BANK CUSTOMER CHURN PREDICTION REPORT

REPORT DATE: JANUARY 7, 2025

SUBMITTED BY: OLAWALE FRANCIS ONAOLAPO

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This report presents the development of a supervised learning model for bank customer churn prediction, employing the K-Nearest Neighbors (KNN) algorithm. Customer churn, a critical issue in various industries, leads to significant revenue losses. Using a dataset of 10,000 bank customers, key features such as customer demographics, transaction history, and account activity were analyzed to identify churn patterns. After addressing class imbalance through the Synthetic Minority Oversampling Technique (SMOTE) and optimizing hyperparameters via grid search, the model achieved a recall of 94%, an F1-score of 90%, and an AUC of 0.91. These results underscore the model's reliability and utility for predicting churn and informing retention strategies. Comparative performance analysis demonstrated the model's superiority over similar studies. While the current implementation excels in identifying potential churners, opportunities exist to enhance precision and cost-efficiency. This study provides a framework for leveraging machine learning to mitigate customer churn in the banking sector.

1.0 INTRODUCTION AND BACKGROUND

This report focuses on a bank customer churn analysis conducted to build a machine learning model capable of predicting which bank customers might churn using a supervised learning algorithm.

Customer churn occurs when a customer or group of customers ceases transactions with a business entity. This phenomenon results in significant revenue losses for businesses, as acquiring new customers is more expensive than retaining existing ones. Several factors can lead to customer churn, including dissatisfaction with the quality of products or services provided by the company.

Customer churn is a widespread issue across various industries, hindering business growth and, in severe cases, leading to business closure. This analysis was inspired by my previous professional experience of over 10 years as a Quality Engineer. In that role, I was responsible for ensuring customer satisfaction with the company's products and services and had access to customer records. During this time, I observed instances of customer churn, which were often identified too late to take preventive actions.

The implementation of this customer churn analysis allowed me to reinforce the knowledge acquired during the Data Science Foundation lectures. It has also enhanced my technical skills in predicting customer churn by analyzing customer records, which will enable me to advise businesses on strategies to prevent customer attrition.

It is important to note that customer churn is not a challenge faced solely by banks. Other industries, such as telecommunications, retail, and real estate, also contend with customer churn. Bank customer data was selected for this analysis due to my interest in pursuing a career in the financial sector. The skills reinforced through this project will enable me to contribute effectively to reducing customer churn in financial institutions.

A supervised machine learning algorithm was used to build the predictive model. Such algorithms are particularly effective for classification tasks like this because most companies maintain databases containing detailed records of customers who have churned and those who continue to transact with the company.

Various supervised machine learning algorithms have been applied by researchers to analyze customer churn in different industries. These include Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM), among others.

2.0 LITERATURE REVIEW

Several studies on customer churn analysis were reviewed to understand the methodologies employed and the performance of the models developed.

Tékouabou et al. (2022) conducted research on bank customer churn, utilizing various algorithms such as Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), and Logistic Regression (LR) for model building. The study applied Z-score normalization for data scaling and the Synthetic Minority Oversampling Technique (SMOTE) to address data imbalance issues. Among the models, RF demonstrated the best performance, achieving an accuracy of 0.86 and an F1 score of 0.86. In contrast, SVM, KNN, DT, and LR underperformed, with KNN achieving an accuracy of 0.68 and an F1 score of 0.70, while SVM recorded an accuracy of 0.57 and an F1 score of 0.64. The study emphasized that data imbalance is a common challenge in classification problems, contributing to the lower performance of SVM, KNN, DT, and LR algorithms.

Tran, Le, and Nguyen (2023) explored credit card bank customer churn prediction using SVM, RF, DT, LR, and KNN. For scaling numerical features, they employed standardization for normally distributed features and normalization for non-normally distributed ones. Categorical features were processed using one-hot encoding and label encoding. SMOTE was used for data balancing, and the dataset was divided into clusters using the K-means algorithm before model building. RF and SVM were the overall top performers, while KNN exhibited the best recall performance.

Chen, Liu, and Wang (2023) investigated customer churn prediction for a European bank. Their study applied Min-Max normalization to numerical features and one-hot encoding to categorical features. SMOTE was used for data balancing. The algorithms employed included Logistic Regression (LR), Support Vector Classifier (SVC), Gradient Boosted Decision Tree (GBDT), Random Forest (RF), and AdaBoost. Among these, the AdaBoost classifier achieved the best performance, with a recall of 0.718 and an AUC score of 0.776. SVC followed, with a recall of 0.712 and an AUC score of 0.727.

Given the restrictions on the types of models that can be selected within the framework of this project—limited to KNN and SVM for supervised classification tasks—and based on the reviewed literature, despite SVM had slightly higher accuracy than the KNN algorithm, the KNN algorithm was chosen for building the model due to its relatively high recall values compared to SVM in the referenced studies, highlighting its potential utility in customer churn prediction scenarios where recall is critical.

KNN is an algorithm that classifies new data points by comparing their similarity to neighboring points in the training dataset. It assigns a class to the new data point based on the majority class of the nearest neighbors it resembles.

The feature engineering and oversampling techniques used in this research were selected based on the model performance assessments in the reviewed literature, ensuring the development of a model capable of making accurate customer churn predictions.

3.0 TECHNICAL IMPLEMENTATION OF THE MODEL

The technical implementation of the model followed the data science process cycle as illustrated in **Figure 1 (The implemented Data Science Process Cycle)**.

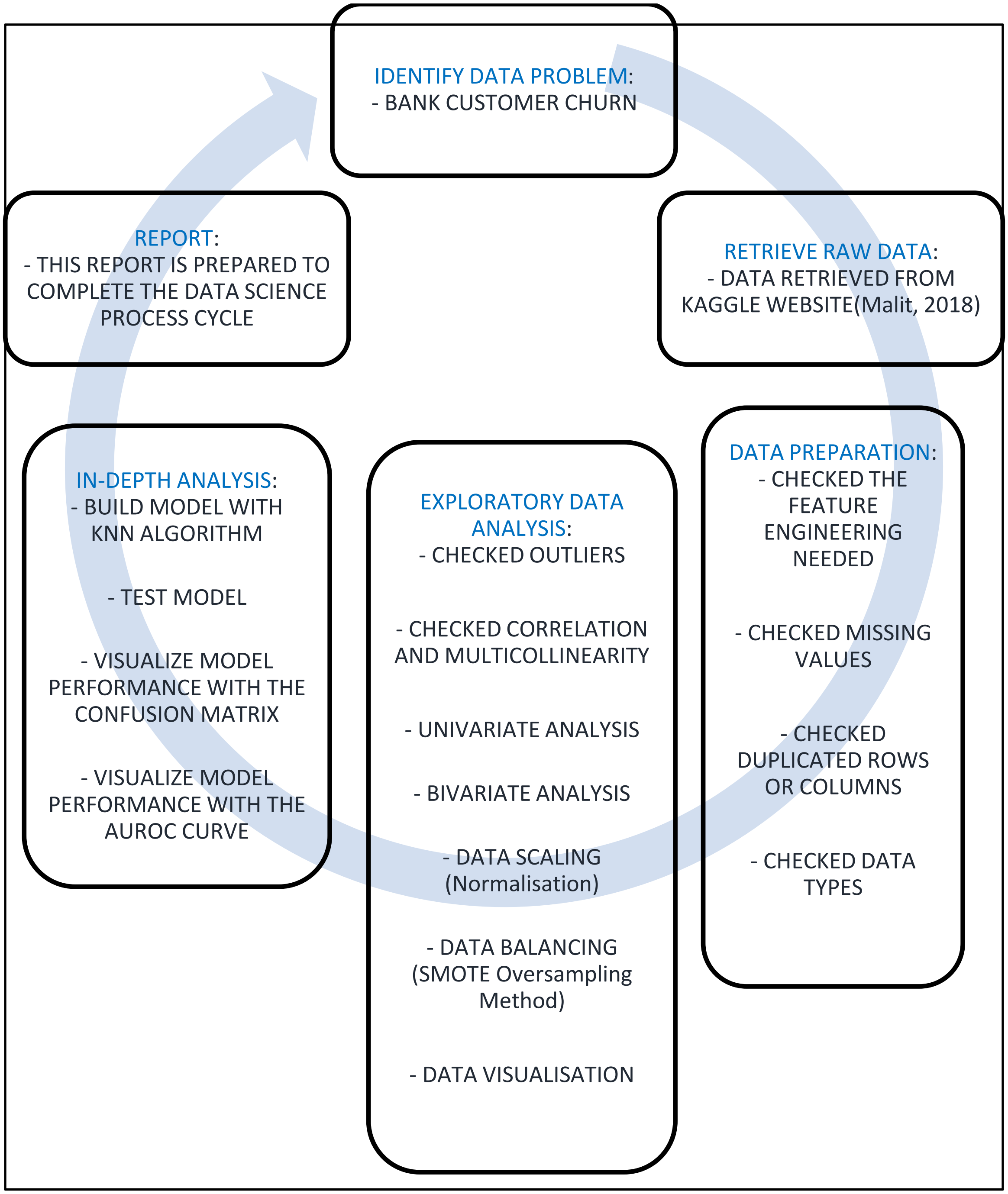

IDENTIFY DATA PROBLEM:
- BANK CUSTOMER CHURN
RETRIEVE RAW DATA:
- DATA RETRIEVED FROM KAGGLE WEBSITE(Malit, 2018)
DATA PREPARATION:
- CHECKED THE FEATURE ENGINEERING NEEDED
- CHECKED MISSING VALUES
- CHECKED DUPLICATED ROWS OR COLUMNS
- CHECKED DATA TYPES
EXPLORATORY DATA ANALYSIS:
- CHECKED OUTLIERS
- CHECKED CORRELATION AND MULTICOLLINEARITY
- UNIVARIATE ANALYSIS
- BIVARIATE ANALYSIS
- DATA SCALING (Normalisation)
- DATA BALANCING (SMOTE Oversampling Method)
- DATA VISUALISATION
IN-DEPTH ANALYSIS:
- BUILD MODEL WITH KNN ALGORITHM
- TEST MODEL
- VISUALIZE MODEL PERFORMANCE WITH THE CONFUSION MATRIX
- VISUALIZE MODEL PERFORMANCE WITH THE AUROC CURVE
REPORT:
- THIS REPORT IS PREPARED TO COMPLETE THE DATA SCIENCE PROCESS CYCLE

Figure 1: The implemented Data Science Process Cycle

3.1 Data Source and Description

The data used to build the customer churn prediction model was obtained from the Kaggle website (Malit, 2018). The dataset contains 14 columns and 10,000 rows. The target variable is in the column named Exited.

Features such as RowNumber, CustomerID, and Surname were removed as they do not provide predictive value for the target variable. The remaining 11 features are described in Table 1 (Data Description).

S/N	COLUMN NAME	DESCRIPTION	ATTRIBUTE
1	CreditScore	Customer credit score (range: 350–850)	Numeric - Discrete Quantitative Data
2	Age	Customer age (range: 18–92 years)	
3	Tenure	Customer's years of transaction with the bank (range: 0–10 years)	
4	NumOfProducts	Number of products subscribed by the customer (range: 1–4 products)	
5	Balance	Customer account balance (range: 0–250,898 Euros)	Numeric - Continuous Quantitative Data
6	Estimated Salary	Customer's estimated salary (range: 11.58–199,992.48 Euros)	
7	Geography	Customer's location (France, Germany, Spain)	Categorical - Nominal Qualitative Data Type
8	Gender	Customer gender (male or female)	
9	HasCrCard	Boolean: 0 = No credit card, 1 = Has credit card	
10	IsActiveMember	Boolean: 0 = Not active, 1 = Active	
11	Exited	Boolean: 0 = Customer still transacting, 1 = Customer churned.	Boolean - Target Variable

Table 1: Data Description

3.2 Data Preparation (cleaning / pre-process raw data)

- No missing values or duplicate rows/columns were found in the dataset.

3.3 Exploratory Data Analysis (EDA)

3.3.1 Outliers

The Z-score method (threshold: 3 standard deviations) was applied to the numeric features and it identified potential outliers in the **CreditScore**, **Age**, and **NumOfProducts features**. A visual inspection confirmed that these potential outliers fell within the typical range for bank customers and were retained.

3..3.2 Correlation and Multicollinearity:

- No strong correlations were observed among the numerical features, as illustrated in the correlation matrix (Figure 2). For the bivariate analysis, a correlation value of ±0.8 is used as the threshold for identifying strong relationships between variables. However, the correlation value between account balance and number of products is -0.30, indicating a weak inverse relationship, as signified by the negative sign.
- The Variance Inflation Factor (VIF), computed using a Generalized Linear Model (GLM), confirmed the absence of multicollinearity among the independent features (Figure 3: VIF Bar Chart). A VIF score of 1 indicates no multicollinearity, while a score greater than 10 is generally considered to indicate high multicollinearity.

For further details on the interpretation and application of multicollinearity and correlation, refer to Chan et al. (2022).

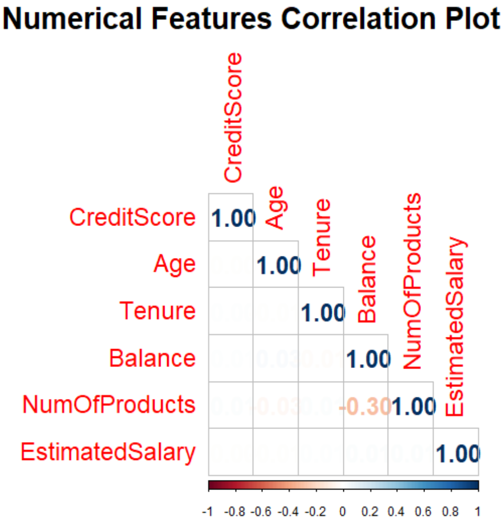


Figure 2: Correlation Matrix

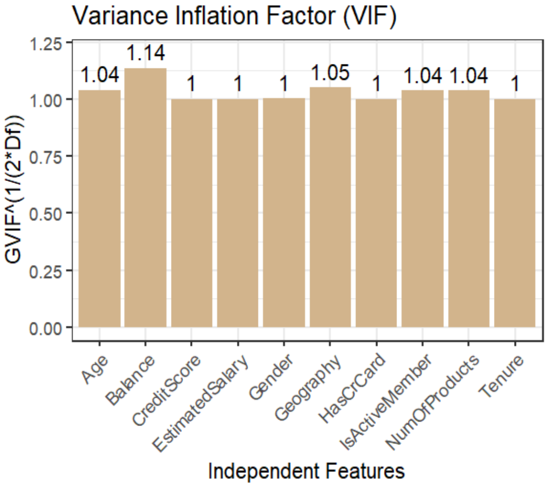


Figure 3: VIF Bar Chart

3.3.3 Univariate Analysis:

- The distributions of all features were analyzed to ensure they aligned with expected patterns for bank customer data (Figures 4–7 – samples of the univariate analysis visualizations).
- The chart types used for the univariate analysis and the reasons for using the chart types are as follows:
    - Bar Charts (Figures 4 and 5): These were used to analyze the distribution of attributes in categorical features, allowing for visual comparison of their frequencies, and allow to see categorical features with zero-variance or very high-variance with no predictive influence.
    - Histogram (Figure 7): This was used to provide insights into the distribution of numeric data by grouping values into bins, making patterns and trends easier to observe.
    - Boxplot (Figure 6): This was employed to examine the distribution of numeric data attributes in terms of percentiles, such as the median and quartiles, and to re-confirm any potential outliers in the data.
- Key observations include, approximately 20% of customers churned (Figure 4), male customers conducted more transactions with the bank (Figure 5), the median estimated salary was around 100,000 Euros, with most customers earning between 50,000–150,000 Euros (Figure 6), and the majority of customers were aged between 25–45 years (Figure 7) among other univariate analysis observations.
- All analyzed features aligned with expected patterns for bank customer data.
- The target variable, **Exited**, was found to be imbalanced (Figure 4). To address this, the minority class (customers who exited) was oversampled.

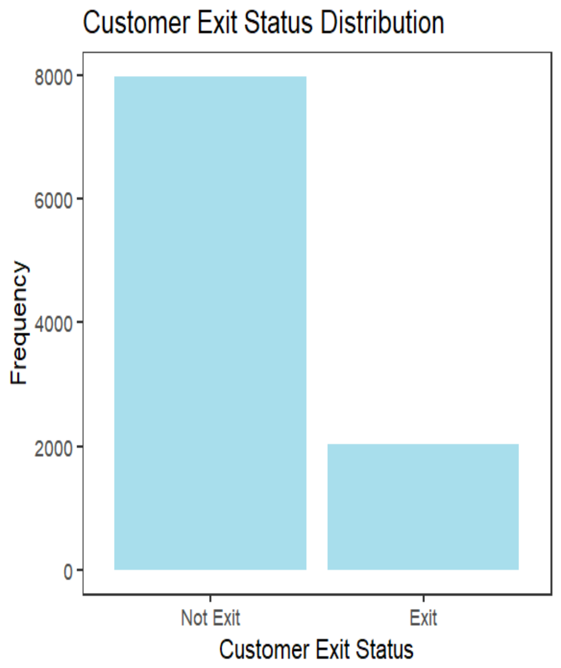


Figure 4: Distribution of the Target Variable

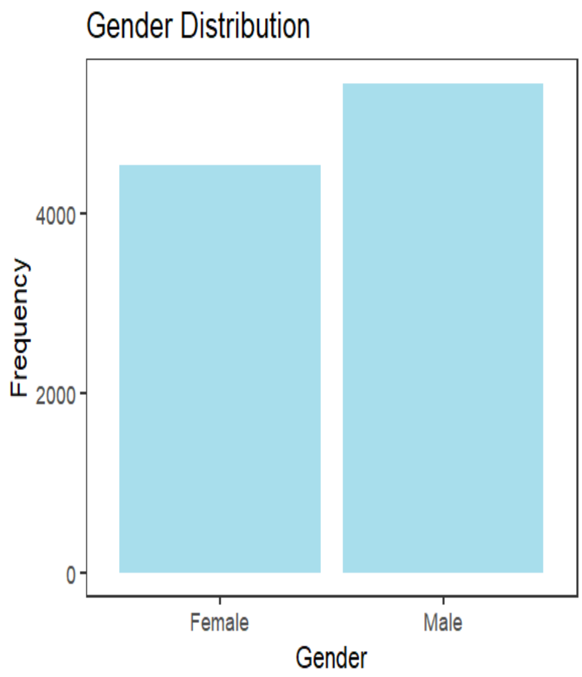


Figure 5: Gender Distribution

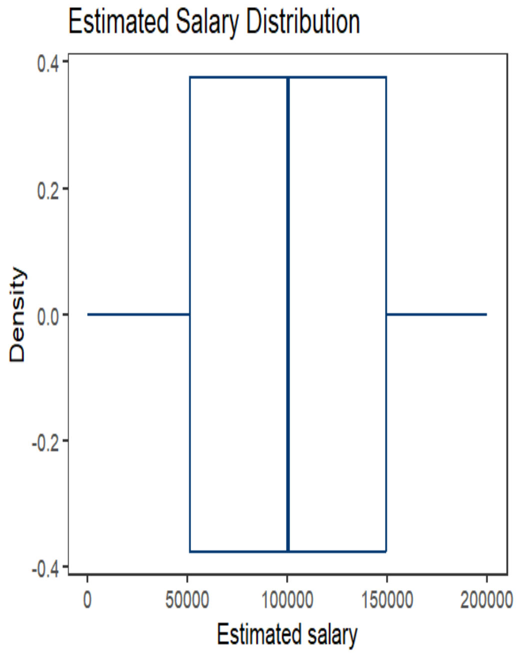


Figure 6: Estimated Salary Distribution

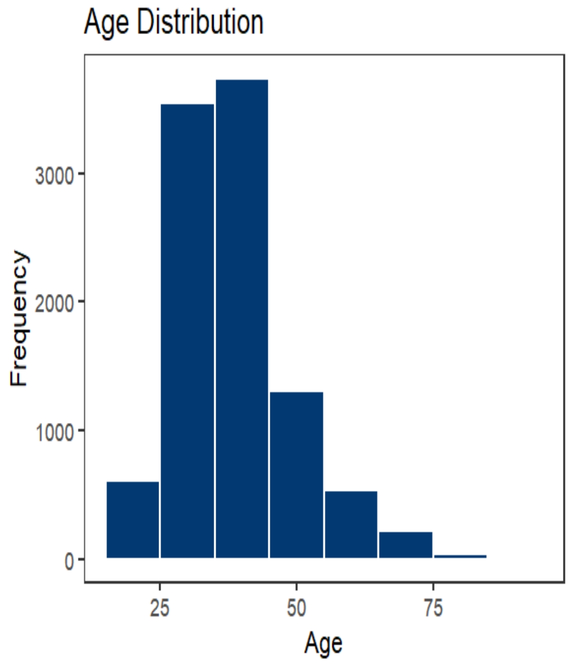


Figure 7: Age Distribution

3.3.4 **Bivariate Analysis**:

- Relationships between the target variable and independent features were thoroughly explored.
- The chart types used for the exploration and the reasons selecting the chart types are as follows:
    - Bar Charts with Normalized Frequency (Figures 8 and 12): These were used to facilitate direct comparisons of categorical feature attributes concerning their potential influence on customer churn.
    - Bar Charts with Absolute Frequency (Figures 9 and 13): These were used to reveal the actual number of customers within each categorical feature attribute, highlighting their potential influence on customer churn.
    - Histograms with Normalized Frequency (Figure 15): These enabled direct comparisons of the distribution of numeric data across histogram bins, providing insights into the potential influence of numeric features on customer churn.
    - Histograms with Absolute Frequency (Figure 14): These were used to display the actual number of customers within each histogram bin of the numeric data, highlighting their potential influence on customer churn.
    - Boxplots (Figures 10 and 11): These were employed to examine the distribution of numeric data attributes in terms of percentiles and to detect potential outliers concerning customer exit (churn) status. Identical boxplots suggest that the numeric feature might have minimal or no influence on customer churn, whereas non-identical boxplots indicate otherwise.
- Some of the key observations from the exploration of the available data include:
    - Approximately 30% of customers in Germany churned, compared to about 15% of customers in France and Spain (Figure 8).
    - About 1,300 non-active customers churned compared to 3,700 non-active customers who did not churn. In contrast, around 800 active customers churned compared to 4,300 active customers who did not churn (Figure 9). This suggests that non-active customers are more likely to churn.
    - The median age of churned customers is approximately 45 years, with most churned customers aged between 40–50 years. Meanwhile, the median age of non-churned customers is around 35 years, with most aged between 30–40 years. This indicates that older customers are more likely to churn (Figure 10).
    - The median estimated salary for both churned and non-churned customers is nearly identical at approximately 100,000 Euros. Both groups predominantly earn between 50,000–150,000 Euros, suggesting that **Estimated Salary** has minimal to no influence on customer churn (Figure 11).

  - The proportion of customers who churned and those who did not churn, with respect to credit card possession, is approximately equal at around 20% (Figure 12). This suggests that credit card possession has minimal to no influence on customer churn.

- Following the exploration of bivariate relationships, it was concluded that features such as **IsActiveMember, Gender, Geography, Age, Balance, CreditScore,** and **NumOfProducts** demonstrated strong predictive potential for the target variable (Figures 8, 9, 10, 13, 14 and 15 – samples of the bivariate analysis visualizations). Conversely, features **like Tenure, EstimatedSalary,** and **HasCrCard** appeared to have limited predictive influence (Figures 11 and 12 – samples of the bivariate analysis visualizations).

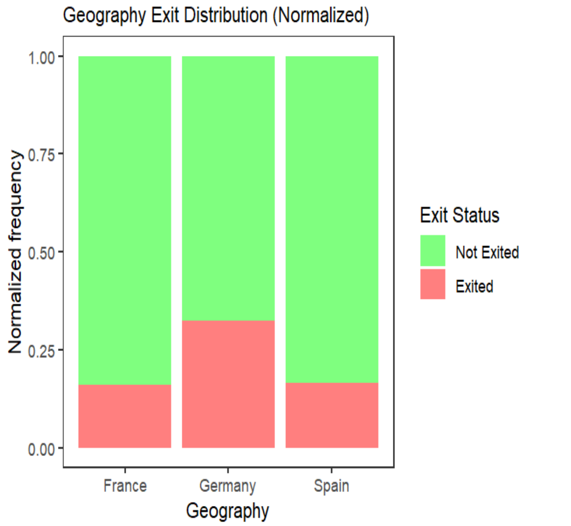


Figure 8: Geography Bivariate Analysis with the Target Feature – shows varied influence across the 100% stacked bar chart columns

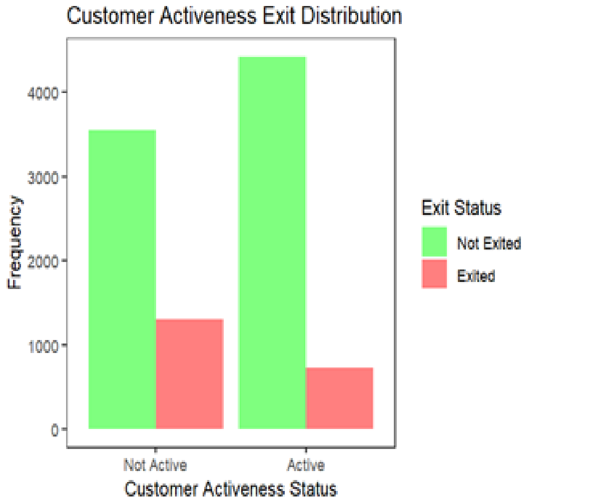


Figure 9: Customer Activeness Bivariate Analysis with the Target Feature – shows varied influence across the clustered bar chart columns

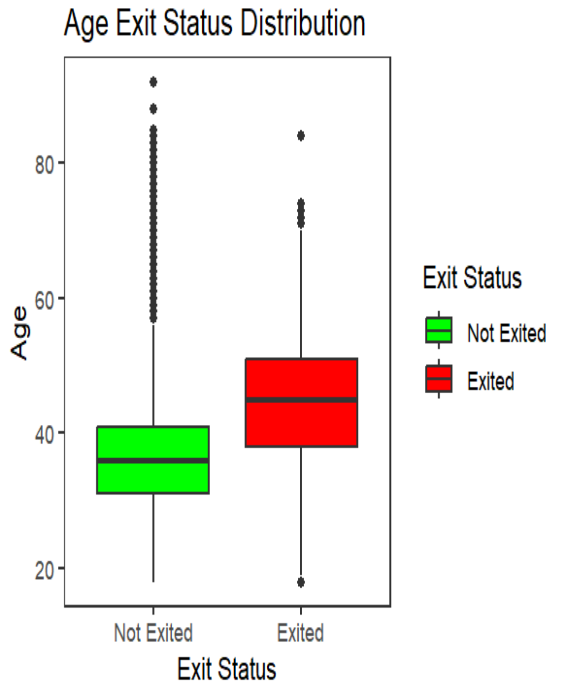


Figure 10: Age Bivariate Analysis with the Target Feature – shows varied influence across the boxplots

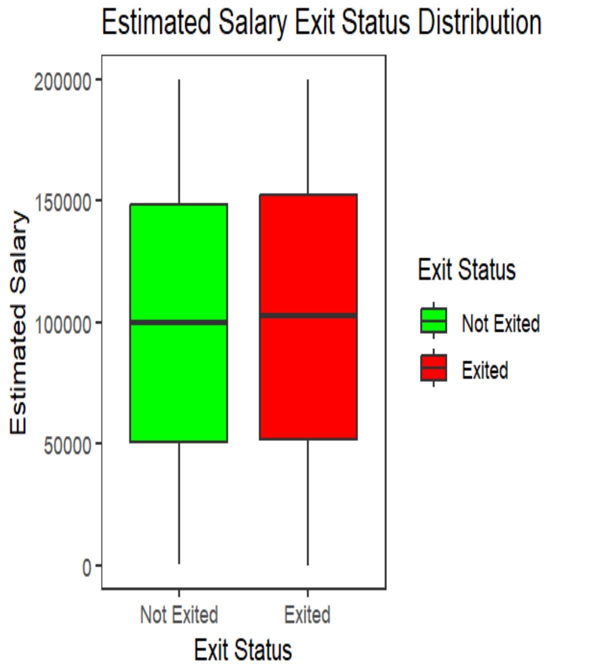


Figure 11: Estimated Salary Bivariate Analysis with the Target Feature – shows similar influence across the boxplots

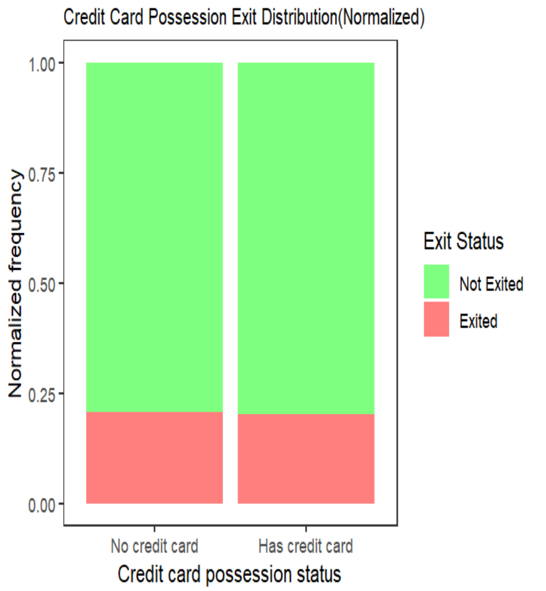


Figure 12: Credit Card Status Bivariate Analysis with the Target Feature - shows similar influence across the 100% stacked bar chart columns.

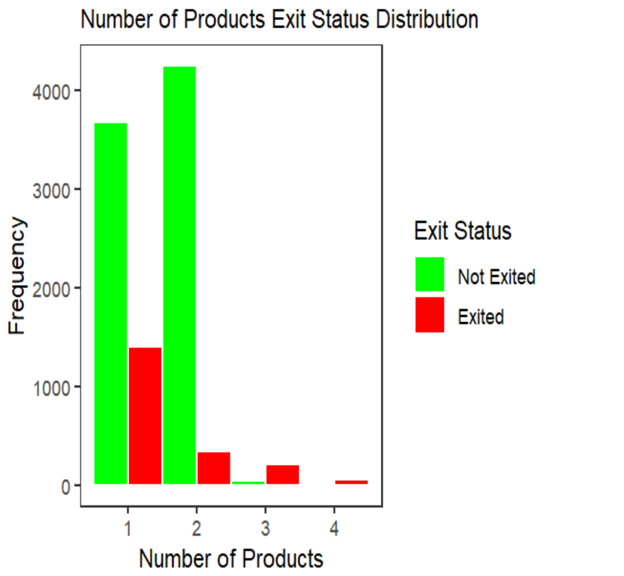


Figure 13: Number of Products Bivariate Analysis with the Target Feature - shows varied influence across the clustered histogram chart bins.

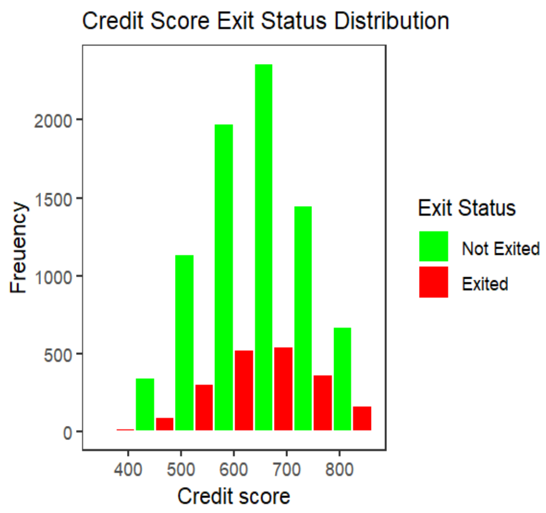


Figure 14: Credit Score Bivariate Analysis with the Target Feature - shows varied influence across the histogram chart bins.

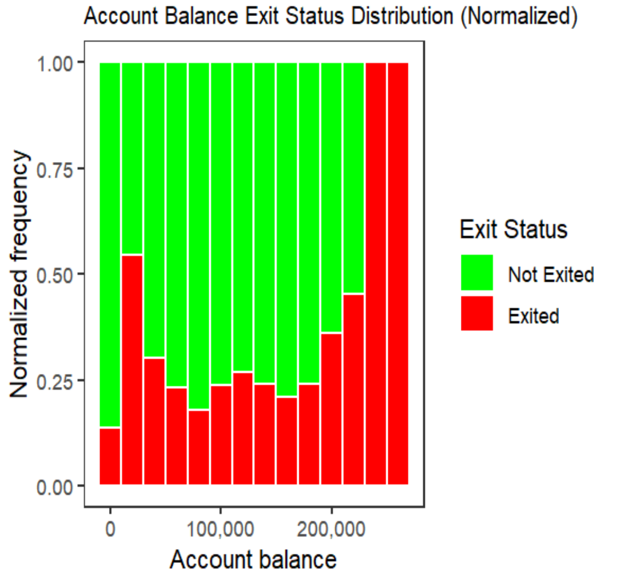
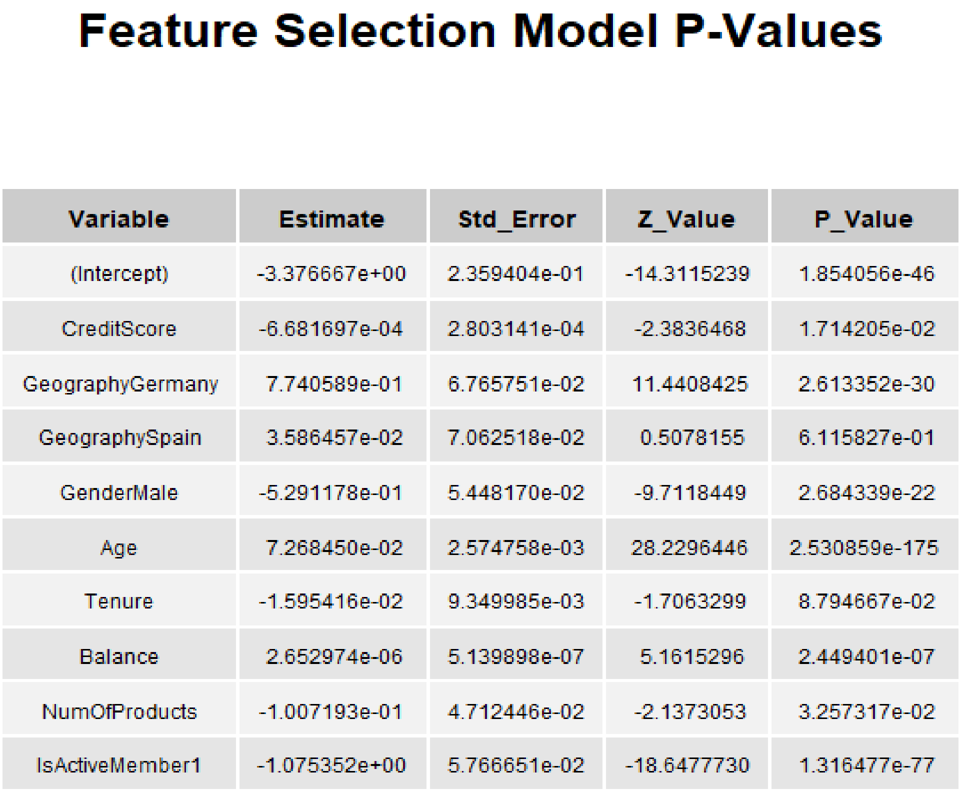


Figure 15: Account Balance Bivariate Analysis with the Target Feature - shows varied influence across the normalized histogram chart bins.

3.3.5 Statistical Significance for Feature Selection:

A Generalized Linear Model (GLM) was employed to evaluate the statistical significance of each independent feature in predicting the target variable. The result largely corroborated the findings from the bivariate analysis, confirming the predictive relevance of several features (Figure 16: P-values of Variables). Specifically, **CreditScore**, **Geography**, **Gender**, **Age**, **Balance**, **NumOfProducts**, and **IsActiveMember** had p-values less than 0.05, indicating statistical significance. Features with a p-value below 0.05 are generally considered significant predictors, while those with p-values above 0.05 are deemed less significant. For further details on the interpretation and application of p-values, refer to Pineda and Sirota (2018) and Chén et al. (2023).



Feature Selection Model P-Values

Variable	Estimate	Std_Error	Z_Value	P_Value
(Intercept)	-3.376667e+00	2.359404e-01	-14.3115239	1.854056e-46
CreditScore	-6.681697e-04	2.803141e-04	-2.3836468	1.714205e-02
GeographyGermany	7.740589e-01	6.765751e-02	11.4408425	2.613352e-30
GeographySpain	3.586457e-02	7.062518e-02	0.5078155	6.115827e-01
GenderMale	-5.291178e-01	5.448170e-02	-9.7118449	2.684339e-22
Age	7.268450e-02	2.574758e-03	28.2296446	2.530859e-175
Tenure	-1.595416e-02	9.349985e-03	-1.7063299	8.794667e-02
Balance	2.652974e-06	5.139898e-07	5.1615296	2.449401e-07
NumOfProducts	-1.007193e-01	4.712446e-02	-2.1373053	3.257317e-02
IsActiveMember1	-1.075352e+00	5.766651e-02	-18.6477730	1.316477e-77

Figure 16: P-values of Variables

3.4 Feature Selection and Pre-Processing for Model Training

3.4.1 Feature Selection

All 10 independent features were retained for model training. This decision was driven by the domain knowledge of the banking industry, which suggests that all features may hold some predictive relevance.

3.4.2 Data Scaling and Transformation

- **Min-Max Scaling**: Min-max scaling was applied to the numerical features to normalize their values within the range (0 - 1), ensuring uniformity in feature magnitudes.
- **One-Hot Encoding**: Categorical features were transformed using one-hot encoding to convert them into a numerical format suitable for machine learning algorithms.

3.4.3 Data Oversampling

To address the class imbalance in the target variable, the Synthetic Minority Oversampling Technique (SMOTE) was employed. This increased the minority class representation from 21% of the total dataset to 42%.

- **Figure 17**: 100% stacked bar chart showing exit status distribution before oversampling.
- **Figure 18**: 100% stacked bar chart showing exit status distribution after oversampling.

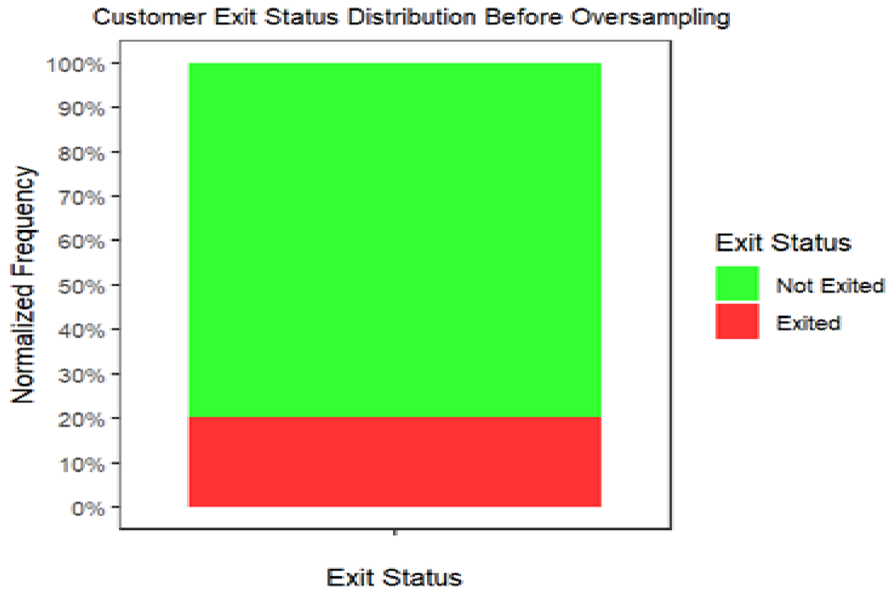


Figure 17: 100% stacked bar chart showing exit status distribution before oversampling

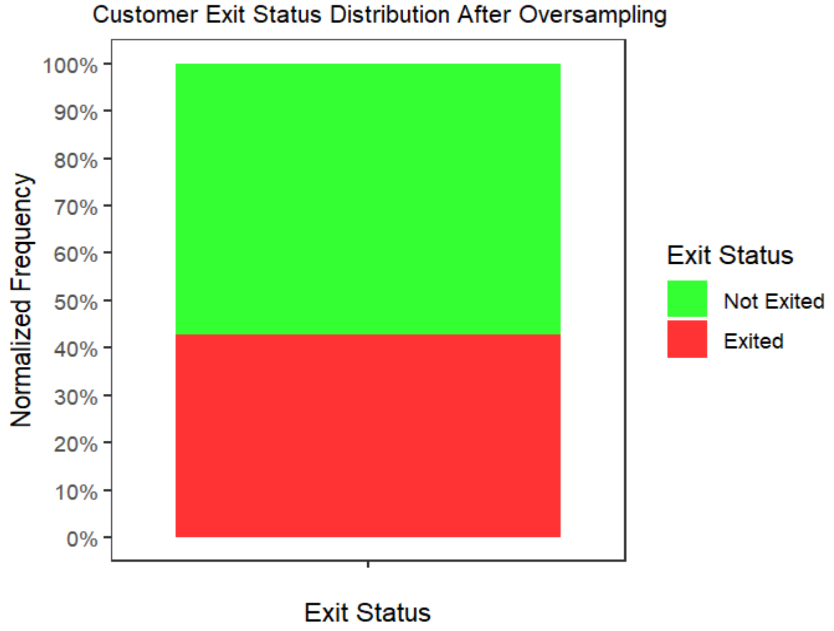


Figure 18: 100% stacked bar chart showing exit status distribution after oversampling

3.4.4 Data Splitting

The dataset was split into training and testing sets using a random sampling method, maintaining a 70:30 ratio. The class distribution in the target variable was analyzed post-splitting to confirm that the split datasets were representative of the overall dataset.

- **Figure 19**: 100% stacked bar chart showing Exit status distribution in the training dataset

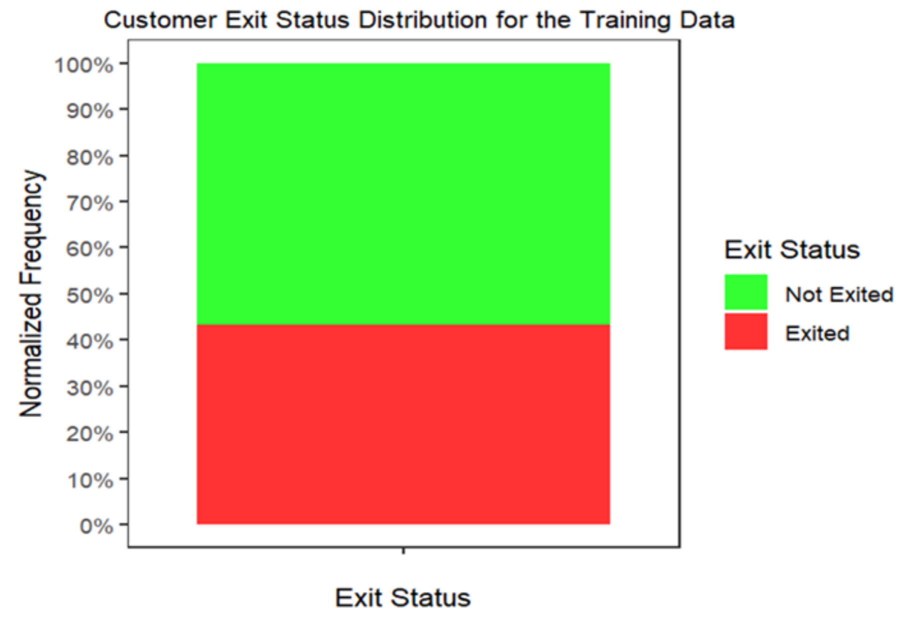


Figure 19: 100% stacked bar chart showing Exit status distribution in the training dataset

3.5 Model Configuration Details

3.5.1 Model Training and Validation

The training and validation of the customer churn prediction KNN classification model were performed using the train function from the caret package. A separate dataset was not allocated for validation, maximizing the utilization of the training dataset. Instead, a 10-fold cross-validation method was employed for model evaluation during training. This ensured robust hyperparameter tuning and improved generalization.

3.5.2 Class Probability and AUC Evaluation

The class probability setting was enabled to facilitate the plotting of the AUROC (Area Under the Receiver Operating Characteristic) curve and the calculation of the AUC (Area Under the Curve) value.

3.5.3 Hyperparameter Tuning

Hyperparameter tuning was conducted using a grid search method to identify the optimal value for the KNN hyperparameter (number of neighbors, k). The initial tuning process explored odd values between 1 and 40, and the optimal k value was determined to be 1 based on the model's averaged performance across the 10 validation sets, providing an estimate of the model's generalization ability.

To reduce computational time, the hyperparameter search range was subsequently refined to odd values between 1 and 10 in the source code. For reproducibility and efficiency, the optimal k value (k=1) can be directly set, further minimizing computational requirements.

3.5.4 Model Testing / Evaluation

After training and hyperparameter tuning, the model's performance was evaluated on the test dataset. The evaluation results were presented using a confusion matrix to analyze classification accuracy across the class labels and an AUROC curve to interpret the model's ability to distinguish between classes.

4.0 PERFORMANCE EVALUATION

The evaluation of the KNN classification model for customer churn prediction is presented as follows:

4.1 Confusion Matrix

The confusion matrix (Figure 20) provides a breakdown of the model's predictions while Figure 21 and table 2 shows details of the metrics derived from the confusion matrix:

- **True Positive (TP)**: 1682 customers correctly identified as churners.
- **True Negative (TN)**: 2214 customers correctly identified as non-churners.
- **False Positive (FP)**: 275 non-churners incorrectly identified as churners (Type I Error).
- **False Negative (FN)**: 107 churners incorrectly identified as non-churners (Type II Error).

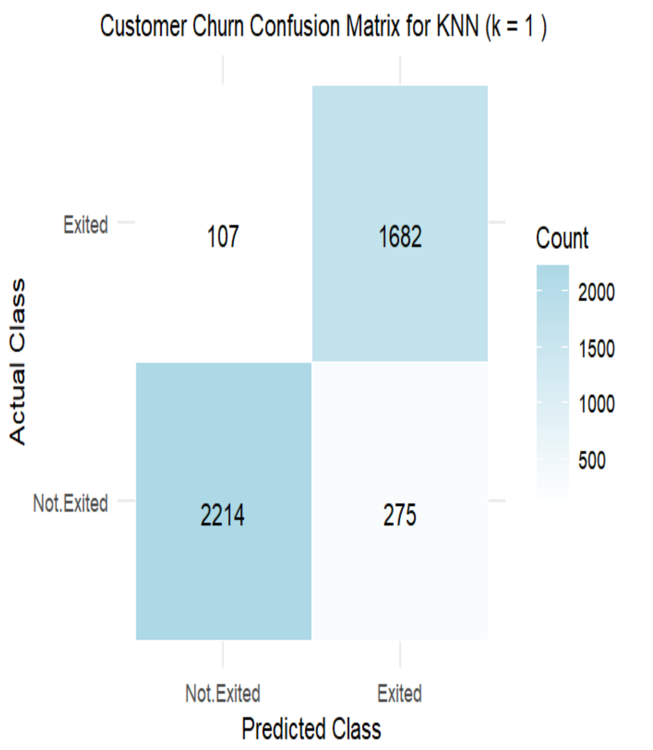
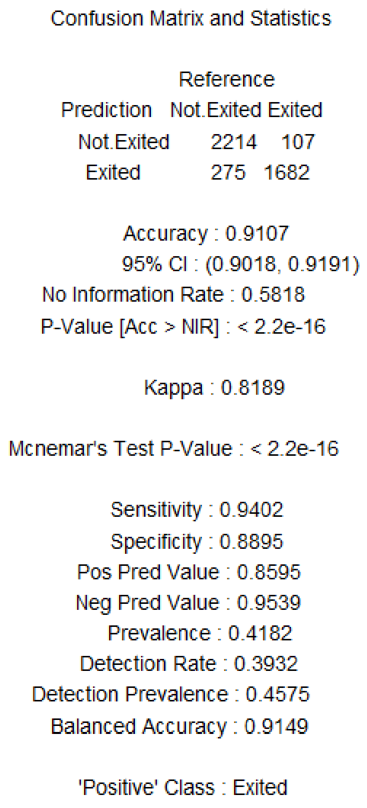


Figure 20: Confusion Matrix



```
Confusion Matrix and Statistics

          Reference
Prediction  Not.Exited Exited
  Not.Exited      2214    107
  Exited           275   1682

               Accuracy : 0.9107
                 95% CI : (0.9018, 0.9191)
    No Information Rate : 0.5818
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8189

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9402
            Specificity : 0.8895
         Pos Pred Value : 0.8595
         Neg Pred Value : 0.9539
             Prevalence : 0.4182
         Detection Rate : 0.3932
   Detection Prevalence : 0.4575
      Balanced Accuracy : 0.9149

       'Positive' Class : Exited
```

Figure 21: Confusion Matrix and Statistics

CONFUSION MATRIX DERIVED METRICS			
S/N	**Metrics**	**Value**	**Interpretation**
1	Accuracy	0.91	91% of the overall predictions were correct. However, accuracy can be misleading for imbalanced dataset. $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
2	Prevalence (Churn Rate)	0.42	42% of the test dataset were churned customers. The imbalance necessitates using other metrics like sensitivity, specificity, precision, F1- score and AUC for a robust evaluation of the model's performance. $Prevalence = \frac{TP+FN}{TP+TN+FP+FN}$
3	Sensitivity (Recall or True Positive Rate)	0.94	94% of the actual churners were correctly predicted. The metric is valuable for retention strategies targeting churn-prone customers. $Sensitivity = \frac{TP}{TP + FN}$
4	Specificity (True Negative Rate)	0.89	89% of the actual non-churners were predicted correctly. High specificity helps avoid unnecessary retention efforts for customers unlikely to churn. $Specificity = \frac{TN}{TN + FP}$
5	Precision (Positive Predictive Value)	0.86	86% of the predicted churned customers were churners. The cost implication for false positives is will be assessed. $Precision = \frac{TP}{TP+TN}$
6	Negative Predictive Value (NPV)	0.95	95% of predicted non-churners were non-churners. High NPV means the model prediction of non-churners are reliable. $NPV = \frac{TN}{TN + FN}$
7	F1-Score	0.90	The harmonic mean of precision and sensitivity is 90%. Helps to balance false positives and false negatives in imbalanced datasets. $F1 = \frac{2\ x\ Precision\ x\ Sensitivity}{Precision+Sensitivity}$
8	Balanced Accuracy	0.92	The average of sensitivity and specificity is 92%. Provides fair evaluation for imbalanced datasets by giving equal importance to both the positive and negative classes. $Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2}$
9	False Positive Rate (FPR)	0.11	11% of actual non-churners were incorrectly predicted as churners. The cost for customer retention will be assessed. (1 – Specificity = 0.11)

			$FPR = \frac{FP}{FP+TN}$
10	False Negative Rate (FNR)	0.06	6% of actual churners were incorrectly predicted as non-churners. This represents missed opportunities to retain customers. (1 – Sensitivity = 0.06) $FNR = \frac{FN}{TP+FN}$
11	False Discovery Rate (FDR)	0.14	14% of the predicted churners were non-churners. (1 − Precision = 0.14) $FDR = \frac{FP}{FP+TP}$
12	Detection Rate (Churn Detection Rate)	0.39	39% of the total customers were accurately predicted to churn, compared to the actual 42% of the total customers who churned. $Detection\ Rate = \frac{TP}{TP + TN + FP + FN}$
13	Detection Prevalence	0.46	46% likelihood for the model to predict a customer to churn, compared to the training baseline likelihood of 42% churning rate. $Detection\ Prevalence = \frac{TP+FP}{TP+TN+FP+FN}$
14	Kappa (Cohen’s Kappa)	0.82	Measures the agreement between the model's predictions and actual outcomes, adjusted for chance. A high Kappa value indicates strong performance.

Table 2: Confusion Matrix Derived Metrics

4.2 AUROC Curve

The AUROC (Area Under Receiver Operating Characteristic) curve (Figure 22) demonstrates the trade-off between the true positive rate (TPR) and false positive rate (FPR) at various thresholds.

AUC Value of 0.91 was obtained. The high AUC value indicates excellent discriminative ability, showcasing the model's effectiveness in distinguishing churned from non-churned customers, even with imbalanced data.

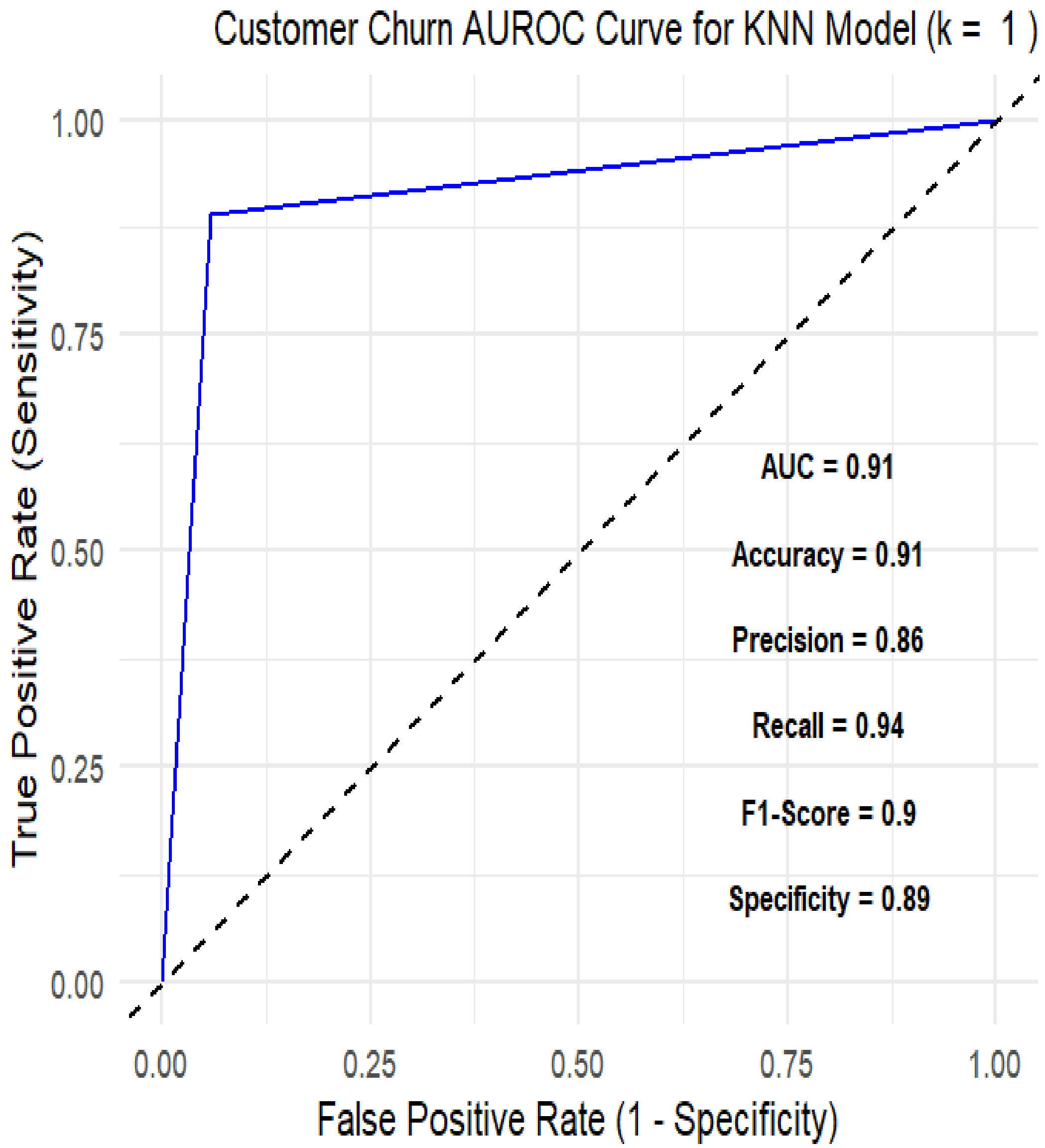


Figure 22: AUROC Curve

5 TECHNICAL DISCUSSIONS AND CONCLUSIONS

5.1 Performance Comparison

This bank customer churn prediction model outperformed the previously reviewed models that utilized the same dataset, regardless of whether the authors applied KNN or other machine learning algorithms. The performance comparisons are detailed in Table 3.

AUTHOR(S)	MODEL AND PERFORMANCE METRICS FOR AUTHORS WHO ANALYZED THE SAME DATASET						REMARKS
	Model	Accuracy	AUC	Recall	Precision	F1 - Score	
Onaolapo	KNN	0.91	0.91	0.94	0.86	0.90	New model
Tékouabou et al. (2022)	KNN	0.68				0.70	Reviewed Models (Published by other authors)
	SVM	0.57				0.64	
	RF	0.86				0.86	
	ET	0.86				0.86	
Zhang (2022)	XGBoost	0.84		0.84	0.83	0.84	
Chen, Liu, and Wang (2023)	AdaBoost	0.81	0.78	0.72	0.51	0.60	
	RF	0.84	0.76	0.64	0.57	0.60	
	Logistics	0.72	0.72	0.72	0.39	0.50	

Table 3: Performance Comparisons

5.2 Conclusions

The KNN classification model demonstrated excellent predictive capabilities in detecting bank customer churn, achieving an accuracy of 91%, a recall of 94%, and an AUC score of 0.91. These results underscore its potential to support proactive customer retention strategies, particularly due to its high sensitivity in identifying potential churners. Despite the robust performance, the model's misclassification rates—11% false positives and 6% false negatives—highlight areas where further refinement could reduce operational costs and improve precision. Overall, the model's performance positions it as a valuable tool for mitigating customer churn in the banking sector.

5.3 Future Directions

- Cost-Sensitive Machine Learning: Incorporating cost-sensitive techniques can enhance model performance by explicitly addressing the financial implications of false positives and false negatives.
- Churn Analysis and Insights: Analyzing data from churned customers can reveal key factors influencing their decisions to leave. Machine learning models that integrate these insights can identify churn triggers, enabling the design of targeted retention strategies. By proactively mitigating these issues, banks can strengthen customer relationships, lower churn rates, and foster long-term profitability and loyalty.

5.4 Potential Security Vulnerabilities

This section provides guidance for securely reproducing this work or to deploy the model for real-world usage. By default, Rtools42 implements Address Space Layout Randomization (ASLR) and Data Execution Prevention (DEP), enhancing the security of R and its packages on Windows. These features protect against common memory-based attacks, creating a safer environment for R users and developers (Comprehensive R Archive Network, 2024).

However, security threats remain, particularly those associated with the unsafe use of R packages. A notable risk is the potential injection of malicious code into packages uploaded to repositories such as CRAN or GitHub by attackers. To mitigate these risks, it is essential to:

- Download packages only from trusted and verified repositories.
- Regularly maintain and update all packages and their dependencies.
- Review the source code of packages prior to installation to identify potential vulnerabilities.

Adopting these measures is critical for minimizing security risks and safeguarding the R working environment. This is particularly vital for predictive models, such as those used in customer churn prediction, where both data confidentiality and system reliability are paramount.

REFERENCES

Chan, J.Y.-L. *et al.* (2022) 'Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review', *Mathematics*, 10, pp.1283. Available at: https://doi.org/10.3390/math10081283.

Chén, O.Y. *et al.* (2023) 'The roles, challenges, and merits of the p value', *Patterns*, 4. Available at: https://doi.org/10.1016/j.patter.2023.100878.

Chen, P., Liu, N. and Wang, B. (2023) 'Evaluation of Customer Behaviour with Machine Learning for Churn Prediction: The Case of Bank *Customer Churn in Europe', in Proceedings of the International Conference on Financial Innovation, FinTech and Information Technology, FFIT 2022*, Shenzhen, October 28-30, 2022. Available at: https://doi.org/10.4108/eai.28-10-2022.2328450.

Malit, K. (2018) *Bank Customer Churn Prediction.* Available at: https://www.kaggle.com/code/kmalit/bank-customer-churn-prediction/notebook (Accessed: 20 October 2024).

Pineda, S. and Sirota, M. (2018) 'Determining Significance in the New Era for *P* Values', *Journal of Pediatric Gastroenterology and Nutrition*, 67(5), pp. 547 – 548. Available at: https://doi.org/10.1097/MOG.0000000000002120.

Tékouabou, S.C.K. et al. (2022) 'Towards Explainable Machine Learning for Bank Churn Prediction Using Data Balancing and Ensemble-Based Methods', *Mathematics*, 10, pp. 2379. Available at: https://doi.org/10.3390/math10142379.

the Comprehensive R Archive Network (2024) *R News*. Available at: https://cran.r-project.org/bin/windows/base/NEWS.R-4.4.2.html (Accessed: 24 December 2024)

Tran, H., Le, N. and Nguyen, V.-H. (2023) 'Customer Churn Prediction in the Banking Sector Using Machine Learning-Based Classification Models', *Interdisciplinary Journal of Information, Knowledge, and Management*, 18, pp. 87–105. Available at: https://doi.org/10.28945/5086.

Zhang, T. (2022) 'Prediction and Clustering of Bank Customer Churn Based on XGBoost and K-means', *BCP Business & Management*, 23, pp. 360 – 366. Available at: https://doi.org/10.54691/bcpbm.v23i.1373.

APPENDIX – ATTACHED R CODE

```
##### GENERAL INFORMATION
#################################################################################

# MODEL: BANK CUSTOMER CHURN PREDICTION

# DATA SOURCE: https://www.kaggle.com/code/kmalit/bank-customer-churn-prediction/data

# PREPARED BY: OLAWALE FRANCIS ONAOLAPO


##### SETTING THE WORKING DIRECTORY
######################################################################

setwd("C:/Users/OMEN 16/Desktop/dsfica")

getwd()


##### LOADING THE REQUIRED PACKAGES
######################################################################

if (!require("tidyverse")) { install.packages("tidyverse", dependencies = TRUE); library(tidyverse) } else { library(tidyverse) }

if (!require("corrplot")) { install.packages("corrplot", dependencies = TRUE); library(corrplot) } else { library(corrplot) }

if (!require("car")) { install.packages("car", dependencies = TRUE); library(car) } else { library(car) }

if (!require("caret")) { install.packages("caret", dependencies = TRUE); library(caret) } else { library(caret) }

if (!require("pROC")) { install.packages("pROC", dependencies = TRUE); library(pROC) } else { library(pROC) }

if (!require("reshape2")) { install.packages("reshape2", dependencies = TRUE); library(reshape2) } else { library(reshape2) }

if (!require("gridExtra")) { install.packages("gridExtra", dependencies = TRUE); library(gridExtra) } else { library(gridExtra) }

if (!require("MASS")) { install.packages("MASS", dependencies = TRUE); library(MASS) } else { library(MASS) }
if (!require("grid")) { install.packages("grid", dependencies = TRUE); library(grid) } else {
library(grid) }

if (!require("DMwR")) { install.packages("DMwR", dependencies = TRUE); library(DMwR) } else {
library(DMwR) }


##### IMPORTING THE CUSTOMER CHURN DATASET
#######################################################

customer_churn <- read_csv("churn_modelling.csv")


##### DATA PREPARATION - DATA CLEANING / DATA PREPROCESSING (1)
###################################

# TO VIEW THE IMPORTED DATASET

View(customer_churn)


# TO CHECK THE NUMBER OF ROWS AND COLUMNNS IN THE IMPORTED DATASET

dim(customer_churn)


# TO CHECK FOR DUPLICATES

sum(duplicated(customer_churn)) # No duplicate


# TO CHECK FOR DUPLICATES WITH THE CUSTOMER ID

length(unique(customer_churn$CustomerId)) # No duplicate - 10000 unique IDs


# TO CHECK FOR MISSING VALUES

colSums(is.na(customer_churn)) # No missing value


# TO CHECK THE STRUCTURE SUMMARY
```

```
str(customer_churn)


# TO CHECK THE DESCRIPTIVE SUMMARY

summary(customer_churn)


# TO CHECK THE NUMBER OF UNIQUE VALUES IN THE DEPENDENT VARIABLE

length(unique(customer_churn$Exited)) # TO VERIFY THE NUMBERS OF UNIQUE VALUES IN THE
EXITED COLUMN


# TO REMOVE SOME COLUMNS NOT NEEDED

customer_churn$RowNumber <- NULL

customer_churn$CustomerId <- NULL

customer_churn$Surname <- NULL


# TO CONVERT CATEGORICAL VARIABLES IN NUMERIC TO FACTOR

customer_churn$HasCrCard <- as.factor(customer_churn$HasCrCard)

customer_churn$IsActiveMember <- as.factor(customer_churn$IsActiveMember)

customer_churn$Exited <- as.factor(customer_churn$Exited)


str(customer_churn)


##### EXPLORATORY DATA ANALYSIS (1)
#############################################################################

# DETECTING OUTLIERS USING Z-SCORES WITH 3 STANDARD DEVIATIONS FROM THE MEAN

customer_churn_numeric <- customer_churn %>%                    # Selecting the numeric columns

  select_if(is.numeric)
```

```
customer_churn_z_scores <- scale(customer_churn_numeric)        # Calculate z-scores


customer_churn_outliers <- lapply(1:ncol(customer_churn_z_scores), function(x) {


  outlier_indices <- which(abs(customer_churn_z_scores[, x]) > 3) # Get the indices of the
outliers


  outlier_values <- customer_churn_numeric[outlier_indices, x]    # Get the values of the outliers


  data.frame(Row = outlier_indices, Value = outlier_values)       # Return both the row numbers
and values

})


customer_churn_outliers      # To show the outliers and their row number


# TO CHECK FOR MULTICOLLINEARITY

# Correlation between numerical variables

customer_churn_num_corr <- cor(customer_churn_numeric)

corrplot(customer_churn_num_corr, main = "\nNumerical Features Correlation Plot", type =
"lower",

        method = "number", cl.cex = 0.5)


# Using Variance Inflation Factor (VIF) to detect multicollinearity on a generalized linear model
of the dataset

set.seed(42)

customer_churn_vif_model <- glm(Exited ~ ., data = customer_churn, family = binomial)

vif(customer_churn_vif_model)          # To calculate the GVIF^(1/(2*Df))
```

```
# TO PLOT THE VIF

plot_customer_churn_vif <- vif(customer_churn_vif_model)


plot_customer_churn_vif <- data.frame(plot_customer_churn_vif)


colnames(plot_customer_churn_vif)[3]<- "customer_churn_independent_features_vif"


ggplot(plot_customer_churn_vif, aes(x=rownames(plot_customer_churn_vif),
y=(customer_churn_independent_features_vif)))+
  geom_bar(stat = "identity", fill = "tan") +
  geom_text(aes(label = round(customer_churn_independent_features_vif, 2)), , vjust = -0.5) +
  labs(title = "Variance Inflation Factor (VIF)", x = "Independent Features", y = "GVIF^(1/(2*Df))")
+
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_cartesian(ylim = c(0, 1.2))


##### EXPLORATORY DATA ANALYSIS (2) ##### UNIVARIATE ANALYSIS
##########################################################

# TO CHECK THE DATA DISTRIBUTION FOR THE CATEGORICAL VARIABLES / VARIABLES WITH
QUALITATIVE DATA


# To check the gender data distribution

customer_churn %>%
  ggplot(aes(x = Gender)) +
  geom_bar(position = "dodge", alpha = 0.5, fill = "#52beda") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Gender Distribution", x = "Gender", y = "Frequency")

# To check the geography data distribution
customer_churn %>%
  ggplot(aes(x = Geography)) +
  geom_bar(position = "dodge", alpha = 0.5, fill = "#52beda") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Distribution across the Geography", x = "Geography", y = "Frequency")

# To check the data distribution for credit card ownership
customer_churn %>%
  ggplot(aes(x = HasCrCard)) +
  geom_bar(position = "dodge", alpha = 0.5, fill = "#52beda") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Credit Card Ownership Distribution", x = "Credit Card Ownership", y =
"Frequency")+
  scale_x_discrete(labels = c("0" = "No credit card", "1" = "Has credit card"))

# To check the data distribution for the customer activeness
customer_churn %>%
  ggplot(aes(x = IsActiveMember)) +
  geom_bar(position = "dodge", alpha = 0.5, fill = "#52beda") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Customer Activeness Distribution", x = "Customer Activeness", y = "Frequency")+
  scale_x_discrete(labels = c("0" = "Not Active", "1" = "Active"))

# To check the customer exit distribution
customer_churn %>%
  ggplot(aes(x = Exited)) +
  geom_bar(position = "dodge", alpha = 0.5, fill = "#52beda") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Customer Exit Status Distribution", x = "Customer Exit Status", y = "Frequency")+
  scale_x_discrete(labels = c("0" = "Not Exit", "1" = "Exit"))

##### EXPLORATORY DATA ANALYSIS (3) ##### UNIVARIATE ANALYSIS
################################################
# TO CHECK THE DATA DISTRIBUTION FOR THE QUANTITATIVE VARIABLES USING HISTOGRAM
AND BOX PLOT

##### USING HISTOGRAM #####
# Histogram illustrating the number of products distribution
customer_churn %>%
  ggplot(aes(x = NumOfProducts)) +
  geom_histogram(binwidth = 1, color = "white", fill = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Number of Products Distribution", x = "Number of Products", y = "Frequency")
```

```
# Histogram illustrating the tenure distribution
customer_churn %>%
  ggplot(aes(x = Tenure)) +
  geom_histogram(binwidth = 2, color = "white", fill = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Tenure Distribution", x = "Tenure", y = "Frequency")

# Histogram illustrating the age distribution
customer_churn %>%
  ggplot(aes(x = Age)) +
  geom_histogram(binwidth = 10, color = "white", fill = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Age Distribution", x = "Age", y = "Frequency")

# Histogram illustrating the credit score distribution
customer_churn %>%
  ggplot(aes(x = CreditScore)) +
  geom_histogram(binwidth = 75, color = "white", fill = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Credit Score Distribution", x = "Credit score", y = "Frequency")

# Histogram illustrating the estimated salary distribution
```

```
customer_churn %>%
  ggplot(aes(x = EstimatedSalary)) +
  geom_histogram(binwidth = 10000, color = "white", fill = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Estimated Salary Distribution", x = "Estimated salary", y = "Frequency")

# Histogram illustrating the account balance distribution
customer_churn %>%
  ggplot(aes(x = Balance)) +
  geom_histogram(binwidth = 20000, color = "white", fill = "#023972") +
  scale_x_continuous(labels = scales::comma) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Account Balance Distribution", x = "Balance", y = "Frequency")

######## USING BOXPLOT #####
# THE OUTLIERS DETECTED BY THE BOXPLOT CHECKED AND WERE NOT OUTLIERS IN THIS CONTEXT

# Boxplot illustrating the age distribution
customer_churn %>%
  ggplot(aes(x = Age)) +
  geom_boxplot(color = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Age Distribution", x = "Age", y = "Density")

# Boxplot illustrating the tenure distribution
customer_churn %>%
  ggplot(aes(x = Tenure)) +
  geom_boxplot(color = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Tenure Distribution", x = "Tenure", y = "Density")

# Boxplot illustrating the account balance distribution
customer_churn %>%
  ggplot(aes(x = Balance)) +
  geom_boxplot(color = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Account Balance Distribution", x = "Balance", y = "Density")

# Boxplot illustrating the distribution for the number of products by the customer
customer_churn %>%
  ggplot(aes(x = NumOfProducts)) +
  geom_boxplot(color = "#023972") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Distribution of the Number of Products", x = "Number of products", y = "Density")
```

```
# Boxplot illustrating the distribution for the estimated salary by the customer

customer_churn %>%

  ggplot(aes(x = EstimatedSalary)) +

  geom_boxplot(color = "#023972") +

  theme_bw() +

  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +

  labs(title = "Estimated Salary Distribution", x = "Estimated salary", y = "Density")


# Boxplot illustrating the credit score distribution

customer_churn %>%

  ggplot(aes(x = CreditScore)) +

  geom_boxplot(color = "#023972") +

  theme_bw() +

  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +

  labs(title = "Credit Score Distribution", x = "Credit score", y = "Density")


##### EXPLORATORY DATA ANALYSIS (4) ##### BIVARIATE ANALYSIS
########################################################

# TO CHECK THE EXIT STATUS DISTRIBUTION WITH REFERENCE TO BOTH THE CATEGORICAL
FEATURES AND THE NUMERICAL FEATURES


##### BIVARIATE ANALYSIS ### EXIT STATUS DISTRIBUTION WITH REFERENCE TO THE
CATEGORICAL FEATURES ####


# To check the the Exit Distribution with regards to the gender

customer_churn %>%

  filter(Gender == "Female" | Gender == "Male") %>%
  ggplot(aes(x = Gender, fill = (Exited))) +
  geom_bar(position = "dodge", alpha = 0.5) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Gender Exit Distribution",
       x = "Gender", y = "Frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the the normalized Exit Distribution with regards to the gender
customer_churn %>%
  filter(Gender == "Female" | Gender == "Male") %>%
  ggplot(aes(x = Gender, fill = (Exited))) +
  geom_bar(position = "fill", alpha = 0.5) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Gender Exit Distribution (Normalized)", fill = "Exit Status",
       x = "Gender", y = "Normalized frequency")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the the Exit Distribution with regards to geography
customer_churn %>%
  filter(Geography == "France" | Geography == "Spain" | Geography == "Germany") %>%
  ggplot(aes(x = Geography, fill = (Exited))) +
  geom_bar(position = "dodge", alpha = 0.5) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Geography Exit Distribution",
      x = "Geography", y = "Frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the the normalized Exit Distribution with regards to the geography
customer_churn %>%
  filter(Geography == "France" | Geography == "Spain" | Geography == "Germany") %>%
  ggplot(aes(x = Geography, fill = (Exited))) +
  geom_bar(position = "fill", alpha = 0.5) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
       plot.title = element_text(size = 11)) +
  labs(title = "Geography Exit Distribution (Normalized)",
      x = "Geography", y = "Normalized frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the the Exit Distribution of the customer activeness status
customer_churn %>%
  filter(IsActiveMember == 1 | IsActiveMember == 0) %>%
  ggplot(aes(x = IsActiveMember, fill = (Exited))) +
  geom_bar(position = "dodge", alpha = 0.5) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Customer Activeness Exit Distribution",
      x = "Customer Activeness Status", y = "Frequency", fill = "Exit Status")+
  scale_x_discrete(labels = c("0" = "Not Active", "1" = "Active"))+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the the normalized Exit Distribution of the customer activeness status
customer_churn %>%
  filter(IsActiveMember == 1 | IsActiveMember == 0) %>%
  ggplot(aes(x = IsActiveMember, fill = (Exited))) +
  geom_bar(position = "fill", alpha = 0.5) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 10.5)) +
  labs(title = "Customer Activeness Exit Distribution (Normalized)",
       x = "Customer Activeness", y = "Normalized frequency", fill = "Exit Status")+
  scale_x_discrete(labels = c("0" = "Not Active", "1" = "Active"))+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the the Exit Distribution of the customer who has credit card
customer_churn %>%
  filter(HasCrCard == 1 | HasCrCard == 0) %>%
  ggplot(aes(x = HasCrCard, fill = (Exited))) +
  geom_bar(position = "dodge", alpha = 0.5) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 12)) +
  labs(title = "Credit Card Possession Exit Distribution",
       x = "Credit card possession status", y = "Frequency", fill = "Exit Status")+
  scale_x_discrete(labels = c("0" = "No credit card", "1" = "Has credit card"))+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the the normalized Exit Distribution of the customer who has credit card
customer_churn %>%
  filter(HasCrCard == 1 | HasCrCard == 0) %>%
  ggplot(aes(x = HasCrCard, fill = (Exited))) +
  geom_bar(position = "fill", alpha = 0.5) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 10)) +
  labs(title = "Credit Card Possession Exit Distribution(Normalized)",
       x = "Credit card possession status", y = "Normalized frequency", fill = "Exit Status")+
  scale_x_discrete(labels = c("0" = "No credit card", "1" = "Has credit card"))+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

##### BIVARIATE ANALYSIS ### EXIT STATUS DISTRIBUTION WITH REFERENCE TO THE
QUANTITATIVE FEATURES ### USING BOXPLOT #####
# To check the exit status distribution across age using a boxplot
customer_churn %>%
  ggplot(aes(x = (Exited), y = Age, fill = (Exited))) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited")) +
  scale_x_discrete(labels = c("0" = "Not Exited", "1" = "Exited"))+
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Age Exit Status Distribution",
      x = "Exit Status", y = "Age", fill = "Exit Status")

# To check the exit status distribution across credit score using boxplot
customer_churn %>%
  ggplot(aes(x = (Exited), y = CreditScore, fill = (Exited))) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited")) +
  scale_x_discrete(labels = c("0" = "Not Exited", "1" = "Exited"))+
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Credit Score Exit Status Distribution",
      x = "Exit Status", y = "Credit Score", fill = "Exit Status")

# To check the exit status distribution for the account balance using boxplot
customer_churn %>%
  ggplot(aes(x = (Exited), y = Balance, fill = (Exited))) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited")) +
  scale_x_discrete(labels = c("0" = "Not Exited", "1" = "Exited"))+
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Balance Exit Status Distribution",
      x = "Exit Status", y = "Balance", fill = "Exit Status")

# To check the exit status distribution for the estimated salary using boxplot
customer_churn %>%
  ggplot(aes(x = (Exited), y = EstimatedSalary, fill = (Exited))) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited")) +
  scale_x_discrete(labels = c("0" = "Not Exited", "1" = "Exited"))+
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Estimated Salary Exit Status Distribution",
       x = "Exit Status", y = "Estimated Salary", fill = "Exit Status")

# To check the exit status distribution for the number of products using boxplot
customer_churn %>%
  ggplot(aes(x = (Exited), y = NumOfProducts, fill = (Exited))) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited")) +
  scale_x_discrete(labels = c("0" = "Not Exited", "1" = "Exited"))+
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Number Of Products Exit Status Distribution",
       x = "Exit Status", y = "Number Of Products", fill = "Exit Status")

# To check the exit status distribution for the tenure using boxplot
customer_churn %>%
  ggplot(aes(x = (Exited), y = Tenure, fill = (Exited))) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited")) +
  scale_x_discrete(labels = c("0" = "Not Exited", "1" = "Exited"))+
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Tenure Exit Status Distribution",
       x = "Exit Status", y = "Tenure", fill = "Exit Status")
```

```
##### BIVARIATE ANALYSIS ### EXIT STATUS DISTRIBUTION WITH REFERENCE TO THE QUANTITATIVE FEATURES ##### USING HISTOGRAM #####
# To check the exit status distribution for the credit score using histogram
customer_churn %>%
  ggplot(aes(x = CreditScore, fill = (Exited))) +
  geom_histogram(binwidth = 75, position = 'dodge', color = "white") +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited")) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 12)) +
  labs(title = "Credit Score Exit Status Distribution",
       x = "Credit score", y = "Frequency", fill = "Exit Status")
```

```
# To check the normalized exit status distribution for the credit score using histogram
customer_churn %>%
  ggplot(aes(x = CreditScore, fill = (Exited))) +
  geom_histogram(binwidth = 75, position = 'fill', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 11)) +
  labs(title = "Credit Score Exit Status Distribution (Normalized)",
      x = "Credit score", y = "Normalized freuency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the exit status distribution for the age using histogram
customer_churn %>%
  ggplot(aes(x = Age, fill = (Exited))) +
  geom_histogram(binwidth = 10, position = 'dodge', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Age Exit Status Distribution",
      x = "Age", y = "Frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the normalized exit status distribution for the age using histogram
customer_churn %>%
  ggplot(aes(x = Age, fill = (Exited))) +
  geom_histogram(binwidth = 10, position = 'fill', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Age Exit Status Distribution (Normalized)",
      x = "Age", y = "Normalized frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the exit status distribution for the tenure using histogram
customer_churn %>%
  ggplot(aes(x = Tenure, fill = (Exited))) +
  geom_histogram(binwidth = 2, position = 'dodge', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Tenure Exit Status Distribution",
       x = "Tenure", y = "Frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the normalized exit status distribution for the tenure using histogram
customer_churn %>%
  ggplot(aes(x = Tenure, fill = (Exited))) +
  geom_histogram(binwidth = 2, position = 'fill', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 12)) +
  labs(title = "Tenure Exit Status Distribution (Normalized)",
       x = "Tenure", y = "Normalized frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the exit status distribution for the account balance using histogram
customer_churn %>%
  ggplot(aes(x = Balance, fill = (Exited))) +
  geom_histogram(binwidth = 20000, position = 'dodge', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Account Balance Exit Status Distribution",
       x = "Account balance", y = "Frequency", fill = "Exit Status")+
  scale_x_continuous(labels = scales::comma) +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the normalized exit status distribution for the account balance using histogram
customer_churn %>%
  ggplot(aes(x = Balance, fill = (Exited))) +
  geom_histogram(binwidth = 20000, position = 'fill', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 10)) +
  labs(title = "Account Balance Exit Status Distribution (Normalized)",
       x = "Account balance", y = "Normalized frequency", fill = "Exit Status")+
  scale_x_continuous(labels = scales::comma) +
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the exit status distribution for the estimated salary using histogram
customer_churn %>%
  ggplot(aes(x = EstimatedSalary, fill = (Exited))) +
  geom_histogram(binwidth = 10000, position = 'dodge', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(title = "Estimated Salary Exit Status Distribution",
       x = "Customer Estimated Salary", y = "Frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the normalized exit status distribution for the estimated salary using histogram
```

```
customer_churn %>%
  ggplot(aes(x = EstimatedSalary, fill = (Exited))) +
  geom_histogram(binwidth = 10000, position = 'fill', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 10)) +
  labs(title = "Estimated Salary Exit Status Distribution (Normalized)",
       x = "Customer Estimated Salary", y = "Normalized frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the exit status distribution for the number of products using histogram
customer_churn %>%
  ggplot(aes(x = NumOfProducts, fill = (Exited))) +
  geom_histogram(binwidth = 1, position = 'dodge', color = "white") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        plot.title = element_text(size = 11)) +
  labs(title = "Number of Products Exit Status Distribution",
       x = "Number of Products", y = "Frequency", fill = "Exit Status")+
  scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))

# To check the normalized exit status distribution for the number of products using histogram
customer_churn %>%
  ggplot(aes(x = NumOfProducts, fill = (Exited))) +
  geom_histogram(binwidth = 1, position = 'fill', color = "white") +
  theme_bw() +
 theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
       plot.title = element_text(hjust = 0.2, size = 10)) +
 labs(title = "Number of Products Exit Status Distribution (Normalized)",
      x = "Number of Products", y = "Normalized frequency", fill = "Exit Status")+
 scale_fill_manual(values = c("0" = "green", "1" = "red"), labels = c("Not Exited", "Exited"))


##### TO CHECK STATISTICALLY SIGNIFICANT VARIABLES USING THE GLM FUNCTION
######################################
# To change the variable data type as required
customer_churn$Gender <- as.factor(customer_churn$Gender)
customer_churn$Geography <- as.factor(customer_churn$Geography)


str(customer_churn)


# Fitting the generalized linear model
set.seed(42)
customer_churn_feat_sel <- glm(Exited ~ ., data = customer_churn, family = binomial)


# Stepwise selection using AIC (backward selection)
customer_churn_feat_sel_step_model <- stepAIC(customer_churn_feat_sel, direction =
"backward")


summary(customer_churn_feat_sel_step_model)   # Summary of the final model


# Extracting the summary of the final model
customer_churn_feat_sel_step_model_summary <-
summary(customer_churn_feat_sel_step_model)
```

```
# Extracting coefficients and related statistics
customer_churn_feat_selection_coefs <-
as.data.frame(customer_churn_feat_sel_step_model_summary$coefficients)
customer_churn_feat_selection_coefs <- customer_churn_feat_selection_coefs %>%
  rownames_to_column(var = "Variable") %>%
  rename(Estimate = Estimate,
    Std_Error = `Std. Error`,
    Z_Value = `z value`,
    P_Value = `Pr(>|z|)`
  )

customer_churn_custom_theme <- ttheme_default(core = list(fg_params = list(cex = 0.5)),
  colhead = list(fg_params = list(cex = 0.6)))

customer_churn_table_grob <- tableGrob(customer_churn_feat_selection_coefs,
  rows = NULL, theme = customer_churn_custom_theme)

customer_churn_title_grob <- textGrob("Feature Selection Model P-Values",
  gp = gpar(fontsize = 14, fontface = "bold"))

customer_churn_combined_grob <- grid.arrange(customer_churn_title_grob,
  customer_churn_table_grob, nrow = 2, heights = c(0.2, 1))

#grid.newpage()
#grid.draw(customer_churn_combined_grob)
```

```
############################################################################

customer_churn$Exited <- factor(customer_churn$Exited, levels = c(0, 1), labels =
c("Not.Exited", "Exited"))


customer_churn_features <- customer_churn[, !(names(customer_churn) %in% "Exited")] #
Separate features and target variable

customer_churn_target <- customer_churn$Exited



customer_churn_numerical_columns <- sapply(customer_churn_features, is.numeric) #
Numerical and categorical columns

customer_churn_categorical_columns <- !customer_churn_numerical_columns


#View(customer_churn_numerical_columns)

#View(customer_churn_categorical_columns)


# Define normalization function for numerical features using the min-max scaling method

customer_churn_normalize_function <- function(x) {return((x - min(x)) / (max(x) - min(x)))}


customer_churn_normalized_features <- as.data.frame(lapply(customer_churn_features[,
customer_churn_numerical_columns],

                                                            customer_churn_normalize_function)) # Normalize numerical
features


# Apply one-hot encoding to categorical features

customer_churn_categorical_features <- customer_churn_features[,
customer_churn_categorical_columns]
```

```
customer_churn_encoded_features <- as.data.frame(model.matrix(~ . - 1, data = customer_churn_categorical_features))


# Combine normalized numerical features and one-hot encoded categorical features

customer_churn_scaled_features <- cbind(customer_churn_normalized_features, customer_churn_encoded_features)


# Combine scaled features and target variable into a single data frame

customer_churn_scaled <- data.frame(customer_churn_scaled_features, Exited = customer_churn_target)


# Create the stacked bar chart with normalized frequencies before oversampling

ggplot(customer_churn_scaled, aes(x = "", fill = factor(Exited))) +

  geom_bar(position = "fill", alpha = 0.8) +

  scale_y_continuous(labels = scales::percent, breaks = seq(0, 1, by = 0.1)) +

  labs(title = "Customer Exit Status Distribution Before Oversampling",

       x = "Exit Status", y = "Normalized Frequency", fill = "Exit Status") +

  scale_fill_manual(values = c("green", "red"), labels = c("Not Exited", "Exited")) +

  theme_bw() +

  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.3, size = 10))


# Apply SMOTE oversampling (200%)

set.seed(42)

customer_churn_smote <- SMOTE(Exited ~ ., data = customer_churn_scaled, K = 5, perc.over = 200)


# Create the stacked bar chart with normalized frequencies after oversampling
```

```
ggplot(customer_churn_smote, aes(x = "", fill = factor(Exited))) +
  geom_bar(position = "fill", alpha = 0.8) +
  scale_y_continuous(labels = scales::percent, breaks = seq(0, 1, by = 0.1)) +
  labs(title = "Customer Exit Status Distribution After Oversampling",
       x = "Exit Status", y = "Normalized Frequency", fill = "Exit Status") +
  scale_fill_manual(values = c("green", "red"), labels = c("Not Exited", "Exited")) +
  theme_bw() +
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.3, size = 10))

# Split data into training and testing sets (70:30 split)
set.seed(42)
customer_churn_intrain <- sample(1:nrow(customer_churn_smote), size = floor(0.7 * nrow(customer_churn_smote)), replace = FALSE)
customer_churn_train_data <- customer_churn_smote[customer_churn_intrain, ]
customer_churn_test_data <- customer_churn_smote[-customer_churn_intrain, ]
#View(customer_churn_train_data)

# Create the stacked bar chart with normalized frequencies for the training data after random sampling split
ggplot(customer_churn_train_data, aes(x = "", fill = factor(Exited))) +
  geom_bar(position = "fill", alpha = 0.8) +
  scale_y_continuous(labels = scales::percent, breaks = seq(0, 1, by = 0.1)) +
  labs(title = "Customer Exit Status Distribution for the Training Data",
       x = "Exit Status", y = "Normalized Frequency", fill = "Exit Status") +
  scale_fill_manual(values = c("green", "red"), labels = c("Not Exited", "Exited")) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
     plot.title = element_text(hjust = 0.3, vjust = 0.2, size = 10))

# Define train control for KNN
customer_churn_train_control <- trainControl(method = "cv", number = 10, classProbs = TRUE)

# To define the range of k values for grid search
customer_churn_k_values <- data.frame(k = seq(1, 10, by = 2))

# Train the KNN model
set.seed(42)
customer_churn_knn_model <- train(Exited ~ ., data = customer_churn_train_data, method = "knn",
  trControl = customer_churn_train_control, tuneGrid = customer_churn_k_values)

customer_churn_best_k <- customer_churn_knn_model$bestTune$k
cat("Optimal k:", customer_churn_best_k, "\n") # Best k-value

# Predictions on the test dataset
customer_churn_predicted_probs <- predict(customer_churn_knn_model, customer_churn_test_data, type = "prob")
customer_churn_predictions <- predict(customer_churn_knn_model, customer_churn_test_data)

# Evaluate the model
customer_churn_conf_matrix <- confusionMatrix(customer_churn_predictions, customer_churn_test_data$Exited, positive = "Exited")
```

```
print(customer_churn_conf_matrix)

# To output customer_churn_coef_matrix as image
customer_churn_stats_text <- capture.output(print(customer_churn_conf_matrix))
customer_churn_stats_grob <- textGrob(paste(customer_churn_stats_text,
                                      collapse = "\n"), x = 0.5, y = 0.5, just = "center", gp = gpar(fontsize = 7))
grid.newpage()
grid.draw(customer_churn_stats_grob)

# Extract metrics
customer_churn_accuracy <- customer_churn_conf_matrix$overall["Accuracy"]
customer_churn_precision <- customer_churn_conf_matrix$byClass["Precision"]
customer_churn_recall <- customer_churn_conf_matrix$byClass["Recall"]
customer_churn_f1_score <- customer_churn_conf_matrix$byClass["F1"]
customer_churn_specificity <- customer_churn_conf_matrix$byClass["Specificity"]
cat("Accuracy:", customer_churn_accuracy, "\nPrecision:", customer_churn_precision,
    "\nRecall:", customer_churn_recall, "\nF1-Score:", customer_churn_f1_score,
    "\nSpecificity:", customer_churn_specificity, "\n")

# To Plot the Confusion Matrix
customer_churn_conf_matrix_melt <- melt(as.table(customer_churn_conf_matrix$table))
customer_churn_conf_matrix_melt
colnames(customer_churn_conf_matrix_melt) <- c("Predicted", "Actual", "Count")
ggplot(customer_churn_conf_matrix_melt, aes(x = Predicted, y = Actual, fill = Count)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Count), vjust = 1) +
  scale_fill_gradient(low = "white", high = "lightblue") +
  labs(title = paste("Customer Churn Confusion Matrix for KNN (k =", customer_churn_best_k,
")"),
      x = "Predicted Class", y = "Actual Class") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.3, size = 11))


# Plot AUROC curve
customer_churn_roc_curve <- roc(customer_churn_test_data$Exited,
customer_churn_predicted_probs[, "Exited"],
  levels = rev(levels(customer_churn_test_data$Exited)), direction = ">")
customer_churn_auc_value <- auc(customer_churn_roc_curve)
customer_churn_roc_plot <- ggplot(data.frame(FPR = 1 -
customer_churn_roc_curve$specificities,
          TPR = customer_churn_roc_curve$sensitivities)) +
  geom_line(aes(x = FPR, y = TPR), color = "blue") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  annotate("text", x = 0.80, y = 0.60, label = paste("AUC =", round(customer_churn_auc_value,
2)),
        color = "black", size = 3, fontface = "bold") +
  annotate("text", x = 0.80, y = 0.50, label = paste("Accuracy =",
round(customer_churn_accuracy, 2)),
        color = "black", size = 3, fontface = "bold") +
  annotate("text", x = 0.80, y = 0.40, label = paste("Precision =",
round(customer_churn_precision, 2)),
        color = "black", size = 3, fontface = "bold") +
  annotate("text", x = 0.80, y = 0.30, label = paste("Recall =", round(customer_churn_recall, 2)),
        color = "black", size = 3, fontface = "bold") +
  annotate("text", x = 0.80, y = 0.20, label = paste("F1-Score =",
round(customer_churn_f1_score, 2)),
           color = "black", size = 3, fontface = "bold") +
  annotate("text", x = 0.80, y = 0.10, label = paste("Specificity =",
round(customer_churn_specificity, 2)),
           color = "black", size = 3, fontface = "bold") +
  labs(title = paste("Customer Churn AUROC Curve for KNN Model (k = ",
customer_churn_best_k, ")"),
    x = "False Positive Rate (1 - Specificity)", y = "True Positive Rate (Sensitivity)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 1, size = 11))


print(customer_churn_roc_plot)
```