1 Model Selection.....	80
7.2 Model Theory.....	81
7.3 Model Assumptions and Limitations	82
7.4 Model Sensitivity to Key Drivers	83
8.0 Conclusion and Recommendations	85
8.1 Impacts on Business Problem	86
8.2 Recommended next steps.....	87
References	89

1.0 Executive Summary

1.1 Executive Introduction

This report involves the systematic study of Sapphire Bank which is an America banking institution operating in Germany, France and Spain. Various factors have been researched that influence the customer decisions to leave and developed predictive models to identify at-risk customers and the essential strategies to mitigate churn and enhance customer loyalty using Python. The dataset has been modelled using Logistic Regression, Decision Tree, Random Forest and Gradient Boosting Machines and the most effective model has been selected based on the performance metrics.

In a highly competitive banking industry, customer retention is a crucial factor to maintain success and profitability. The Sapphire Bank is facing an ongoing challenge of understanding and mitigating customer churn. Churn is defined as the rate at which the customers discontinue their banking relationships, leading to significant financial implications which can include losing revenue, increased cost of acquisition and negative impacts on brand reputation (CHENG, 2024).

This report delves into a comprehensive analytical approach that predicts the customer churn at The Sapphire Bank. It uses advanced machine learning techniques, and the objective is to identify key drivers of churn and develop predictive model which enables proactive customer retention strategies. The Sapphire bank aims to enhance its customer relationship management and strengthen its market position in European banking sector.

Each model's performance in the study is assessed on metrics such as accuracy, ROC-AUC score, and F1 score. The study is focussed on understanding the sensitivity to key drivers of churn. Additionally, the analysis explores the impact of reducing the feature sets to improve

model efficiency without compromising the predictive power. Through this report, The Sapphire Bank seeks to transform its customer data into actionable insights which provides a strategic framework for reducing the churn and fostering long-term customer loyalty. The findings and recommendations presented in the report are intended to guide the bank's decision-making processes which are customer-centric and ensures that it aligns with the business objective of The Sapphire Bank and its competitive landscape.

1.2 Executive Objective

The primary objective of this project is to develop a robust predictive model that identifies the customers which are at risk of churning at The Sapphire Bank. By accurately predicting the customer churn, the bank aims to implement targeted retention strategies which enhance customer loyalty and minimize revenue loss.

The following are the specific objectives of the project:

- a) Data Analysis and Preparation: To ensure that the data is suitable for modelling, it is first cleaned, pre-processed and analysed. This involves the process that handles the missing values (if any). Categorical variables need to be encoded and numerical features must be normalized.
- b) Model Development and Evaluation: Multiple predictive models are developed that includes Logistic Regression, Decision Tree, Random Forest and Gradient Boosting Machines. Each model is evaluated based on the performance metrics such as accuracy, ROC-AUC score, F1 score. Based on these metrics, the most effective model is selected for predicting the customer churn.

- c) Feature Importance Analysis: Key drivers are identified that significantly influence the customer retention and to assess the impact of various customer attributes on churn. This analysis will guide the bank in focussing on critical areas for intervention.
- d) Model Optimization: To ensure an optimal balance between the predictive accuracy and computational efficiency, model performance is enhanced by tuning the hyperparameters and reducing the feature sets.
- e) Business Insights and Recommendations: To translate the predictive model's outcome to actionable insights, certain strategic business recommendations are provided to The Sapphire Bank to reduce the customer churn and improve customer satisfaction.
- f) Implementation Framework: To propose a practical framework for integrating the predictive model into bank's existing customer relationship management system, enabling real-time churn prediction and proactive retention efforts.

1.3 Executive Model Description:

The Sapphire Bank has employed a suite of sophisticated machine learning models in pursuit of minimizing customer churn and enhancing retention strategies. These models are designed in such a manner that they predict which customers are more likely to leave the bank. This will allow timely and targeted interventions. The models which are evaluated include Logistic Regression, Decision Tree, Random Forest and Gradient Boosting Machines (GBM). These models have unique advantages and offers useful insights from a business perspective.

- a) **Logistic Regression:** This model estimates the probability of customer churn on various predictor variables. It is simple and transparent which makes it an excellent tool for understanding the baseline factors that drive churn. The model's coefficients highlight the most significant predictors which allows the bank to focus its retention efforts on the variables that are more influential.
- b) **Decision Tree:** The decision tree offers a visual and intuitive approach to churn prediction. By splitting the data on key features, it creates a straightforward decision-making framework. It helps in segmenting customers into distinct groups for the Sapphire Bank. This segmentation enables the bank to tailor its marketing and retention strategies to specific customer segments which addresses their unique needs and increase the effectiveness of interventions.
- c) **Random Forest:** Random Forest is an ensemble learning method which averages the results of multiple decision trees and enhance the prediction accuracy. For the Sapphire Bank, it helps in the identification if relative importance of different features along with it gives the highest accuracy.
- d) **Gradient Boosting Machines:** This model builds sequentially with each new model that corrects the error made by the previous ones. (Saini, 2024) The iterative approach captures the complex patterns in the data that makes it highly effective for predictive tasks. For The Sapphire Bank, this model demonstrates a strong ability to predict customer churn with high ROC-AUC score. This enables the bank to implement precise and timely retention measures which significantly reduce the churn rates.

1.4 Executive Recommendations

The Sapphire Bank formulates several key recommendations based on the comprehensive analysis and predictive modelling to address the customer churn. These recommendations aim to leverage the insights gained from the machine learning models to implement effective customer retention strategies, enhancing satisfaction for customers and improve the business revenue.

- a) Implement a Targeted Retention Strategy: Using the predictive power of random forest and gradient boosting machines, the bank focusses retention efforts on the customers through personalised offers, loyalty programs and proactive engagement. Interventions are tailored according to the important features such as age, number of products held, customer tenure which increases the effectiveness of these strategies.
- b) Enhance Customer Experience: Invest in improving the overall customer experience by addressing pain points identified through model insights. For example, offer more attractive product bundles and ensure excellent customer service. By addressing the specific needs, the bank can improve satisfaction and reduce churn rates.
- c) Leverage Feature Importance Insights: The bank needs to focus on enhancing features that have the most significant impact on customer retention. It can include offering incentives to the customers who have been with the bank for a shorter period or those with lower credit scores to foster loyalty and reduce their churning.
- d) Integrate Predictive Models into CRM systems: Integrate the developed predictive models into The Sapphire Bank's existing Customer Relationship Management System (CRM). This will enable real-time churn prediction and allow for timely and proactive customer interventions. This ensures that CRM is equipped to handle and act on the predictive insights provide by the models.

- e) Monitor and Refine Models Regularly: Customer behaviour and market conditions can change overtime, so it is important to keep the models updated and relevant. Continuous monitoring the performance of the predictive models and refine them as necessary to maintain their predictive accuracy and effectiveness.

2.0 Introduction

2.1 Background

While it is vital to gain new customers for any bank, it is equally important to retain the customers. Generally, the financial sector has a retention rate of around 75%, which also has a scope of potential improvement. A Forrester report found that 87% of banking customers who feel valued will stay with their bank. PWC found that 46% of customers would leave a financial service or banking brand because of a product or service. (Puga, 2024)

In banking sector, the prediction of customer churn is crucial as retaining customers can significantly impact profitability. The model helps in identifying the customers who are likely to leave the Sapphire bank and the proactive measures to retain them. The dataset which is used in this analysis includes features like age, credit score, balance, number of products among others. Understanding and mitigating churn is important for maintaining profitability, reducing acquisition costs, and fostering customer loyalty.

2.2 Problem Statement

The business problem is to develop a predictive model that uses customer data to predict whether a customer will churn in a bank or not. If a customer churns, it means they have left the bank and took their account somewhere else. It is necessary to predict the customers who are likely to churn. By doing so, the bank can take measures to retain them. This can include providing promotions, discounts or incentives to boost customer satisfaction, and therefore, retention.

The Sapphire Bank faces the following challenges:

- a) Identifying at-risk customers: Determining which factors are likely to churn based on their transaction history, demographics, product usage and interaction with the bank.
- b) Understanding Key Drivers of Churn: Identifying the critical factors that influence customer decision to leave the bank.
- c) Developing Accurate Predictive Models: Building and validating machine learning models that can reliably predict churn with high accuracy and provide actionable insights.
- d) Implementing Targeted Retention Strategies: Development and deployment of targeted interventions using predictive insights which are aimed at retaining high-risk customers and improving overall customer satisfaction.

2.3 Objectives and Measurement

The primary objective of this analysis is to develop and evaluate different machine learning models to predict customer churn. The specific objectives include:

- a) Preprocessing and analysing the dataset
- b) Implementing multiple machine learning models including Logistic Regression, Decision Trees, Random Forest and Gradient Boosting Machines.
- c) Evaluation of the models using performance metrics such as accuracy, precision, recall, F1-score and ROC-AUC score.
- d) Identifying the best model based on the evaluation metrics and interpreting its results to provide actionable insights.

2.4 Assumptions and Limitations

The following are the assumptions for the analysis:

- a) The dataset used for the analysis is assumed to be accurate, clean and free from inconsistencies. The preprocessing steps including handling missing values and encoding categorical variables are correctly implemented and do not introduce bias into the models.
- b) The features in the dataset are assumed to be relevant and sufficient in predicting customer churn. No additional external factors are considered in this analysis.
- c) The data splitting process which divides the dataset into training and testing sets is assumed to be done randomly and appropriately that ensures both sets are representative of the entire dataset.

- d) The choice of models and their hyperparameters are assumed to be appropriate for this task. The method that is used to find the best parameters is GridSearchCV assuming it explores the hyperparameter space.
- e) The creation of interaction terms (e.g. Balance_Age and CreditScore_NumOfProducts) are assumed to be relevant and contribute positively to the performance of the model.
- f) The treatment of outliers based on z-scores is assumed to be appropriate and from business perspective the outliers does not skew the results.
- g) Since the dataset does not contain any date or year, it is assumed that time-based factors do not influence customer churn in this analysis. The models are built on static data without considering the trends which are temporary.

The following are the limitations for the analysis:

- a) The absence of date or year information limits the ability to analyze trends and seasonality effects.
- b) The models are trained or tested on the provided dataset, and their performance may not generalize well to other dataset or future data. The predictiveness can vary with new and unseen data.
- c) The models are evaluated using specific metrics (accuracy, precision, recall, F1-score, and ROC-AUC). While these metrics provide insights into model performance, they may not capture all the aspects of model effectiveness, such as economic impact or customer satisfaction.
- d) The analysis does not account for other external factors such as market conditions, competitive actions or macroeconomic trends.

3.0 Data Sources

The data for this analysis is sourced from Kaggle. It does not have any name for bank. The dataset has various attributes which are about the customers of the bank which includes demographics, account information, and transactional behaviour. It is assumed that the data has been collected from some bank's internal system. The link to the data source: <https://www.kaggle.com/code/murilozangari/customer-bank-churn-prediction>

3.1 Data Set Introduction

The dataset has 10,000 customer records and 14 columns that provide valuable information to the profiles of the customers and their banking behaviour. The columns include details such as age, gender, account related information like balance and the number of products held. The columns also indicate whether the customer is an active member or not. 'Exited' is the target variable where the customer has churned (1) or not (0). (ZANGARI, n.d.)

The 'df.info()' function provides a concise summary of the dataset which is useful for understanding the structure of the data and identifying any potential issues. The analysis shows that all columns have 10,000 non-null entries reflecting that there are no missing values in the dataset. The dtype displays the data type of each columns which includes: (ZANGARI, n.d.)

- 1) 'int64': Integer values for 'RowNumber', 'Customer ID', 'CreditScore', 'Age', 'Tenure', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'Exited'.
- 2) 'float64': Floating point values for 'Balance', 'Estimated Salary'.
- 3) 'Object': String or categorical variables for 'Surname', 'Geography', 'Gender'.

The analysis uses 'df.isnull().sum()' function that calculates the total number of missing values for each column in the dataset. The function is important to identify any data quality issues which is related to the missing values. Since there are no missing values, therefore the output for all the variables is 0. The absence of missing values indicate good data quality, ensuring that all the records are complete and can be used for analysis and modelling without concerns about data gaps.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   RowNumber             10000 non-null  int64  
 1   CustomerId            10000 non-null  int64  
 2   Surname               10000 non-null  object  
 3   CreditScore           10000 non-null  int64  
 4   Geography             10000 non-null  object  
 5   Gender               10000 non-null  object  
 6   Age                  10000 non-null  int64  
 7   Tenure               10000 non-null  int64  
 8   Balance              10000 non-null  float64 
 9   NumOfProducts        10000 non-null  int64  
10   HasCrCard            10000 non-null  int64  
11   IsActiveMember       10000 non-null  int64  
12   EstimatedSalary      10000 non-null  float64 
13   Exited               10000 non-null  int64  
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

```
df.isnull().sum()

RowNumber      0
CustomerId     0
Surname        0
CreditScore    0
Geography      0
Gender         0
Age           0
Tenure        0
Balance       0
NumOfProducts 0
HasCrCard     0
IsActiveMember 0
EstimatedSalary 0
Exited        0
dtype: int64

There are no null values.
```

3.2 Exclusions

There are some data points and attributes that must be excluded from the analysis due to their irrelevance, redundancy, or lack of sufficient information. These exclusions are vital as they ensure that the dataset is focussed on the most appropriate and informative attributes for predicting customer churn. Any columns that do not contribute to the prediction must be dropped.

The code 'df2= df.drop(columns=["RowNumber","CustomerId","Surname"])' is used to remove the specific columns from the dataset. These columns are dropped because they do not

provide meaningful information for the analysis or modelling process. RowNumber is an index that simply put numbers in the rows of the dataset. It does not provide any additional information about the customers and is not useful for analysis or modelling. The column CustomerId uniquely identifies each customer and does not contribute to understanding customer behaviour or predicting outcomes like churn. It is a unique identifier that has no predictive power. The column 'Surname' is not relevant for analysis or predictive modelling. It does not influence any of the target variables or provide insights into customer behaviour.

By dropping these columns, I can simplify the dataset and focus on the variables that are more likely to provide valuable insights for analysis and modelling. The cleaned dataset 'df2' now contains only the relevant features which can be used for further exploratory data analysis, feature engineering and building predictive models.

```
[ ] df2= df.drop(columns=["RowNumber","CustomerId","Surname"])
```

The columns "RowNumber", "CustomerId", and "Surname" were dropped from the dataset because they are unique identifiers that do not provide meaningful insights for analysis. These columns do not contribute to understanding customer behavior or predicting churn. Removing them simplifies the dataset, focusing on relevant features.

```
[ ] df2.head()
```

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

3.3 Data Dictionary

The initial data was previewed using ‘df.head()’ which displays the first five rows of the dataset. The dataset contains several key attributes for each customer including:

Number	Variable Name	Meaning
1	RowNumber	A sequential number indicating the row in the dataset.
2	CustomerId	A unique identifier for each customer
3	Surname	The customer’s last name
4	CreditScore	A numerical value representing the customer’s credit score
5	Geography	The country where the customer lives in
6	Gender	The customer’s gender
7	Age	The customer’s age
8	Tenure	The number of years the customer has been with the bank.
9	Balance	The customer’s account balance
10	NumOfProd	The number of bank’s products that the customer uses.
11	HasCrCard	Whether the customer has a credit card (1) or not (0)
12	IsActiveMember	Whether the customer is an active member (1) or not (0)
13	EstimatedSalary	The estimated salary of the customer
14	Exited	Whether the customer has churned (1) or not (0). (Target Variable)

(ZANGARI, n.d.)

4.0 Data Exploration

The first step in data analysis that involves the use of data visualisation tools and statistical techniques to uncover data set characteristics and initial pattern is called data exploration. In this section, raw data is analysed for similarities, patterns, outliers and relationships are identified between different variables (Robinson, n.d.).

Stakeholders visualise the data more easily in the form of charts rather than numerical data. It can be challenging for decision-makers to uncover meaningful insights through raw data. Data visualisation and exploration transforms complex data into intuitive visual representations. (Bozkurt, 2023)

This step is important for forming hypothesis and guiding the subsequent stages of analysis. The exploration phase set the foundation for building predictive models. The insights gained from this phase guided the feature engineering and model selection process. This ensures that the models which are developed are both accurate and interpretable.

4.1 Data Exploration Techniques

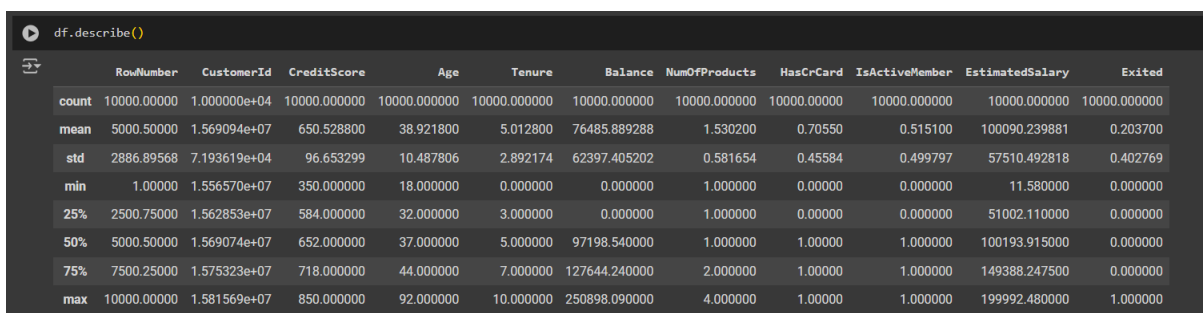
4.1.1 Descriptive Statistics

To summarize the central tendency, dispersion and the shape of the dataset's distribution, descriptive statistics were used. Key metrics such as mean, median, standard deviation and percentiles were calculated for numerical features like 'CreditScore', 'Age', 'Balance', 'NumOfProducts' and 'EstimatedSalary'.

The 'df.describe()' function generates descriptive statistics for the numerical columns in the dataset. This function is vital for understanding the central tendency, dispersion and the shape of the data distribution.

1. Count: Count is the number of non-null entries for each column. There are 10,000 entries in the dataset.
2. Mean: The average value of each column.
3. Std (Standard Deviation): A measure of the amount of variation or dispersion in the data.
4. Min: The minimum value in each column.
5. 25% (1st Quartile): The value below which 25% of the data falls.
6. 50% (Median): The middle value of the data.
7. 75% (3rd Quartile): The value below which 75% of the data falls.
8. Max: The maximum value in each column.

(Hayes, 2024)



	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

Interpretation:

- ➔ The credit card scores are fairly distributed with a peak around the mean and some variation.
- ➔ Most customers are middle-aged, with a few younger or older customers.
- ➔ Customers have varying tenures, indicating a mix of new and long-term customers.
- ➔ There is a wide range of balances, with many customers having zero balance and a few with very high balances.
- ➔ Most customers have one or two products with few having three or four.
- ➔ The majority of the customers possess a credit card.
- ➔ The distribution between active and inactive members is almost even.
- ➔ Customers have a wide range of estimated salaries, indicating diverse income levels.
- ➔ A minority of customers have churned, with the majority remaining with the bank (20.37%)

4.1.2 Outlier Detection

An outlier is an extremely high or extremely low data point relative to the nearest data point and the rest of the neighbouring co-existing values in a data graph or data set that is being worked. These are extreme values that stand out from the overall pattern of values in the dataset or graph. (Lemonaki, 2021)

This is a critical step in data preprocessing. This step identifies and handles anomalous data points that deviate significantly from the rest of the dataset. In this analysis, outliers were detected statistically through boxplot and z-scores which measures the number of standard deviation a data point is from the mean.

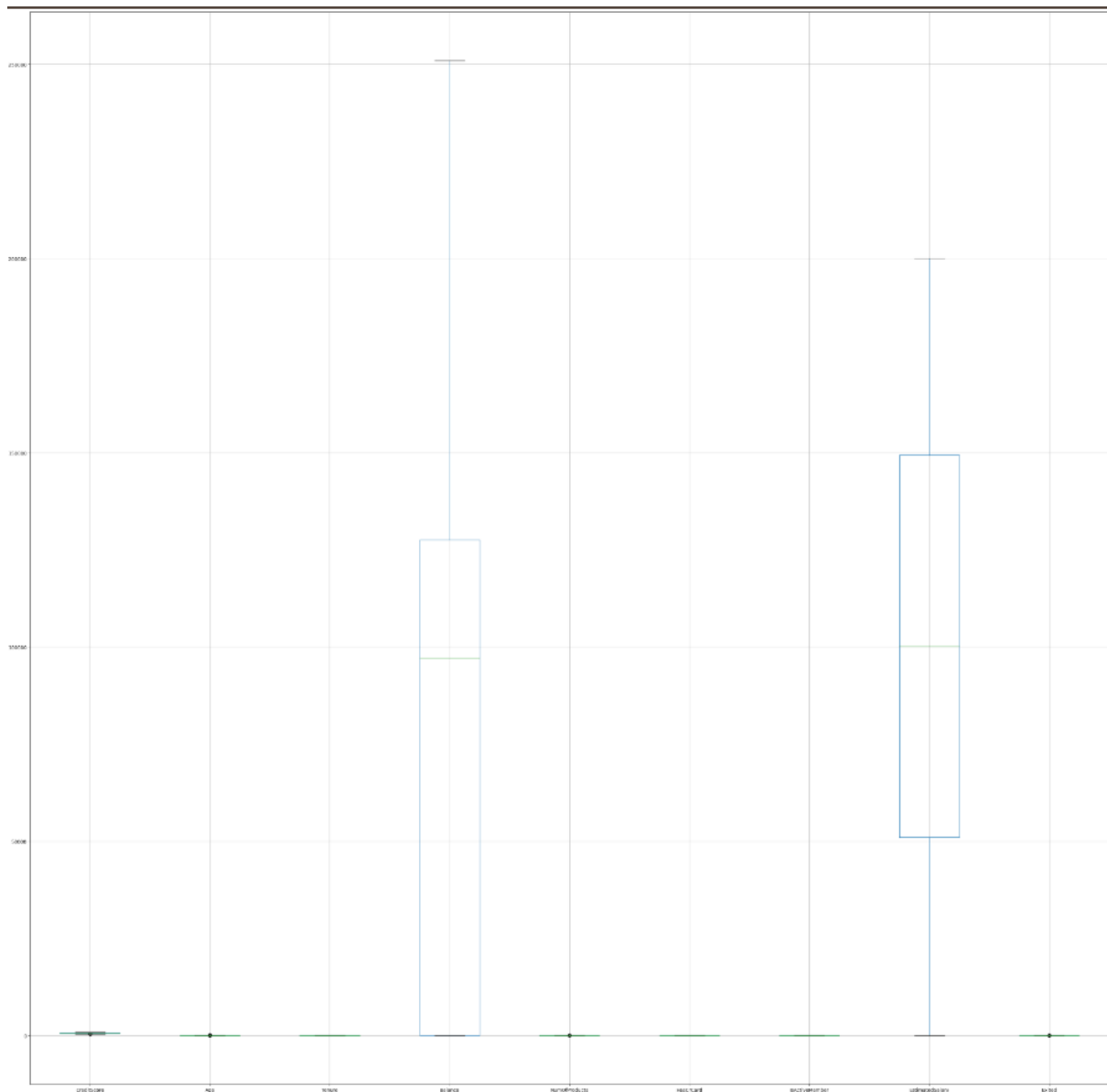
4.1.2.1 Boxplot Method

Box plots display the distribution of data based on minimum, first quartile (Q1), median, third quartile (Q3), and maximum. The 'figsize' parameter determines the size of the figure which is set to 40 x 40 to accommodate the box plots for all columns in the dataset.

```
# boxplot to find out outliers
fig, ax = plt.subplots(figsize=(40, 40))

# Plot the box plots for all columns
df2.boxplot(ax=ax)

# Show the plot
plt.show()
```



Statistical Analysis: The variables like CreditScore, Age, Tenure, NumOfProducts has a narrow IQR, with outliers detected at both ends of the distributions. Balance variable has a broader IQR, indicating significant variability among the customer balances. Many customers have zero balance and there are substantial outliers at the higher end, suggests that few customers have large balances. Likewise, EstimatedSalary has a broader IQR, which indicate wide variability in customer salaries.

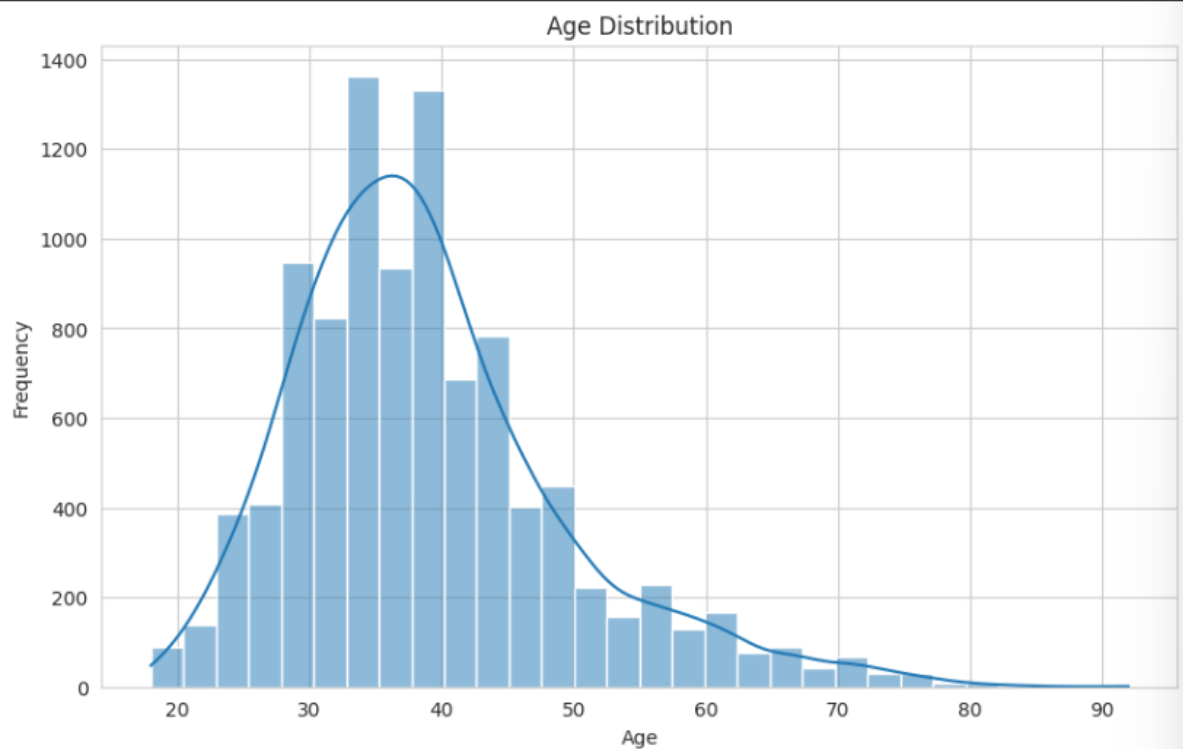
4.1.2.2 Z-Score Method

From the below code, z-scores are checked if they are greater than 2 or 3, indicating that they are more than 2 standard deviations away from the mean.

The code identifies the outliers in the dataset by counting the number of data points that are beyond 2&3 standard deviations from the mean for each numerical feature.

1. CreditScore and Age have a significant number of outliers beyond both 2 and 3 standard deviations, indicating these features have a substantial number of extreme values.
2. Balance and NumOfProducts also have a notable outliers beyond 2 standard deviations but fewer beyond 3 standard deviations.
3. Tenure and EstimatedSalary have no outliers beyond both 2 and 3 standard deviations, suggesting these features have a values that are more uniformly distributed around the mean.

```
plt.figure(figsize=(10, 6))
sns.histplot(df2['Age'], bins=30, kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



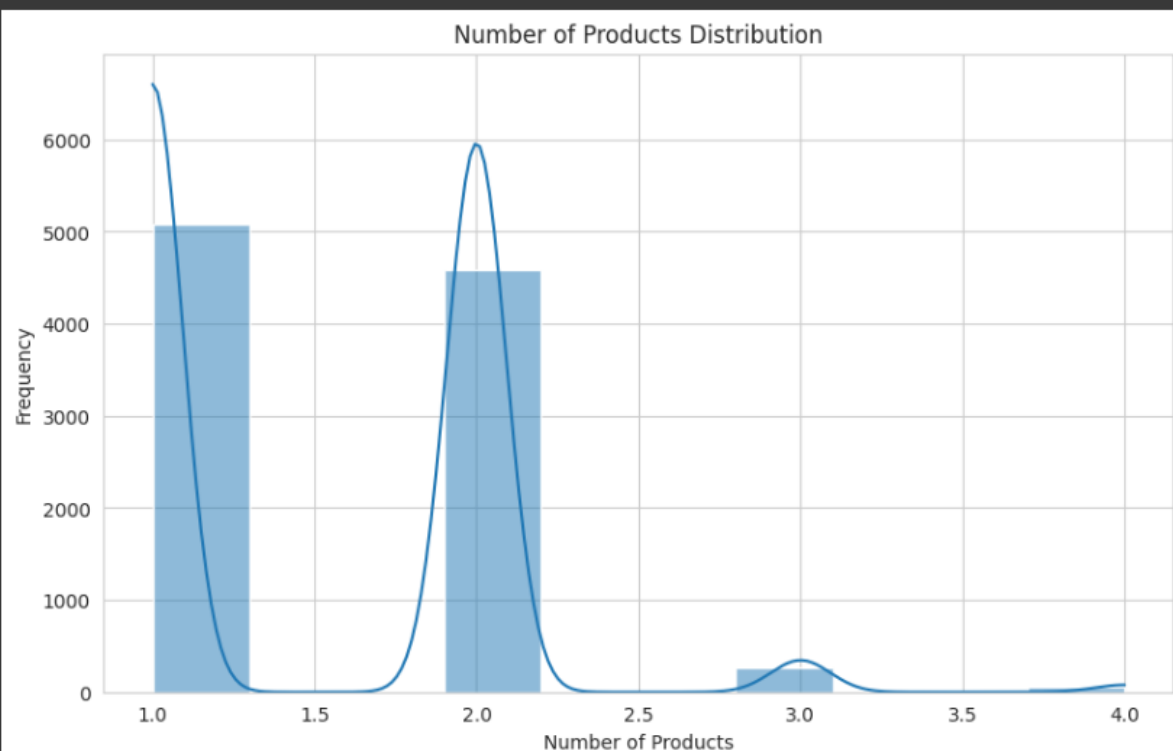
Age Distribution: The histogram shows that the majority of customers fall within the age range of 20 to 60, with a peak around the age of 40. There are fewer customers beyond the age of 60. Outliers beyond 3 standard deviations (133 customers) are significantly older or younger compared to the majority. These outliers might include very young customers or very senior customers.

Older Customers: Customers who are significantly older may represent a valuable segment due to their potentially higher financial stability and loyalty. Removing these outliers would disregard an important customer base.

Younger Customers: Very young customers may represent new market entrants or future long-term customers. They might have different banking needs and preferences that could provide insights into product offerings tailored to younger demographics.

While ages above 60 or below 20 might be statistical outliers, from a business perspective, they can represent important segments of the customer base, such as retirees or young adults. These age groups could have different financial needs and behaviours.

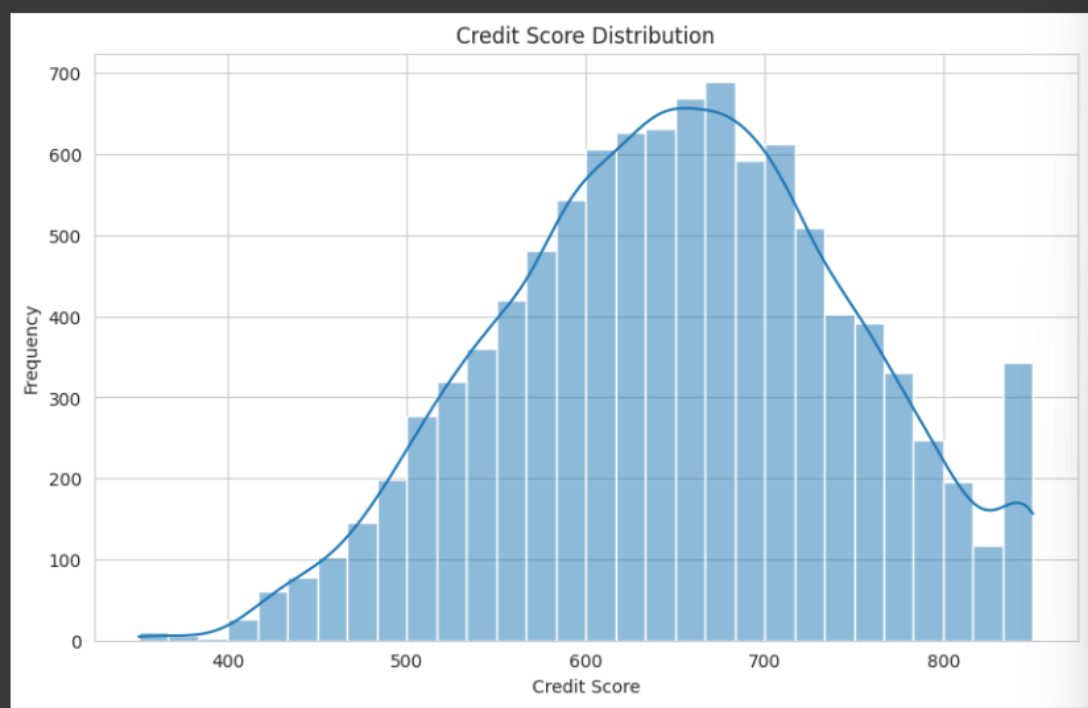
```
plt.figure(figsize=(10, 6))
sns.histplot(df2['NumOfProducts'], bins=10, kde=True)
plt.title('Number of Products Distribution')
plt.xlabel('Number of Products')
plt.ylabel('Frequency')
plt.show()
```



The distribution shows that most customers have 1 or 2 products. There are fewer customers with 3 or more products. Outliers beyond 3 standard deviations (60 customers) likely have an unusually high number of products, which might represent a very small segment of highly engaged or affluent customers.

Customers with a high number of products are typically more engaged and valuable to the bank. They might be receiving tailored services or packages that encourage them to adopt multiple products. Removing these outliers would ignore a critical segment that demonstrates high engagement and potential profitability. Customers with a very high number of products might be rare, but they represent highly engaged customers. This information is valuable for cross-selling and customer relationship management.

```
plt.figure(figsize=(10, 6))
sns.histplot(df2['CreditScore'], bins=30, kde=True)
plt.title('Credit Score Distribution')
plt.xlabel('Credit Score')
plt.ylabel('Frequency')
plt.show()
```



The credit score distribution is slightly skewed towards higher scores, with most customers having scores between 500 and 800. A few customers have scores beyond 800 or below 400. Outliers beyond 3 standard deviations (8 customers) include those with very high or very low credit scores.

High Credit Scores: Customers with exceptionally high credit scores often represent low risk and high profitability for the bank. They might be eligible for premium services and products.

Low Credit Scores: Customers with very low credit scores might be at higher risk but could also be targeted with specialized financial products aimed at credit rebuilding. Removing these outliers would eliminate insights into how to manage high-risk customers and leverage opportunities for offering tailored financial solutions.

CreditScore is typically ranged between 300 and 850. Outliers at the lower end (below 400) might represent high-risk customers, while scores above 800 indicate very low-risk customers. These extremes can be important for decision-making in lending or risk management rather than being treated as statistical anomalies.

Tenure: Customers with very short or very long tenures could be outliers statistically, but they represent new customers or loyal, long-term customers, respectively. Understanding these groups is crucial for customer retention strategies.

Balance: Extremely high balances might be outliers statistically, but from a business perspective, these customers could be very valuable. Similarly, customers with zero balance could represent a risk of churn or different account usage patterns that are important to understand.

EstimatedSalary: High salaries might be statistical outliers, but these customers could be key targets for premium services. Similarly, customers with very low salaries might need different financial products and services.

4.1.3 Visualization

Graphical representation that can uncover patterns, trends, and insights play an important role in the analysis and interpretation of data. In this project, various visualization techniques were employed to understand the distribution and relationships within the dataset. Histograms were used to explore the distribution of key features like which reveals the insights into customer demographics and behaviour. Count plots are also used in this study to examine the variables in relation to the target variable ‘Exited’, showing the churn rate across the different segments. Correlation matrices are visualised through heatmaps, helped identify the strength and direction of relationships between numerical features. The visualisation of decision tree offers an intuitive way to understand the decision making process of the model, while ROC curves illustrate the performance of the classification models.

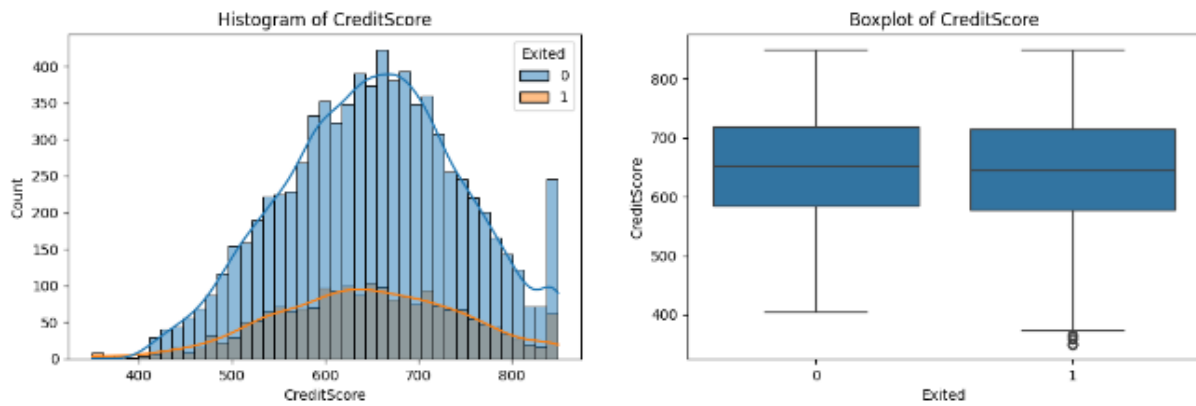
```
[12] # Set up the figure with subplots
fig, axes = plt.subplots(len(continuous_vars), 2, figsize=(12, len(continuous_vars) * 4))

for i, var in enumerate(continuous_vars):
    # Histogram
    sns.histplot(data=df2, x=var, hue='Exited', ax=axes[i, 0], kde=True)
    axes[i, 0].set_title(f'Histogram of {var}')

    # Boxplot
    sns.boxplot(data=df2, x='Exited', y=var, ax=axes[i, 1])
    axes[i, 1].set_title(f'Boxplot of {var}')

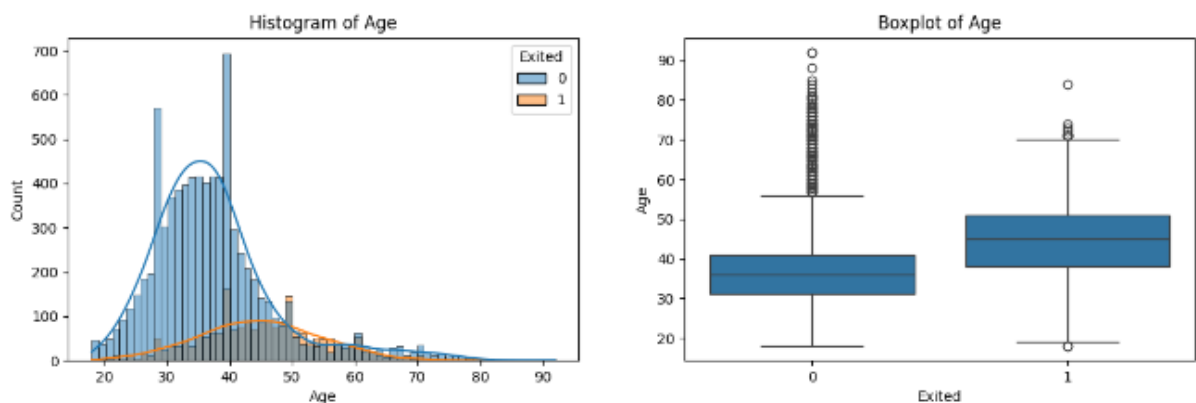
plt.tight_layout()
plt.show()
```

Each continuous variable gets a row in the figure, with the left column showing the histograms and boxplots. Each continuous variable gets a row in the figure, with the left column showing the histogram and the right column showing the box plot grouped by ‘Exited’ column. This allows for a detailed comparison of the distributions and potential relationships between continuous variables and the ‘Exited’ status.



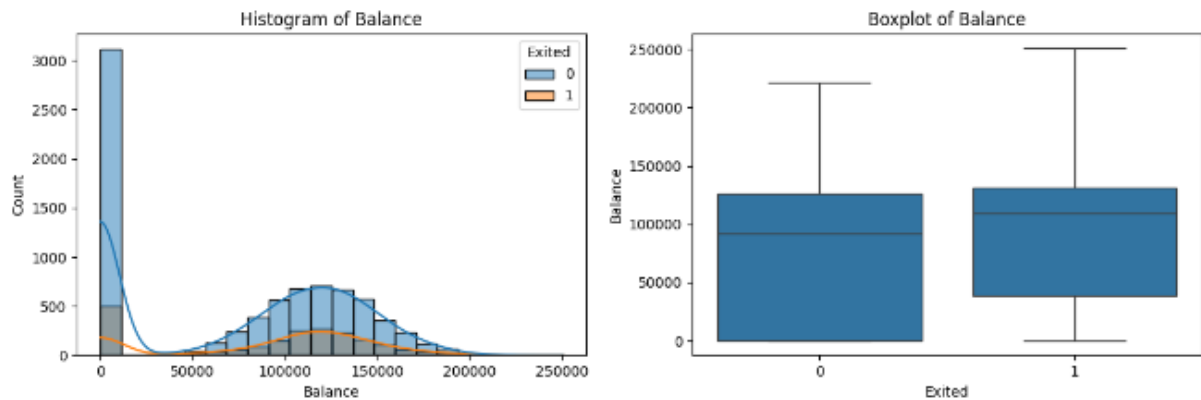
The distribution of credit scores shows a peak around 600-700. Customers who have exited (Exited=1) generally have lower credit scores compared to those who have not exited (Exited=0).

From the boxplot, The median credit scores is slightly higher for the customers who stayed. There are several outliers on the lower end for both exited and not exited customers, but more pronounced for the ones who exited.



The age distribution peaks around 30-40 years. A significant proportion of customers who exited are older (above 50 years) which suggests that age might influence churn.

From the boxplot, The median age is higher for the customers who exited. There are many outliers in the age distribution for both exited and non-exited customers, especially among older age groups.



Many customers have a balance close to zero. Customers with higher balances, above 100,000 are more likely to have exited.

From the boxplot, the median balance is higher for customers who have exited, but the overall distribution is similar for both groups. There are several high-value outliers in both the groups, indicating significant variability in account balances.



The estimated salary is uniformly distributed across the range. The exit rate does not appear to vary significantly across the range. This indicates that the salary might not be a strong predictor of churn.

For the boxplot, the median estimated salary is almost identical for both exited and non-exited customers. There are no significant outliers in the salary distribution and interquartile range is similar for both the groups.

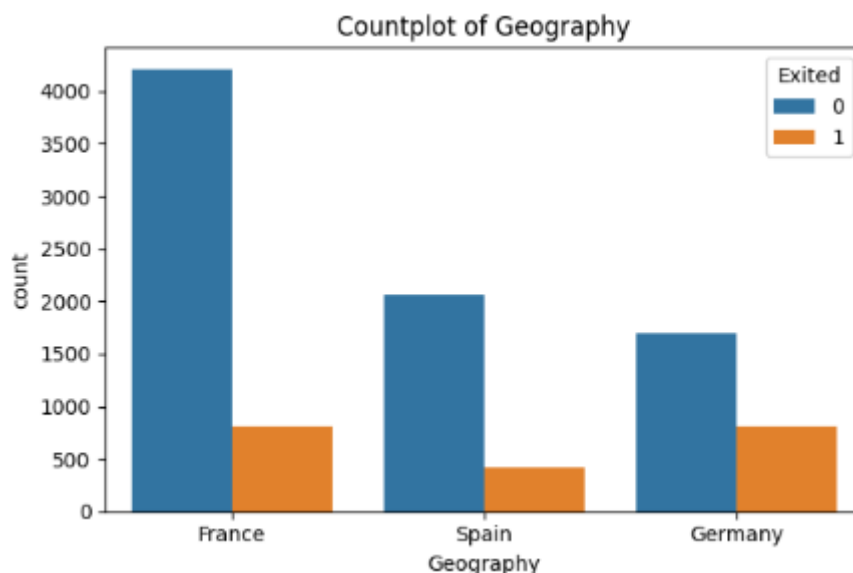
Visualization of the distribution of categorical variables in dataset using count plots.

```
# Set up the figure with subplots
fig, axes = plt.subplots(3, 2, figsize=(12, 12))

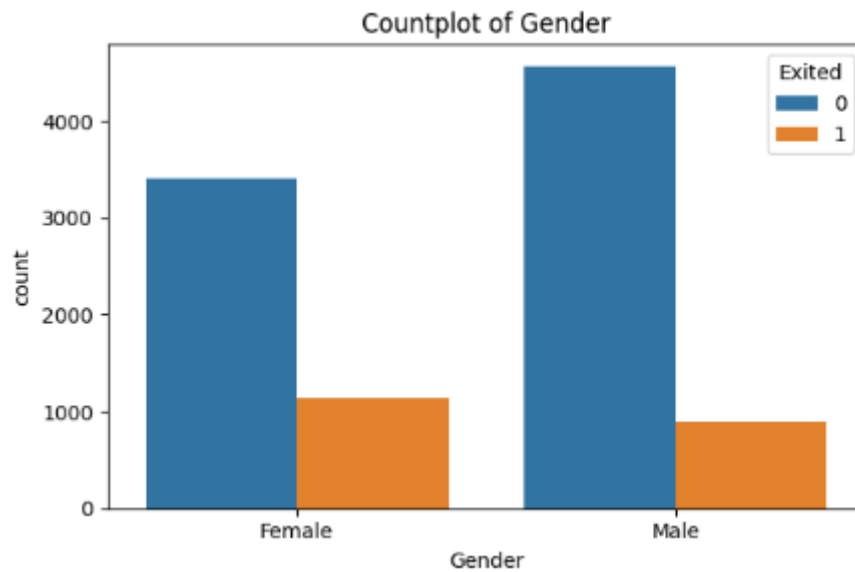
for i, var in enumerate(categorical_vars):
    row, col = divmod(i, 2) # Determine the row and column position
    sns.countplot(data=df2, x=var, hue='Exited', ax=axes[row, col])
    axes[row, col].set_title(f'Countplot of {var}')

plt.tight_layout()
plt.show()
```

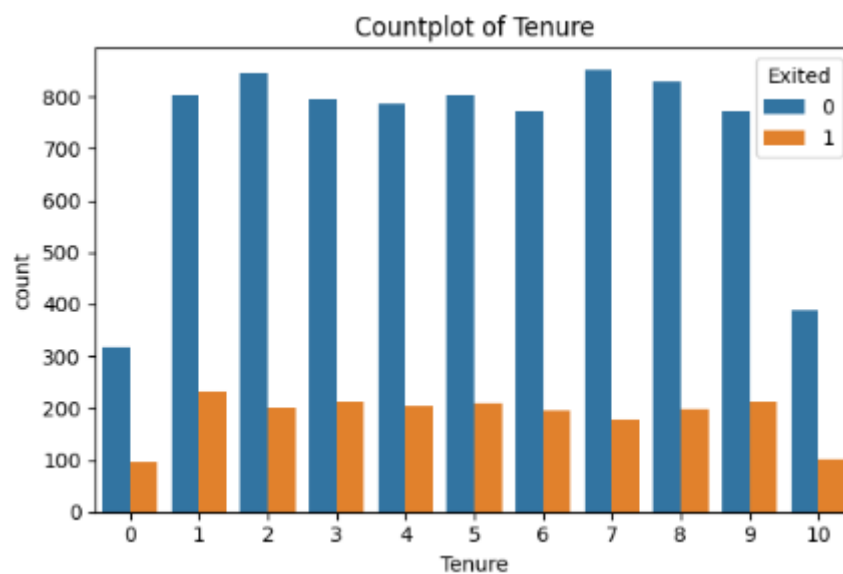
Each categorical variable gets a subplot in a 3 x 2 grid with the count plots showing the distribution of each variable and the 'Exited' status. This allows for the detailed comparison of the distributions and potential relationships between the categorical variables and the 'Exited' status that help to identify the patterns and insights related to customer churn.



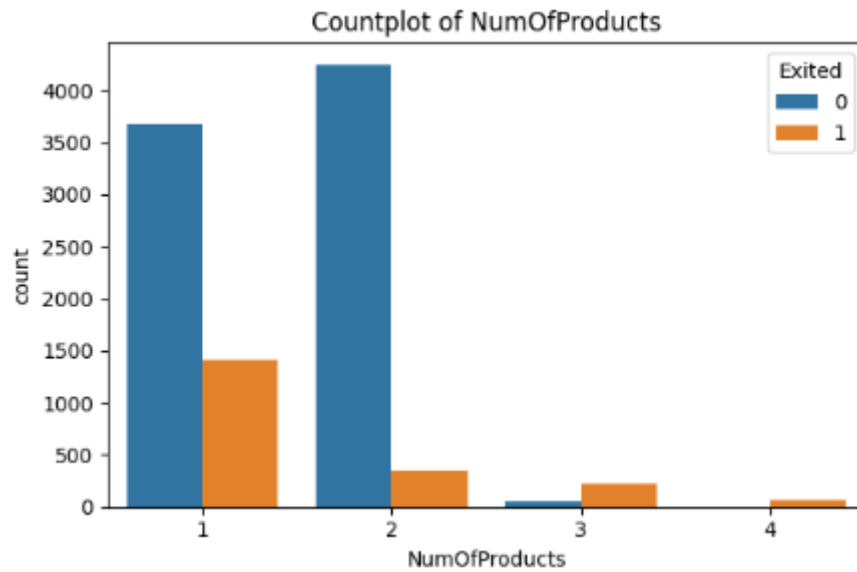
Most customers are from France, with a relatively smaller proportion exiting. Fewer customers compared to France, but the exit rate is similar. Germany has fewer customers than France but a higher proportion of exits, indicating a higher churn rate.



The count of female customers is lower than male customers, but the proportion of exits is higher. There are more male customers overall, with a lower proportion exiting compared to females.



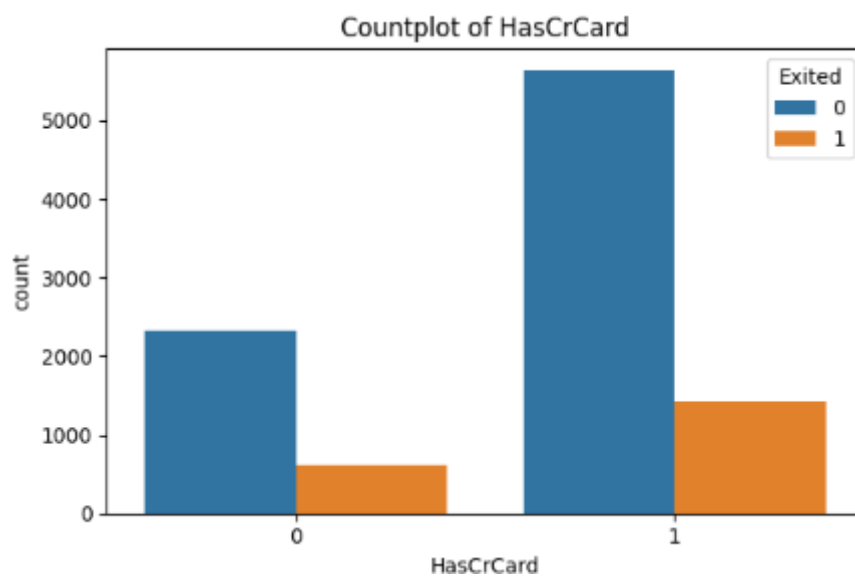
The distribution of tenure is fairly even, but exits are higher for customers with very short (0-1 years) and very long (9-10 years) tenures. This suggests that new customers and those with the longest relationships are more likely to churn.



1 Product: The majority of customers have only one product, with a notable number exiting.

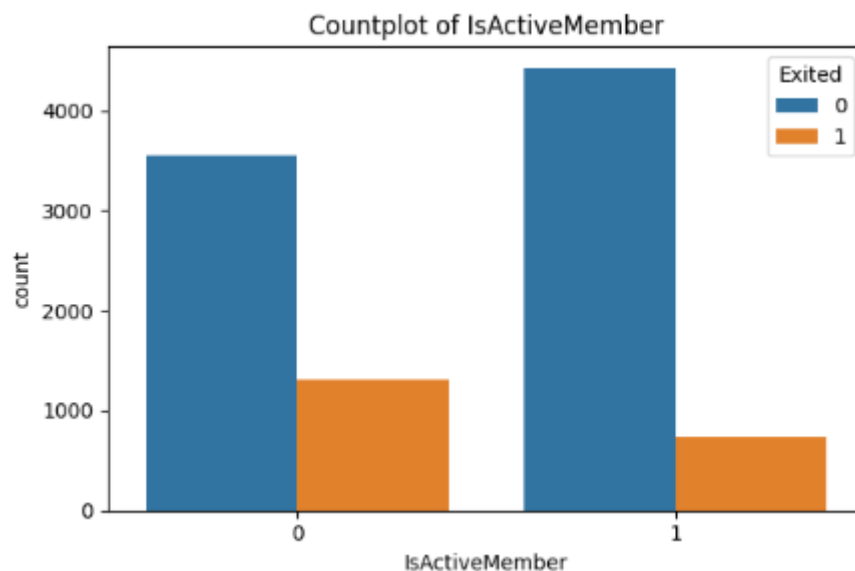
2 Products: A smaller number of customers have two products, but the exit rate is also significant.

3 or 4 Products: Very few customers have three or four products, with a low exit rate, indicating high engagement and lower churn.



No Credit Card: Customers without a credit card have a higher exit rate compared to those with a credit card.

Has Credit Card: The majority of customers have a credit card, and their exit rate is lower.



Not Active: Customers who are not active members have a higher exit rate.

Active Member: Active members are less likely to exit, indicating engagement is crucial for retention.

Conclusion:

Geography: German customers have a higher churn rate, suggesting that region-specific factors might influence customer retention.

Gender: Female customers are more likely to churn than male customers.

Tenure: Both very new and long-term customers show higher churn rates, possibly indicating onboarding issues and changing needs over time.

NumOfProducts: Customers with only one product are more likely to churn, while those with more products are more engaged and less likely to leave.

HasCrCard: Having a credit card is associated with lower churn, possibly due to higher engagement or satisfaction.

IsActiveMember: Active members are less likely to churn, highlighting the importance of customer engagement.

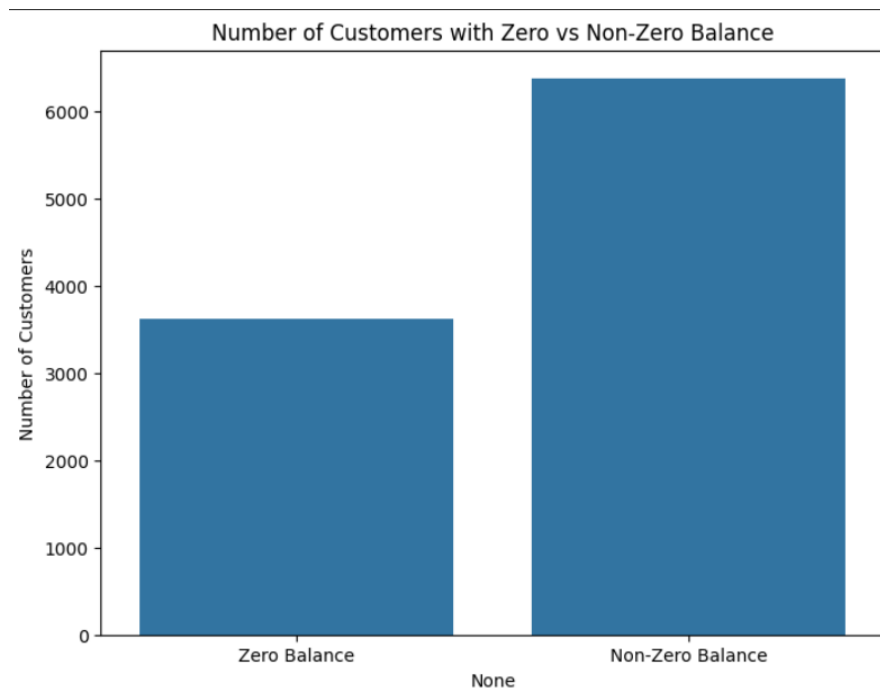
```
#Count people having 0 balance in their accounts.
zero_balance_count = df2[df2['Balance'] == 0].shape[0]
total_customers = df2.shape[0]
zero_balance_percentage = (zero_balance_count / total_customers) * 100

print(f"Number of customers with zero balance: {zero_balance_count}")
print(f"Percentage of customers with zero balance: {zero_balance_percentage:.2f}%")
```

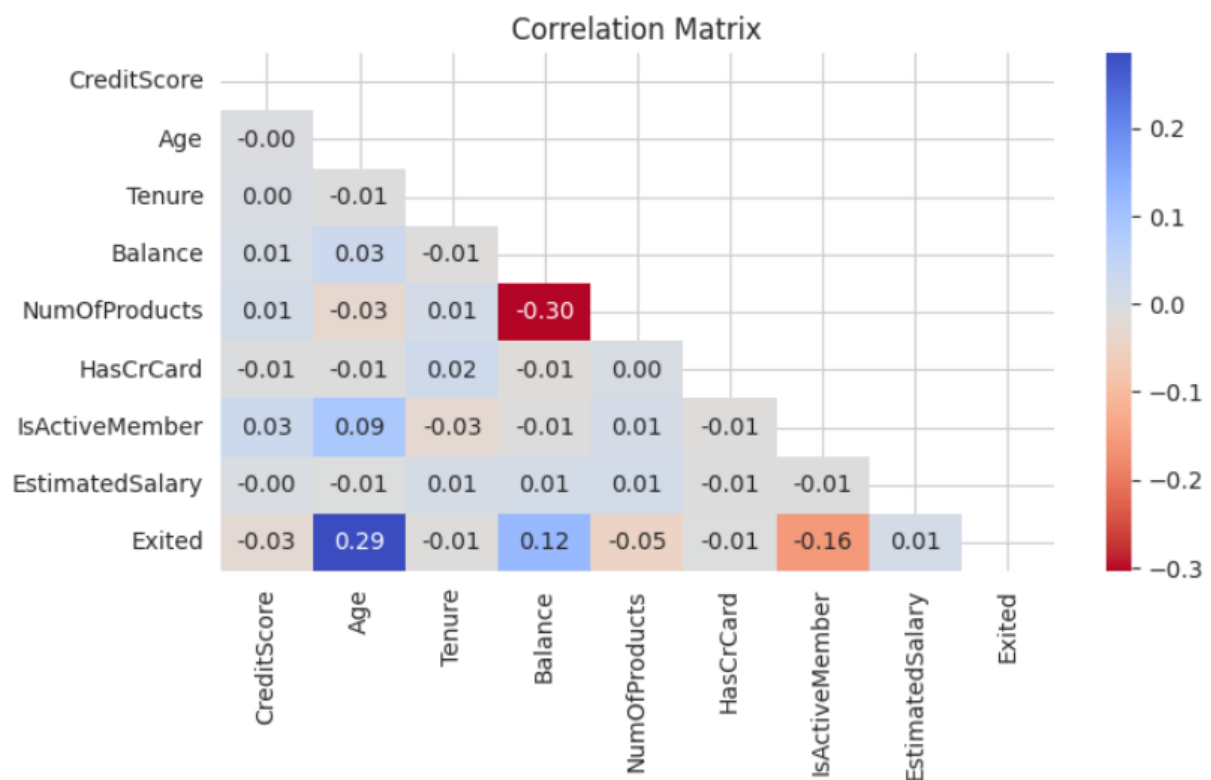
Number of customers with zero balance: 3617
Percentage of customers with zero balance: 36.17%

Customers with zero balance might be at a higher risk of churning. They may have already moved their funds into another bank or are planning to close their account. Identifying these customers can help in understanding the potential reasons for churn and developing targeted retention strategies. Analysing the proportion of customers with zero balance helps in segmenting the customer base. This segmentation can be useful in tailoring the marketing and retention efforts to different groups based on their account balances. By knowing how many customers have zero balance accounts, the bank can allocate the resources more efficiently. For example, more effort can be directed towards engaging these customers and understanding their needs to prevent churn.

From the analysis, it is found that 3617 customers had zero balance in their account.



a) Visualisation of the correlation matrix through heatmap



Analyzing the correlation matrix, we can make the following conclusions:

Age and Exited (0.29): A very weak positive correlation between a customer's age and the likelihood of churn is observed. This highlights that older customers may be slightly more likely to churn compared to younger customers.

Balance and NumOfProducts (-0.30): There is a moderate negative correlation between the number of products a customer has and their account balance. This indicates that customers with more products tend to have lower account balances.

Balance and Exited (0.12): There is a weak positive correlation between a customer's account balance and the likelihood of churn. This suggests that customers with higher account balances may be slightly more likely to churn.

IsActiveMember and Exited (-0.16): There is a weak negative correlation between whether a customer is an active member and the likelihood of churn. This indicates that inactive members may be slightly more likely to churn.

Remaining correlations close to 0: The remaining correlations being close to 0 indicate that there is no correlation between them. This does not mean that there is no relationship at all, as there could still be nonlinear relationships or interactions between variables that are not captured by the correlation coefficient.

```
z_scores = np.abs((X[numerical_features] - X[numerical_features].mean()) / X[numerical_features].std())
```

Z-scores standardize the data by centring it around zero and scaling it to have a standard deviation of one. This makes it easier to compare values across different features. It also helps in identifying the outliers. Typically, z-scores greater than 3 (or less than -3) are considered outliers because they lie more than three standard deviations away from the mean.

```

outliers_2_std = (z_scores > 2).sum()
print("Outliers beyond 2 standard deviations:")
print(outliers_2_std)

```

```

Outliers beyond 2 standard deviations:
CreditScore      500
Age              526
Tenure           0
Balance          30
NumOfProducts   326
EstimatedSalary  0
dtype: int64

```

```

outliers_3_std = (z_scores > 3).sum()
print("Outliers beyond 3 standard deviations:")
print(outliers_3_std)

```

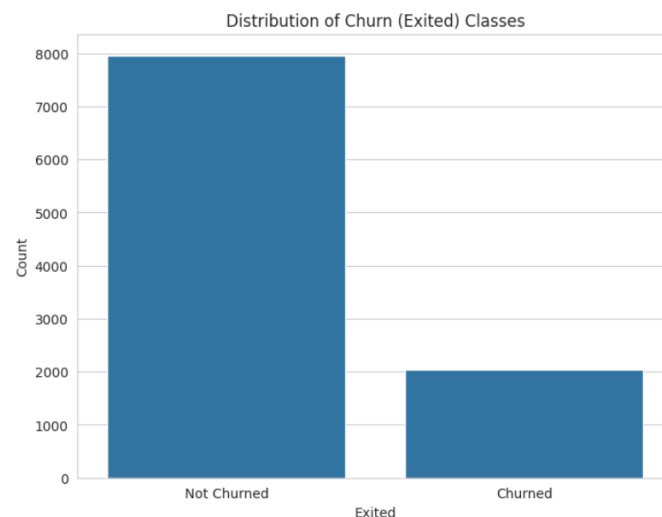
```

Outliers beyond 3 standard deviations:
CreditScore      8
Age             133
Tenure           0
Balance          0
NumOfProducts    60
EstimatedSalary  0
dtype: int64

```

4.1.4 Class Imbalance

Visualising the distribution of the target variable 'Exited' helps to identify the class imbalance. Class imbalance occurs when the number of instances of one class significantly outweighs the number of instances of other class. Checking for the class imbalance is important because it can affect the performance of machine learning models. Many models assume balanced classes and imbalance can cause to biased predictions favouring the majority class. (10 Techniques to Solve Imbalanced Classes in Machine Learning (Updated 2024), 2024)



The bar plot visualises the distribution of the target variable 'Exited' to check for class imbalance. The x-axis represents the 'Exited' status of the customers. It has two categories: Not churned (0): Customer who have not churned, and Churned(1) : Customers who have churned. The y-axis represents the count of customers in each category. The height of each bar shows the number of customers who either churned or not. The bar for not churned is significantly higher than that of churned indicating that the majority of the customers have not churned.

```
[ ] # Print the counts of each class to check for imbalance
    class_counts = y.value_counts()
    print("Class Counts:")
    print(class_counts)
    print("\nClass Percentages:")
    print(class_counts / len(y) * 100)
```

```
↔ Class Counts:
   Exited
0      7963
1      2037
Name: count, dtype: int64

Class Percentages:
   Exited
0      79.63
1      20.37
Name: count, dtype: float64
```

The above code has been used to calculate and print the counts and percentage of each class in the target variable. There are 7963 (79.63%) customers who did not churn and 2037 (20.37%) customers who churned. The dataset does not exhibit class imbalance based on 10% criterion. Both classes are well-represented. Class balance is a crucial aspect of preparing data for machine learning especially in classification tasks. In an imbalanced dataset, a model might achieve high accuracy by simply predicting the majority class, ignoring the minority class. This can be misleading because the model does not truly learn to differentiate between the classes.

With balanced classes, evaluation metrics such as precision, recall, and F1-score become a more reliable indicators of model performance. Balanced dataset provides a better foundation for training models, ensuring that the model learns to recognize the patterns for both classes. Since there was no imbalance, techniques like oversampling, under sampling or SMOTE has not been used in the analysis.

4.2 Summary

The initial data exploration involved previewing the dataset which include its key attributes like CustomerId, CreditScore, Geography, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, and Exited. Descriptive statistics were computed revealing several key insights. The average credit score is around 650 and a standard deviation indicating a moderate variability. Most customers are middle-aged, with a significant proportion falling between 30 and 40 years. The tenure distribution shows a mix of new and long-term customers, with a median tenure of 5 years. Account balances vary widely, with many customers having a balance close to zero and some with very high balances, indicating significant variability.

Statistically, the outliers were detected using box plots and z-scores identifying extreme values in features like CreditScore, Age, Balance, and NumOfProducts. However, a decision was made to now cap or remove the outliers as they were important from business perspective. Removing older customers (above 50 years) would mean losing an important customer base, rather than focussing on retention strategies. Various visualisations were used which provided insights into customer demographics and behaviors. For example, count plots showed that customers from Germany have a higher churn compared to those from Fance and Spain. Female customers are more likely to churn than the male customers.

Customers with one product are more likely to churn while those with more products are less likely to leave. Credit card holders have lesser chances to leave the bank and active members

are less likely to churn. All these factors highlight the importance of customer engagement. To ensure that the model training and evaluation is robust, class imbalance was checked revealing that 20.37% of customers have churned, while 79.63% have not. The data exploration phase provided crucial insights that guided feature engineering and model selection which is vital to make accurate and interpretable models.

5.0 Data Preparation and Feature Engineering

Data Preparation and feature engineering are critical processes in any data analysis or machine learning project. These steps ensure that the raw data is transformed into a suitable format for modelling. This allows for more accurate and reliable predictions.

5.1 Excluding Irrelevant column

This process involves cleaning and transforming raw data to make it suitable for analysis. As discussed earlier in the report, there were no missing values or inconsistent data. Only the columns RowNumber, CustomerId and Surname was dropped since they were unique identifiers for the customers and were not required for the analysis. These variables has no or little predictive power.

5.2 Feature Engineering

The following code has been used to create the feature 'X' by dropping the 'Exited' column which is the target variable. The purpose of this code is to prepare the data for machine learning modelling by defining the features and target variable and to provide an overview of the dataset structure. The categorical variables, such as 'Geography' and 'Gender' needs to be encoded into numerical values before they can be used in a machine learning model.


```
# Define the features and target variable
X = df2.drop(['Exited'], axis=1)
y = df2['Exited']

# Unique value counts for each column
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CreditScore            10000 non-null  int64
1   Geography              10000 non-null  object
2   Gender                 10000 non-null  object
3   Age                   10000 non-null  int64
4   Tenure                 10000 non-null  int64
5   Balance                10000 non-null  float64
6   NumOfProducts          10000 non-null  int64
7   HasCrCard              10000 non-null  int64
8   IsActiveMember         10000 non-null  int64
9   EstimatedSalary        10000 non-null  float64
dtypes: float64(2), int64(6), object(2)
memory usage: 781.4+ KB
```

5.3 Transformations

One-hot encoding- Categorical variables are features that represent categories or groups. Machine learning algorithms require numerical input, so it is necessary to convert the categorical features to a numerical format. The method used in the project is One-hot encoding. It is applied to the ‘Geography’ and ‘Gender’ columns. This method converts each category in these columns into a separate binary column (0 or 1).

In the code of `pd.get_dummies`, ‘`drop_first=True`’ parameter is used that helps in avoiding the dummy variable trap by dropping one category. This reduces multicollinearity. Encoding categorical variables is not only crucial for the purpose of modelling, but also important from business perspective. It allows the model to interpret and use categorical information effectively, leading to better predictions. For instance, understanding the geographical distribution of the customers can help in tailoring market strategies.

```
X = pd.get_dummies(X, columns=['Geography', 'Gender'], drop_first=True)
```

Normalizing the numerical features:

Normalization is the process of scaling the numerical variables to ensure that they have a standard range, usually with a mean of 0 and a standard deviation of 1. This step helps in improving the performance and convergence speed of the machine learning algorithms.

This project uses the method of 'StandardScaler' from 'sklearn.preprocessing' module to normalize the numerical features like 'CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'EstimatedSalary'. Normalizing the features ensure that all numerical inputs contribute equally to the model, preventing any single feature from disproportionately influencing the model's predictions. This aids in taking fair and balanced decision such as evaluating the credit score or customer tenures uniformly.

```
from sklearn.preprocessing import StandardScaler

# Initialize the StandardScaler
scaler = StandardScaler()

# Select the numerical columns for scaling
numerical_features = ['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'EstimatedSalary']

# Apply scaling
X[numerical_features] = scaler.fit_transform(X[numerical_features])
```

Check Skewness of the Variables:

```
# Calculate skewness for each feature in the training data
skewness = X_train.apply(lambda x: skew(x.dropna()))

# Display skewness for each feature
print(skewness)
```

CreditScore	-0.070494
Age	0.989627
Tenure	0.023312
Balance	-0.131164
NumOfProducts	0.751658
HasCrCard	-0.902053
IsActiveMember	-0.059455
EstimatedSalary	-0.010800
Geography_Germany	1.161755
Geography_Spain	1.152942
Gender_Male	-0.187678
dtype:	float64

Skewness is a measure of asymmetry of the distribution of the values in a dataset. The range for acceptable skewness is $\pm 2\%$.

The features EstimatedSalary and Tenure fall within or very close to the acceptable skew range. There is a slight deviation in Tenure, but it does not impact the model's performance, given its minimal skewness. Several features including Age, Balance, NumofProducts and Geography indicate that they exhibit the skewness out of $\pm 2\%$. However, no log transformation is applied as it is assumed that any impact of skewness on the model prediction is minimal.

6.0 Model Exploration

6.1 Modelling Approach

The modelling approach for this project involves the use of various machine learning algorithms to predict the target variable 'Exited' based on the given dataset. The primary goal is to identify the most effective model for the prediction task. The evaluation metrics that have been used are accuracy, precision, recall, F1-score, and ROC-AUC score. The models used in this section are Logistic Regression (Full, Forward, Backward, Stepwise), Decision tree, Random Forest(with and without reduced features), Gradient boosting (with and without reduced features). Each model has undergone hyperparameter tuning using GridSearchCV to find the best parameters that yield the highest cross-validation accuracy.

6.2 Model Technique #1: Full Logistic Regression

Full logistic regression involves using all available predictor variables in the model to predict the outcome. This model includes all the variables regardless of their statistical significance or

contribution to the model's performance. It is a comprehensive approach where no variable is excluded initially. It is simple to implement since it uses all the data and also captures all potential predictors. It can lead to overfitting and may include irrelevant variables which reduces model interpretability.

```
[ ] #Full Logistic Regression
    param_grid_logistic_regression = {
        'C': [0.01, 0.1, 1, 10, 100],
        'solver': ['newton-cg', 'lbfgs']
    }
```

This code is designed to identify the best logistic regression model by tuning the hyperparameters 'C' and 'solver'. 'C' is the inverse of regularization strength in logistic regression. A smaller value specifies stronger regularization. The algorithm to use for the optimization 'newton-cg' and 'lbfgs' are both solvers that handle multiclass classification well and are robust in logistic regression contexts. The model has been initialized with a maximum of 1000 iterations ensuring that the algorithm has a sufficient iterations to converge and a random seed of 42. Following this, hyperparameter tuning is done GridSearchCV which exhaustively search over the parameter grid. The number of cross-validation folds are 5 in this case. The model is fit on the training data to discover the best combination of hyperparameters.

```
print("Optimal parameters for Logistic Regression:", grid_search_logistic_regression.best_params_)
print("Highest cross-validation accuracy for Logistic Regression:", grid_search_logistic_regression.best_score_ * 100)

Optimal parameters for Logistic Regression: {'C': 0.1, 'solver': 'newton-cg'}
Highest cross-validation accuracy for Logistic Regression: 80.92857142857144
```

The optimal value for the regularization parameter 'c' was found to be 0.1. The 'c' parameter controls the trade-off between achieving a low error on the training data and minimizing the norm of the coefficients. A smaller value for 'c' prevents overfitting. The optimal solver is 'newton-cg'. The highest cross-validation accuracy achieved during the hyperparameter tuning process was approximately 80.93%. This score provides an estimate of how well the model is

expected to perform on the unseen data, which ensures that the chosen hyperparameters lead to a robust and generalizable model.

The 'predict' method was used to generate the class predictions based on the learned model parameters. The 'predict_proba' method of the logistic regression model returns the probability estimates for all classes. The variable 'pred_y_proba_logistic_regression' stores the predicted probability for the positive class (class 1) for the test set. Similar to the test set, class predictions are generated based on the learned model parameters as well. The predictions on the test set are used to evaluate the model's performance on unseen data. This helps in assessing the generalization ability of the model. The predictions are used to evaluate the model's performance on the data it was trained on. This helps to understand how well the model has learned from the training data and to detect any potential overfitting.

Model Evaluation

Model evaluation is important in the development of a machine learning model. It involves assessing how well the model performs on a give dataset. The goal is to ensure that the model generalizes well to the unseen data.

The evaluation metrics for confusion matrix are as follows which is true for all the models:

1. True Positives and True Negatives- This indicates the number of correct predictions made by the model.

2. If a model incorrectly predicts the positive class, then a false positive occurs. In this case, the model identifies an instance as belonging to the positive class when it actually belongs to the negative class.
3. when a model incorrectly predicts the negative class then a false negative happens. The model identifies an instance as belonging to the negative class when it actually belongs to a positive class.

Other metrics are:

1. Accuracy: The proportion of correctly predicted instances out of the total instances.
2. Precision: The proportion of positive predictions that are actually correct.
3. Recall: The proportion of actual positives that are correctly identified.
4. F1-Score: The harmonic mean of precision and recall.
5. ROC-AUC Score: measures the model's ability to distinguish between the classes. It is the area under the Receiver Operating Characteristic (ROC) curve.

```
#classification report for the test set
print("Full Logistic Regression")
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred_y_logistic_regression))
print("\nClassification Report:")
print(classification_report(y_test, pred_y_logistic_regression))
print("\nROC-AUC Score:", roc_auc_score(y_test, pred_y_proba_logistic_regression))
```

Full Logistic Regression

Confusion Matrix:

```
[[2328  88]
 [ 471 113]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.96	0.89	2416
1	0.56	0.19	0.29	584
accuracy			0.81	3000
macro avg	0.70	0.58	0.59	3000
weighted avg	0.78	0.81	0.78	3000

ROC-AUC Score: 0.7741065556109952

The confusion matrix is a table used to evaluate the performance of a classification model. It provides a summary of the prediction results on a classification problem.

Insights:

- ➔ 2328 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 88 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 471 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 113 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn. The Sapphire bank can focus on retention strategies on these customers, such as offering special deals or personalised services to encourage them to stay.

Performance metrics insights:

For class 0 (no churn), the precision is 0.83, (83% of the instances predicted as no churn were correct). The value for recall is 0.96, that means model correctly identified 96% of the actual no churn instances. The value of F1 score is 0.89.

For class 1 (churn), the precision is 0.56, which means that 56% of the instances predicted as churn were correct. The recall for class 1 is 0.19, that means only 19% of actual churn instances were correctly identified. The F1-score for class 1 is 0.28.

The overall accuracy of the model. is 0.81, signifying that 81% of all predictions made by the model were correct.

```
#classification report for the training set
print("Training Set Evaluation")
print("Confusion Matrix:")
print(confusion_matrix(y_train, pred_y_logistic_regression_train))
print("\nClassification Report:")
print(classification_report(y_train, pred_y_logistic_regression_train))
print("\nROC-AUC Score:", roc_auc_score(y_train, pred_y_proba_logistic_regression_train))
```

```
Training Set Evaluation
Confusion Matrix:
[[5357  190]
 [1147  306]]

Classification Report:
              precision    recall  f1-score   support

     0       0.82        0.97       0.89       5547
     1       0.62        0.21       0.31       1453

 accuracy          0.81        0.81        0.81       7000
 macro avg         0.72        0.59        0.60       7000
 weighted avg      0.78        0.81        0.77       7000

ROC-AUC Score: 0.7656321361186662
```

Insights:

- ➔ 5357 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 190 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 1147 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 306 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn. The Sapphire bank

can focus on retention strategies on these customers, such as offering special deals or personalised services to encourage them to stay.

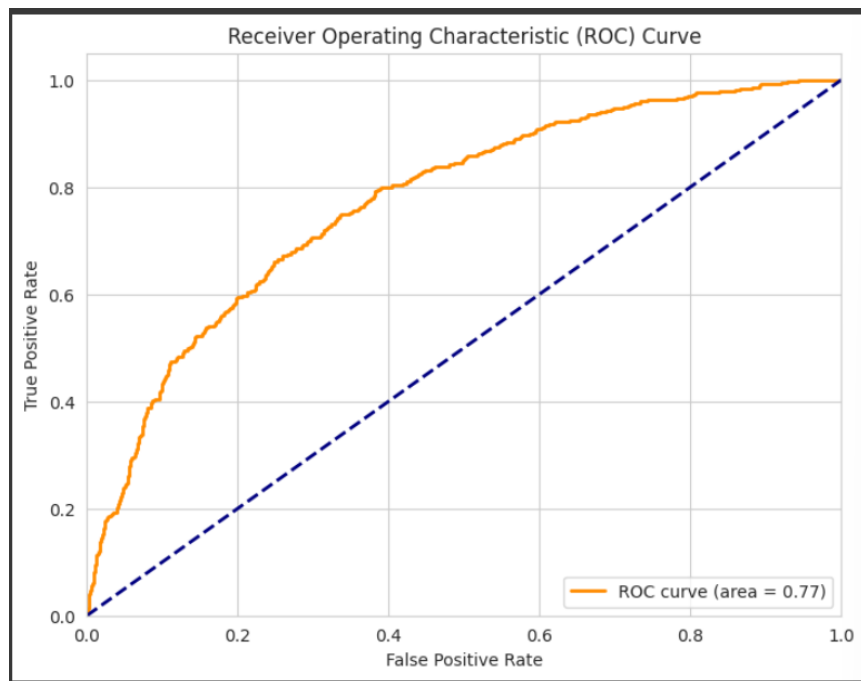
For class 0 (no churn), the precision is 0.82, (82% of the instances predicted as no churn were correct). The value for recall is 0.97, that means model correctly identified 97% of the actual no churn instances. The value of F1 score is 0.89.

For class 1 (churn), the precision is 0.62, which means that 62% of the instances predicted as churn were correct. The recall for class 1 is 0.21, that means only 21% of actual churn instances were correctly identified. The F1-score for class 1 is 0.31.

The overall accuracy of the model. is 0.81, signifying that 81% of all predictions made by the model were correct.

These metrics show that that while the model performs well in predicting the no churn class, it has significant room for improvement in predicting the churn class on the training set as well.

The ROC-AUC score for the training set is 0.7656, further indicating the model's ability to distinguish between the classes.



In the above graph, on the x-axis lies FPR (False Positive Rate) and on the y-axis lies TPR (True Positive Rate) also known as recall or sensitivity.

The blue dashed diagonal line represents a random classifier having an area under the curve (AUC) of 0. The orange curve is the ROC curve of the logistic regression model. The closer this curve follows the left-hand border and then the top border of the ROC space, the better the model's performance.

The value of ROC-AUC is 0.77, who shows that there is a 77% chance that the model will be able to distinguish between a randomly chosen positive instance and a randomly chosen negative instance. This value is good for this model, but there can be a room of improvement.

From a business perspective, the full logistic regression model's performance in predicting customer churn is moderately effective, with an overall accuracy of 81% and an ROC-AUC score of 0.77. The high precision (83%) and recall (96%) for predicting non-churning

customers suggest that the model is reliable in identifying customers who are likely to stay, enabling targeted retention strategies. However, the model's lower precision (56%) and recall (19%) for predicting churning customers indicate room for improvement in detecting potential churners. This insight highlights the need for more refined models or additional data features to better identify at-risk customers.

6.3 Model Technique #2: Forward Logistic Regression

Forward Logistic Regression is an approach where the model starts with no predictors and variables are added one at a time. At each step, the predictor that improves the model the most is added until no significant improvement is observed.

The parametric grid and hyperparametric tuning remains same for this case as well.

```
#Evaluate the model on the test set
print("Forward Selection Logistic Regression")
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred_y_logistic_regression_forward))
print("\nClassification Report:")
print(classification_report(y_test, pred_y_logistic_regression_forward))
print("\nROC-AUC Score:", roc_auc_score(y_test, pred_y_proba_logistic_regression_forward))
```

Forward Selection Logistic Regression
Confusion Matrix:
[[2322 94]
 [469 115]]

Classification Report:				
	precision	recall	f1-score	support
0	0.83	0.96	0.89	2416
1	0.55	0.20	0.29	584
accuracy			0.81	3000
macro avg	0.69	0.58	0.59	3000
weighted avg	0.78	0.81	0.77	3000

ROC-AUC Score: 0.7740966331760863

- ➔ 2322 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 94 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.

- ➔ 469 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 115 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn. The Sapphire bank can focus on retention strategies on these customers, such as offering special deals or personalised services to encourage them to stay.

Performance metrics insights:

For class 0 (no churn), the precision is 0.83, (83% of the instances predicted as no churn were correct). The value for recall is 0.96, that means model correctly identified 96% of the actual no churn instances. The value of F1 score is 0.89.

For class 1 (churn), the precision is 0.55, which means that 55% of the instances predicted as churn were correct. The recall for class 1 is 0.20, that means only 20% of actual churn instances were correctly identified. The F1-score for class 1 is 0.29 with ROC-AUC score of 0.77 approximately.

```
#evaluate the model on the training set
print("Forward Selection Logistic Regression - Training Set")
print("Confusion Matrix:")
print(confusion_matrix(y_train, pred_y_logistic_regression_forward_train))
print("\nClassification Report:")
print(classification_report(y_train, pred_y_logistic_regression_forward_train))
print("\nROC-AUC Score:", roc_auc_score(y_train, pred_y_proba_logistic_regression_forward_train))
```

```
Forward Selection Logistic Regression - Training Set
Confusion Matrix:
[[5350  197]
 [1129  324]]

Classification Report:
              precision    recall  f1-score   support

     0       0.83       0.96       0.89       5547
     1       0.62       0.22       0.33       1453

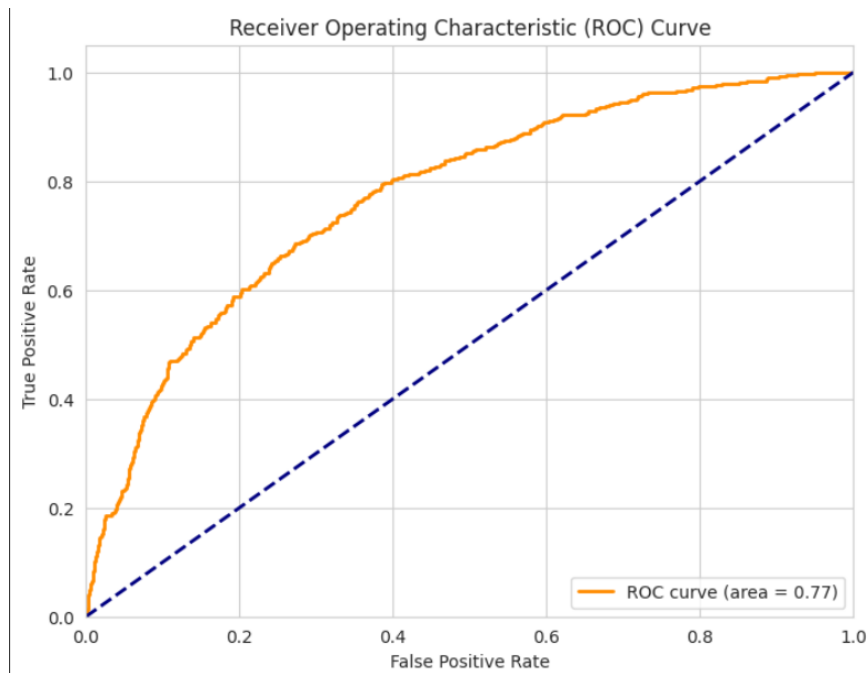
   accuracy       0.81       0.81       0.81       7000
  macro avg       0.72       0.59       0.61       7000
weighted avg       0.78       0.81       0.77       7000

ROC-AUC Score: 0.7649853451534908
```

- ➔ 5350 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 197 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 1129 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 324 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn.

For class 0 (no churn), the precision is 0.83, (83% of the instances predicted as no churn were correct). The value for recall is 0.96, that means model correctly identified 96% of the actual no churn instances. The value of F1 score is 0.89.

For class 1 (churn), the precision is 0.62, which means that 62% of the instances predicted as churn were correct. The recall for class 1 is 0.22, that means only 22% of actual churn instances were correctly identified. The F1-score for class 1 is 0.33 with ROC-AUC score of 0.76 approximately. The overall accuracy of the model is 81%.



The ROC curve for the Forward Selection Logistic Regression model shows the trade-off between the true positive rate and the false positive rate. The orange line represents the model's performance, with an AUC of 0.77. This indicates moderate effectiveness in distinguishing between customers who will churn and those who will not. An AUC of 1 would be perfect, while an AUC of 0.5 would indicate no discrimination ability. Therefore, this model performs better than random guessing but has room for improvement.

6.4 Model Technique #3: Backward Logistic Regression

Backward logistic regression starts with the full model, that includes all the predictors and removes the least significant predictors one at a time. The removal continues until all remaining predictors are statistically significant.

```
#Classification report for the test set
print("Backward Selection Logistic Regression")
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred_y_logistic_regression_backward))
print("\nClassification Report:")
print(classification_report(y_test, pred_y_logistic_regression_backward))
print("\nROC-AUC Score:", roc_auc_score(y_test, pred_y_proba_logistic_regression_backward))

Backward Selection Logistic Regression
Confusion Matrix:
[[2322  94]
 [ 469 115]]

Classification Report:
              precision    recall  f1-score   support

     0       0.83       0.96       0.89       2416
     1       0.55       0.20       0.29        584

 accuracy          0.81          0.81          0.81       3000
 macro avg          0.69          0.58          0.59       3000
 weighted avg       0.78          0.81          0.77       3000

ROC-AUC Score: 0.7740966331760863
```

- ➔ 2322 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 94 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 469 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 115 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn.

For class 0 (no churn), the precision is 0.83, (83% of the instances predicted as no churn were correct). The value for recall is 0.96, that means model correctly identified 96% of the actual no churn instances. The value of F1 score is 0.89.

For class 1 (churn), the precision is 0.55, which means that 55% of the instances predicted as churn were correct. The recall for class 1 is 0.20, that means only 20% of actual churn instances were correctly identified. The F1-score for class 1 is 0.29 with ROC-AUC score of 0.77 approximately. The overall accuracy of the model is 81%.

```
#classification report on training set
print("Backward Selection Logistic Regression - Training Set")
print("Confusion Matrix:")
print(confusion_matrix(y_train, pred_y_logistic_regression_backward_train))
print("\nClassification Report:")
print(classification_report(y_train, pred_y_logistic_regression_backward_train))
print("\nROC-AUC Score:", roc_auc_score(y_train, pred_y_proba_logistic_regression_backward_train))
```

Backward Selection Logistic Regression - Training Set
Confusion Matrix:
[[5350 197]
 [1129 324]]

Classification Report:

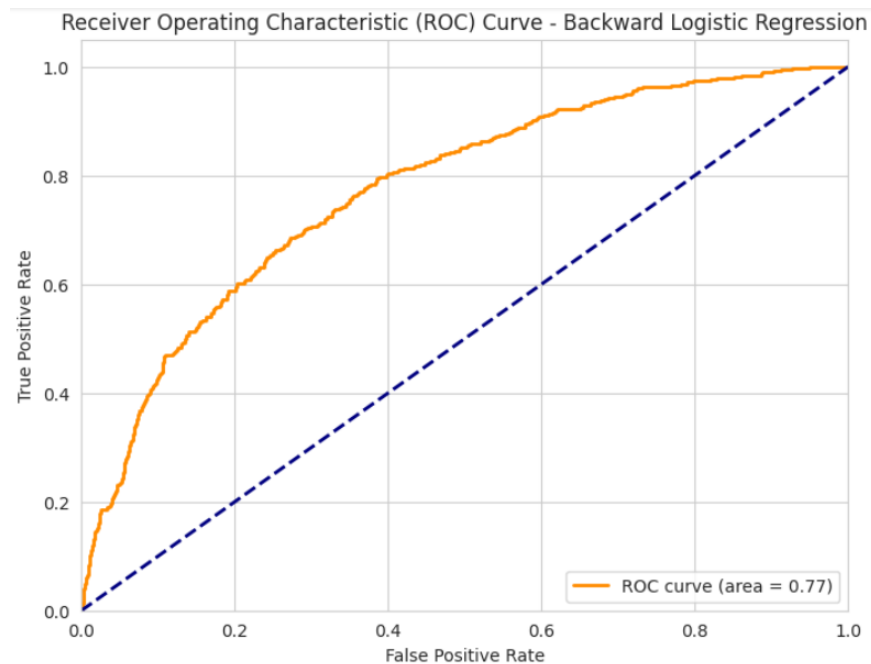
	precision	recall	f1-score	support
0	0.83	0.96	0.89	5547
1	0.62	0.22	0.33	1453
accuracy			0.81	7000
macro avg	0.72	0.59	0.61	7000
weighted avg	0.78	0.81	0.77	7000

ROC-AUC Score: 0.7649853451534908

- ➔ 5350 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 197 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 1129 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 324 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn.

For class 0 (no churn), the precision is 0.83, (83% of the instances predicted as no churn were correct). The value for recall is 0.96, that means model correctly identified 96% of the actual no churn instances. The value of F1 score is 0.89.

For class 1 (churn), the precision is 0.62, which means that 62% of the instances predicted as churn were correct. The recall for class 1 is 0.22, that means only 22% of actual churn instances were correctly identified. The F1-score for class 1 is 0.33 with ROC-AUC score of 0.76 approximately. The overall accuracy of the model is 81%.



The ROC curve for the backward logistic regression model illustrates the performance of the model in distinguishing between the churn and no churn classes. The curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) across different threshold settings. The area under the curve (AUC) is 0.77, indicating a good level of separability, as a value of 0.5 would signify no discriminative ability (the model is guessing randomly) and a value of 1 would indicate perfect discrimination. The curve being above the diagonal line suggests that the model has a better than random chance of correctly classifying churn versus no churn instances. An AUC of 0.77 means the model has a 77% chance of distinguishing between positive and negative classes, which is quite reasonable for this application.

6.5 Model Technique #4: Stepwise Logistic Regression

Stepwise logistic regression combines elements of both forward and backward selection. At each step, the method considers adding or removing predictors based on specific criterion.

```
#classification report for the test set
print("Stepwise Selection Logistic Regression")
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred_y_logistic_regression_stepwise))
print("\nClassification Report:")
print(classification_report(y_test, pred_y_logistic_regression_stepwise))
print("\nROC-AUC Score:", roc_auc_score(y_test, pred_y_proba_logistic_regression_stepwise))
```

Stepwise Selection Logistic Regression
Confusion Matrix:
[[2322 94]
 [469 115]]

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.96	0.89	2416
1	0.55	0.20	0.29	584
accuracy			0.81	3000
macro avg	0.69	0.58	0.59	3000
weighted avg	0.78	0.81	0.77	3000

ROC-AUC Score: 0.7740966331760863

- ➔ 2322 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 94 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 469 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 115 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn.

For class 0 (no churn), the precision is 0.83, (83% of the instances predicted as no churn were correct). The value for recall is 0.96, that means model correctly identified 96% of the actual no churn instances. The value of F1 score is 0.89.

For class 1 (churn), the precision is 0.55, which means that 55% of the instances predicted as churn were correct. The recall for class 1 is 0.20, that means only 20% of actual churn instances were correctly identified. The F1-score for class 1 is 0.29 with ROC-AUC score of 0.77 approximately. The overall accuracy of the model is 81%.

```
#classification report on training set
print("Stepwise Selection Logistic Regression - Training Set")
print("Confusion Matrix:")
print(confusion_matrix(y_train, pred_y_logistic_regression_stepwise_train))
print("\nClassification Report:")
print(classification_report(y_train, pred_y_logistic_regression_stepwise_train))
print("\nROC-AUC Score:", roc_auc_score(y_train, pred_y_proba_logistic_regression_stepwise_train))
```

Stepwise Selection Logistic Regression - Training Set
Confusion Matrix:
[[5350 197]
[1129 324]]

Classification Report:

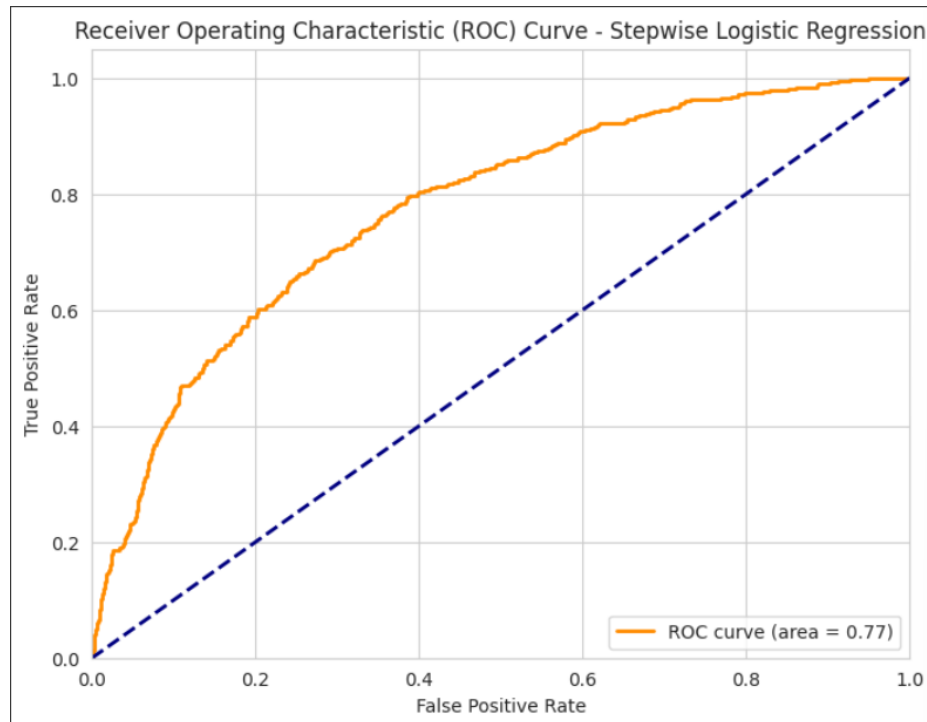
	precision	recall	f1-score	support
0	0.83	0.96	0.89	5547
1	0.62	0.22	0.33	1453
accuracy			0.81	7000
macro avg	0.72	0.59	0.61	7000
weighted avg	0.78	0.81	0.77	7000

ROC-AUC Score: 0.7649853451534908

- ➔ 5350 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 197 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 1129 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 324 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn.

For class 0 (no churn), the precision is 0.83, (83% of the instances predicted as no churn were correct). The value for recall is 0.96, that means model correctly identified 96% of the actual no churn instances. The value of F1 score is 0.89.

For class 1 (churn), the precision is 0.62, which means that 62% of the instances predicted as churn were correct. The recall for class 1 is 0.22, that means only 22% of actual churn instances were correctly identified. The F1-score for class 1 is 0.33 with ROC-AUC score of 0.76 approximately. The overall accuracy of the model is 81%.



The ROC curve for the stepwise logistic regression model displays the trade-off between the true positive rate and the false positive rate. With an area under the curve (AUC) of 0.77, the model shows a good capability to differentiate between positive and negative instances. An AUC of 0.77 indicates that the model correctly distinguishes between a randomly chosen positive and negative instance 77% of the time. The curve's position above the diagonal line (representing random guessing) confirms that the model performs better than random classification.

All logistic regression models showed strong performance in predicting non-churn customers but faced challenges in accurately identifying churn customers. The consistent ROC-AUC score of 0.77 across models indicates that they have a comparable ability to differentiate between churn and non-churn instances. For business, these models can be useful in reducing false positives for non-churn customers, but further refinement may be needed to improve churn prediction accuracy.

6.6 Model Technique #5: Decision Tree model

A decision tree is a supervised learning algorithm used for both classification and regression. It is a tree-like structure where each internal node represents a ‘decision’ based on the value of a feature, each branch represents the outcome of that decision and each leaf node represents a final output.

The next technique that is used to model the data is decision trees. It starts with defining the hyperparametric grid, where the maximum depth of the tree is 3,6,7 and 10. The parameter also defines the minimum number of samples required. Possible values are 2 and 10. The possible values for the parameter that specifies the minimum number of samples required to be a leaf node is 1 and 4. The method that is used to perform hyperparameter tuning is GridSearchCV.

```
#classification report for the test set
print("Decision Tree")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_dt))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_dt))
print("\nROC-AUC Score:", roc_auc_score(y_test, y_pred_proba_dt))
```

Decision Tree
Confusion Matrix:
[[2308 108]
 [309 275]]

Classification Report:

	precision	recall	f1-score	support
0	0.88	0.96	0.92	2416
1	0.72	0.47	0.57	584
accuracy			0.86	3000
macro avg	0.80	0.71	0.74	3000
weighted avg	0.85	0.86	0.85	3000

ROC-AUC Score: 0.834492013857389

The insight from the confusion matrix is as follows:

- ➔ 2308 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 108 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 309 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 275 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn. The Sapphire bank can focus on retention strategies on these customers, such as offering special deals or personalised services to encourage them to stay.

For the test set, the Decision Tree model demonstrates a strong ability to correctly identify non-churn customers, with a precision of 0.88 and a recall of 0.96 for class 0. The F1-score for class 0 is 0.92, indicating a high level of accuracy in predicting non-churn instances. However, the model's performance for identifying churn customers is less robust, with a precision of 0.72 and a recall of 0.47, resulting in an F1-score of 0.57. The overall accuracy of the model on the test set is 0.86. The macro averages for precision, recall, and F1-score are 0.80, 0.71, and 0.74, respectively, while the weighted averages are 0.85, 0.86, and 0.85. The ROC-AUC score of 0.834 indicates that the model has a good overall ability to distinguish between churn and non-churn customers.

```
#classification report for the training set
print("Decision Tree - Training Set")
print("Confusion Matrix:")
print(confusion_matrix(y_train, y_pred_dt_train))
print("\nClassification Report:")
print(classification_report(y_train, y_pred_dt_train))
print("\nROC-AUC Score:", roc_auc_score(y_train, y_pred_proba_dt_train))
```

Decision Tree - Training Set
Confusion Matrix:
[[5337 210]
 [680 773]]

Classification Report:

	precision	recall	f1-score	support
0	0.89	0.96	0.92	5547
1	0.79	0.53	0.63	1453
accuracy			0.87	7000
macro avg	0.84	0.75	0.78	7000
weighted avg	0.87	0.87	0.86	7000

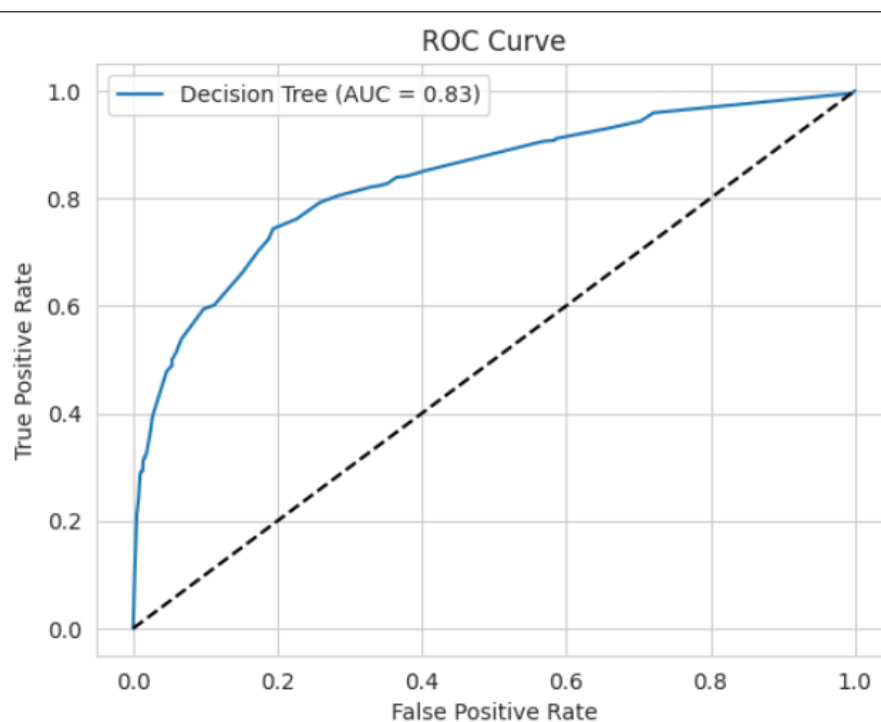
ROC-AUC Score: 0.8817478021452418

The insight from the confusion matrix is as follows:

- ➔ 5337 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 210 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 680 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 773 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn. The Sapphire bank can focus on retention strategies on these customers, such as offering special deals or personalised services to encourage them to stay.

On the training set, the Decision Tree model maintains high performance, with a precision of 0.89 and a recall of 0.96 for class 0, resulting in an F1-score of 0.92. For class 1, the precision is 0.79, and the recall is 0.53, with an F1-score of 0.63. The model's accuracy on

the training set is 0.87, slightly higher than on the test set. The macro averages for precision, recall, and F1-score are 0.84, 0.75, and 0.78, respectively, while the weighted averages are 0.87, 0.87, and 0.86. The ROC-AUC score of 0.881 indicates even stronger performance on the training set compared to the test set, reflecting the model's ability to effectively distinguish between churn and non-churn customers in the training data.



The ROC curve shows that the Decision Tree model has a good true positive rate for a range of false positive rates. For example, at a false positive rate of 0.2, the true positive rate is approximately 0.8. This means that when the model incorrectly identifies 20% of negatives as positives, it correctly identifies 80% of positives. Customer Churn Prediction: The ROC curve and AUC score provide a clear indication that the Decision Tree model is effective at predicting customer churn. With an AUC of 0.83, the model can be used to identify customers who are at risk of churning with a reasonable degree of accuracy.

6.7 Model Technique #6: Random Forest model

Random Forest is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees during training and outputting the mode of the classes. It combines multiple decision trees to create a stronger model. Random forest combines multiple decision trees to create a stronger model. (What is random forest?, n.d.)

Before building the model, parameter grid has been defined for Random Forest, where 'n_estimators' are the number of trees in the forest. The grid search will try 50, 100 and 200 trees. The maximum depth of tree is taken as none, 10 and 20. It means the nodes are expanded until all the leaves are pure or until all leaves contain less than 'min_sample_split' which is the minimum number of samples required to split an internal node. The grid search is performed on the training data and it is split into 5 folds. The model is then trained on 4 folds and validated on the 5th fold for each combination of hyperparameters.

```
#classification report for the test set
print("Random Forest")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_rf))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_rf))
print("\nROC-AUC Score:", roc_auc_score(y_test, y_pred_proba_rf))
```

Random Forest
Confusion Matrix:
[[2339 77]
 [318 266]]

Classification Report:

	precision	recall	f1-score	support
0	0.88	0.97	0.92	2416
1	0.78	0.46	0.57	584
accuracy			0.87	3000
macro avg	0.83	0.71	0.75	3000
weighted avg	0.86	0.87	0.85	3000

ROC-AUC Score: 0.8635913260909008

The insight from the confusion matrix is as follows:

- ➔ 2339 instances were true negatives where the model correctly predicted the negative class (0). These are the customers who were correctly identified as not likely to churn.
- ➔ 77 instances were false positive where the model incorrectly predicted the positive class (1) for actual negatives (0). These are customers who were incorrectly predicted to churn but will actually stay.
- ➔ 318 instances were false negative where the model incorrectly predicted the negative class (0) for actual positives (1). These are the customers who were incorrectly predicted to stay but will actually churn.
- ➔ 266 instances were true positives where the model correctly predicted the positive class (1). These are the customers who were correctly predicted to churn.

For class 0 (no churn), the precision is 0.88, (88% of the instances predicted as no churn were correct). The value for recall is 0.97, that means model correctly identified 97% of the actual no churn instances. The value of F1 score is 0.92.

For class 1 (churn), the precision is 0.78, which means that 78% of the instances predicted as churn were correct. The recall for class 1 is 0.46, that means only 46% of actual churn instances were correctly identified. The F1-score for class 1 is 0.57 with ROC-AUC score of 0.86 approximately. The overall accuracy of the model is 87%.

```
#classification report for the training set
print("Random Forest")
print("Confusion Matrix:")
print(confusion_matrix(y_train, y_pred_rf_train))
print("\nClassification Report:")
print(classification_report(y_train, y_pred_rf_train))
print("\nROC-AUC Score:", roc_auc_score(y_train, y_pred_proba_rf_train))

Random Forest
Confusion Matrix:
[[5474  73]
 [ 564 889]]

Classification Report:
              precision    recall  f1-score   support

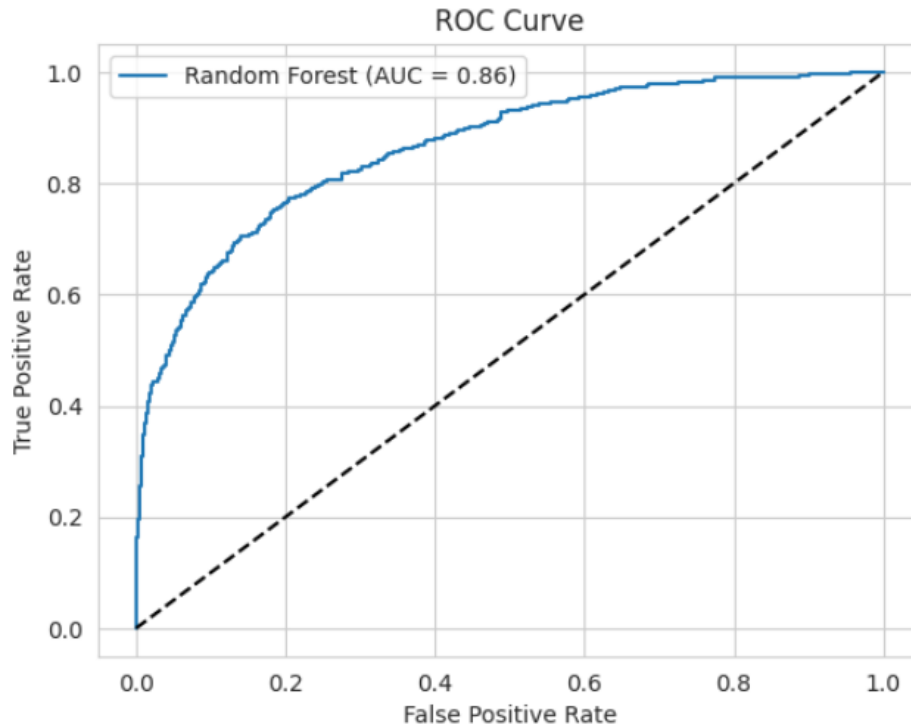
     0       0.91       0.99       0.95       5547
     1       0.92       0.61       0.74       1453

   accuracy       0.91
  macro avg       0.92       0.80       0.84       7000
weighted avg       0.91       0.91       0.90       7000

ROC-AUC Score: 0.9783261129227793
```

For class 0 (non-churn), it achieves a precision of 0.91 and a recall of 0.99, resulting in an F1-score of 0.95. This indicates that the model is highly effective at correctly identifying non-churn customers. For class 1 (churn), the model attains a precision of 0.92 and a recall of 0.61, leading to an F1-score of 0.74. This shows that while the model is very precise in predicting churn, it has some limitations in recall, meaning it misses a notable portion of actual churn instances.

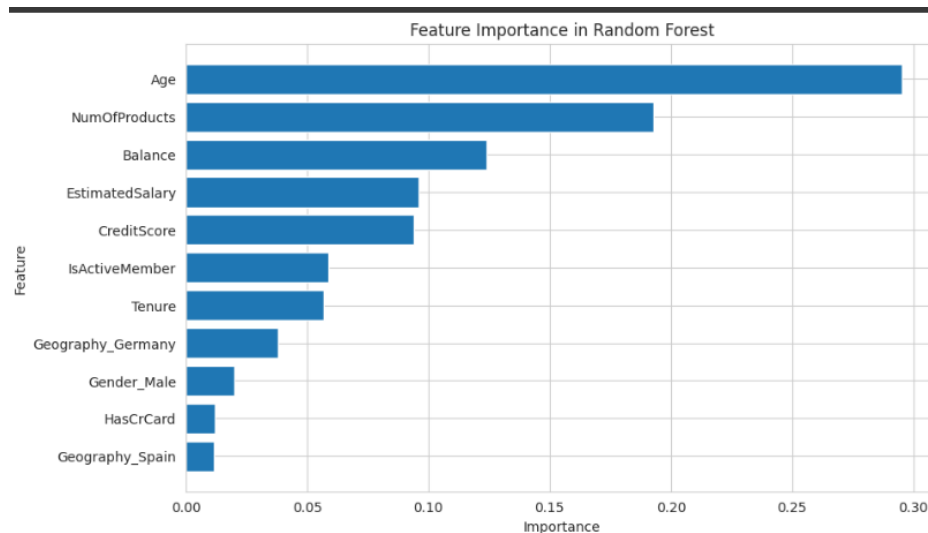
The overall accuracy of the model is 0.91, reflecting its high performance in distinguishing between churn and non-churn customers. The macro average for precision is 0.92, for recall is 0.80, and for F1-score is 0.84, indicating balanced performance across both classes. The weighted averages, which account for the imbalance in the dataset, are 0.91 for precision, 0.91 for recall, and 0.90 for F1-score. The ROC-AUC score is 0.978, signifying excellent overall ability to differentiate between the two classes. This high ROC-AUC score highlights the model's effectiveness in identifying both churn and non-churn customers on the training data.



The AUC (Area Under the Curve) score of 0.86 indicates a strong ability of the model to discriminate between the two classes. An AUC score of 0.86 suggests that there is an 86% chance that the model will correctly distinguish between a randomly chosen churned customer and a randomly chosen non-churned customer. In practical terms, the closer the ROC curve is to the top left corner of the plot, the better the model's performance. The diagonal line represents a model with no discriminative ability, where the true positive rate equals the false positive rate (AUC = 0.5).

Since the ROC curve for the Random Forest model is well above this diagonal line, it confirms the model's effectiveness. From a business perspective, this high AUC score means that the Random Forest model is reliable for predicting customer churn. It can be effectively used to identify customers who are at risk of churning, allowing the bank to take proactive measures to retain these customers. The model's strong performance can aid in targeted marketing and personalized customer engagement strategies to improve customer retention.

To understand which features are most influential in making the predictions for better interpretation and insight into model's decision-making process, feature importance has been calculated and plotted.



The bar chart titled "Feature Importance in Random Forest" visually represents the importance of each feature used in the Random Forest model for predicting customer churn. The importance of a feature indicates how much weight the model places on that feature when making its predictions.

Age: This is the most important feature in the model, with an importance score of around 0.295. This implies that the age of the customers has the most significant influence on predicting whether they will churn or not.

NumOfProducts: The number of products a customer has is the second most important feature, with a score of about 0.192. This suggests that customers with fewer products are more likely to churn. **Balance:** The balance in a customer's account is the third most important feature, with an importance score of approximately 0.124. Lower balances are associated with a higher likelihood of churn.

EstimatedSalary: This feature also plays a significant role, with an importance score of around 0.095. It indicates that the estimated salary of a customer affects their likelihood of staying or leaving.

CreditScore: Credit score has a notable impact with an importance score of about 0.093. Customers with lower credit scores are more likely to churn.

IsActiveMember: Being an active member has an importance score of around 0.058, showing that active members are less likely to churn. **Tenure:** The duration of time a customer has been with the bank (tenure) has a lower importance score of approximately 0.057, but it still plays a role in predicting churn.

Geography_Germany: This feature represents customers located in Germany and has an importance score of around 0.038, indicating geographical location can influence churn.

Gender_Male: Gender (specifically being male) has a smaller importance score of about 0.020, showing some influence on churn prediction.

HasCrCard: Whether a customer has a credit card has a very low importance score of around 0.012, indicating minimal influence on churn.

Geography_Spain: Similar to HasCrCard, this feature has a very low importance score, around 0.011, suggesting that being located in Spain has minimal impact on churn prediction.

Business Perspective: From this analysis, it is clear that demographic factors like age and the number of products held by a customer are critical in predicting churn. Customers who are younger and have fewer products with the bank are more likely to leave. Additionally, financial metrics such as account balance and estimated salary also play significant roles. Active membership status and credit score are moderately important, while geographical location and

whether the customer has a credit card have the least influence on churn predictions. These insights can guide the bank in designing targeted interventions to reduce churn by focusing on these key areas.

To reduce the dimensionality of the dataset, the decision of removing the least 3 important features which were 'Gender_Male', 'Geography_Spain' and 'HasCrCard' and re-run the model of random forest with reduced features.

```
# Print the classification report and ROC-AUC score for the test set
print("Random Forest after removing least important features")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_rf))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_rf))
print("\nROC-AUC Score:", roc_auc_score(y_test, y_pred_proba_rf))
```

Random Forest after removing least important features
Confusion Matrix:
[[2343 73]
 [313 271]]

Classification Report:		precision	recall	f1-score	support
0	0.88	0.97	0.92	2416	
1	0.79	0.46	0.58	584	
accuracy			0.87	3000	
macro avg	0.83	0.72	0.75	3000	
weighted avg	0.86	0.87	0.86	3000	

ROC-AUC Score: 0.8567537053206932

```
# Print the classification report and ROC-AUC score for the training set
print("\nClassification report for the training set")
print("Confusion Matrix:")
print(confusion_matrix(y_train, y_pred_rf_train))
print("\nClassification Report:")
print(classification_report(y_train, y_pred_rf_train))
print("\nROC-AUC Score:", roc_auc_score(y_train, y_pred_proba_rf_train))
```

Classification report for the training set
Confusion Matrix:
[[5482 65]
 [509 944]]

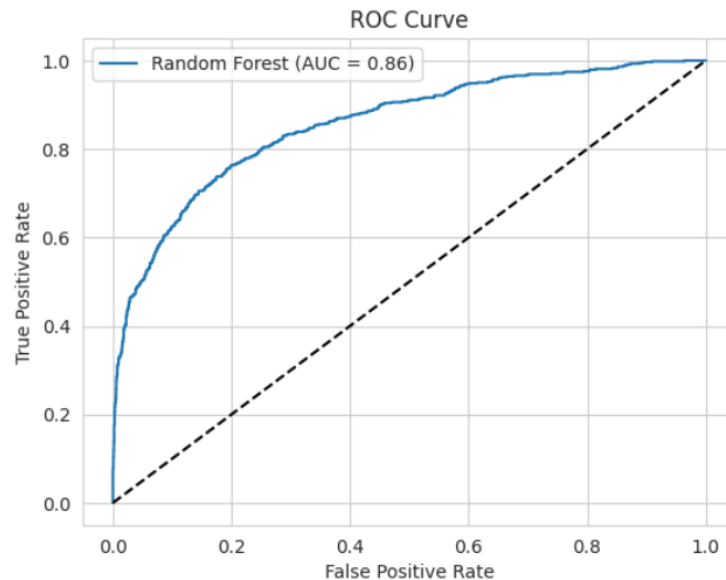
Classification Report:		precision	recall	f1-score	support
0	0.92	0.99	0.95	5547	
1	0.94	0.65	0.77	1453	
accuracy			0.92	7000	
macro avg	0.93	0.82	0.86	7000	
weighted avg	0.92	0.92	0.91	7000	

ROC-AUC Score: 0.9839657132548474

After removing the least important features, the test set results showed a precision of 0.88 for class 0 and an improved 0.79 for class 1. The recall remained the same for class 0 at 0.97, while it was 0.46 for class 1. The F1-scores were slightly better at 0.92 for class 0 and 0.58 for class 1.

The overall accuracy remained the same at 0.87, with a slight decrease in the ROC-AUC score to 0.8568. For the training set, the precision was 0.92 for class 0 and 0.94 for class 1, with recall values of 0.99 for class 0 and 0.65 for class 1. The F1-scores were consistent with the non-reduced model at 0.95 for class 0 and 0.77 for class 1. The overall accuracy was 0.92, and the ROC-AUC score was 0.9839. In summary, the feature reduction did not significantly alter the performance metrics of the Random Forest model. The model maintained similar levels of

precision, recall, and F1-scores while potentially simplifying the model and reducing computational complexity. The minor decrease in the ROC-AUC score suggests a slight trade-off in model performance, but the overall accuracy remained stable. This indicates that the removed features were not contributing significantly to the model's predictive power.



This refined model can be effectively used to predict customer churn, helping the business focus retention efforts on the most likely churners, ultimately improving customer retention strategies and reducing churn-related losses. The removal of the least important features make it more efficient and saves cost as well.

6.8 Model Technique #7: Gradient Boosting Model

Gradient boosting is an ensemble machine learning technique which is used for regression and classification tasks, which builds a model in a stage-wise manner from decision trees and combines them to form a strong predictive model. Gradient boosting often provides highly accurate predictions, especially for structured or tabular data.

Before training the model, hyperparameters are specified which are to be tuned.


```
#Gradient Boosting
from sklearn.ensemble import GradientBoostingClassifier

param_grid_gbm = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7]
}
```

Here ‘n_estimators’ refer to the number of boosting stages to run (i.e. the number of trees) , ‘learning_rate’ shrinks the contribution of each tree. The parameter ‘max_depth’ is the maximum depth of the individual regression estimators.

```
#classification report for test set
print("Gradient Boosting Machines")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_gbm))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_gbm))
print("\nROC-AUC Score:", roc_auc_score(y_test, y_pred_proba_gbm))

Gradient Boosting Machines
Confusion Matrix:
[[2339  77]
 [ 316 268]]

Classification Report:
              precision    recall  f1-score   support

     0       0.88        0.97        0.92        2416
     1       0.78        0.46        0.58         584

 accuracy          0.87        3000
 macro avg          0.83        0.71        0.75        3000
 weighted avg       0.86        0.87        0.86        3000

ROC-AUC Score: 0.861118513562551
```

For the test set, the Gradient Boosting Machine (GBM) model exhibits strong performance. The confusion matrix shows that out of 2416 instances of non-churn, 2339 were correctly identified, and 77 were misclassified. For churn instances, 268 were correctly predicted out of 584, while 316 were incorrectly classified as non-churn. The precision for non-churn (class 0) is 0.88, indicating that 88% of the instances predicted as non-churn were correct. The recall for non-churn is 0.97, meaning the model correctly identified 97% of the actual non-churn instances. The F1-score for non-churn is 0.92, representing a balance between precision and recall. For churn (class 1), the precision is 0.78, recall is 0.46, and F1-score is 0.58. The overall

accuracy of the model is 0.87, with a macro average F1-score of 0.75 and a weighted average F1-score of 0.86. The ROC-AUC score is 0.8611, indicating good model performance.

```
#Classification report for the train set
print("Gradient Boosting Machines")
print("Confusion Matrix:")
print(confusion_matrix(y_train, y_pred_gbm_train))
print("\nClassification Report:")
print(classification_report(y_train, y_pred_gbm_train))
print("\nROC-AUC Score:", roc_auc_score(y_train, y_pred_gbm_train))
```

Gradient Boosting Machines
Confusion Matrix:
[[5462 85]
 [607 846]]

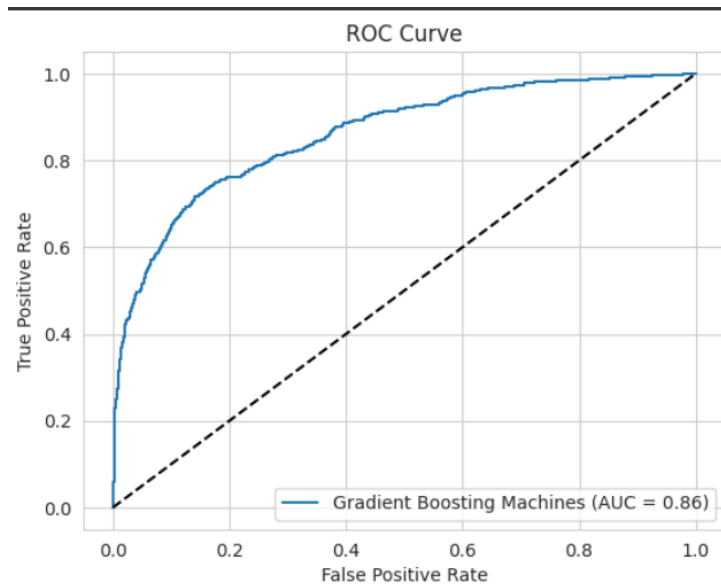
Classification Report:

	precision	recall	f1-score	support
0	0.90	0.98	0.94	5547
1	0.91	0.58	0.71	1453
accuracy			0.90	7000
macro avg	0.90	0.78	0.83	7000
weighted avg	0.90	0.90	0.89	7000

ROC-AUC Score: 0.7834600177597658

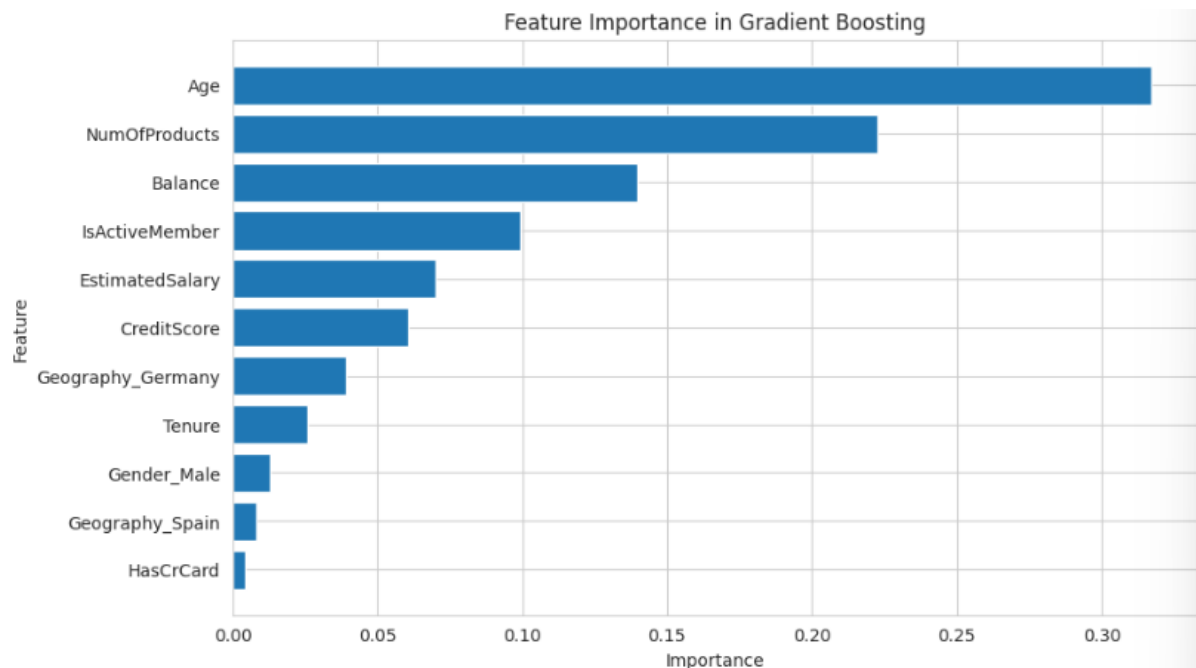
For the training set, the GBM model shows high performance with an accuracy of 0.90. The confusion matrix indicates that 5547 out of 5547 non-churn instances were correctly classified, and 85 were misclassified. For churn instances, 846 out of 1453 were correctly identified, and 607 were incorrectly classified. The precision for non-churn is 0.90, recall is 0.98, and F1-score is 0.94. For churn, the precision is 0.91, recall is 0.58, and F1-score is 0.71. The macro average F1-score is 0.83, and the weighted average F1-score is 0.89. The ROC-AUC score is 0.7835, indicating robust model performance.

These results suggest that the GBM model performs well in predicting customer churn, providing a reliable tool for identifying potential churners. This can help the business in taking proactive measures to retain customers, thereby improving customer loyalty and reducing churn rates.



The area under the curve (AUC) is 0.86, indicating that the model has a strong ability to discriminate between the two classes. An AUC value of 0.5 suggests no discrimination (random guessing), while an AUC value of 1.0 indicates perfect discrimination. Therefore, an AUC of 0.86 means the GBM model is quite effective at predicting customer churn. The ROC curve shows a good true positive rate for a range of false positive rates. For example, at a false positive rate of 0.2, the true positive rate is approximately 0.7. This means that when the model incorrectly identifies 20% of negatives as positives, it correctly identifies 70% of positives. This balance between sensitivity and specificity highlights the GBM model's robustness in identifying customers who are at risk of churn while minimizing false alarms.

To understand which features are most influential in making the predictions for better interpretation and insight into model's decision-making process, feature importance has been calculated and plotted.



Age is the most influential feature, with the highest importance score. This indicates that age plays a significant role in predicting whether a customer will churn. Number of Products is the second most important feature. The number of products a customer has is a strong indicator of their likelihood to churn. Balance also has a high importance score, suggesting that the account balance of a customer is crucial in determining their churn risk. IsActiveMember follows closely, indicating that whether a customer is an active member significantly impacts churn predictions. Estimated Salary and Credit Score have moderate importance, implying they are relevant but not as critical as the top features. Geography_Germany and Tenure have lower importance scores but still contribute to the model's predictions. Gender_Male, Geography_Spain, and HasCrCard have the least importance, indicating they have a minimal effect on the churn prediction.

To reduce the dimensionality of the dataset, the decision of removing the least 3 important features which were 'Gender_Male', 'Geography_Spain' and 'HasCrCard' and re-run the model of gradient boosting with reduced features.

```
# Evaluate the model on the test set
print("Gradient Boosting Machines (Reduced Features)")
print("Confusion Matrix (Test Set):")
print(confusion_matrix(y_test, y_pred_gbm_reduced))
print("\nClassification Report (Test Set):")
print(classification_report(y_test, y_pred_gbm_reduced))
print("\nROC-AUC Score (Test Set):", roc_auc_score(y_test, y_pred_proba_gbm_reduced))
```

```
Gradient Boosting Machines (Reduced Features)
Confusion Matrix (Test Set):
[[2322  94]
 [ 311 273]]

Classification Report (Test Set):
      precision    recall  f1-score   support

     0       0.88       0.96       0.92       2416
     1       0.74       0.47       0.57        584

 accuracy       0.86       0.86       0.86       3000
  macro avg       0.81       0.71       0.75       3000
 weighted avg       0.86       0.86       0.85       3000

ROC-AUC Score (Test Set): 0.8577300020411867
```

```
# Evaluate the model on the training set
print("\nConfusion Matrix (Training Set):")
print(confusion_matrix(y_train, y_pred_gbm_train_reduced))
print("\nClassification Report (Training Set):")
print(classification_report(y_train, y_pred_gbm_train_reduced))
print("\nROC-AUC Score (Training Set):", roc_auc_score(y_train, y_pred_proba_gbm_train_reduced))
```

```
Confusion Matrix (Training Set):
[[5432 115]
 [ 616 837]]

Classification Report (Training Set):
      precision    recall  f1-score   support

     0       0.90       0.98       0.94       5547
     1       0.88       0.58       0.70       1453

 accuracy       0.89       0.78       0.90       7000
  macro avg       0.89       0.78       0.82       7000
 weighted avg       0.89       0.90       0.89       7000

ROC-AUC Score (Training Set): 0.9328055777128712
```

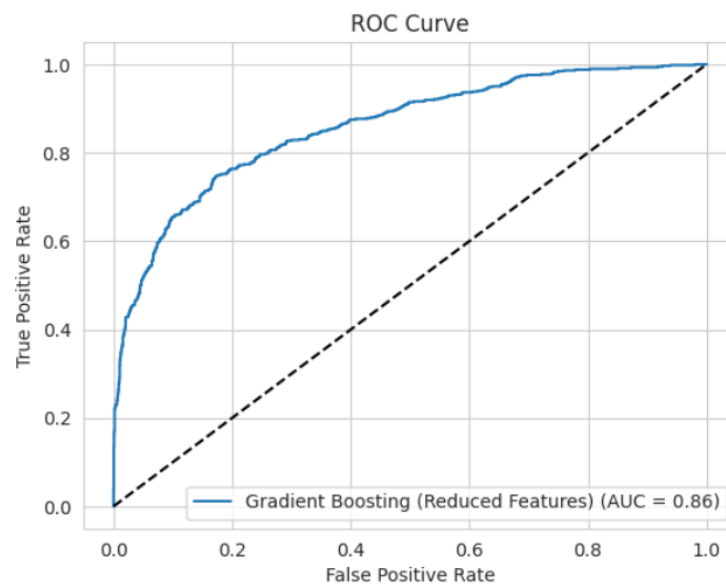
The comparison between the Gradient Boosting Machines (GBM) with all features and the GBM with reduced features reveals some insightful observations. For the test set performance, both models exhibit similar metrics.

The GBM with reduced features demonstrates a precision of 0.88 and a recall of 0.96 for class 0 (No Churn), resulting in an F1-score of 0.92. For class 1 (Churn), it shows a precision of 0.74, recall of 0.47, and F1-score of 0.57, with an overall accuracy of 0.86 and a ROC-AUC score of 0.8577. These metrics are quite comparable to the GBM with all features, which has a precision of 0.88, recall of 0.97, and F1-score of 0.92 for class 0, and a precision of 0.78, recall of 0.46, and F1-score of 0.58 for class 1, with an overall accuracy of 0.87 and a ROC-AUC score of 0.8611.

For the training set performance, both models maintain high precision and recall for class 0. The GBM with reduced features exhibits a precision of 0.90, recall of 0.98, and F1-score of 0.94 for class 0, and a precision of 0.88, recall of 0.58, and F1-score of 0.70 for class 1, with

an overall accuracy of 0.90 and a higher ROC-AUC score of 0.9328. In contrast, the GBM with all features shows a precision of 0.90, recall of 0.98, and F1-score of 0.94 for class 0, and a precision of 0.91, recall of 0.58, and F1-score of 0.71 for class 1, with an overall accuracy of 0.90 and a ROC-AUC score of 0.7835.

In summary, the GBM with reduced features achieves nearly identical performance to the model with all features on the test set while simplifying the model and potentially lowering computation costs. This efficiency gain makes the reduced features model a more practical choice for business applications, maintaining high accuracy and generalization capability.



6.9 Model Comparison

Model comparison is the process of evaluating multiple machine learning models that determine which model performs best on the given dataset. It involves the comparison of metrics such as accuracy, precision, recall, F1 score and ROC-AUC score to assess each model's effectiveness in predicting outcomes. The goal is to identify the model that not only fits the training data well but also generalizes effectively to the new and unseen data. This comparison is crucial for The Sapphire Bank to predict customer churn. Accurate prediction of

churn is vital for the bank to take proactive measures to retain the customers and improve the overall customer satisfaction.

Among all these models, the Random Forest model shows the best performance with a ROC-AUC score of 0.86 , an F1 score for churn of 0.58 and an accuracy of 0.91. The high score of ROC suggests that the model has a good balance between sensitivity (true positive rate) and specificity (true negative rate) which is vital for making informed decisions in churn prediction. For the churn class, the F1 score of 0.58 indicates that the Random Forest model effectively balances precision (the accuracy of positive predictions) and recall (the ability to find all the positive instances). This balance is important for the Sapphire Bank as it helps in minimizing both missed churn predictions and false alarms, that ensures the customer retention efforts are accurately targeted. An accuracy of 0.91 means that 91% of the predictions made by the Random Forest Model are correct. This high accuracy indicates that the model is reliable and effective in overall classification that is critical for the bank to trust the model's performance and base strategic decisions on them.

Model	ROC-AUC Score	F1-Score(Churn)	Accuracy
Full Logistic Regression	0.774107	0.28	0.81
Forward Logistic Regression	0.774097	0.29	0.81
Backward Logistic Regression	0.774097	0.29	0.81
Stepwise Logistic Regression	0.774097	0.29	0.81
Decision Tree	0.834492	0.57	0.86

Random Forest	0.856754	0.58	0.91
Random Forest (Reduced Features)	0.861119	0.58	0.87
Gradient Boosting Machines	0.857730	0.58	0.87
Gradient Boosting Machines (Reduced Features)	0.857730	0.57	0.86

The reduced feature versions of both Random Forest and Gradient Boosting Machines highlights that the bank can achieve efficient and cost-effective predictions without significant loss of performance. This is beneficial for operational efficiency that allows the bank to streamline its data processing while maintaining high predictive accuracy.

7.0 Model Recommendations

7.1 Model Selection

In the process of model selection for predicting customer churn at The Sapphire Bank, various models were evaluated, which includes Logistic Regression, Decision Tree, Random Forest and Gradient Boosting Machines along with their feature reduced versions. While gradient boosting machines showed a high ROC-AUC score of 0.86 and F1 score of 0.58, the Random Forest model outperformed in terms of accuracy with a score of 0.91, and a similar ROC-AUC and F1 scores which makes it the best model.

7.2 Model Theory

The models evaluated in this study leverages both traditional statistical techniques and advanced machine learning algorithms which predicts the customer churn for the Sapphire Bank. Each model brings unique strengths and limitations, which have been tailored through specific codes and parameters to optimize their performance for the bank's needs.

1. **Logistic Regression:** This is a traditional statistical model that estimates the probability of a binary outcome, such as customer churn, based on various predictor variables.
2. **Decision Tree:** This model works by splitting the data based on feature values to predict target variable. These models are intuitive and easy to interpret which makes it useful for the bank to identify specific customer segments at risk of churn. For the Sapphire Bank, decision trees help in understanding decision rules that lead to customer churn.
3. **Random Forest:** This is an ensemble learning method which is builds multiple decision trees and averages their predictions to enhance the accuracy and reduces overfitting. It is robust and provides insights to feature importance, which is crucial for the Sapphire Bank to prioritize factors that affect churn.
4. **Gradient Boosting Machines:** It is a powerful machine learning algorithm that sequentially builds models to correct errors from previous iterations. This method is effective in capturing complex patterns within the data. For the bank, GBM provides a sophisticated approach to churn prediction.

From the Sapphire Bank's perspective, the selection of these models is aimed at leveraging both interpretability and predictive power to address customer churn. Logistic Regression offers simplicity and ease of interpretation, which helps bank executives to understand the impact of different variables on churn. Decision trees provide clear decision rules that can be translated into actionable insights. Random Forest balances prediction accuracy and

robustness, making it a more reliable choice for identifying at-risk customers GBM, with its ability to handle complex data patterns, offers the highest precision, with the need for careful tuning.

7.3 Model Assumptions and Limitations

It is assumed that the models chosen are robust to skewed data distributions, particularly the tree-based models which can handle non-normal distributions. It is hypothesized that the predictive power of these features remains significant as indicated by their importance in the model analysis. Given the satisfactory performance of the models during testing, no log-transformations is applied. It is assumed that any impact of skewness on model predictions is minimal or offset by model's complexity.

Each model has specified assumptions and limitations which were addressed during the modelling process:

1. Logistic Regression: It assumes a linear relationship between features and the log odds of the outcome. It is limited in capturing non-linearities. Various selection methods (full, forward, backward, stepwise) were used to optimize the feature selection.
2. Decision Tree: This is prone to overfitting, requiring pruning and depth. It has limitations in generalisation.
3. Random Forest: This is computationally intensive with large datasets but mitigates overfitting and handles high-dimensional data effectively. The ensemble approach and feature importance evaluation helped in understanding the key drivers of churn.
4. Gradient Boosting Machines: This model is effective for classification but it is sensitive to hyperparameters, that requires careful tuning to avoid overfitting.

7.4 Model Sensitivity to Key Drivers

Analysing key drivers are important as it helps businesses to understand the underlying elements or variables that have the most significant impact on a specific outcome or interest, such as customer satisfaction, purchase intent, or brand loyalty. (What Is Key Driver Analysis and How To Use It in Your Customer Research, n.d.)

Different models – Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting Machines (GBM) highlight various factors with varying levels of importance. The Sapphire Bank can derive insights into which factors are most influential in predicting the churn and can prioritize business strategies accordingly.

From Logistic Regression, the most significant factor includes Age (Coefficient:0.733435), which means that older customers are more likely to churn. Followed by this, customers from Germany have a higher likelihood of churning compared to the other regions (0.725420). Even higher account balances are associated with increased churn (0.176369). There is a slight positive correlation of Estimated Salary with churn which indicates that salary is not a strong predictor. This suggests that the Sapphire Bank should focus on age demographic and region specific strategies to mitigate churn.

From Decision tree, age is again the most critical factor along with the number of products (0.281632). This means that the variety and number of products that a customer holds significantly impact their likelihood to stay with the bank. These findings indicate that the bank should enhance the customer engagement specially focussing on cross-selling and ensuring active membership to retain customers.

From Random Forest, the most crucial factor is age. The number of products is still a vital predictor and even the balance continues to be a significant driver (0.124153). The feature of

Estimated Salary is 0.0959 which gains a little more importance compared to the other models, followed by the CreditScore (0.093957) which indicates credit score as a notable factor, highlighting customers with lower credit scores may churn more. This suggests that the bank should focus on customer's financial behaviour and product portfolio to tailor retention strategies.

For Gradient Boosting Model, the most significant factor is age (0.317272) which has been consistent in all the models. Even the number of products remain the second most critical factor with importance of 0.222813. Balance happens to be consistently significant with importance level of 0.139641.

The consistent identification of age, number of products, balance and regional factors across the models suggests that the Sapphire Bank should develop targeted strategies for older customers, potentially offering specialized products and services. It should also focus on cross-selling and increasing the number of products held by each customer to enhance customer loyalty. The Sapphire bank can address regional disparities by tailoring marketing and customer service strategies to specific geographies and ensure active customer engagement is there as active members are less likely to churn. They also need to monitor financial behaviour indicators like balance and credit score to identify at-risk customers early and implement pre-emptive retention measure.

8.0 Conclusion and Recommendations

In this comprehensive study aimed at predicting customer churn for The Sapphire Bank, by utilizing a variety of machine learning models to assess and determine the most effective predictive tool. Among the models tested, the Random Forest model demonstrated superior performance achieving the highest accuracy of 0.91, a robust ROC-AUC score of 0.857730 and competitive F1 score of churn of 0.58. These results highlight the Random Forest model's capability to handle complex datasets with numerous features offering a balance between prediction accuracy and model robustness. The Random Forest model's ability to average multiple decision trees reduce the risk of overfitting, making it a reliable choice for the bank's predictive analytics needs. (What is random forest?, n.d.)

The evaluation also focusses on critical drivers of customer churn. Factors such as age, number of products held, account balance and regional differences are consistent across all the models and they were identified as significant predictors of churn. These findings underscore the importance of understanding customer demographics and behaviour to effectively tailor retention strategies.

1. Age: Older customers were found to have a higher likelihood to churn. This insight can guide the Sapphire Bank to develop specialised services and products aimed at retaining this customer segment.
2. Number of products: Customers with fewer products are more likely to churn. Enhancing cross-selling and bundling strategies can help in increasing product holdings per customer, thereby reducing churn.
3. Account Balance: Lower account balance were associated with higher churn rates. Monitoring financial behaviour and providing personalised financial advice can mitigate the risk.

4. Regional differences: The analysis revealed significant regional variations in churn rates particularly in Germany and Spain. Addressing these disparities with localised strategies can enhance customer retention in these areas.

These insights can inform various aspects of the Sapphire Bank's operation from marketing and customer service to product development and financial advisory services.

8.1 Impacts on Business Problem

Addressing customer churn is the top-most priority for the Sapphire Bank, as high churn rates can have significant negative impacts on business performance and profitability.

1. Revenue Decline and Volatility: If customer churn is not addressed then it can lead to significant decline in revenue. The loss of customers means a direct loss in the income streams, which could lead to financial instability. If the churn rate remains high, the bank could lose 3000 customers annually. With an average revenue per customer of \$1,000, this translates to a \$2 million loss in revenue in each year which can impact severely.
2. Reduced customer lifetime value: It is the measurement of how valuable a customer is to the company, not just on purchase-by-purchase basis but across the entire customer relationships. (What is customer lifetime value (CLV) and how can you increase it?, n.d.). The reduction in CLV implies that bank is not fully capitalizing on the potential revenue from each customer, causing in diminished long-term profitability. For example, if the average customer relationship duration decreases from 5 years to 3 years, the CLV will significantly drop, reducing the bank's ability to generate sustained profits from its customer base.

3. **Decreased customer satisfaction and loyalty:** Without targeted retention strategies, customer satisfaction and loyalty are likely to decline. Dissatisfied customers are more inclined to switch to competitors. Lack of loyalty programs and personalized services will alienate customers which makes it difficult to foster long-term relationships. For example, the absence of loyalty programs could cause customers to feel undervalued, prompting them to explore better offers from competitors. This dissatisfaction can lead to an increase in negative word-of-mouth, damaging the reputation of the bank.
4. **Increased marketing and acquisition costs:** The constant need for new customer acquisition is costly and inefficient. The bank will incur substantial expenses in marketing campaigns, sales and promotions.
5. **Weakened Competitive Position:** Failing to retain the customers can weaken the Sapphire Bank's competitive position. Competitors with effective retention strategies will attract more customers. This loss of market share can be difficult to recover and lead to the loss of primary stakeholders as well.

8.2 Recommended next steps

To mitigate the risk of customer churn and capitalize on the insights gained from the predictive models, The Sapphire Bank should consider the following strategic actions:

- a) **Implement targeted retention strategies:** develop and deploy targeted retention campaigns based on the key predictors of churn. Focus on high-risk customers with tailored offers, personalized communications and enhanced service experiences to improve satisfaction and loyalty.
- b) **Enhance customer experience:** Invest more on improving the overall customer experience by addressing pain points and ensuring consistent, high quality interactions.

This involves conducting regular customer feedback surveys, training customer representatives and streamlining processes to reduce wait times and enhance service delivery.

- c) Leverage data analytics for continuous improvement: Utilize data analytics to monitor and analyse customer behaviour, preferences and feedback. Implement an iterative process to refine retention strategies based on data-driven insights. This involves setting up a dedicated analytics team to track customer metrics and churn rates. Use advanced analytics tool to identify emerging trends and patterns.
- d) Strengthen customer relationships: Build stronger relationships with customers by fostering trust and engagement through transparent communication and value-added services. This involves communicating clearly about new products, services and changes. Offer financial education and advisory services to help customers make informed decisions. Create community initiatives and events to engage customers and build brand loyalty.

References

- Bozkurt, C. D. (2023, July 3). *The Power of Data Visualization: Enhancing Decision Making and Stakeholder Satisfaction*. Retrieved from Medium:
<https://medium.com/@candelil.bozkurt/the-power-of-data-visualization-enhancing-decision-making-and-stakeholder-satisfaction-b91f6e3aca09>
- CHENG, M. (2024, March 21). *Churn Rate: What It Means, Examples, and Calculations*. Retrieved from Investopedia: <https://www.investopedia.com/terms/c/churnrate.asp>
- Lemonaki, D. (2021, August 24). *What is an Outlier? Definition and How to Find Outliers in Statistics*. Retrieved from freeCodeCamp: <https://www.freecodecamp.org/news/what-is-an-outlier-definition-and-how-to-find-outliers-in-statistics/>
- Puga, J. (2024, February 8). *Customer retention strategies for banks – what works?* Retrieved from unblu: <https://www.unblu.com/en/blog/customer-retention-strategies-for-banks-what-works/>
- Robinson, S. (n.d.). *Data Exploration*. Retrieved from TechTarget:
<https://www.techtarget.com/searchbusinessanalytics/definition/data-exploration>
- What is customer lifetime value (CLV) and how can you increase it?* (n.d.). Retrieved from Qualtrics: <https://www.qualtrics.com/experience-management/customer/customer-lifetime-value/>
- What Is Key Driver Analysis and How To Use It in Your Customer Research*. (n.d.). Retrieved from quantilope: <https://www.quantilope.com/resources/what-is-key-driver-analysis-and-how-to-use-it-in-your-customer-research#:~:text=Key%20Driver%20analysis%20will%20help,appeal%2C%20popularity%2C%20and%20profitability.>
- What is random forest?* (n.d.). Retrieved from IBM: <https://www.ibm.com/topics/random-forest>