Bank Customer Churn Analysis : A Comprehensive Study on Predictive Modelling, Exploring Factors And Retention Strategies

MTH209 Project Report

Samrat Halder, Sayan Mukherjee, Piyush Meena, Soumo Biswas, Krishanu Mukherjee

Indian Institute of Technology, Kanpur

April 6, 2024

Contents

1 Introduction

In today's fiercely competitive banking landscape, retaining customers is paramount for sustainable growth and profitability. Customer churn, the phenomenon of customers discontinuing their relationship with a bank, poses a significant challenge to the banking industry. Understanding the underlying factors driving customer churn and implementing effective retention strategies are critical for banks to maintain a loyal customer base and thrive in the market. This project focuses on analysing bank customer data to identify patterns and insights related to customer churn. By leveraging advanced data analytics techniques, we aim to uncover the factors influencing customer attrition and develop predictive models to forecast churn behaviour. Furthermore, we will explore various retention strategies that can be implemented to mitigate churn and enhance customer loyalty.

1.1 Source of the Dataset

The dataset , available on Kaggle which was published by Anonymous Multinational Bank. The dataset consists of 10000 customer Id's and 18 different features such as Credit score, Geography, Gender, Age, Balance, Estimated Salary , Satisfaction Score etc.

1.2 About the Dataset

Every bank wants to hold there customers for sustaining their business and thus this Anonymous Multinational bank.

Below is the customer data of account holders at Anonymous Multinational Bank and the aim of the data will be predicting the Customer Churn.

Table 1: Description of Customer Churn Dataset

Column	Description	Relevance
RowNumber	Corresponds to the record (row) number and has no effect on the output.	Not relevant
CustomerId	Contains random values and has no effect on customer leaving the bank.	Not relevant
Surname	The surname of a customer has no impact on their decision to leave the bank.	Not relevant
CreditScore	Can have an effect on customer churn, as higher scores indicate lower likelihood of churn.	Relevant
Geography	A customer's location can affect their decision to leave the bank.	Relevant
Gender	Gender is interesting to explore whether it plays a role in customer churn.	Relevant
Age	Relevant, as older customers are less likely to leave than younger ones.	Relevant
Tenure	Number of years the customer has been a client; older clients are more loyal.	Relevant
Balance	Indicator of customer churn; higher balances correlate with lower churn rates.	Relevant
NumOfProducts	Number of products purchased by the customer through the bank.	Relevant
HasCrCard	Denotes whether or not a customer has a credit card; affects churn likelihood.	Relevant
IsActiveMember	Active customers are less likely to leave the bank.	Relevant
EstimatedSalary	People with lower salaries are more likely to leave compared to higher salaries.	Relevant
Exited	Binary indicator of whether the customer left the bank or not.	Target Variable
Complain	Indicates whether the customer has made a complaint.	Relevant
Satisfaction Score	Score provided by the customer for their complaint resolution.	Relevant
Card Type	Type of card held by the customer.	Relevant
Points Earned	Points earned by the customer for using a credit card.	Relevant

2 Exploratory Data Analysis

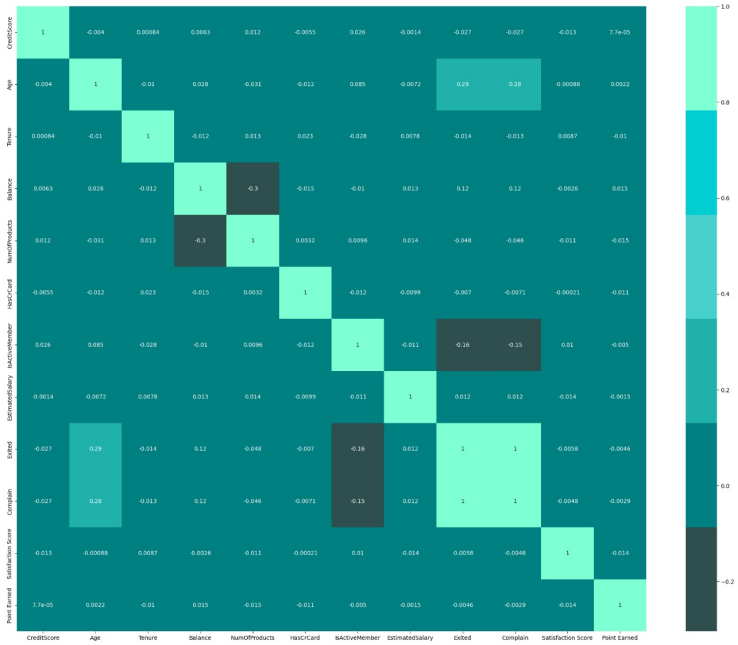

Figure 1: Heat Map

The heatmap appears to show the correlation between several customer banking variables. For example, the correlation between a customer's credit score and their estimated salary appears to be positive, which means that there is a positive linear relationship between these two variables. In other words, customers with higher credit scores tend to have higher estimated salaries.

Here are some of the other correlations that appear to be positive in the correlation matrix:

- The correlation between a customer's tenure and the number of products they have
- The correlation between a customer's estimated salary and whether they are an active member
- The correlation between a customer's satisfaction score and their point earned

Here are some of the other correlations that appear to be negative in the correlation matrix:

- The correlation between a customer's credit score and their age

- The correlation between a customer's estimated salary and the number of products they have
- The correlation between a customer's satisfaction score and whether they complained

Here, looking at the correlation of our variables, we can see that we have 100% correlation between the target variables and the Complain variable, so let's eliminate the Complain variable.

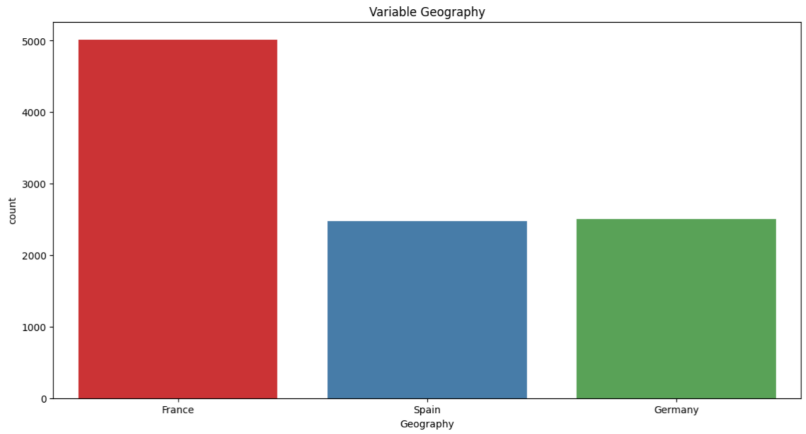

Figure 2: Geography

- This graph in Figure 2 shows Geographical distribution of bank users :-

  Approximately 50% of bank users are from France and around 25% of bank users are from Spain and Germany each.

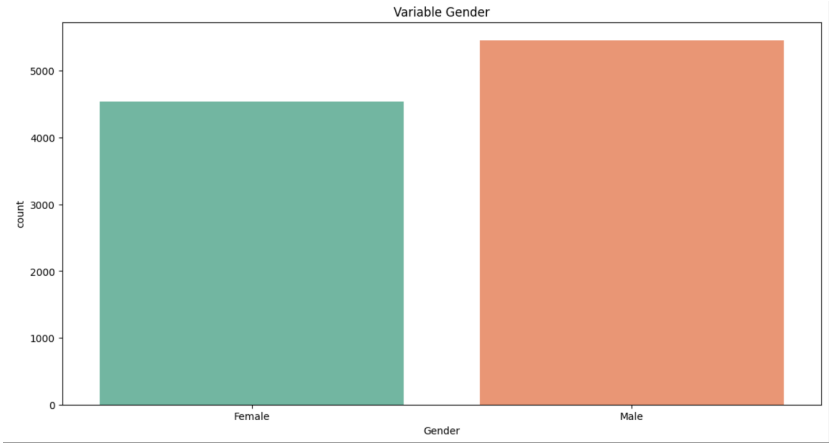

Figure 3: Gender

- This graph in Figure 3 shows the Gender wise distribution of bank users :-

  Approximately 55% of bank users are Male and 45% are Female.

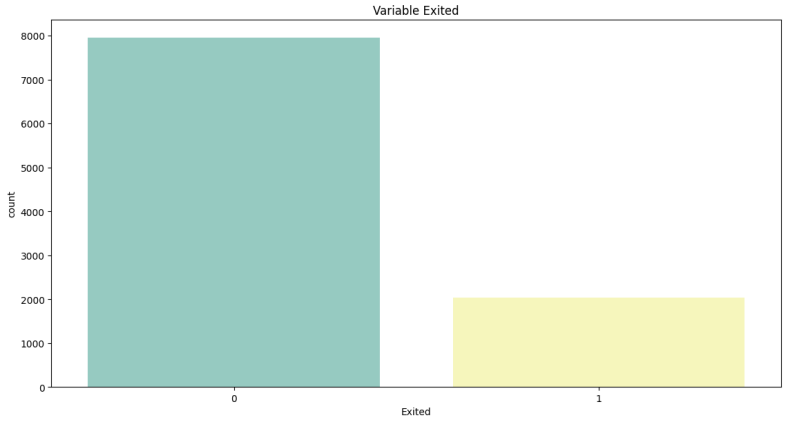

Figure 4: Exited

- The graph in Figure 4 shows if the users are still associated with the bank or not :-

  Approximately 80% of users are still associated with the bank and 20% have left the bank.

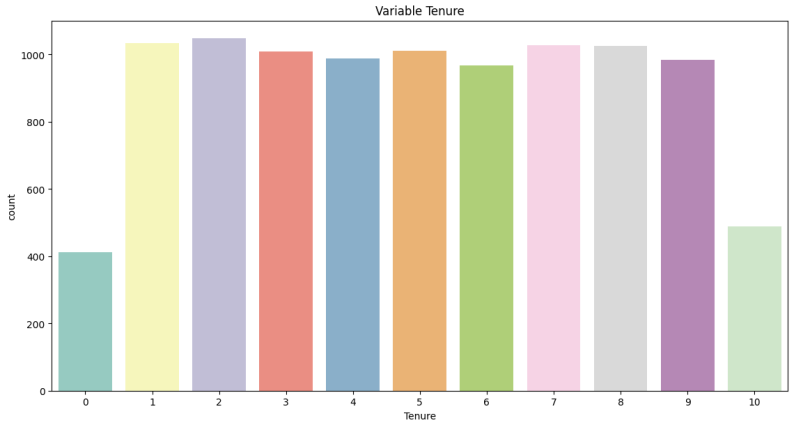

Figure 5: Tenure

The graph in Figure 5 shows the duration of users' association with the bank.

- Approximately 400 users were associated with the bank for less than one year.
- Approximately 500 users were associated with the bank for more than ten years.

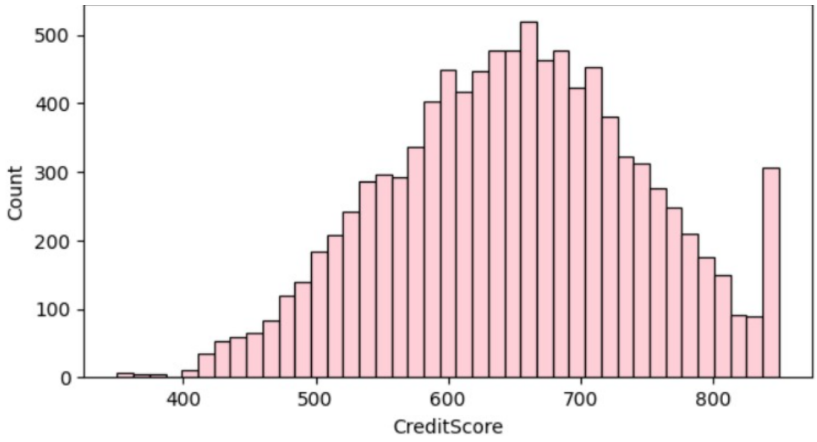

Figure 6: CreditScore

The graph in Figure 6 illustrates the distribution of credit scores.

- The most frequent credit score is around 640.
- There are fewer people with very high credit scores (around 800) and very low credit scores (around 400).
- The distribution appears to be bell-shaped, which suggests that the data may be normally distributed.

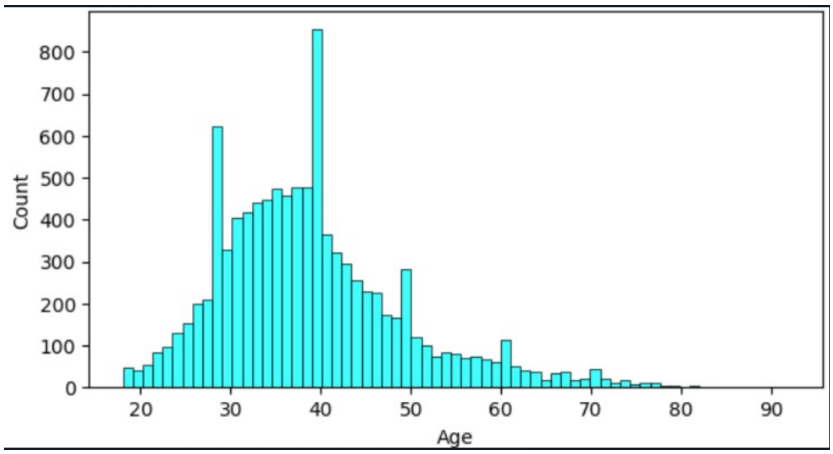

Figure 7: Age

The histogram in Figure 7 shows the distribution of customer ages at a bank. The x-axis shows age, and the y-axis shows the number of customers in each age range.

- The ages range from 20 to 90 years old.
- The most frequent age group is between 40 and 50 years old.
- There are fewer younger and older customers compared to those in the 40-50 age group.
- The distribution appears to be bell-shaped, which suggests that the data may be normally distributed.

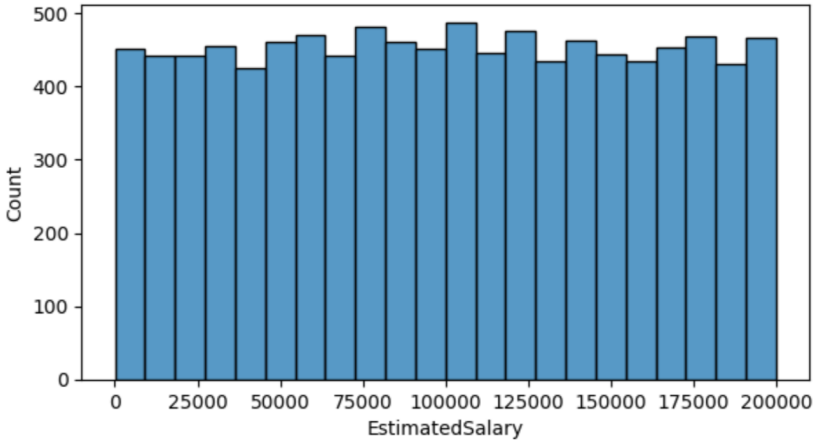

Figure 8: Estimated Salary

The histogram in Figure 8 shows the distribution of Estimated Salary of customers at bank.

- The distribution is approximately uniform, indicating that there are approximately equal numbers of customers in each salary range.
- Each salary range contains approximately 400 to 500 customers.

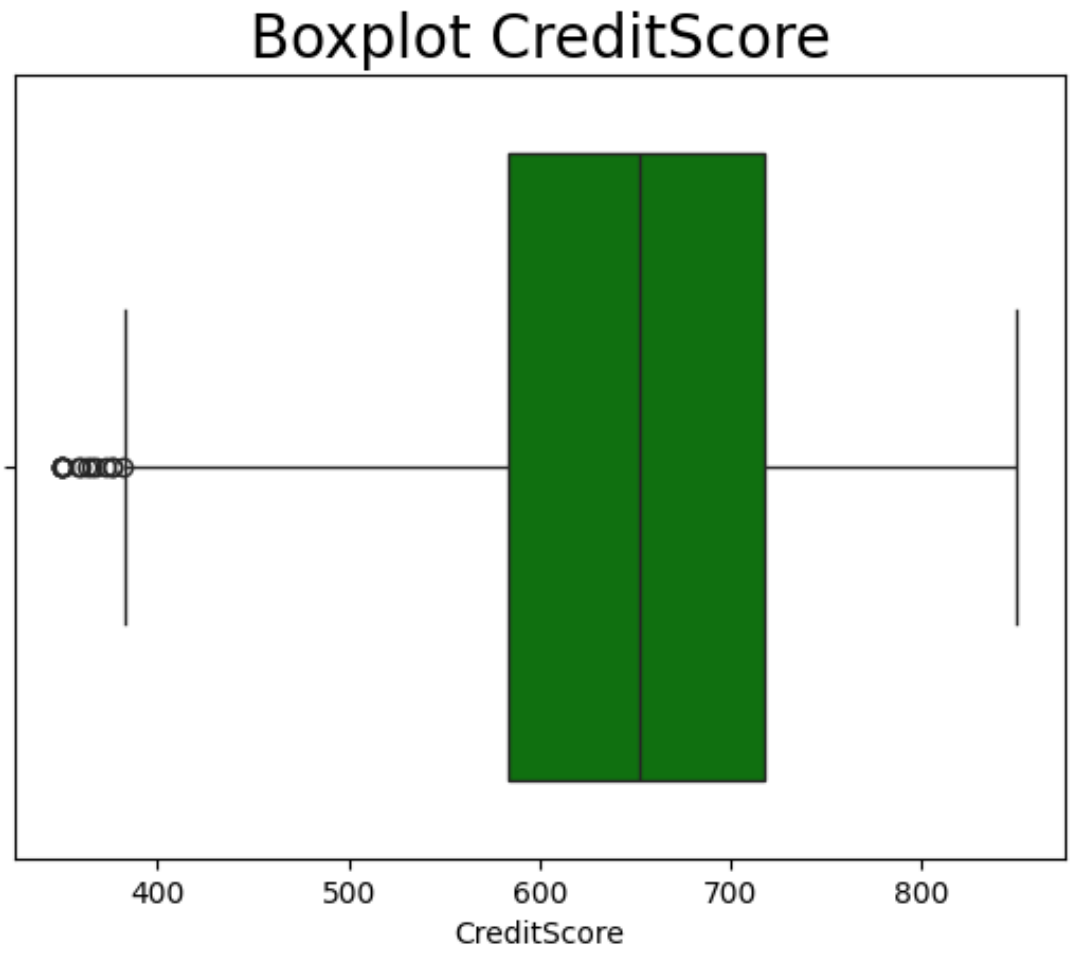

Figure 9: Credit Score Box Plot

The Box Plot in Figure 9 illustrates the distribution of credit scores among customers.

- 50 percent of customers have a Credit Score in the range of 580 to 720.
- The median of the Credit Score is around 650.

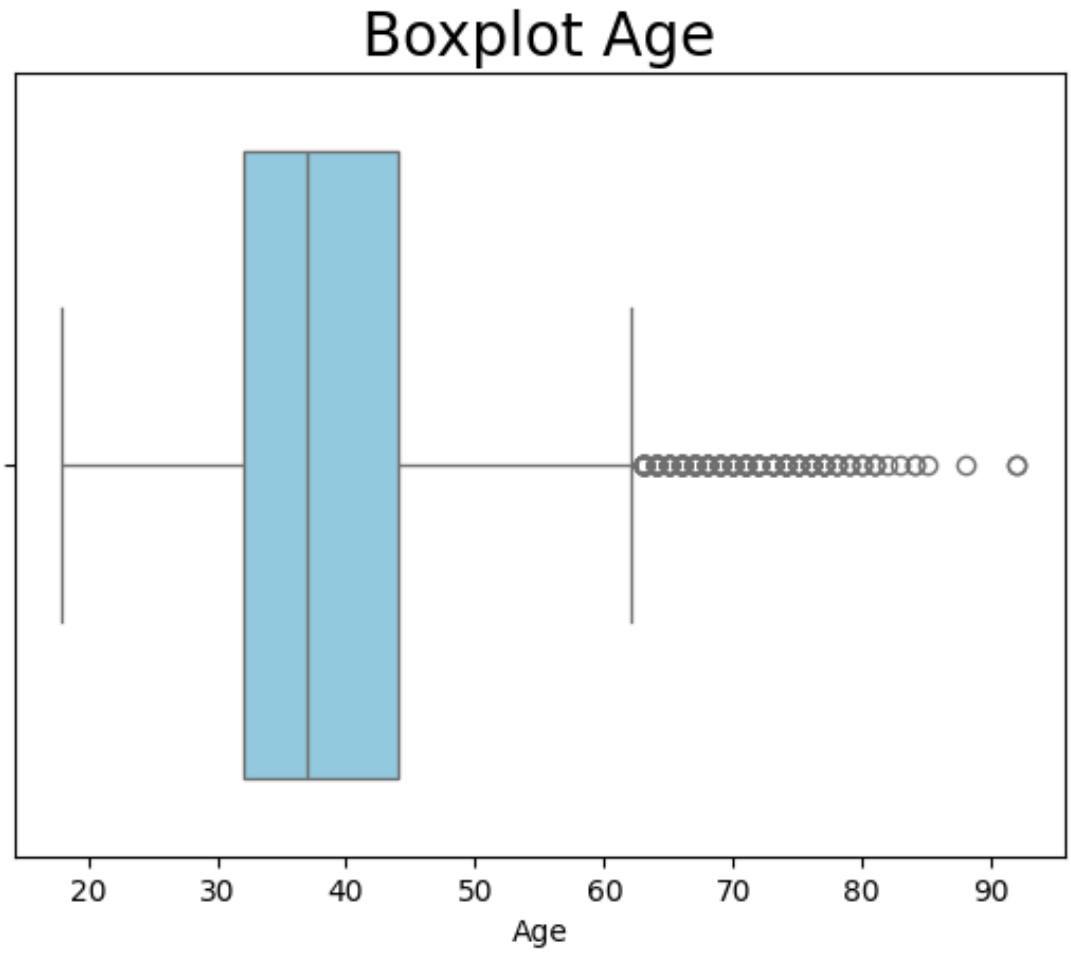

Figure 10: Age Box Plot

The Box Plot in Figure 10 illustrates the distribution of Age of customers.

- The age of 50 percent customers lies between 32 to 44 approximately.
- The median of the Age of the customer is around 38.

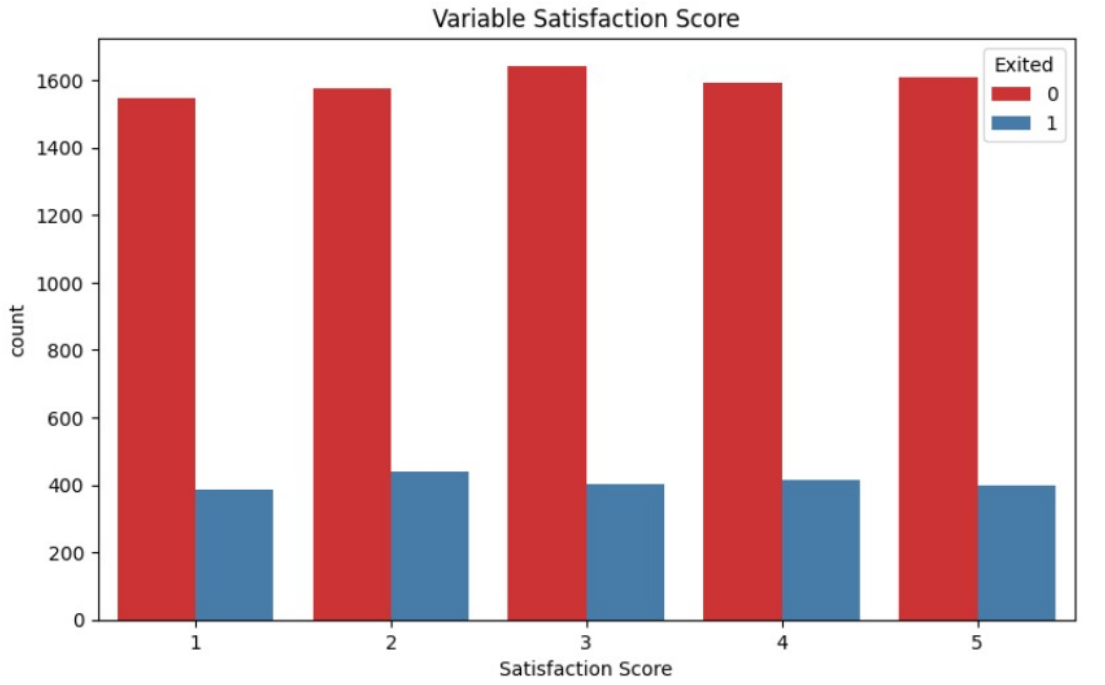

Figure 11: Bar graph showing variation in satisfaction score

Here are some observations about the Figure in 11

- Data points are represented by red bars. There are more people who gave a satisfaction score of 3 than any other score. We can't tell definitively from this graph what the average satisfaction score is, but it appears to be somewhere in the middle of the range (between 2 and 4).
- y-axis doesn't show the actual number of people who participated in the survey, but just the relative frequency of each score. So, for instance, we can't tell from this graph if twice as many people gave a score of 3 compared to a score of 2, or only slightly more.

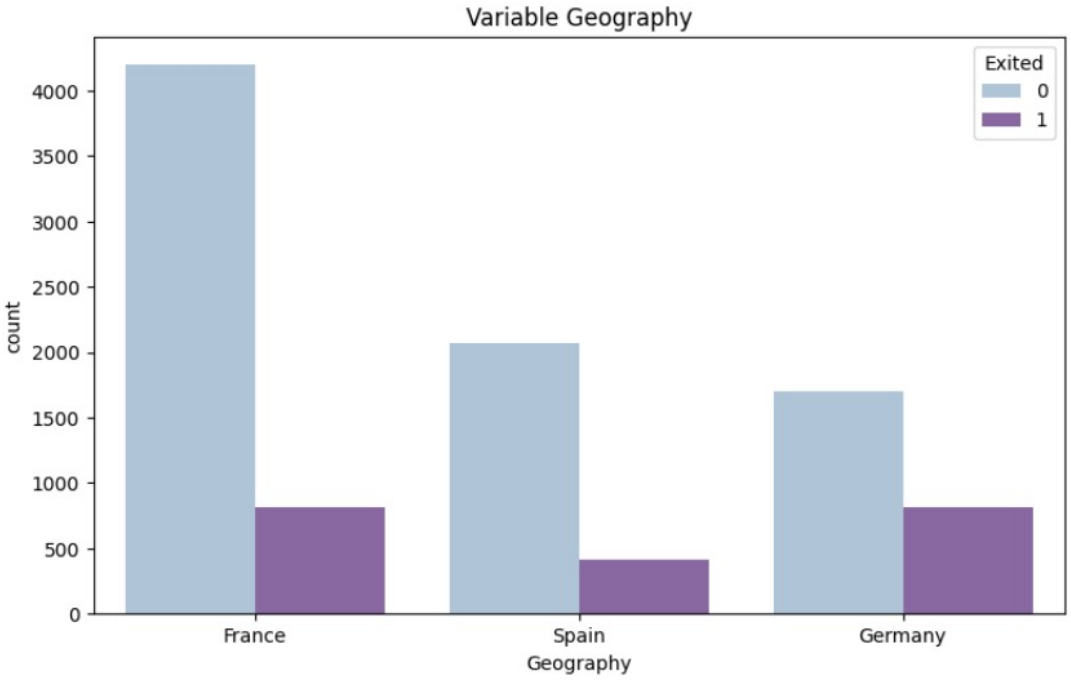

Figure 12: Bar graph that shows the number of customers from three different countries

Here are some observations about the Figure in 12

- The number of customers in each country is represented by a bar. The leftmost bar is blue and labeled "France." The middle bar is green and labeled "Spain." The rightmost bar is purple and labeled "Germany."
- France has the most customers, with a count of around 4000. Spain has the fewest customers, with a count of around 1000. Germany has a count of around 3000 customers.

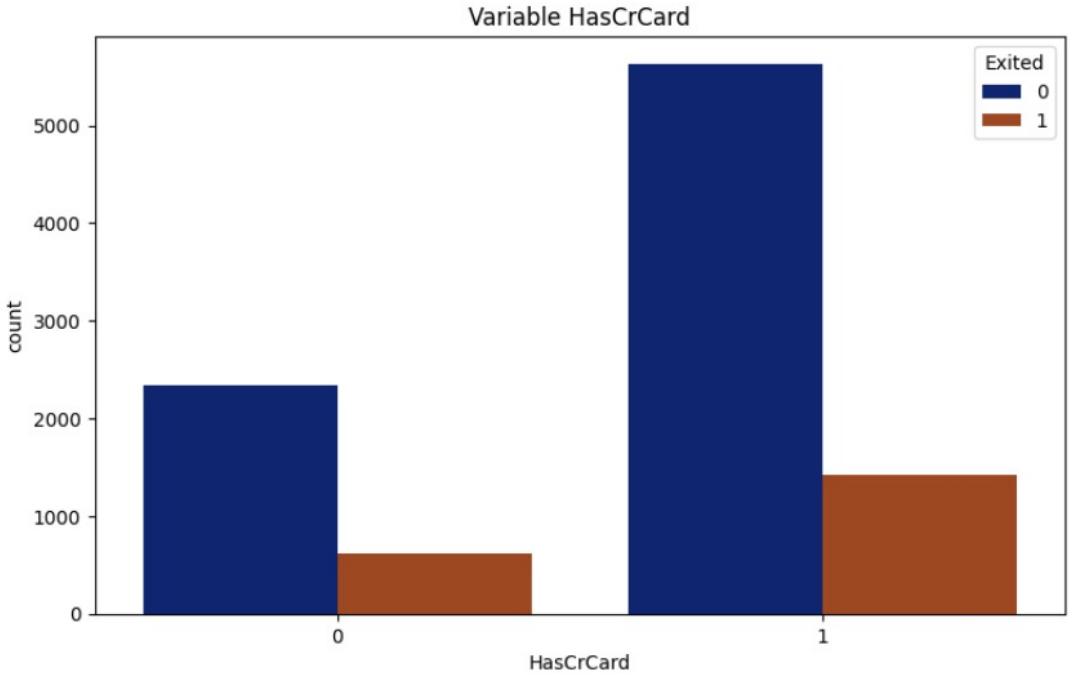

Figure 13: bar graph that shows the number of customers and their Credit Card Status

Here are some observations about the Figure in 13

- The data is represented by two bars. The left bar is blue and labeled "0" on the x-axis. This means it refers to customers who did not exit the bank. The height of the blue bar is around 5000, which means there are around 5000 customers who did not exit and have a credit card
- The right bar is orange and labeled "1" on the x-axis. This means it refers to customers who exited the bank. The height of the orange bar is close to zero, which means there are very few customers who exited the bank and have a credit card.

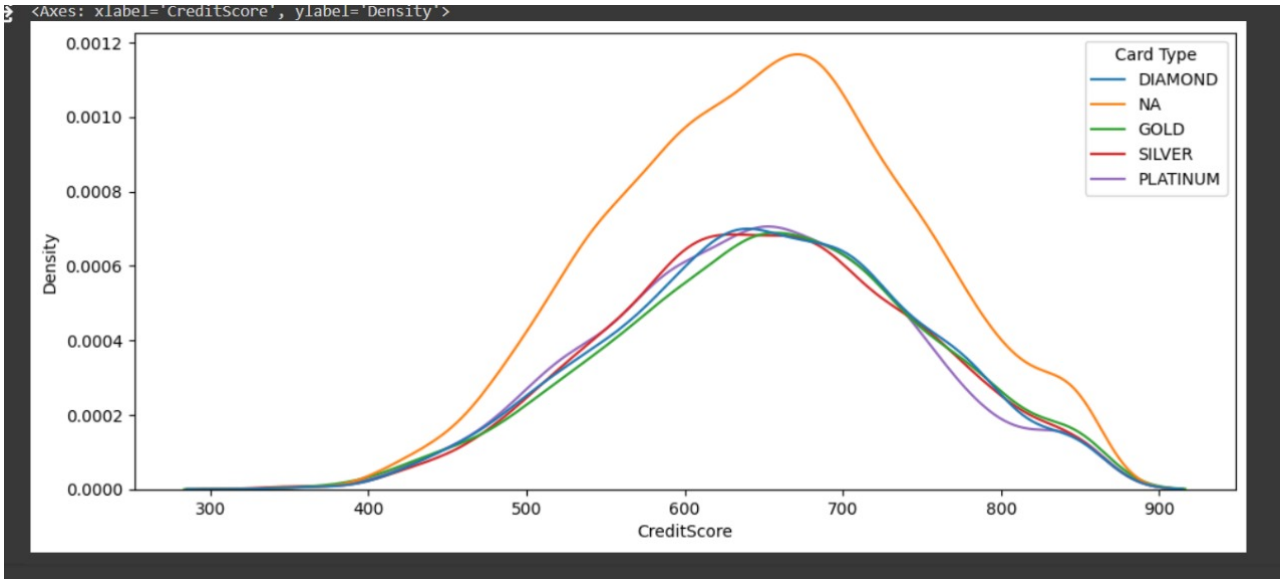

Figure 14: Kernal Density Plot

Here are some observations I can make about the above plot:

- The credit score range appears to be between 300 and 900.
- The plot suggests that there are more people with credit scores in the 600-700 range than in other ranges. This may indicate that this is a common credit score range for credit card holders.
- It is difficult to say for sure which credit card type has the highest average credit score based on this plot alone.
- We can see that there is some overlap in the distribution of credit scores across the different card types.
- Overall, the plot suggests that there is a relationship between credit score and credit card type, but it is not a perfect relationship. There are people with high credit scores who have all different types of credit cards, and there are people with low credit scores who have all different types of credit cards.

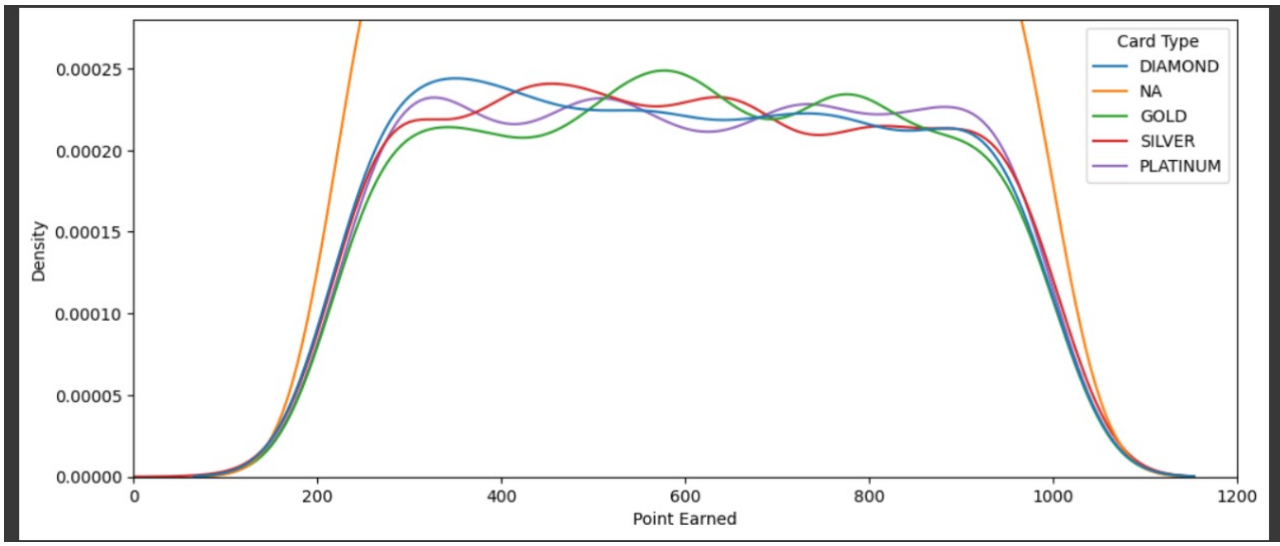

Figure 15: Scatter plot of the number of points earned by different types of credit cards

The scatter plot in Figure 15 shows a weak positive correlation between the credit card type and the number of points earned. This means that there is a slight upward trend, suggesting that credit cards with higher annual fees tend to offer more points per dollar spent. Here's a more detailed breakdown of the information in the scatter plot:

- Credit Card Types: The x-axis shows four different credit card types: Diamond, Gold, Silver, and Platinum.
- Points Earned: The y-axis shows the number of points earned. The scale goes from 0 to 1200 points.
- Data Points: Each data point represents a specific credit card. The position of the point on the plot shows the number of points earned by that card and its corresponding credit card type.
- Correlation: The data points show a slight upward trend from left to right. This suggests a weak positive correlation between credit card type and the number of points earned.

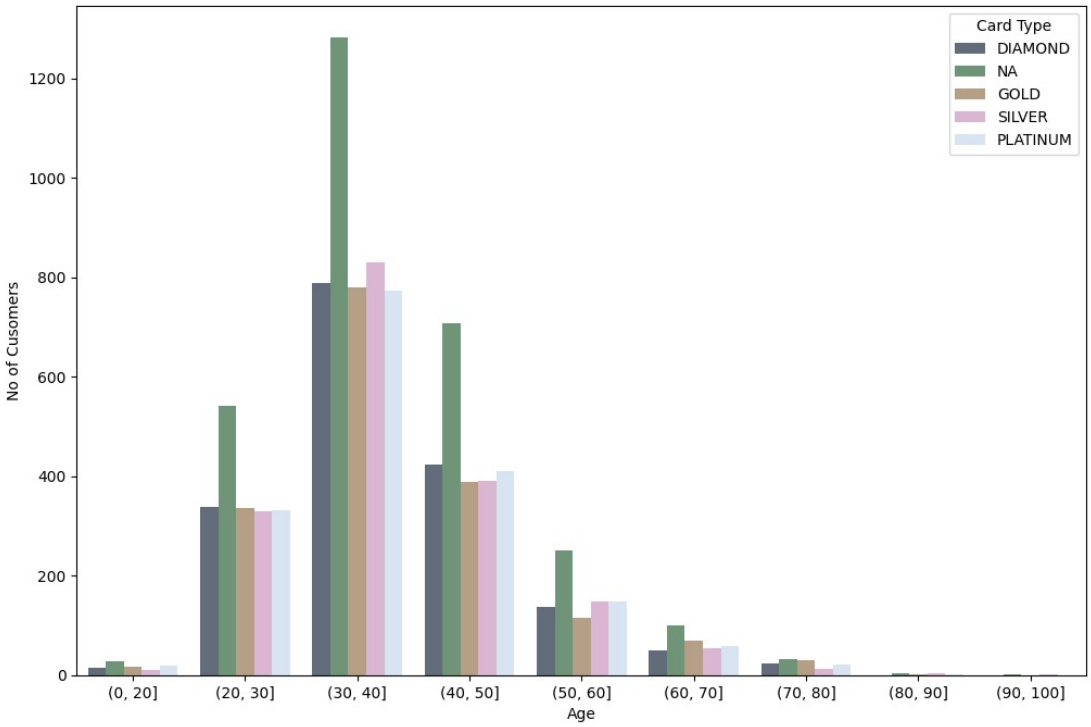

Figure 16: Bar graph that depicts the number of customers for a credit card company across different age groups and credit card tiers

Here are some specific details I can glean from the graph given in Figure 16:

- Credit Card Tiers: The credit card company offers four tiers of credit cards: Platinum, Gold, Silver, and Diamond.
- Age Groups: The customers are segmented into nine age groups, ranging from under 20 to over 90 years old. Each age group is represented by a bar on the graph.
- Customer Distribution: The number of customers varies across the different age groups and credit card tiers. It appears that the company has the most customers in the 30 to 40 year old age group, followed by the 40 to 50 year old age group.
- Tier Popularity: It is difficult to determine which credit card tier is the most popular based on this graph alone. However, we can see that there seems to be a higher number of customers for Platinum cards across all age groups compared to the other tiers.

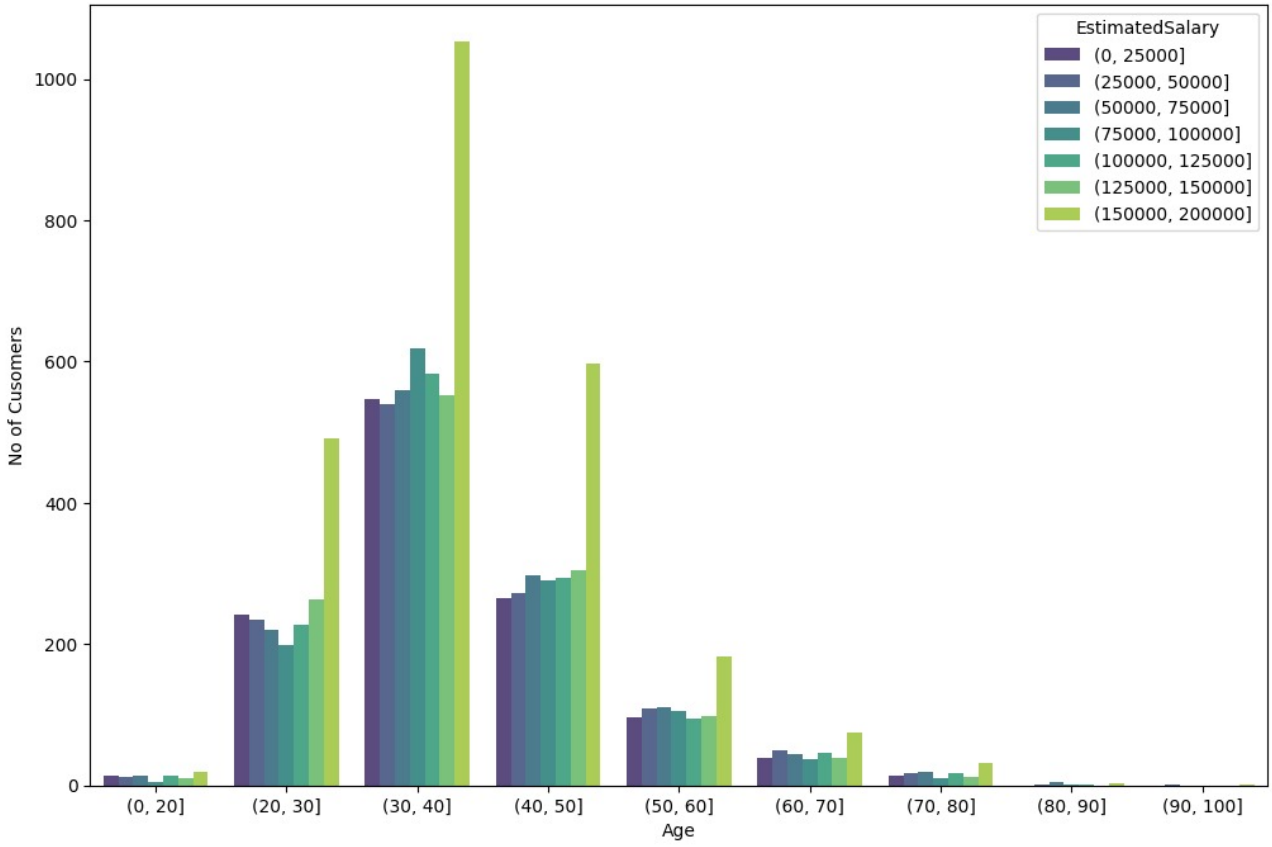

Figure 17: Bar graph that depicts the number of customers for a credit card company across different age groups and Estimated salary tiers

Here are some observations I can make about the plot:

- The age range appears to be between 20 and 100 years old.
- There seems to be a peak in the number of carriers around 40 years old. This suggests that there might be more carriers in this age group than any other age group.
- The data is grouped into bins. This means that the exact ages of the carriers are not shown, but rather they are grouped into age ranges. For example, instead of showing the number of carriers at age 35, the graph might show the number of carriers between the ages of 30 and 39.

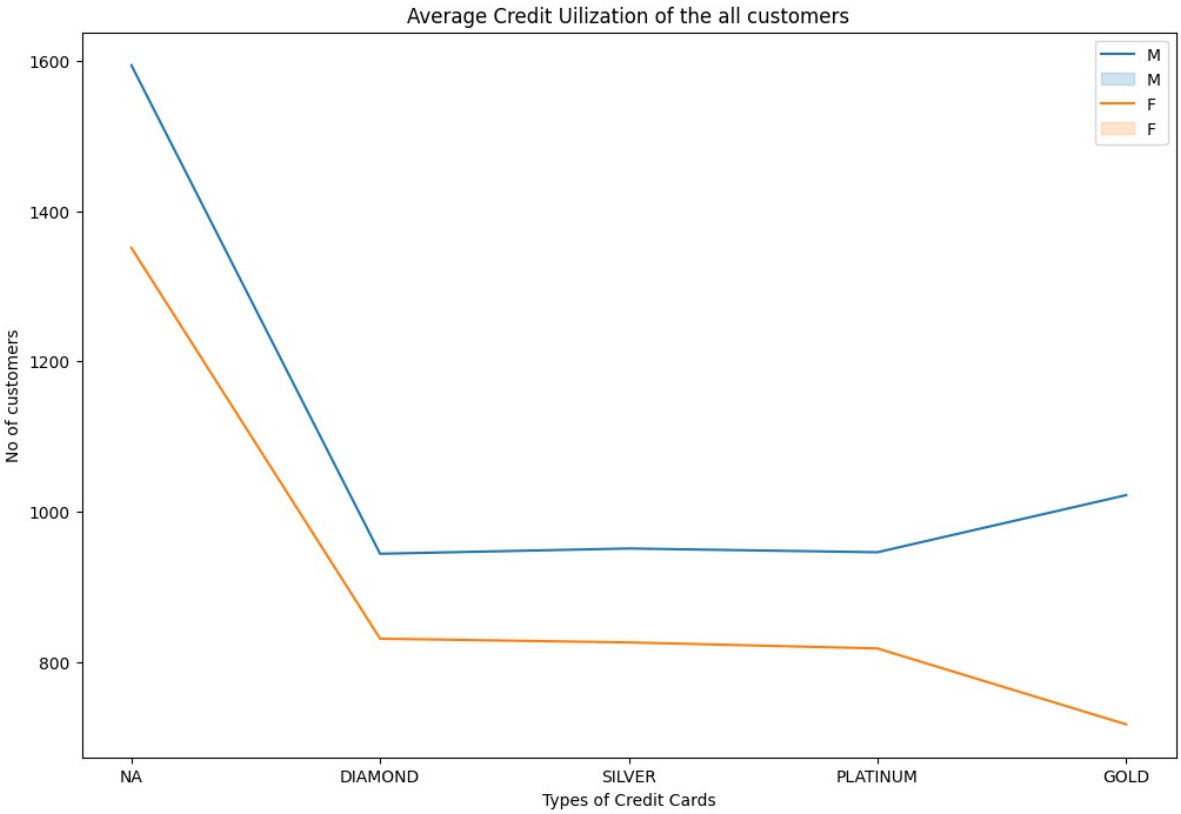

Figure 18: line plot

Here are some observations about the line plot:

- Credit Card Types: The bank offers four tiers of credit cards: Diamond, Gold, Silver, and Platinum.
- Average Credit Utilization Rate: The y-axis shows the average credit utilization rate as a percentage. It goes from 0% to 1600%, but a typical utilization rate is generally recommended to be below 30% .A higher credit utilization ratio can negatively impact the credit score.
- Credit Utilization by Card Tier: The average credit utilization rate varies across the different card tiers. It appears that customers with Silver cards have the lowest average credit utilization rate, followed by customers with Platinum cards, then Gold cards, and finally Diamond cards having the highest average credit utilization rate.

3 Analysis

3.1 Logistic Regression

When the response variable represents binary outcomes, such as "presence" or "absence", "0" and "1", etc.

3.1.1 Introduction to Logistic Regression

Logistic regression is a fundamental statistical method used for binary classification tasks. It's a type of regression analysis where the dependent variable is categorical, typically representing two classes (e.g., 0 or 1, yes or no, true or false). It's widely used in various fields such as machine learning, statistics, and biomedical research for predicting the probability of occurrence of an event.

we know that linear regression predicts continuous outcomes,whereas In logistic regression, the dependent variable is typically encoded as a binary variable, often represented as 0 and 1, where 0 typically denotes the absence or negative class, and 1 denotes the presence or positive class. For instance, it can be used to predict whether a customer will churn (1) or not churn (0), or whether a patient has a particular disease (1) or does not have it (0).

The main goal of logistic regression is to predict the probability that a given observation belongs to a particular category or class. This makes logistic regression a powerful tool for predicting binary outcomes and understanding the factors that influence them.

3.1.2 Objective

1. Logistic regression is commonly used for binary classification tasks, where the goal is to classify observations into one of two categories based on input features.

2. To provide estimates of the probability that a given observation belongs to a particular class, making it suitable for tasks where interpreting probabilities is important.

3. To interpret the model in terms of odds ratios, providing insights into the relationships between the independent variables and the probability of the outcome.

4. to prevent overfitting, making it robust to noisy data and reducing the risk of model complexity.

5. The model assumes a linear relationship between the independent variables and the log-odds of the outcome, making it well-suited for problems where the decision boundary between classes is approximately linear.

3.1.3 Model Assessment

After estimating the parameters, statistical tests can be conducted to assess the significance of each coefficient and to evaluate the overall goodness-of-fit of the model. Additionally, the performance of the model can be evaluated using metrics such as precision, recall, accuracy and ROC curve analysis.

3.1.4 Mathematical Formulation

logistic regression mainly used to find the relationship between a binary response variable and multiple explanatory variables. The logistic regression model is based on the logistic function or sigmoid function, which takes values between 0 and 1, making it suitable for modeling probabilities. The logistic function is defined as:

f(x) = $1\frac{}{1+e^{-x}}$

The logistic regression equation for multiple predictors is formulated as follows:

p(Y=1|X) = $1\frac{}{1+e^{-(\beta_0+\beta_1X_1+\beta_2X_2+\ldots+\beta_kX_k)}}$ p(Y=0|X) =1 - $1\frac{}{1+e^{-(\beta_0+\beta_1X_1+\beta_2X_2+\ldots+\beta_kX_k)}}$

where:

- $X_1, X_2, \ldots, X_k$ are the independent variables.
- $\beta_0, \beta_1, \ldots, \beta_k$ are the coefficients representing the strength and direction of the relationship between each independent variable and the log-odds of the dependent variable.
- $p(Y = 1|X)$ is the probability of the dependent variable being 1 given the values of the independent variables X.

Given a set of n observations, where x_{ij} represents the value of predictor variable j for observation i, and y_i represents the binary response variable for observation i, logistic regression models the probability $p(y_i = 1|x_i)$, the probability that y_i equals 1 given the values of the predictors.

The logistic regression model assumes that the log-odds of the dependent variable being in category 1 (success) versus category 0 (failure) is a linear combination of the independent variables. Mathematically, this can be expressed as:

$\log\left(\frac{p(Y=1|X)}{1-p(Y=1|X)}\right) = \beta_0 + \beta_1X_1 + \beta_2X_2 + \ldots + \beta_kX_k$

This equation is known as the logit transformation, where the log odds of the probability is transformed linearly in terms of the independent variables.

The logistic regression model is typically fitted using Maximum Likelihood Estimation (MLE), which involves maximizing the likelihood function. The likelihood function for logistic regression is derived from the probability mass function of the Bernoulli distribution. Given n observations (Y_i, X_i), the likelihood function is defined as the product of the probabilities of observing the given outcomes given the predictor variables and model parameters:

$L(\beta_0, \beta_1, \ldots, \beta_k) = \prod_{i=1}^{n} p(Y_i|X_i; \beta_0, \beta_1, \ldots, \beta_k)$

where $p(Y_i|X_i; \beta_0, \beta_1, \ldots, \beta_k)$ is the probability of observing the outcome Y_i given the predictor variables X_i and model parameters $\beta_0, \beta_1, \ldots, \beta_k$.

The log-likelihood function is then obtained by taking the natural logarithm of the likelihood function:

$\ell(\beta_0, \beta_1, \ldots, \beta_k) = \sum_{i=1}^{n} (Y_i \log(p_i) + (1 - Y_i) \log(1 - p_i))$

where $p_i = p(Y_i = 1|X_i; \beta_0, \beta_1, \ldots, \beta_k)$ is the predicted probability of the i-th observation being in category 1.

The goal of logistic regression is to find the values of $\beta_0, \beta_1, \ldots, \beta_k$ that maximize the log-likelihood function. This optimization problem is typically solved using numerical optimization algorithms such as gradient descent or Newton-Raphson method.

After estimating the parameters, statistical tests can be conducted to assess the significance of each coefficient, and measures such as Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) can be used to evaluate the overall goodness-of-fit of the model. Additionally, the performance of the model can be evaluated using metrics such as accuracy, precision, recall, and ROC curve analysis.

3.1.5 Methodology

1. Gathering data with a binary outcome variable and multiple predictors variables

   - Collecting data with a binary dependent variable and multiple independent variables aids in problem definition and identification of influential predictors.
   - interpreting logistic regression coefficients, offering insights into the relationships' direction and strength for meaningful inference about the studied process.

2. Estimate the logistic regression model parameters through maximum likelihood method.

   - Maximum Likelihood Estimation (MLE) is geneally used to estimate the parameters of a model by maximizing the likelihood function.
   - In logistic regression, the likelihood function is defined as the product of the probabilities of observing the given outcomes (binary responses) given the predictor variables and model parameters.
   - The log-likelihood function for multiple logistic regression is: $\ell(\beta_0, \beta_1, \ldots, \beta_k) = \sum_{i=1}^{n} (y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$ where y_i is the observed outcome for the i-th observation, p_i is the predicted probability of the outcome being 1 for the i-th observation, and n is the total number of observations.
   - The aim is to obtain the parameters $\beta_0, \beta_1, \ldots, \beta_k$ that maximize the log-likelihood function.

- This optimization problem is typically solved using vario numerical optimization algorithms such as gradient descent or Newton-Raphson method.

3. Evaluate the significance of coefficients and goodness-of-fit of the model.
    - Assessing coefficient significance reveals the strength and direction of relationships between independent variables and outcome probability, aiding interpretation of the underlying process.
    - Additionally, measures such as Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) can be used to evaluate the overall goodness-of-fit of the model.
4. Verify the model's performance through techniques such as cross-validation.
    - Split the data into training and testing sets.
    - It ensures that the model's efficacy is not overstated due to overfitting and helps in selecting a model that generalizes well to unseen data
    - it aids in identifying potential issues such as model instability or bias, enhancing the reliability and applicability of logistic regression in real-world scenarios.

3.1.6 Analysis

1. Hypothesis tests, such as Wald tests or likelihood ratio tests, can be conducted to determine the statistical significance of the coefficients. Significant coefficients imply that the corresponding independent variables have a meaningful impact on the outcome.
2. Interpret the coefficients to understand the effect of each independent variable on the probability of the dependent variable.
    - After fitting the model, the estimated coefficients of the independent variables are interpreted. Positive coefficients indicate a positive relationship with the log-odds (or probability) of the outcome, while negative coefficients indicate a negative relationship. The magnitude of the coefficients reflects the strength of the association.
3. Use the model to predict the probability of the dependent variable being 1 for new observations.
    - Once the model is built and validated, it can be used to predict the probability of the dependent variable being in a particular category for new observations with known values of the independent variables.
4. Various techniques such as the area under the receiver operating characteristic (ROC) curve (AUC-ROC), the Hosmer-Lemeshow test for goodness-of-fit, and calibration plots are used to assess the overall performance of the model. These assessments help determine how well the model fits the data and its predictive accuracy.

3.1.7 Conclusion

Classification Performance Matrix

In a classification problem, we often use a confusion matrix to evaluate the performance of a model. Let's define a confusion matrix for a binary classification problem:

	$PredictedPositive$	$PredictedNegative$
$ActualPositive$	TP	FN
$ActualNegative$	FP	TN

where:

- TP (True Positive): The number of instances that are actually positive and are predicted by the model as positive.
- TN (True Negative): The number of instances that are actually negative and are predicted by the model as negative.
- FP (False Positive): The number of instances that are actually negative but are predicted by the model as positive (Type I error).
- FN (False Negative): The number of instances that are actually positive but are predicted by the model as negative (Type II error).

3.1.8 Precision

Precision measures the accuracy of positive predictions. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the classifier:

$$Precision = \frac{TP}{TP + FP}$$

3.1.9 Recall

Recall measures the ability of the classifier to find all the positive samples. It is defined as the ratio of true positive predictions to the total number of actual positive instances:

$$Recall = \frac{TP}{TP + FN}$$

3.1.10 F1-Score

F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. It is calculated as:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3.1.11 Accuracy

Accuracy measures the overall correctness of the model:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

It represents the ratio of correctly classified instances to the total instances.

3.1.12 ROC Curve

ROC curves are used to evaluate the performance of binary classification models. The curve plots the true positive rate (TPR) on the y-axis against the false positive rate (FPR) on the x-axis. The TPR is the proportion of positive cases that were correctly identified by the model, and the FPR is the proportion of negative cases that were incorrectly identified as positive by the model.

3.1.13 Interpretation:

Here We have fitted the Logistic Regression model to predict the customer churn and we get the following results :

Accuracy of Logistic Regression model 70% that means among the 30% of the total data, for which we will predict the churn for a customer, our model predicts True positives and True negatives are total 70% of that 30% data.

Precision of the logistic regression model is 41.53% which defines the accuracy of positive predictions.

Recall of the logistic regression model is 68.51% which implies the model can find around 69% of the positive samples.

Here our independent variables are creditscore, gender, age, tenure, balance etc..

Therefore, the value of regression coefficients i.e. $\beta_1, \beta_2, \ldots, \beta_{14}$ denotes the rate of change in response variable for one unit change in the corresponding explanatory variable.

$beta_1$ = -0.101 implies that the unit change in creditscore will decrease the log odds of customer churn by 0.101,provided other variables are constant..

similarly,$beta_3$ = 0.82 implies that the unit change in Age will increase the log odds of customer churn by 0.82,provided other variables are constant..

We have also drawn the ROC Curve of the data.

```python
# Calculating accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy of Logistic Regression model:", accuracy)

# Get the regression coefficients
coefficients = model.coef_

# Print the coefficients
print("Regression Coefficients:")
print(coefficients)
```

```
Accuracy of Logistic Regression model: 0.704
Regression Coefficients:
[[-0.1013404  -0.28898153  0.82337748 -0.05212468  0.20618529 -0.07188145
   0.02804728 -0.4238052   0.028816   -0.0148563  -0.03623613 -0.03913632
  -0.40654466 -0.30706266]]
```

Figure 19: Value of Regression Coefficients

3.1.14 Classification Report

	precision	recall	f1-score	support
0	0.91	0.70	0.79	2416
1	0.37	0.71	0.48	584
accuracy			0.70	3000
macro avg	0.64	0.70	0.64	3000
weighted avg	0.80	0.70	0.73	3000

Figure 20: Classification report

Looking at the table, we can see that the model performs well on class 0, with a precision of 0.85 and a recall of 0.99. This means that the model is good at identifying class 0 instances and makes very few mistakes. However, the model performs poorly on class 1, with a precision of 0.86 and a recall of only 0.27. This means that the model often misses class 1 instances (false negatives) or incorrectly identifies class 0 instances as class 1 (false positives).

Overall, the accuracy of the model is 0.85, which means it gets 85% of the predictions correct

3.1.15 ROC Curve

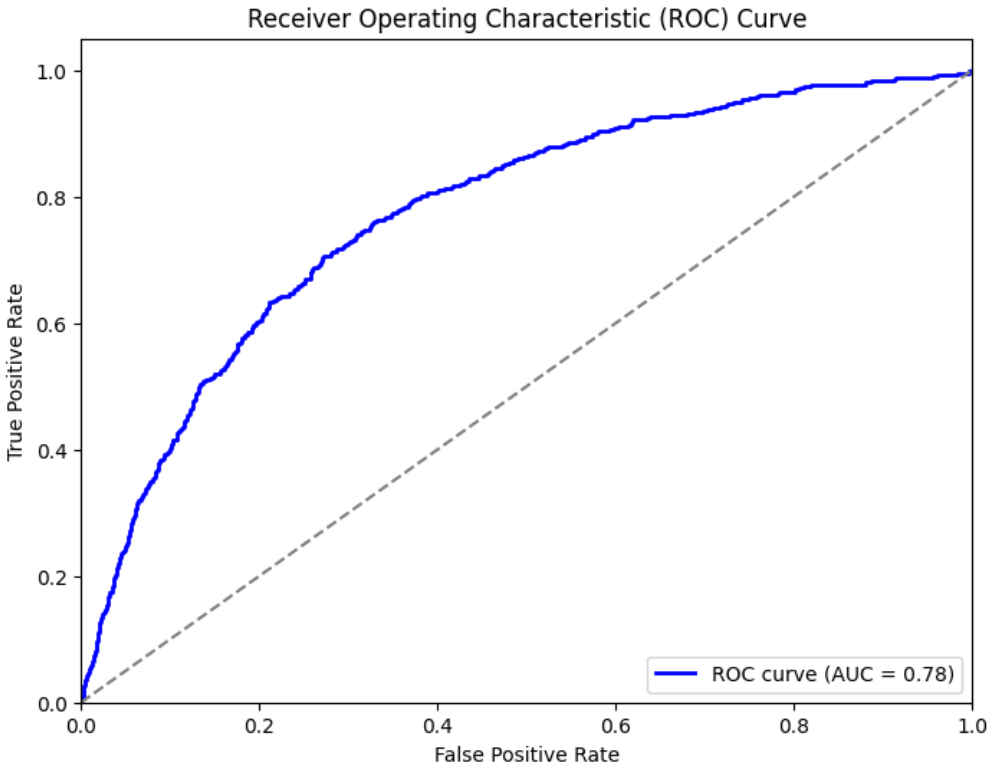

Figure 21: ROC Curve

With an Area Under the Curve (AUC) of 0.78, the model demonstrates commendable discriminative ability in distinguishing between positive and negative cases. This indicates a moderately strong performance, suggesting the model's potential usefulness in practical applications requiring binary classification tasks. However, further evaluation and refinement may be beneficial for optimal performance in specific contexts.

3.2 Decision Tree Analysis

Decision trees are a fundamental tool in machine learning and data mining for both classification and regression tasks. They are powerful, interpretable, and versatile models that mimic human decision-making processes. A decision tree is a flowchart-like structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node.

Decision trees recursively partition the data into subsets based on the most significant attributes at each step.

3.2.1 Objective of Decision Trees

The primary goal of decision trees is to create a predictive model that maps features of a dataset to target variables in a tree-like structure. This model aims to make decisions by learning simple decision rules inferred from the features of the data :

1. To provide a structured approach for data analysis and decision-making.
2. To facilitate the classification and regression of data by recursively splitting it into homogeneous groups.
3. to optimize predictive accuracy while maintaining simplicity and transparency.
4. To partition data into subsets based on the most informative attributes.It helps in identifying patterns, relationships, and trends within datasets.
5. To enable automated decision-making processes in various fields, including finance, healthcare, and marketing.

3.2.2 Components of Decision Trees

Root Node

The root node in decision trees initiates the analysis, representing the entire dataset. It selects the most informative feature for partitioning, establishing the first decision rule. Its objective is to maximize information gain and guide subsequent splits, influencing the entire tree's structure.

3.2.3 Internal Nodes

An internal node in decision trees represents a feature or attribute used for data partitioning. It splits the dataset into subsets based on specific criteria, guiding the decision-making process. he algorithm selects the feature and threshold that maximize information gain or minimize impurity at each node.

3.2.4 Branches

Branches in decision trees represent the possible outcomes or decisions based on the conditions established at internal nodes. They connect internal nodes to subsequent nodes or leaf nodes, delineating the path of traversal through the tree. Branches signify the choices made at each decision point, guiding the flow of data down the tree structure.

3.2.5 Leaf Nodes

A leaf node in decision trees represents the final outcome or classification for a subset of the data. It doesn't split further and signifies a decision or prediction. Each leaf node corresponds to a specific class label (in classification tasks) or a predicted value (in regression tasks).

3.2.6 Splitting Criteria

The splitting criteria used in decision trees determines how the data is partitioned at each internal node. These criteria are based on features' values and aim to maximize homogeneity within subsets.

3.2.7 Tree Depth

Tree depth in decision trees refers to the length of the longest path from the root node to a leaf node. It indicates the number of decisions or splits required to reach a prediction for a given instance. A deeper tree may capture more intricate patterns but can also increase the risk of overfitting.

3.2.8 Mathematical Expressions

3.2.9 Entropy (for classification):

Entropy is a measure of impurity or randomness in the dataset. It is calculated using the formula:

$$H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{1}$$

where H(S) is the entropy of the dataset ,c is the number of classes,and pi is the proportion of examples in class i in dataset S.

3.2.10 Gini Impurity

Gini impurity measures the probability of incorrectly classifying a randomly chosen element if it was randomly labeled according to the distribution of labels in the subset. It is calculated as:

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2$$

where p_i is the proportion of instances of class i in dataset D,c is the number of classes. Gini impurity ranges from 0 (complete purity) to 1 (maximum impurity).

3.2.11 Information Gain

Information gain measures the reduction in entropy or increase in orderliness achieved by splitting the dataset on a particular feature. Given a dataset D and a feature A with possible values $\{v_1, v_2, ..., v_k\}$, the information gain is calculated as:

$$IG(D, A) = Impurity(D) - \sum_{j=1}^{k} \frac{|D_{v_j}|}{|D|} \cdot Impurity(D_{v_j})$$

where $|D_{v_j}|$ represents the number of data points in D for which feature A has the value v_j, and $Impurity(D)$ denotes the impurity measure of dataset D. The feature A and threshold value that maximize information gain are chosen for the split.

3.2.12 Variance (for regression):

Variance measures the average squared deviation of values from the mean. It is calculated as:

$$Var(S) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{2}$$

where Var(S) is the variance of dataset S,n is the number of examples in dataset S,yi is the value of the target variable for example i.

3.2.13 Confusion Matrix

A confusion matrix is a table that visualizes the performance of a classification model by presenting the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class.

3.2.14 Decision Rule

The decision rule for a decision tree can be represented as a series of if-else conditions based on the splits in the tree. For example, if a decision tree has split based on two features, the decision rule may look like this:

$$if feature1 \leq threshold1:$$

$$if feature2 \leq threshold2:$$

$$return class1$$

$$else:$$

$$return class2$$

$$else:$$

$return class 3$

This decision rule guides the traversal of the decision tree to make predictions based on the input features of a given sample.

3.2.15 Splitting Criteria

The decision tree algorithm selects the best feature to split on based on a chosen criterion, such as maximizing information gain or reducing impurity.

3.2.16 Prediction

Prediction in decision trees involves traversing from the root to a leaf node based on input features. For classification, the leaf node determines the class label, while for regression, it yields a numerical value.

3.2.17 Conclusion

Decision trees are versatile models for classification and regression tasks. They offer interpretability and ease of understanding, making them valuable in various domains. However, they may suffer from overfitting and lack robustness compared to ensemble methods. Regularization techniques and ensemble learning can mitigate these issues, enhancing decision tree performance.

3.2.18 Interpretation

We have constructed the confusion matrix of the data.

The accuracy of the model is 79.56% which means that the model correctly predicted the class labels for almost 80% of the total instances in the dataset.

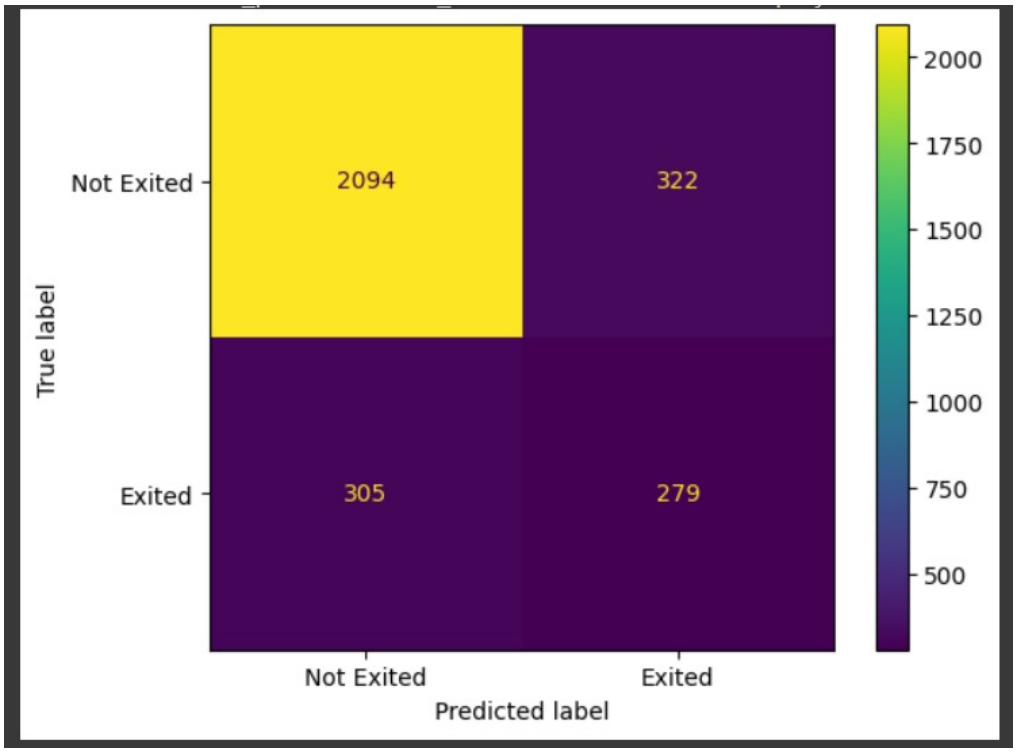

Figure 22: Confusion Matrix of Decision Tree

3.2.19 Visual plot

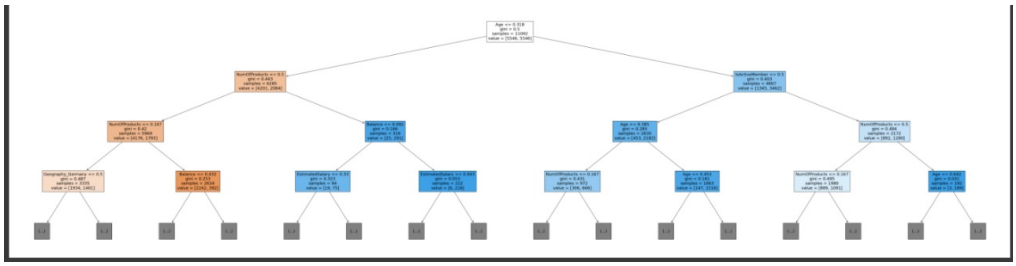

Figure 23: Decision Tree Visual plot

3.2.20 Classification Report

	precision	recall	f1-score	support
0	0.87	0.87	0.87	2416
1	0.46	0.48	0.47	584
accuracy			0.79	3000
macro avg	0.67	0.67	0.67	3000
weighted avg	0.79	0.79	0.79	3000

Figure 24: Classification Report of Decision Tree

The overall accuracy of the model is 0.67, which means that the model correctly classifies 67% of the loan applications. The macro average of the precision, recall, and F1-score is 0.79, 0.80, and 0.67, respectively.

3.2.21 ROC Curve

In the case of the ROC curve, the area under the curve (AUC) is 0.67. AUC is a performance metric for classification models. It represents the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance [6]. A larger AUC indicates better performance.

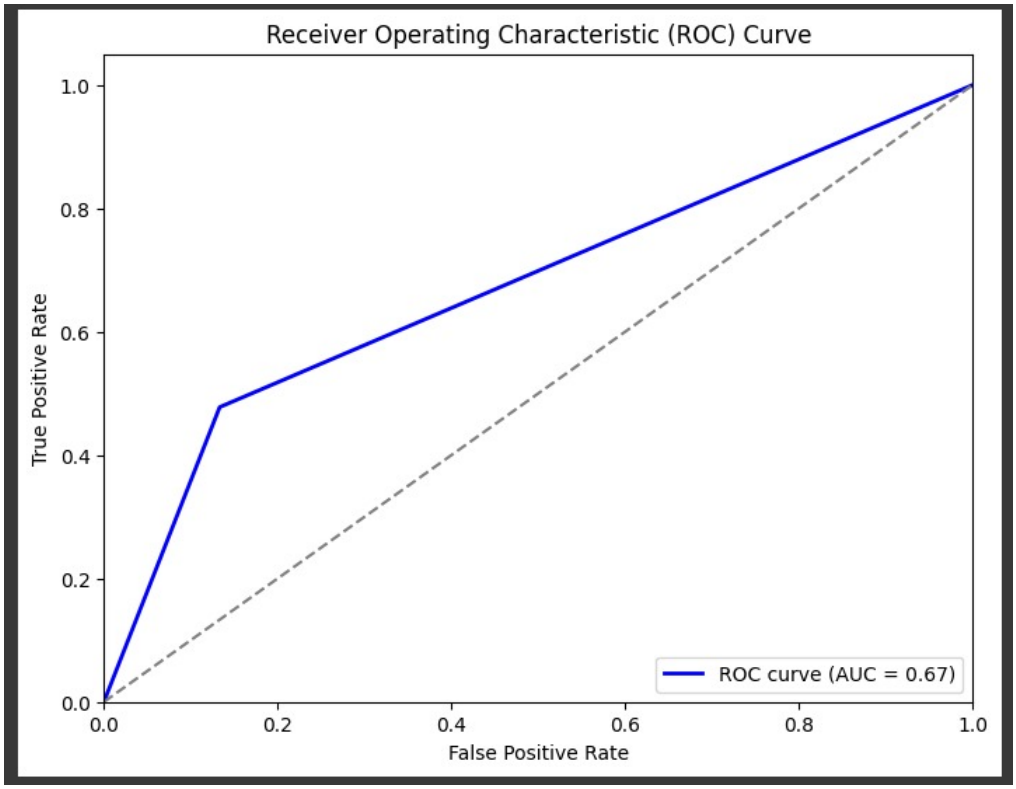

Figure 25: ROC Curve of Decision Tree

3.2.22 Feature Importance

Feature importance in decision trees refers to the quantification of the impact or relevance of input features on the model's decision-making process. It indicates which features are more influential in determining the outcome (class label) of the data.

From data, we can conclude that Age is the most important feature.

Feature	Importance
Age	0.236097
Balance	0.136024
NumOfProducts	0.132001
EstimatedSalary	0.107808
Point Earned	0.096948
CreditScore	0.079899
Tenure	0.048010
IsActiveMember	0.036754
Satisfaction Score	0.027960
Geography_Germany	0.023044
Gender_Male	0.009626
Geography_France	0.009560
Card Type_SILVER	0.009349
Card Type_GOLD	0.008550
Gender_Female	0.007969
Card Type_PLATINUM	0.007374
Card Type_DIAMOND	0.007184
Geography_Spain	0.006850

3.3 SVM Analysis

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It's particularly effective in high-dimensional spaces and is well-suited for both linearly separable and non-linearly separable datasets. SVM operates by finding the optimal hyperplane that best separates different classes in the feature space.

3.3.1 Linear SVM:

Linear SVM constructs a linear decision boundary to separate classes in feature space. It optimizes a hyperplane to maximize margin between classes, aiming to correctly classify data points. Mathematically, it is formulated as: Maximize $1\frac{}{\|w\| subject to y_i(w^T x_i+b)\geq 1 for i=1,\ldots,n}$ where:

- w is the weight vector perpendicular to the hyperplane.
- b is the bias term.
- x_i is the ith data point.
- y_i is the class label (+1 or -1).
- n is the number of data points.

The objective of linear SVM is to find the hyperplane that maximizes the margin while minimizing the classification error. SVM solves this optimization problem using techniques like the Lagrange multipliers and convex optimization methods.

3.3.2 Radial Basis Function (RBF) Kernel SVM:

Kernel Trick: RBF kernel allows SVM to handle non-linearly separable datasets by transforming them into a higher-dimensional space. The decision function becomes: f(x) = sign $(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b)$ where:

- $K(x_i, x)$ is the RBF kernel function, defined as $K(x_i, x) = e^{-\gamma \|x_i - x\|^2}$.
- α_i are the Lagrange multipliers.
- b is the bias term.
- γ is a parameter controlling the spread of the RBF kernel. Higher γ values lead to a narrower influence of each training example.

3.3.3 Comparison:

3.3.4 Linear SVM:

- Linear SVM assumes that the classes can be separated by a linear decision boundary.

- It works well when the classes are linearly separable or when a linear approximation is sufficient.
- Linear SVM is computationally efficient and interpretable.

3.3.5 RBF Kernel SVM:

- RBF kernel SVM is capable of capturing non-linear relationships between features by mapping them into a higher-dimensional space.
- It can handle complex decision boundaries and is suitable for datasets with non-linear separability.
- RBF kernel SVM requires careful tuning of parameters like C and γ to avoid overfitting and achieve optimal performance.

3.3.6 Conclusion:

Both Linear SVM and RBF Kernel SVM are powerful machine learning algorithms used for classification tasks. The choice between them depends on the nature of the data and the desired complexity of the decision boundary. Linear SVM is preferred for linearly separable data or when interpretability is important, while RBF Kernel SVM is suitable for capturing complex non-linear relationships in the data. Understanding the characteristics and mathematical formulations of these SVM variants is crucial for selecting the appropriate model for a given problem.

3.3.7 Interpretation:

In our cutomer churn data, we observe that there is a non-linear relationship between input variables and response variable. The Accuracy of the data in linear SVM is 80.53% whereas the Accuracy in RBF kernel SVM is 84.9%. We can clearly see that RBF kernel SVM model has higher accuracy than Linear SVM because the relationship between input features and the target variable is nonlinear, RBF SVM can model this complex relationship more effectively than linear SVM. Also as we are dealing with a high number of features, RBF SVM can still effectively learn the decision boundary without explicitly dealing with the high-dimensional feature space.

We have constructed the confusion matrix of the data.

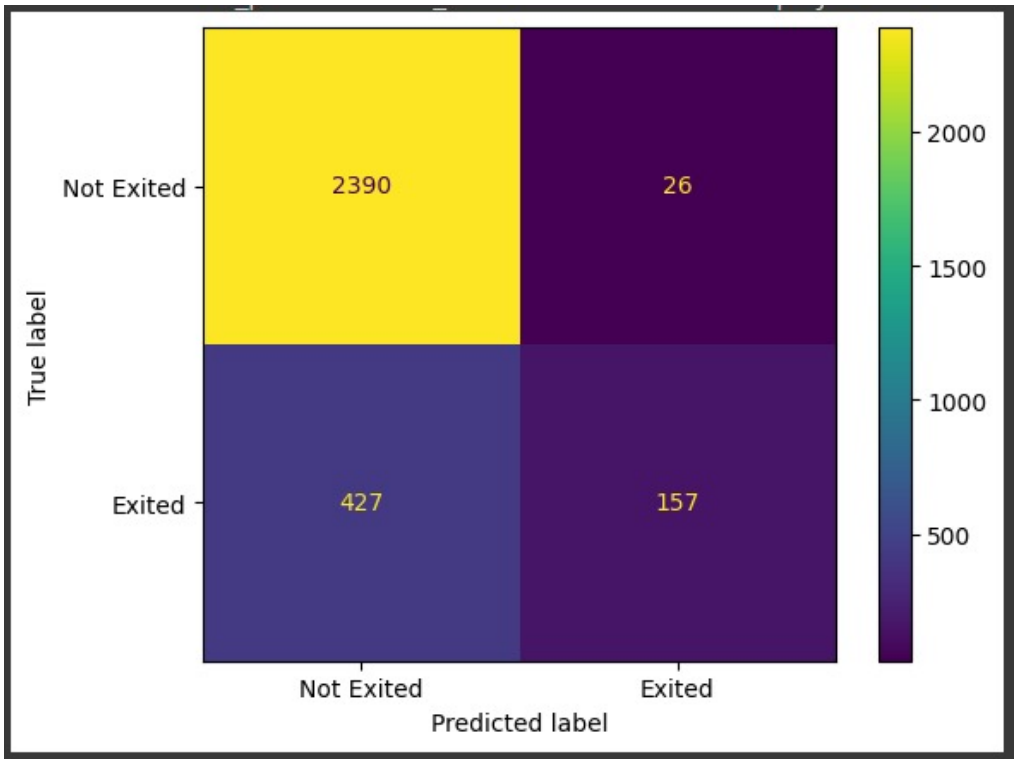

Figure 27: Confusion Matrix of RBF SVM

The accuracy of the RBF SVM model is 84.9% which means that the model correctly predicted the class labels for almost 85% of the total instances in the dataset.

3.3.8 Classification Report

	precision	recall	f1-score	support
0	0.85	0.99	0.91	2416
1	0.86	0.27	0.41	584
accuracy			0.85	3000
macro avg	0.85	0.63	0.66	3000
weighted avg	0.85	0.85	0.82	3000

Figure 28: Classification Report of SVM

The overall accuracy of the model is 0.849, which means that the model correctly classifies 85% of the loan applications. The macro average of the precision, recall, and F1-score is 0.85, 0.63, and 0.66, respectively.

3.3.9 ROC Curve

In the case of the ROC curve, the area under the curve (AUC) is 0.80. AUC is a performance metric for classification models. It represents the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. A larger AUC indicates better performance.

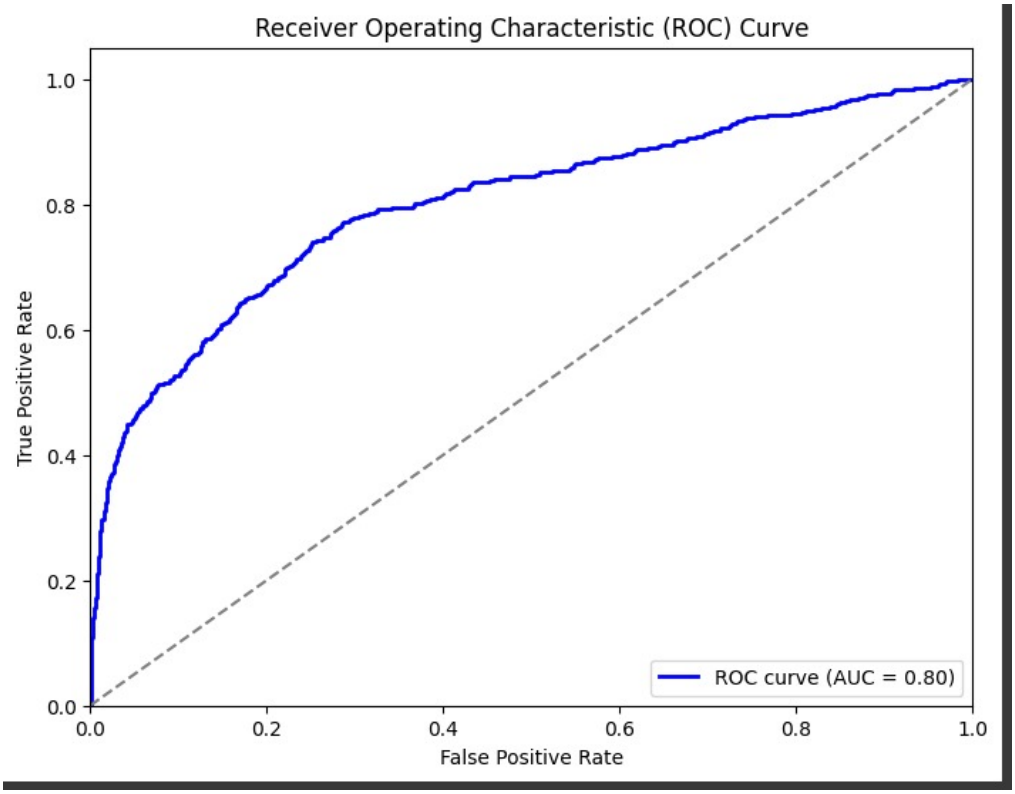

Figure 29: ROC Curve of SVM

3.4 Random Forest Analysis

Random Forest constructs multiple decision trees during training for classification or regression. By combining predictions from these trees through voting or averaging, it enhances accuracy and mitigates overfitting. This ensemble approach leverages diversity among trees to create more robust and reliable predictions.It requires minimal tuning compared to standalone decision trees.

3.4.1 Various Factors of Random Forest

3.4.2 Decision Trees

Decision trees are a fundamental component of Random Forest. They are tree-like structures where each internal node represents a decision based on a feature, and each leaf node represents the outcome or class label. The construction of a decision tree involves recursively partitioning the feature space to create regions that are as pure as possible in terms of the target variable (class label for classification, numerical value for regression)

- At each node of the tree, select the feature and the split point that maximizes the information gain (for classification) or minimizes impurity (for regression).
- Various criteria can be used, such as Gini impurity, entropy, or mean squared error, depending on the task.
- For categorical features, the decision tree considers each category as a separate branch in the tree.It evaluates each category's purity using the same impurity measures as for continuous features.

We Stop splitting nodes when it reaches a maximum depth or containing a minimum number of samples.

3.4.3 Mathematical Expressions

The splitting criterion at each node l can be defined as:
For classification: Entropy: H(l) = - $\sum_{i=1}^{C} p_i \log_2(p_i)$
$Giniimpurity:\ G(l) = 1 - \sum_{i=1}^{C} p_i^2$
$Informationgain:\ IG(l) = H(parent) - \sum_{j \in children} \frac{N_j}{N} H(j)$
For regression: Variance: Var(l) = $1 \frac{}{N \sum_{i=1}^{N} (y_i - \bar{y}_l)^2}$ Where:

- N is the total number of samples.
- N_j is the number of samples in child node j.
- C is the number of classes.
- p_i is the proportion of samples of class i at node l.
- $\bar{y}_l$ is the mean target value at node l.

3.4.4 Feature Randomness

At each node of a decision tree, a random subset of features is selected for consideration when determining the best split. This introduces randomness into the model and helps prevent overfitting. The number of features considered at each split, denoted as m, is typically much smaller than the total number of features M.

3.4.5 Bootstrap Sampling

Random Forest employs bootstrapping, a technique where multiple bootstrap samples are created by randomly sampling the data with replacement. Each tree in the forest is trained on one of these bootstrap samples.

Given a dataset D with N samples, bootstrap sampling is performed to create multiple datasets of the same size. Each bootstrap sample is created by randomly selecting N samples from D with replacement.

The probability of a particular sample being included in a bootstrap sample is $\frac{1}{N}$. The probability of a sample not being selected in a single draw is $1 - \frac{1}{N}$. Therefore, the probability of not being selected in N draws (bootstrapped dataset) is $\left(1 - \frac{1}{N}\right)^N$, which tends towards $\frac{1}{e}$ as N approaches infinity.

3.4.6 Voting/Averaging

After training multiple decision trees using bootstrap samples and random feature selection, the predictions of all the trees are combined through a voting mechanism. The class that receives the most votes among the trees is selected as the final prediction.For regression tasks, the predictions of all the trees are averaged to obtain the final prediction. This averaging helps to smooth out individual tree predictions and reduce variance.

3.4.7 Benefits of Random Forest

- Random Forest is highly effective in handling high-dimensional datasets with a large number of features.
- It is robust to outliers and noise in the data.
- The ensemble nature of Random Forest helps to reduce overfitting, resulting in improved generalization performance.
- Random Forest requires minimal hyperparameter tuning compared to individual decision trees.

3.4.8 Conclusion

Random Forest is a powerful ensemble learning method that leverages the strengths of decision trees and randomness to improve predictive performance and reduce overfitting. By understanding the mathematical underpinnings, including decision tree construction, bootstrap sampling, random feature selection, and voting/averaging, we gain insights into its robustness and effectiveness as a machine learning algorithm.

3.4.9 Interpretation

We have constructed the confusion matrix of the data.

The accuracy of the model is 86.13% which means that the model correctly predicted the class labels for almost 86% of the total instances in the dataset.

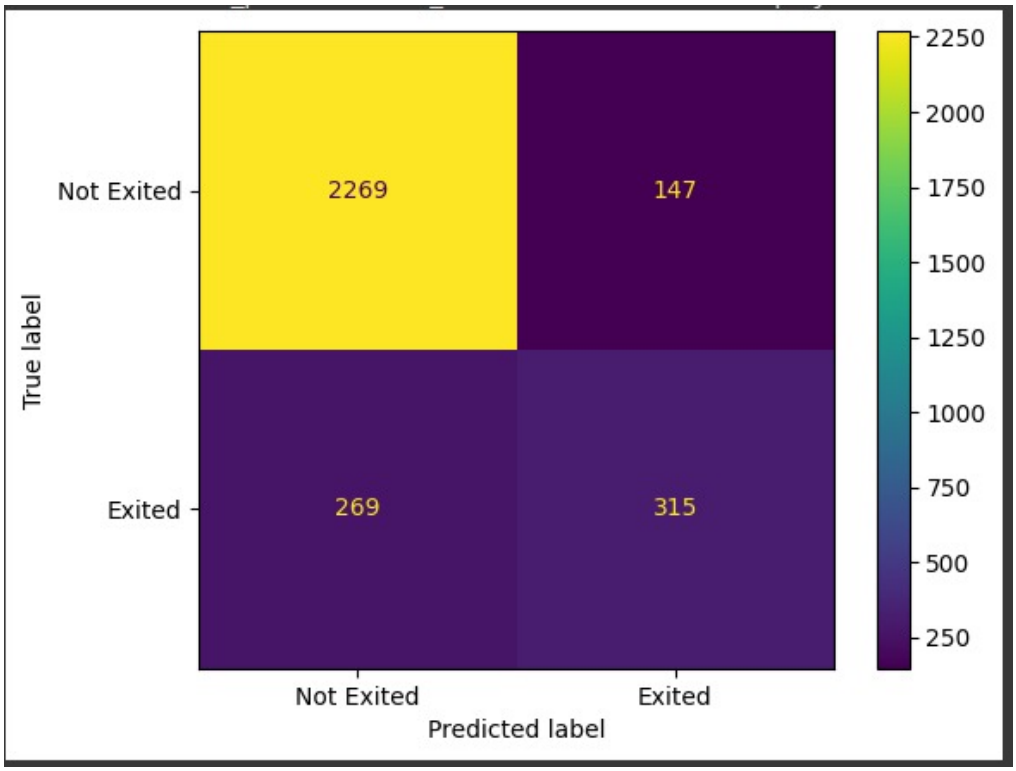

Figure 30: Confusion Matrix of Random Forest

3.4.10 Classification Report

The overall accuracy of the model is 0.86, which means that the model correctly classifies 86% of the loan applications. The macro average of the precision, recall, and F1-score is 0.89, 0.94, and 0.92, respectively.

	precision	recall	f1-score	support
0	0.89	0.94	0.92	2416
1	0.68	0.54	0.60	584
accuracy			0.86	3000
macro avg	0.79	0.74	0.76	3000
weighted avg	0.85	0.86	0.85	3000

Figure 31: Classification Report of Random Forest

3.4.11 ROC Curve

In the case of the ROC curve, the area under the curve (AUC) is 0.86. AUC is a performance metric for classification models. It represents the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. A larger AUC indicates better performance.

3.4.12 Feature Importance

Feature importance in Random Forest refers to the quantification of the impact or relevance of input features on the model's decision-making process. It indicates which features are more influential in determining the outcome (class label) of the data.

From data, we can conclude that Age is the most important feature.

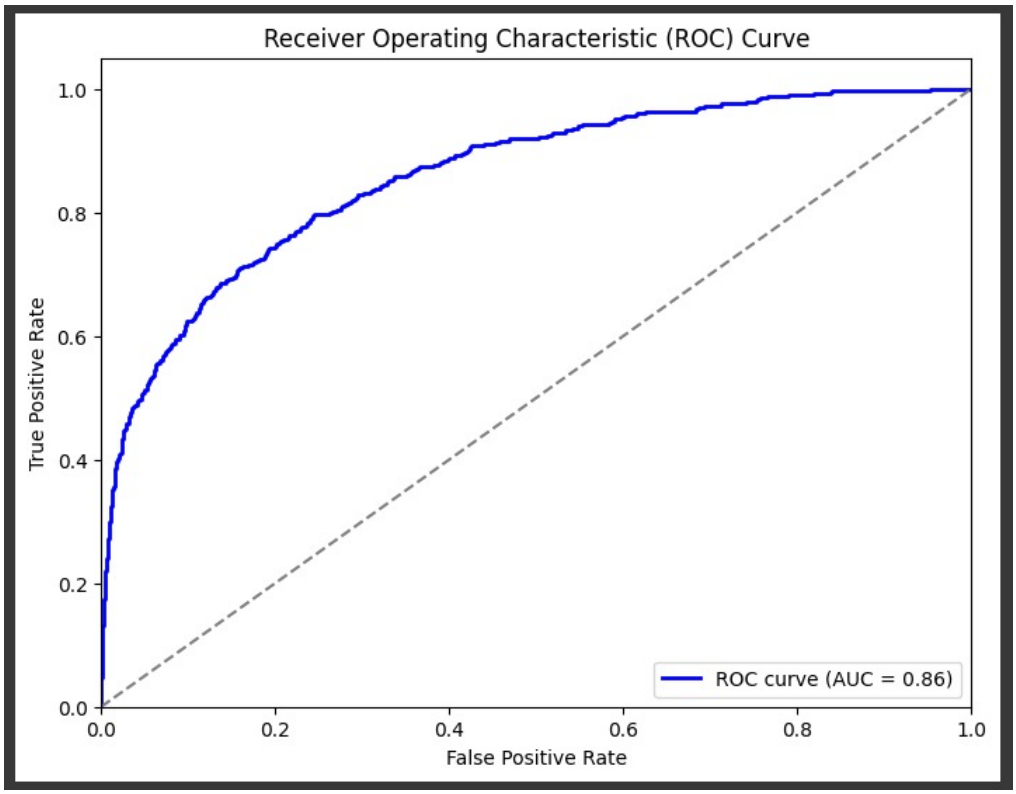

Figure 32: ROC Curve of Random Forest

Feature	Importance
Age	0.229935
NumOfProducts	0.121136
Balance	0.107864
Point Earned	0.097301
EstimatedSalary	0.095395
CreditScore	0.093683
Tenure	0.058619
Satisfaction Score	0.039910
IsActiveMember	0.033881

3.5 KNN Analysis

KNN, or k-nearest neighbors, is a simple yet powerful algorithm used in supervised learning for classification and regression tasks. It belongs to the category of instance-based learning, where the algorithm learns from the instances of the training data itself.

3.5.1 Mathematical Expressions

3.5.2 Given Data

Let $D = (X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$ be the training dataset, where D represents the feature vector of the i^{th} data point, and Y_n is its corresponding class label (for classification) or target value (for regression).

3.5.3 Distance Metric

Define a distance metric d to measure the distance between data points. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance.

3.5.4 Predicting New Data Point

Predicting New Data Point Given a new data point X_{new}, the KNN algorithm predicts its class label (for classification) or target value (for regression) by following these steps:

Calculate the distance $d(X_{new}, X_i)$ between X_{new} and each data point X_i in the training dataset.

Select the K nearest neighbors of X_{new} based on the calculated distances.

For classification, assign the class label of X_{new} as the majority class among its K nearest neighbors.

3.5.5 Methodology

These methods are followed sequentially to successfully construct a KNN model.

Data Collection: Gather data on bank customers, including demographics, transaction history, account status, customer interactions, etc.

Data Preprocessing: Clean and preprocess the data, which may include handling missing values, encoding categorical variables, and scaling numerical features.

Feature Selection: Select relevant features that are likely to influence customer churn, such as account balance, transaction frequency, customer tenure, etc.

Train-Test Split: Divide the dataset into training and testing sets to evaluate the performance of the KNN model.

Model Training: Train the KNN model on the training data. During training, the model learns the relationships between the features and the target variable (churn).

Parameter Tuning: Tune the hyperparameters of the KNN algorithm, such as the number of neighbors (k) and the distance metric, to optimize model performance.

Prediction: Use the trained KNN model to predict the churn probability or class label (churn or not churn) for customers in the testing set.

Evaluation: Evaluate the performance of the KNN model using appropriate evaluation metrics, such as accuracy, precision, recall, F1-score, and ROC-AUC.

Actionable Insights: Analyze the predictions to identify customers who are at high risk of churn. Take proactive measures, such as offering personalized incentives, improving customer service, or providing targeted marketing campaigns, to retain these customers.

3.5.6 Decision Rule:

The decision rule for KNN depends on the majority voting (for classification) or averaging (for regression) among the K nearest neighbors.

In summary, the KNN algorithm makes predictions based on the majority vote or average of the target values of the K nearest neighbors of a new data point, using a specified distance metric to measure proximity. This simple approach makes KNN easy to understand and implement, yet it can be quite effective in practice, especially for small to medium-sized datasets.

3.5.7 Analysis of KNN

Understand the interpretability of the KNN model and analyze which features contribute most to its predictions.

Determine whether the model's predictions align with domain knowledge and intuition about factors influencing customer churn in the banking indus-

try.Evaluate the scalability and efficiency of the KNN algorithm, considering its computational complexity and runtime performance, especially for large datasets.

Consider whether the computational requirements of KNN are feasible for real-time or batch processing in a banking environment.

3.5.8 Conclusion

kNN is a powerful and intuitive algorithm that can be applied to various machine learning tasks. By considering the nearest neighbors, it can capture complex relationships in the data without assuming any specific functional form. However, careful selection of hyperparameters and preprocessing steps is essential for optimal performance.

3.5.9 Advantages of KNN model

1. it's Simple and easy to implement.
2. there are No assumptions about the underlying data distribution.
3. it's Versatile, can be used for both classification and regression tasks.

3.5.10 Interpretation

We have constructed the confusion matrix of the data.

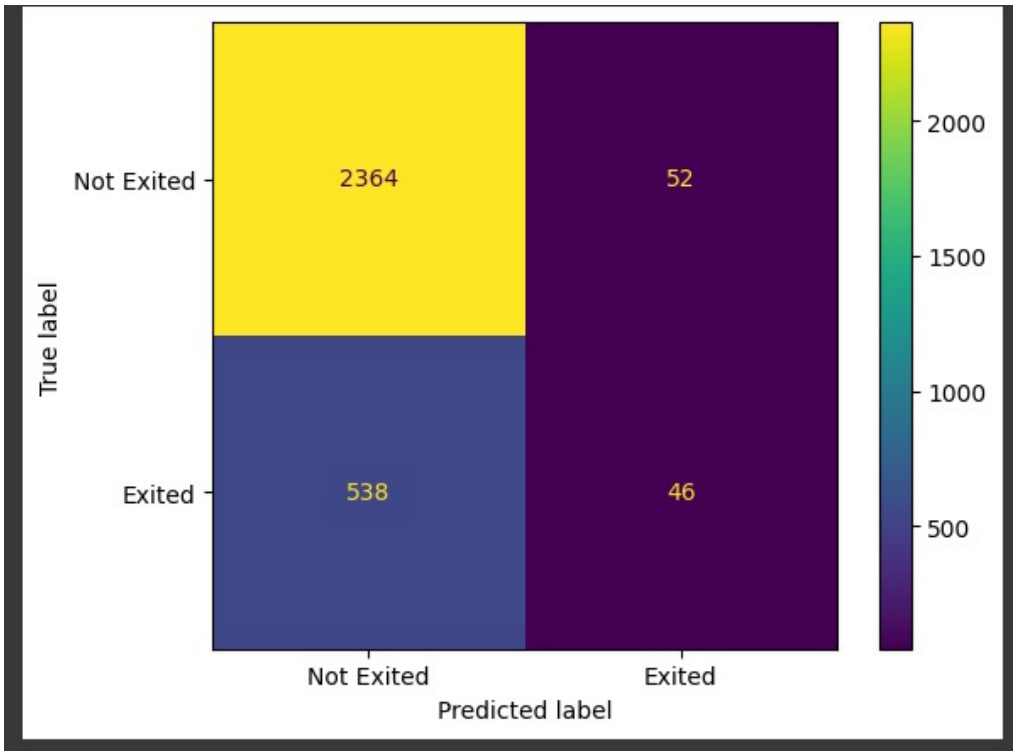

Figure 34: Confusion Matrix of KNN model

3.5.11 Classification Report

Overall, the accuracy of the model is 0.64, which means it gets 64% of the predictions correct. However, this high accuracy is mostly due to the fact that there are many more class 0 instances. The model performs poorly on the minority class (class 1).

	precision	recall	f1-score	support
0	0.81	0.98	0.89	2416
1	0.47	0.08	0.13	584
accuracy			0.80	3000
macro avg	0.64	0.53	0.51	3000
weighted avg	0.75	0.80	0.74	3000

Figure 35: Classification Report of KNN model

4 Accuracy Chart of various models

	Decision Tree	Random Forest	KNN	Logistic Regression	SVM
Model	Decision Tree	Random Forest	KNN	Logistic Regression	SVM
Scaling	Normal Data	Normal Data	Normal	Normal Data	rvf
Type	Gini	Gini	-	-	-
Precision	0.795667	0.861333	0.803333	0.704	0.849

Figure 36: Accuracy Chart

5 Conclusion and Further Scope

The analysis of the bank customer churn dataset reveals several important factors influencing customer attrition. We identified relevant features such as credit score, geography, age, tenure, balance, product holdings, card usage, and customer activity, among others, that significantly impact the likelihood of a customer leaving the bank. Through this study, we gained insights into the characteristics of customers who are more prone to churn, enabling us to develop effective churn prevention strategies.

By leveraging machine learning algorithms on this dataset, we can build predictive models to forecast customer churn with reasonable accuracy. These models can assist banks in proactively identifying at-risk customers and implementing targeted retention initiatives. Such initiatives may include personalized marketing campaigns, loyalty programs, improved customer service, and complaint resolution strategies, all aimed at enhancing customer satisfaction and loyalty.

After extensively evaluating various machine learning algorithms, it became clear from the comparison plot that the Random Forest model demonstrated the highest level of accuracy. The graph representing the Random Forest model closely aligned with the test values, suggesting superior predictive performance compared to other algorithms. Hence, based on this analysis, it can be inferred that the Random Forest algorithm is the most suitable choice for our dataset, offering dependable and precise predictions.

6 Acknowledgments

We would like to express our sincere appreciation to **Prof. Subhajit Dutta**, for his valuable guidance, support, and encouragement throughout this project. We also thank the TAs for this course. Their guidance was invaluable in shaping our work.

Additionally, we would like to thank **IIT Kanpur** for providing resources and facilities that were essential for the completion of this project.

We are grateful for the opportunity to work on this project, as it has been a great learning experience for all of us.

Lastly, we extend our thanks to our classmates and friends for their understanding and encouragement during this project.