PGPDSE FT Capstone Project – Final Report
Chennai-May-2024-Group 5

Project Summary

Batch details	DSE- FT-Chennai, May 2024 – G5
Team members	Ashwin G Chitralekha G Karthick V Mohamed Umar Farook A A Satheesh Kumar G
Domain of Project	BFSI
Proposed project title	Bank Customer Churn Prediction
Group Number	5
Team Leader	Satheesh Kumar G
Mentor Name	P V Subramanian

Date: 02-12-2024

P.V. Subramanian

Signature of the Mentor

Signature of the Team Leader

TABLE OF CONTENTS

ABSTRACT

This project focuses on predicting customer churn for an European MNC bank operating in France, Germany, and Spain. The goal is to identify factors contributing to customer attrition and build predictive models to forecast churn. By leveraging historical banking data, including customer demographics, transaction history, and account activity, machine learning algorithms will be applied to develop a churn prediction model. The insights gained will help the bank implement targeted retention strategies, reduce churn rates, and enhance customer satisfaction across the three key markets.

1) INTRODUCTION

1.1 OBJECTIVE

The objective of this project is to develop a robust customer churn prediction model for the European MNC bank operating in France, Germany, and Spain. By analysing historical data, including customer demographics, transaction patterns, and product usage, the bank aims to accurately predict which customers are at risk of leaving. The model will help the bank implement proactive retention strategies, such as personalized offers, targeted communication, and improved customer service, to reduce churn rates. Ultimately, the goal is to increase customer retention, enhance customer loyalty, minimize revenue loss, and strengthen the bank's competitive position in these markets.

1.2 CURRENT CHALLENGES

European multinational banks operating in France, Germany, and Spain face significant challenges in reducing customer churn due to complex regulatory environments, varying customer expectations, and increasing competition. Stringent GDPR and country-specific regulations complicate data usage and personalization, while cultural and language differences require localized customer engagement strategies. Additionally, legacy systems and slower digital transformation hinder banks' ability to compete with agile fintechs and neobanks offering seamless digital experiences.

Economic uncertainties, such as inflation and fluctuating interest rates, further impact customer trust and engagement. Rising demand for ethical banking practices and ESG compliance adds pressure on banks to align with sustainability goals. Moreover, inconsistent omnichannel experiences and branch closures risk alienating customers who prefer in-person service. Addressing these challenges requires banks to adopt innovative technologies, enhance customer segmentation, and deliver personalized, ethical, and seamless banking experiences across all channels.

1.3 PROPOSED BUSINESS PROBLEM STATEMENT

The European MNC bank faces a significant challenge with customer churn across its branches in France, Germany, and Spain. High churn rates lead to revenue losses and increased customer acquisition costs. Understanding why customers leave is critical to addressing this issue. Factors such as competition, customer dissatisfaction, changing banking needs, and poor service could contribute to churn. The bank seeks to predict which customers are likely to leave to take proactive steps, such as offering personalized services, improving engagement, or addressing pain points. By reducing churn, the bank aims to improve customer retention and strengthen its market position.

1.4 INDUSTRY REVIEW

1.4.1 Current Practices

In the banking industry, customer churn prediction has become a critical focus as banks aim to retain customers in a highly competitive market. Current practices to predict and prevent customer churn generally involve the use of historical data, customer behaviour analysis, and targeted marketing strategies.

Historical Data Analysis: Banks commonly rely on historical customer transaction data to detect patterns that indicate potential churn. Metrics such as account tenure, frequency of account usage, product holdings, and transaction volume are analysed to understand which customers are more likely to leave.

Segmentation and Profiling: Many banks segment their customers based on demographics, transaction behaviour, and product usage. This segmentation allows banks to personalize retention efforts, targeting high-risk segments with tailored offerings.

Customer Feedback and Sentiment Analysis: Banks frequently analyse customer feedback obtained through surveys, complaints, and support interactions. By analysing customer sentiment, banks can proactively address concerns and improve customer experience.

Predictive Analytics Models: Traditional predictive models like logistic regression, decision trees, and random forest classifiers are used by banks to predict churn based on historical data. These models are often deployed to flag customers with a high likelihood of leaving.

Personalized Retention Programs: Once high-risk customers are identified, banks offer personalized retention programs. These programs include loyalty rewards, product bundling, or discounts on fees, aimed at improving customer satisfaction and retention.

1.4.2 Background Research

Research into customer churn prediction has led to advancements in machine learning and data analytics methods specifically applied to the banking sector. Some key areas of background research include:

- **Machine Learning Techniques**: The banking industry has seen a shift from traditional statistical methods to more sophisticated machine learning approaches.

Algorithms like Support Vector Machines (SVM), Gradient Boosting, and Neural Networks have shown higher accuracy and robustness in predicting churn as they can capture complex relationships within customer data.

- **Feature Engineering and Importance**: Research has highlighted the importance of feature engineering to improve model performance. Features such as customer lifetime value, monthly transaction counts, average balance, and even non-traditional features like web/app activity have been found useful in predictive models.

- **Customer Lifecycle Value Models**: A growing area in churn research is understanding customer lifecycle value. Retaining high-value customers is especially critical, so models that predict not only churn but also expected lifetime value help banks prioritize retention efforts effectively.

- **Time-Series Analysis**: Time-series analysis has become a popular approach as it considers changes in customer behavior over time. By applying time-series techniques, banks can detect trends, seasonality, and unusual patterns that may precede churn.

- **Sentiment and Text Mining**: In recent years, text mining and sentiment analysis of unstructured data, such as customer feedback and social media interactions, have provided valuable insights into customer emotions and satisfaction levels. This approach allows banks to address issues that might lead to churn more effectively.

- **Churn Prevention Strategies**: Background research also includes effective churn prevention strategies. Studies have shown that combining predictive analytics with proactive customer service measures—such as early intervention for at-risk customers and personalized retention programs—significantly reduces churn rates.

1.4.3 Literature Survey

1) Predictive Model Development

Early churn models relied on logistic regression for interpretability. However, machine learning models such as decision trees, random forests, and gradient boosting have since outperformed traditional methods in capturing complex patterns (Verbeke et al., 2012). Deep learning models, like LSTMs, effectively detect sequential behaviours in time-series banking data (Shaikh et al., 2019).

2) Feature Engineering

Research identifies customer demographics, account balance, and transaction frequency as core features, with recency-frequency-monetary (RFM) values shown to be reliable predictors (Buckinx & Van den Poel, 2005). More recent studies use NLP for sentiment analysis from feedback, which improves prediction accuracy by capturing customer satisfaction levels (Ahmad & Baig, 2020).

3) Time-Series and Sequential Analysis

Time-series models (e.g., ARIMA) and sequence-based deep learning (LSTMs, CNNs) have advanced churn predictions by leveraging temporal patterns in customer behaviour (Kumar & Ravi, 2016; Zhang et al., 2021).

4) Sentiment Analysis Integration

Incorporating customer sentiment from reviews and complaints via NLP enhances prediction accuracy. Studies show that sentiment is strongly tied to churn rates and enriches models when combined with structured data (Chen et al., 2018).

5) Retention Strategies and Cost-Sensitive Learning

Retention strategies prioritize high-value at-risk customers. Cost-sensitive models improve accuracy and profitability by emphasizing correct classifications of high-value customers (Neslin et al., 2006; Liu & Shi, 2020).

6) Model Evaluation and Interpretability

Besides accuracy, interpretability tools like SHAP and LIME are essential for stakeholder transparency in financial predictions (Ribeiro et al., 2016). These methods clarify which features drive churn predictions, crucial for decision-making in banking.

2 OVERVIEW OF THE FINAL PROCESS

From Kaggle, we obtained the customer data of account holders at Anonymous Multinational Bank and the aim of the data will be predicting the Customer Churn. The dataset contains 10000 rows and 18 columns. We observe that there are no missing values, no duplicates in the dataset. We dropped a few unwanted columns such as RowNumber, CustomerId and Surname before proceeding to analysis. The dataset is not balanced as our target variable, Exited is having 20.38% of observations for the minority class. We treated the data imbalance by applying Synthetic Minority Oversampling Technique, or SMOTE for short.

Detailed Exploratory Data Analysis was done to help us to look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Robust scaling that removes the median and scales the data according to the quantile range was applied to the data to the three continuous numerical columns, namely, Age, Balance and Estimated Salary.

We selected top 8 features, namely, Tenure, Balance, IsActiveMember, EstimatedSalary, Complain, Satisfaction Score, Geography, Gender for building the model by using Recursive Feature Elimination (RFE) that works by searching for a subset of features by starting with all features in the dataset and successfully removing features until the desired number remains.

The most important assumption of absence of multi-collinearity was checked by calculating the Variance Inflation Factor (VIF). All these variables are found to be non-collinear as their VIF value < threshold value, 5 as shown below:

1. Satisfaction Score (3.564209)
2. Tenure (3.039833)
3. Gender (1.998514)
4. IsActiveMember (1.893380)
5. Geography_code (1.708588)
6. Complain (1.247102)
7. Balance (1.112117)
8. EstimatedSalary (1.000601)

We have built the following seven models:

Average of the 10 folds of Recall measure is given within parenthesis.

1. Logistic Regression (99.8039%)
2. Decision Tree (CART) (99.1655%)
3. Random Forest (99.7549%)
4. KNN (99.8039%)
5. Naïve Bayes (99.8039%)
6. XGBoost (99.8039%)
7. AdaBoost (99.8039%)

Before building the Logistic Regression model, all the applicable assumptions are tested and found to be satisfied.

We used K Fold cross validation to assess the performance of the above models. The top priority of this project is to identify if a customer will churn or won't. It's important that we don't predict churning as non-churning customers. That's why the model needs to be evaluated on the "Recall"- metric.

Based on this measure, we observe that all models perform well with the higher recall of above 99%. We had chosen Logistic Regression model as our best model because of its power of interpretation as Logistic regression estimates the probability of an event occurring. We have found that all the assumptions of Logistic Regression model are satisfied.

We observe that the McFadden R square (Pseudo R square) is 98 % and the model fitness is very good. This McFadden approach is one minus the ratio of two log likelihoods. The numerator is the log likelihood of the logit model selected and the denominator is the log likelihood if the model just had an intercept.

The following variables are significant at 5% level of significance:

- IsActiveMember
- Complain

We observe that the probability of customers who churn is 99.99 % if they have complaints. So, the most important variable is Complain among all the independent variables to predict the target variable.

3 Step-by-step walk through of the solution

3.1 Data Dictionary

Please refer to the appendix for more details.

3.2 Variable categorization (count of numeric and categorical)

Numerical variables

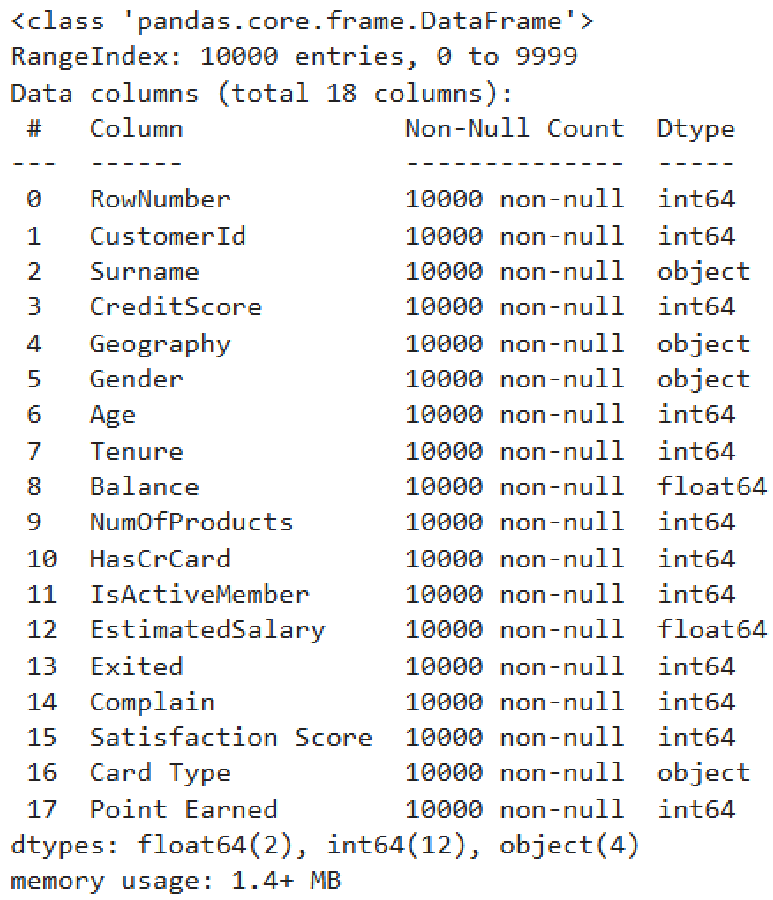
There are 14 numerical variables, inclusive of 1 Target Variable and listed below:

1) RowNumber
2) CustomerId
3) CreditScore
4) Age
5) Tenure
6) Balance
7) NumOfProducts
8) HasCrCard
9) IsActiveMember
10) EstimatedSalary
11) Exited (Target Variable)
12) Complain
13) Satisfaction Score
14) Point Earned

Categorical variables

There are 4 categorical variables and listed below:

1) Surname
2) Geography
3) Gender
4) Card Type

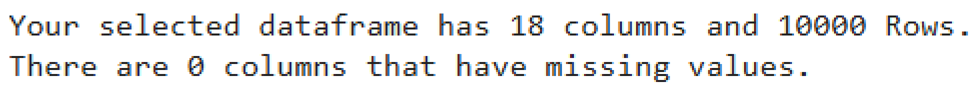


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   RowNumber           10000 non-null  int64  
 1   CustomerId          10000 non-null  int64  
 2   Surname             10000 non-null  object 
 3   CreditScore         10000 non-null  int64  
 4   Geography           10000 non-null  object 
 5   Gender              10000 non-null  object 
 6   Age                 10000 non-null  int64  
 7   Tenure              10000 non-null  int64  
 8   Balance             10000 non-null  float64
 9   NumOfProducts       10000 non-null  int64  
 10  HasCrCard           10000 non-null  int64  
 11  IsActiveMember      10000 non-null  int64  
 12  EstimatedSalary     10000 non-null  float64
 13  Exited              10000 non-null  int64  
 14  Complain            10000 non-null  int64  
 15  Satisfaction Score  10000 non-null  int64  
 16  Card Type           10000 non-null  object 
 17  Point Earned        10000 non-null  int64  
dtypes: float64(2), int64(12), object(4)
memory usage: 1.4+ MB
```

3.3 Pre-Processing Data Analysis (count of missing/ null values, redundant columns, etc.)

3.3.1 Count of Missing values

There are 0 columns that have missing values.



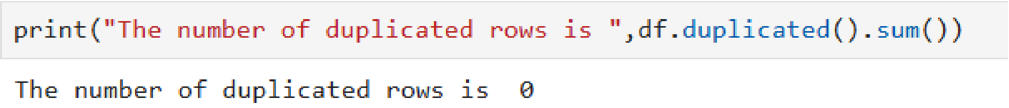
```
Your selected dataframe has 18 columns and 10000 Rows.
There are 0 columns that have missing values.
```

3.3.2 Redundant columns or unwanted columns

Redundant columns are RowNumber, CustomerId and Surname and they ar e dropped. We observe that the ID variables and customer personal details such as RowNumber, CustomerId and Surname etc. will not add value to our analysis and hence we need to remove them for our analysis.

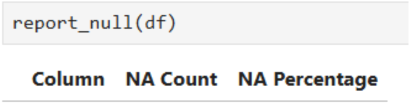
3.3.3 Duplicate rows & Missing Values

There are no duplicates in our dataset & There are no missing values.



```
print("The number of duplicated rows is ",df.duplicated().sum())
```

```
The number of duplicated rows is  0
```

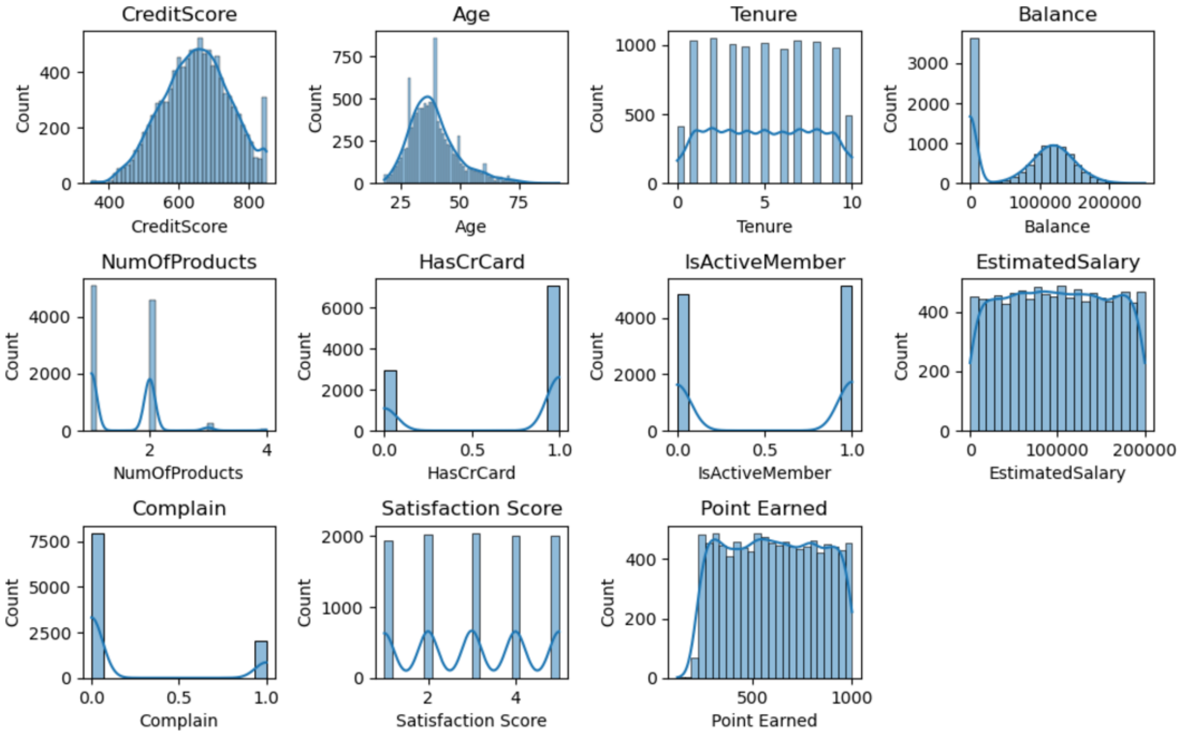


```
report_null(df)
```

Column	NA Count	NA Percentage

3.4 Distribution of Variables

Histograms for the numerical variables



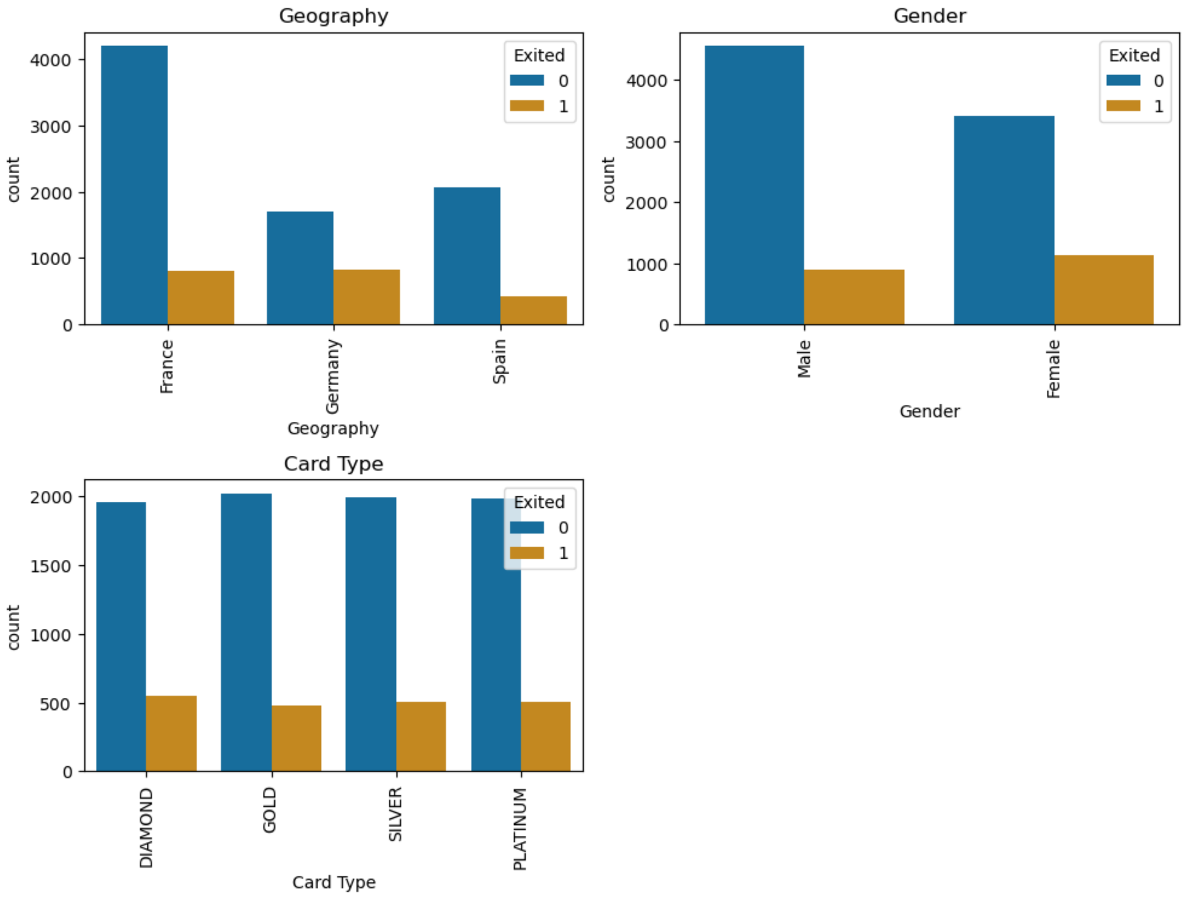
Observations:

None of the Variables are Normally Distributed.

3.4.1 Bivariate Analysis

A. Categorical Variables

Barplots for the categorical variables



Observations:

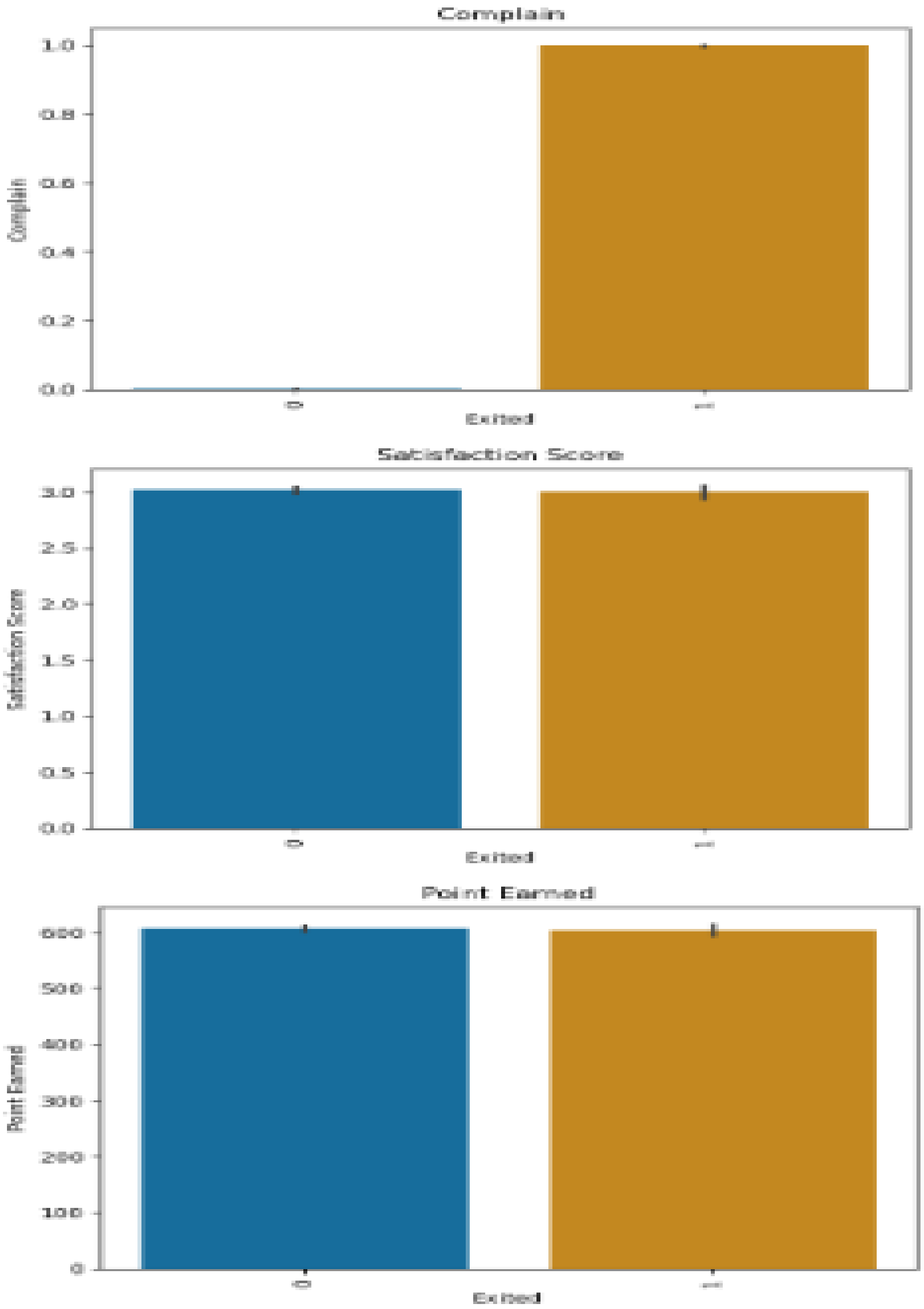
France has the greater number of Customers among the three countries France, Germany & Spain.
Almost one third of the Germany customers are more likely to churn.

Based on Gender, Female customers are more likely to churn than Male.

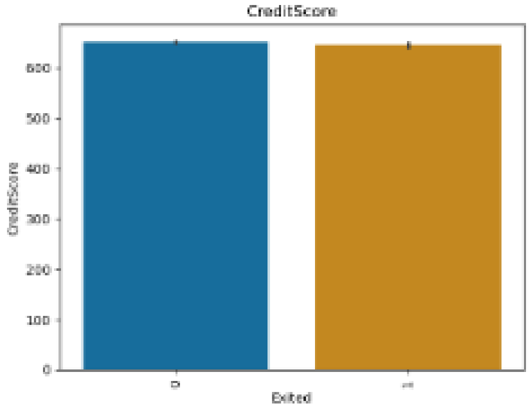
There is so much difference among the Card Type based on Churn.

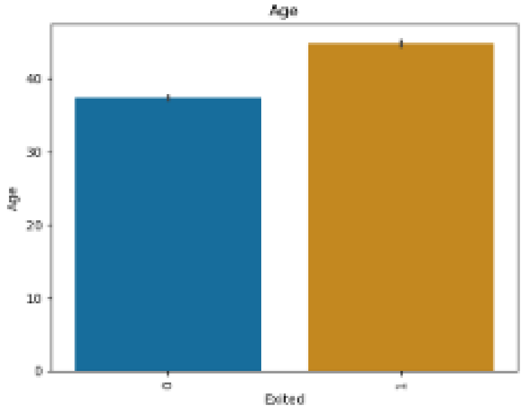
B. Numerical Variables

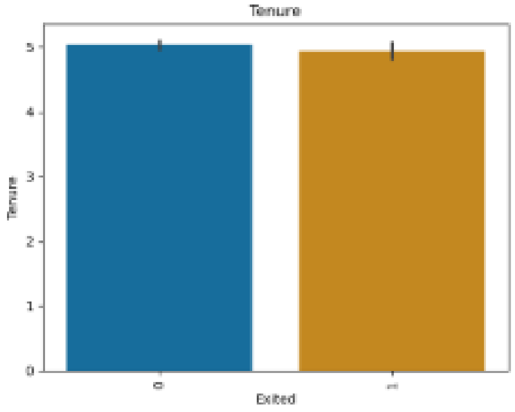
Barplots for discrete numerical variables

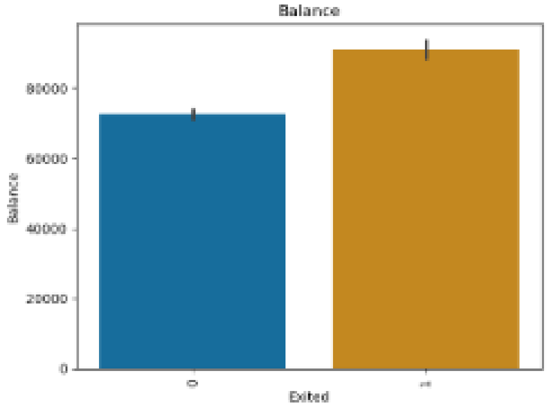


Observations: The Churn rates are more or less the same for Satisfaction Score, Points Earned except for Complain Variable.

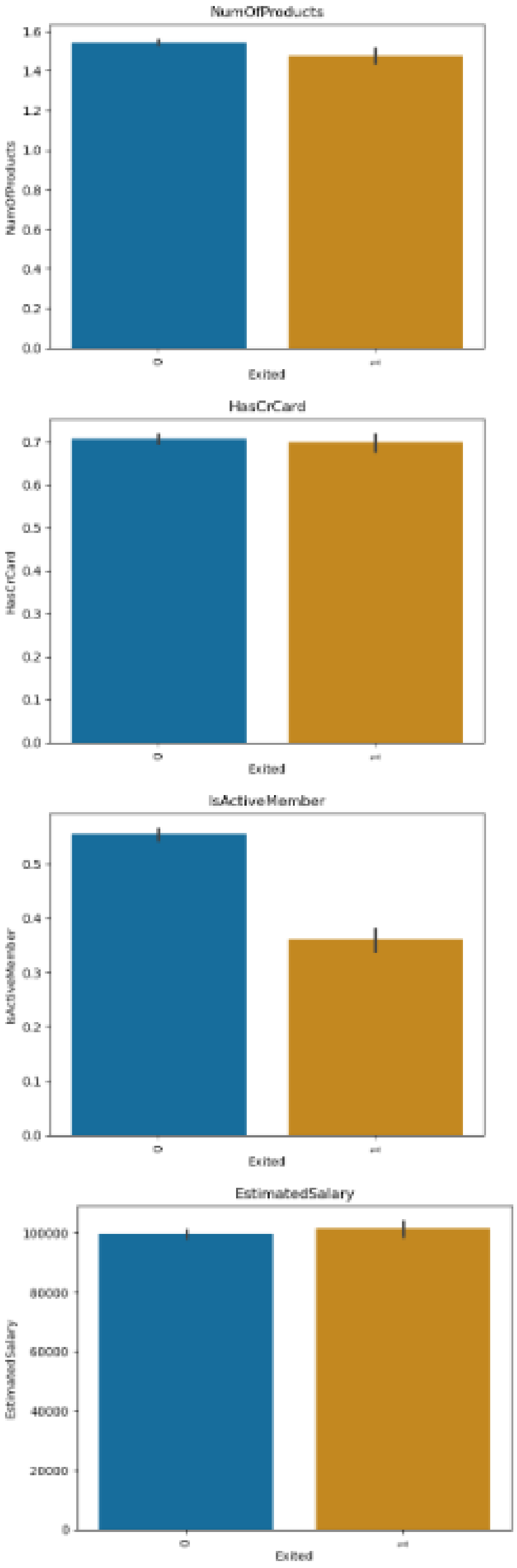








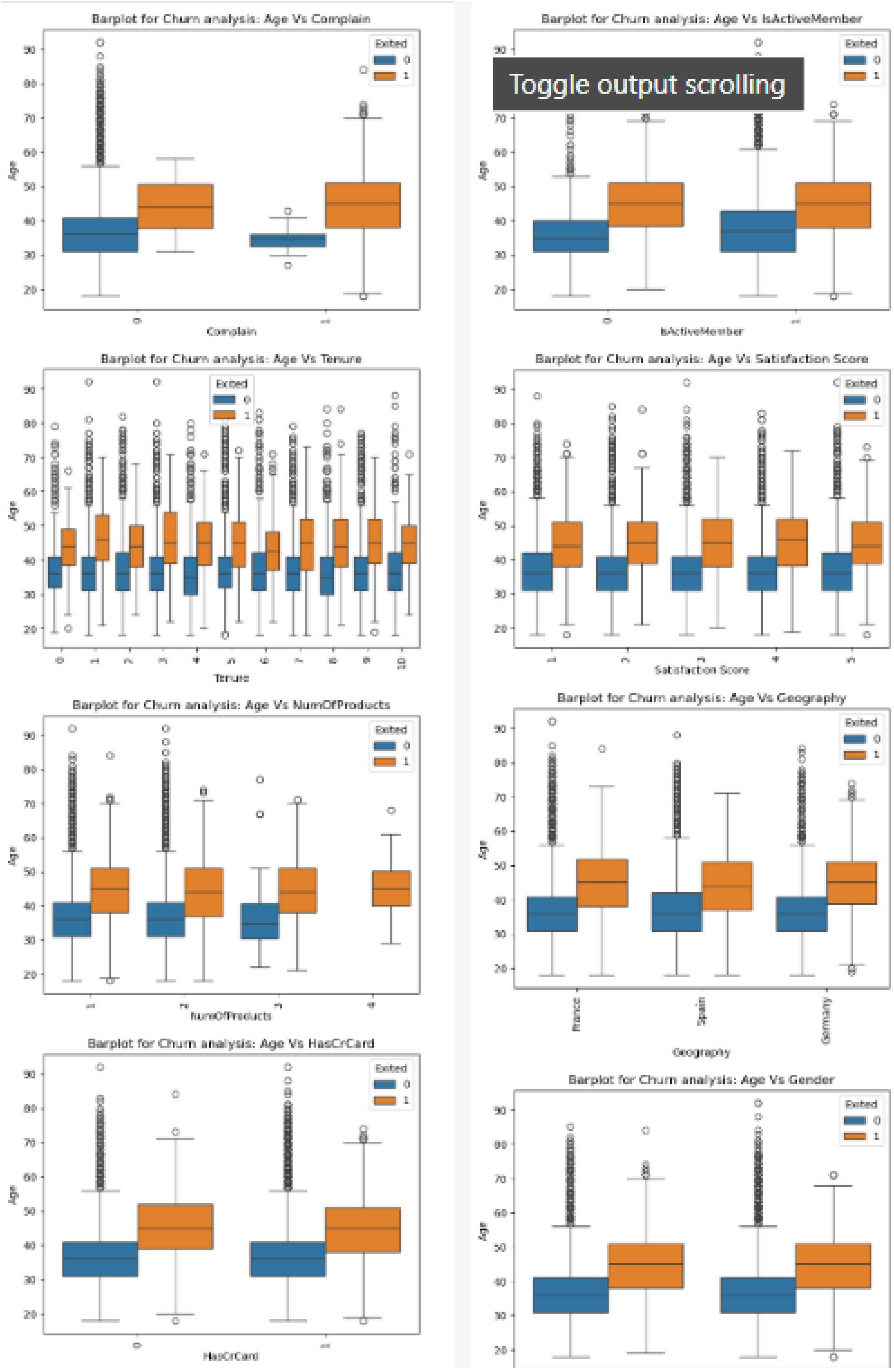
Observations: The Churn rates are more or less the same for CreditScore, Tenure except for Age & Balance



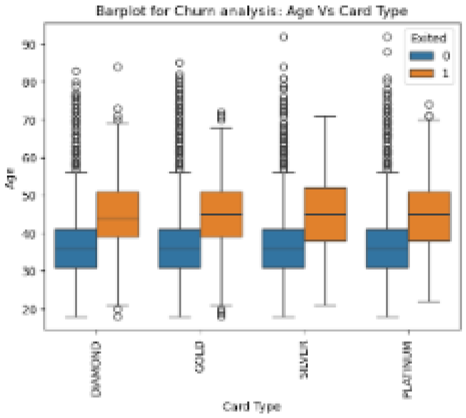
Observations: The Churn rates are more or less the same for EstimatedSalary, NumofProducts, HasCrCard except the IsActiveMember Variable.

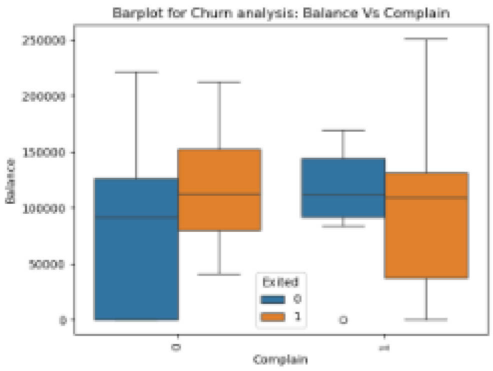
3.4.2 Multivariate Analysis

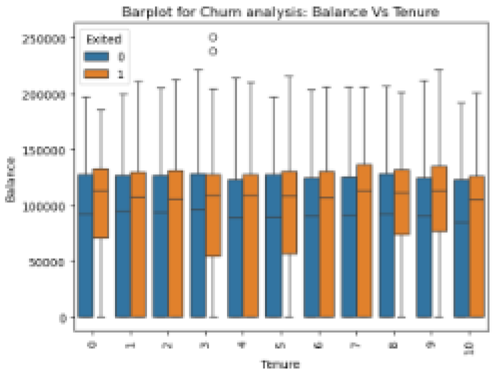
Boxplots for numerical variables group by the Target variable

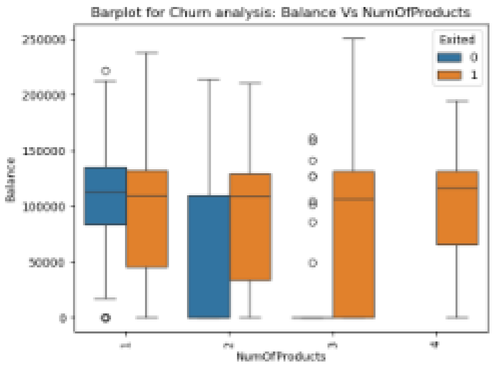


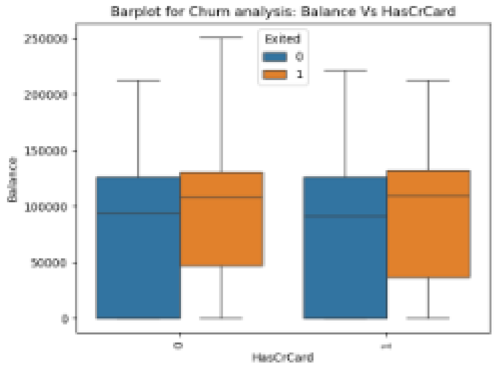
Observations: Customers with Increase in Age are more likely to churn.

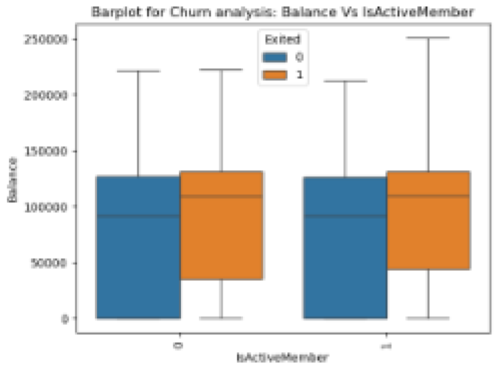


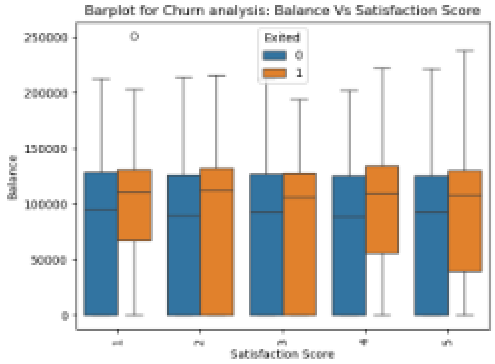


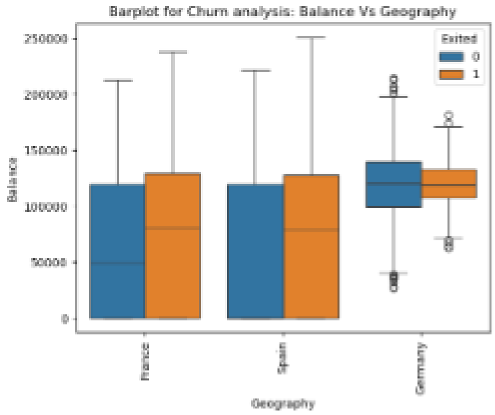






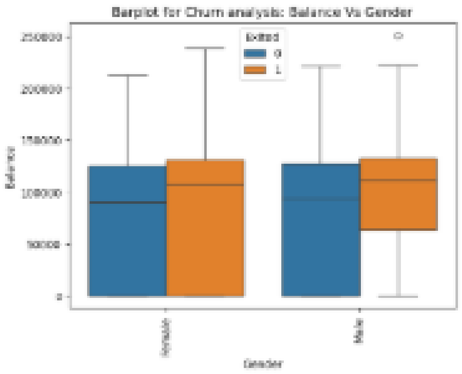


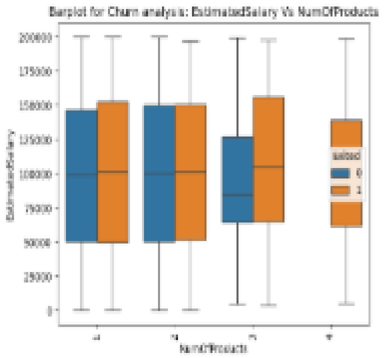


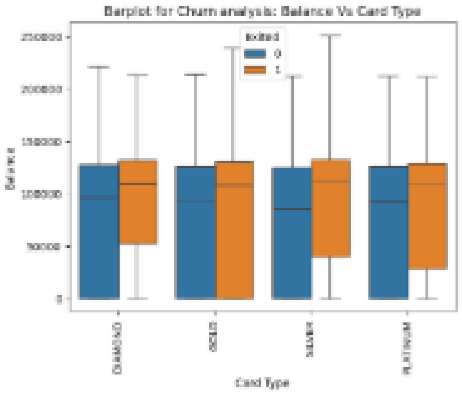


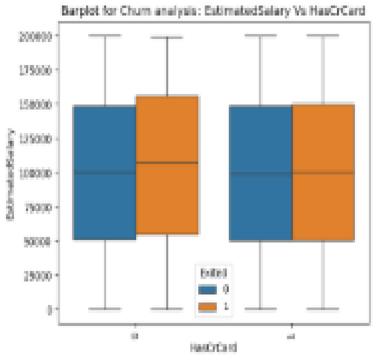
Observations:

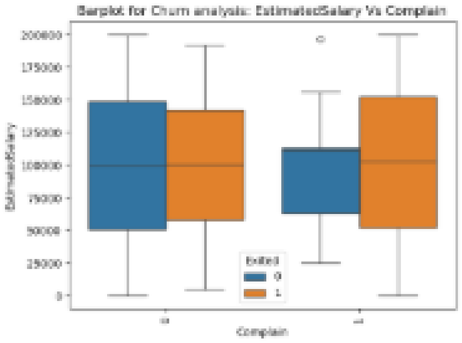
Customers with Increase in Age are more likely to Churn based on Card Type.
Tenure, NumofProducts, HasCrCard, IsActiveMember, Satisfaction Score and Geography does not show a strong impact on churn based on balance.

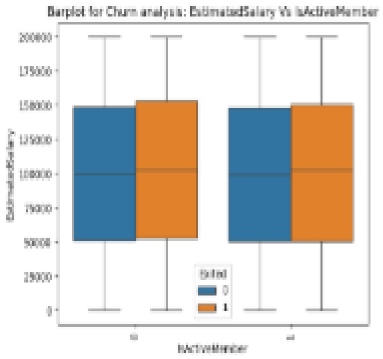


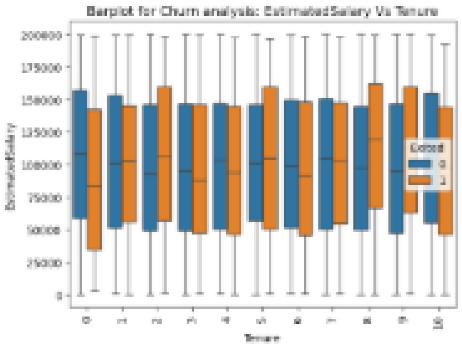


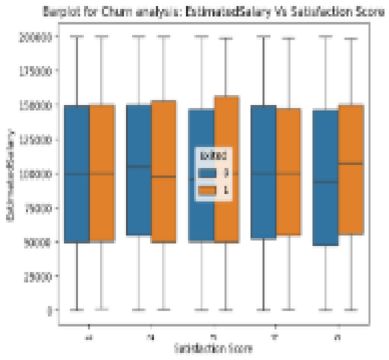






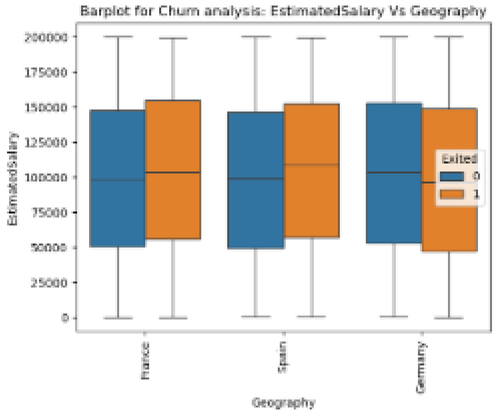


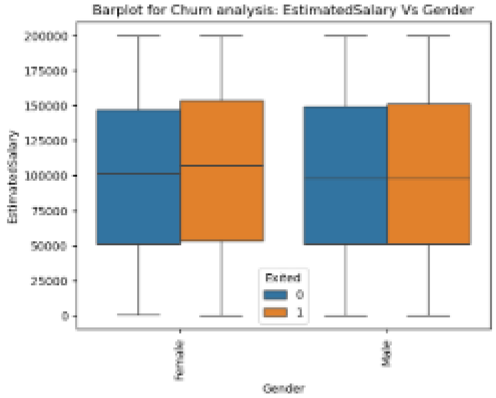


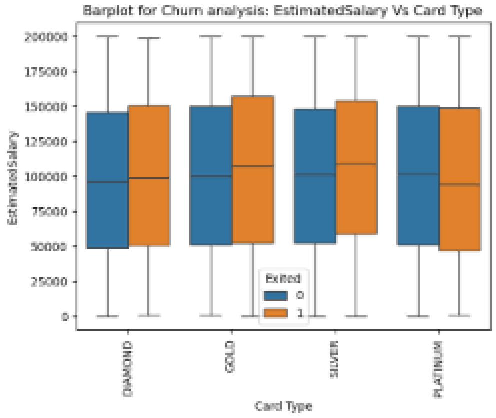


Observations:

Gender & Card Type does not show a strong impact on churn based on balance.
Complaints may correlate with churn.
Tenure does not appear to significantly impact churn based on estimated salary.
Being an active member, NumofProducts, HasCard, Satisfaction Score does not significantly impact churn in relation to estimated salary.







Observations:

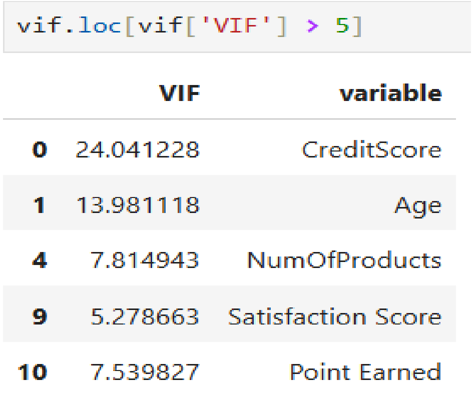
Geography seems not to be a strong differentiating factor for customer churn when considering estimated salary.

Gender does not appear to significantly impact churn based on estimated salary.

Card type seems to have no significant influence on customer churn in relation to estimated salary.

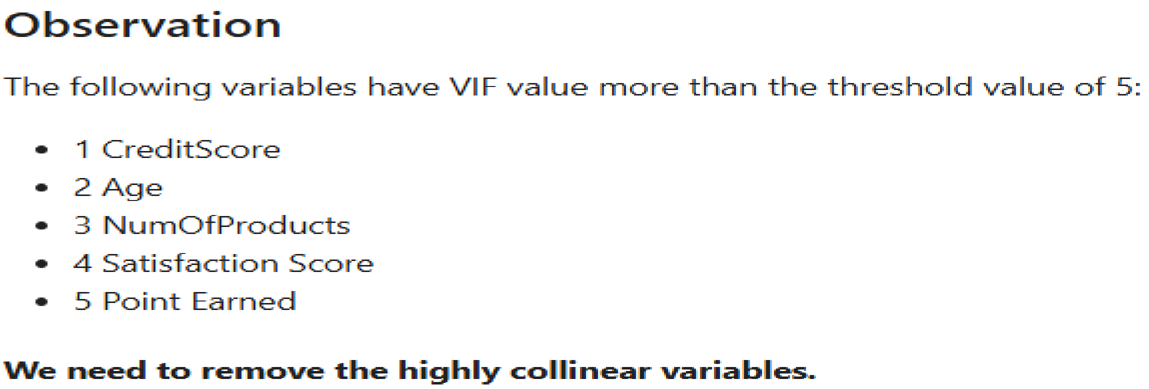
3.5 Check for multi-collinearity

The higher the value, the greater the correlation of the variable with other variables. Values of more than 4 or 5 are sometimes regarded as being moderate to high, with values of 10 or more being regarded as very high. We need to remove variables having greater than the threshold value of 5 from our dataset.



```
vif.loc[vif['VIF'] > 5]
```

	VIF	variable
0	24.041228	CreditScore
1	13.981118	Age
4	7.814943	NumOfProducts
9	5.278663	Satisfaction Score
10	7.539827	Point Earned



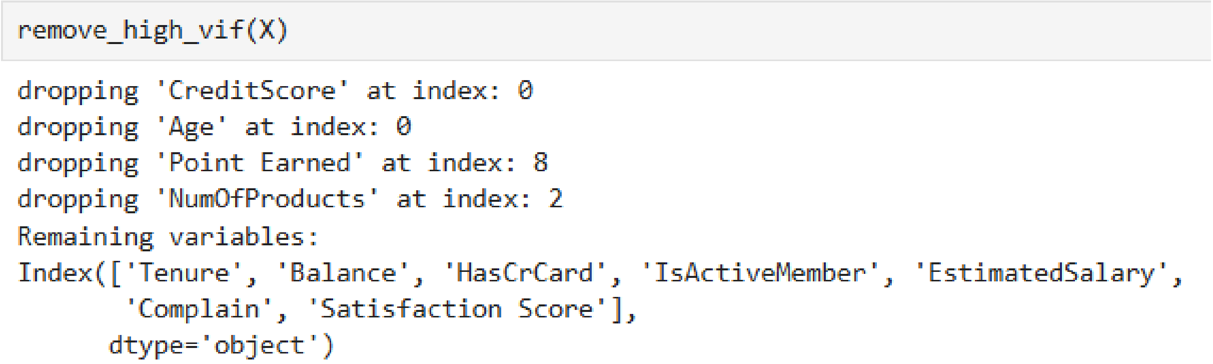
Observation

The following variables have VIF value more than the threshold value of 5:

- 1 CreditScore
- 2 Age
- 3 NumOfProducts
- 4 Satisfaction Score
- 5 Point Earned

We need to remove the highly collinear variables.

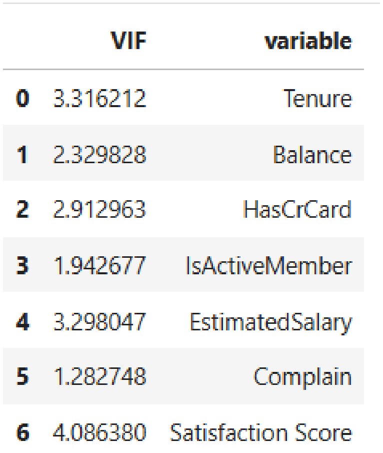
We removed the highly collinear variables programmatically.



```
remove_high_vif(X)
```

```
dropping 'CreditScore' at index: 0
dropping 'Age' at index: 0
dropping 'Point Earned' at index: 8
dropping 'NumOfProducts' at index: 2
Remaining variables:
Index(['Tenure', 'Balance', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary',
       'Complain', 'Satisfaction Score'],
      dtype='object')
```

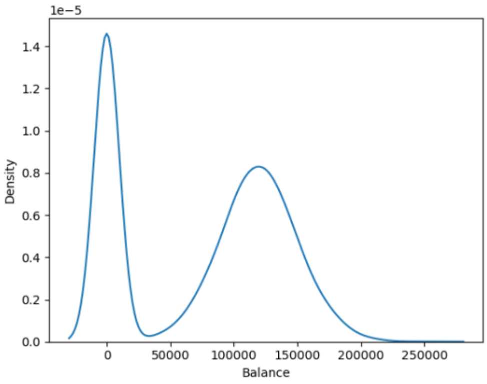
We again check if the multi-collinearity exists in our data.

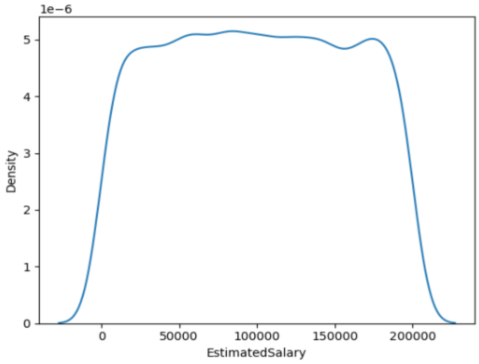


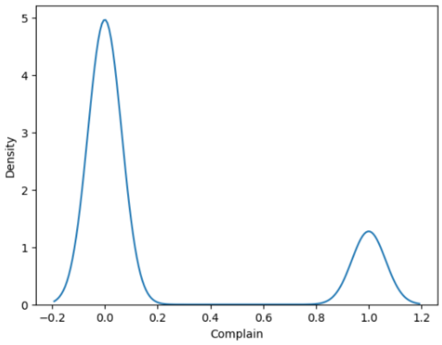
	VIF	variable
0	3.316212	Tenure
1	2.329828	Balance
2	2.912963	HasCrCard
3	1.942677	IsActiveMember
4	3.298047	EstimatedSalary
5	1.282748	Complain
6	4.086380	Satisfaction Score

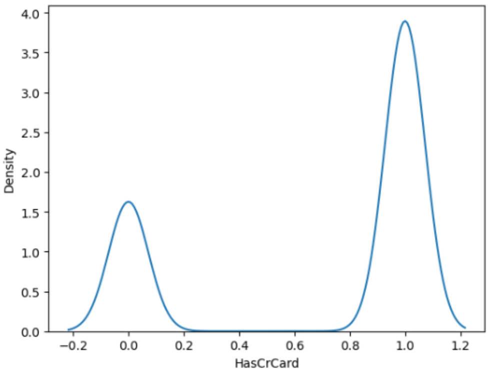
We observe that we have removed multi-collinearity from the data.

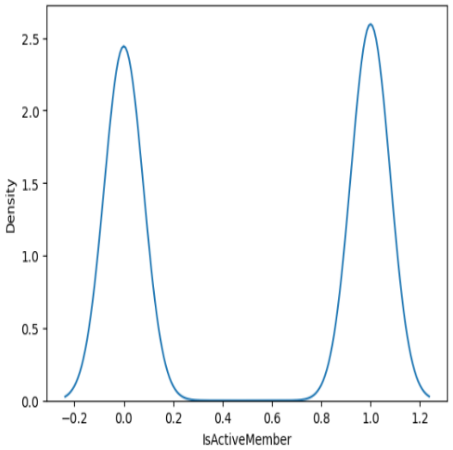
3.6 distribution of variables

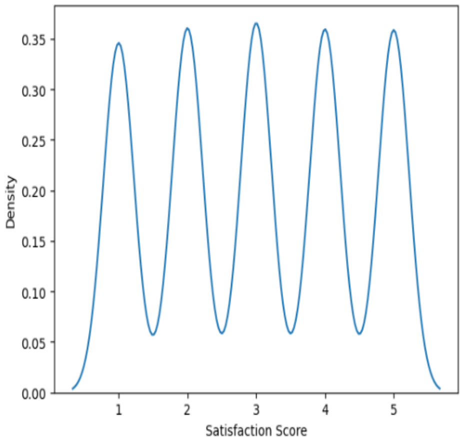












Observations:

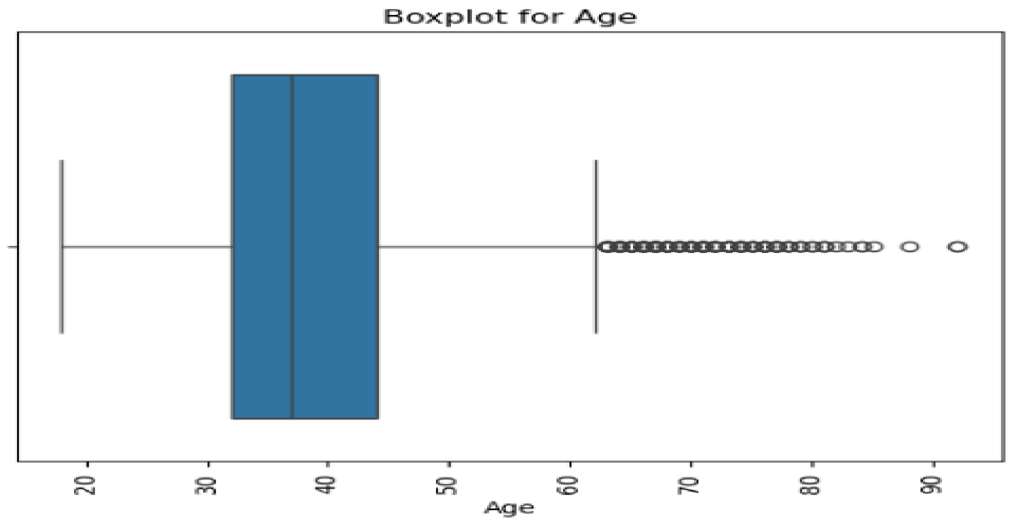
None of the continuous numerical variables are normally distributed.

3.7 Presence of outliers and its treatment

Outliers badly affect mean and standard deviation of the dataset. · It increases the error variance and reduces the power of statistical tests. By applying outlier treatment, machine learning practitioners can handle extreme values effectively. The primary goals of outlier treatment are: Identifying Outliers: Through various statistical methods, such as visualizations and mathematical approaches, outliers can be detected within a dataset. A commonly used rule says that a data point is an outlier if it is more than 1.5 · IQR above the third quartile or below the first quartile.

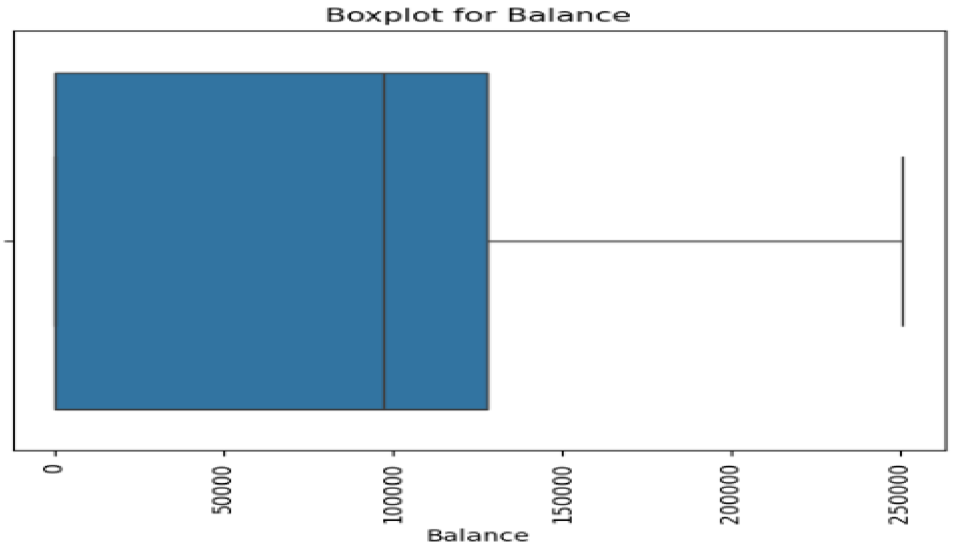
We are interested to identify the outliers in our continuous numerical variables such as 'Age', 'Balance', 'EstimatedSalary' that affects the mean & standard deviation rather than the discrete numerical variables. Discrete variables are typically categorical, meaning they take on a limited number of values or categories.

However, if the outlier is physically possible you should consider it.



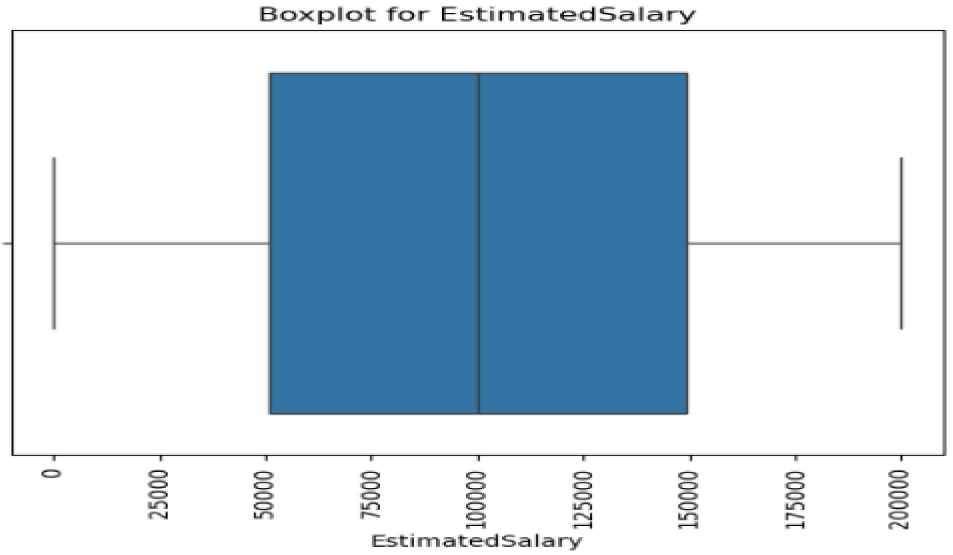
Observations:

The variable, **Age** has outliers outside of the maximum whisker. We observe the minimum and maximum values of Age are 18 and 92 respectively. As 92 years of age is a possible value, we need not remove the outliers but retain them for this variable.



Observations:

The variable, **Balance** has no outliers.



Observations:

The variable, **EstimatedSalary** has no outliers.

3.8 Statistical significance of variables

a) Numerical variables

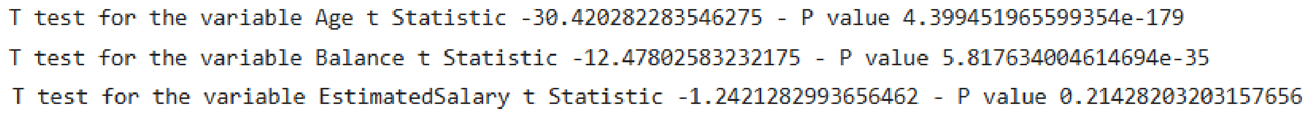
i. T test

An unpaired t-test (also known as an independent t-test) is a statistical procedure that compares the averages/means of two independent or unrelated group of numerical variables to determine if there is a significant difference between the two.

Hypotheses are assumptions about reality whose validity is possible but not yet proven. Two hypotheses are always formulated that assert exactly the opposite. These two hypotheses are the null hypothesis and the alternative hypothesis.

Null hypothesis: H_0: There is no mean difference between the two groups in the population. **μ1 = μ2**

Alternative hypothesis: H_1: The two population means are not equal. The two groups are not from the same population. **μ1≠ μ2**



```
T test for the variable Age t Statistic -30.420282283546275 - P value 4.399451965599354e-179
T test for the variable Balance t Statistic -12.47802583232175 - P value 5.817634004614694e-35
 T test for the variable EstimatedSalary t Statistic -1.2421282993656462 - P value 0.21428203203157656
```

Observations:

Based on the unpaired t test, we find that

There is a statistically significant difference between the mean values of two groups of the Variable, listed below:

1) Age 2) Balance

There is no statistically significant difference between the mean values of two groups of the Variable, listed below:

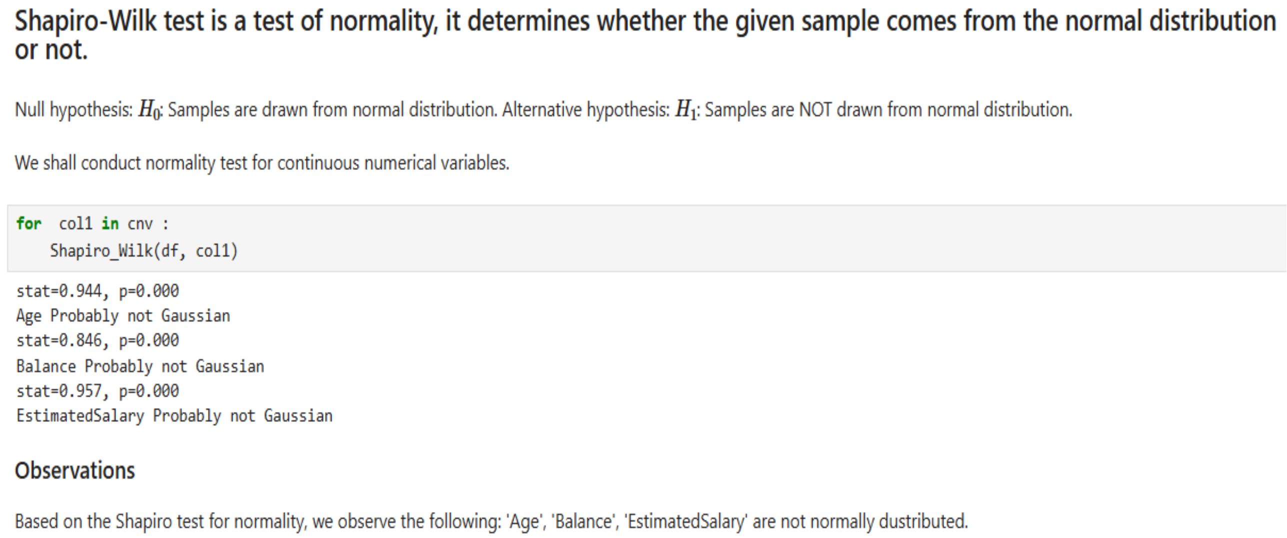
1) EstimatedSalary

ii. Normality Tests

Shapiro-Wilk Test: Tests whether a data sample has a Gaussian distribution.

Assumptions: Observations in each sample are independent and identically distributed (IID).

Interpretation
H0: the sample has a Gaussian distribution.
H1: the sample does not have a Gaussian distribution.



Shapiro-Wilk test is a test of normality, it determines whether the given sample comes from the normal distribution or not.

Null hypothesis: H_0: Samples are drawn from normal distribution. Alternative hypothesis: H_1: Samples are NOT drawn from normal distribution.

We shall conduct normality test for continuous numerical variables.

```
for  col1 in cnv :
    Shapiro_Wilk(df, col1)
```

```
stat=0.944, p=0.000
Age Probably not Gaussian
stat=0.846, p=0.000
Balance Probably not Gaussian
stat=0.957, p=0.000
EstimatedSalary Probably not Gaussian
```

Observations

Based on the Shapiro test for normality, we observe the following: 'Age', 'Balance', 'EstimatedSalary' are not normally dustributed.

iii. Point biserial correlation

Calculate Correlation Between Continuous & Binary Target Variable

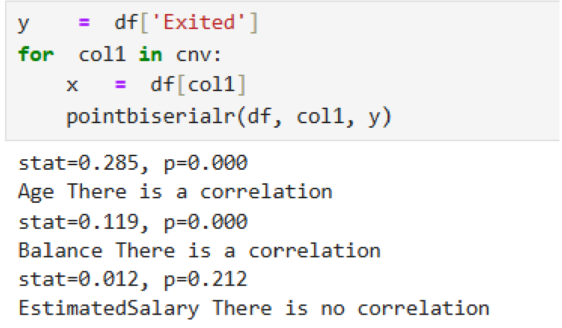
Point biserial correlation is used to calculate the correlation between a binary categorical variable (a variable that can only take on two values) and a continuous variable and has the following properties:

Point biserial correlation can range between -1 and 1. For each group created by the binary variable, it is assumed that the continuous variable is normally distributed with equal variances. For each group created by the binary variable, it is assumed that there are no extreme outliers.

The hypotheses for point biserial correlation thus result in:

Null hypothesis: The correlation coefficient r = 0 (There is no correlation)

Alternative hypothesis: The correlation coefficient r ≠ 0 (There is a correlation)



```
y    =  df['Exited']
for  col1 in cnv:
    x   =  df[col1]
    pointbiserialr(df, col1, y)
```

```
stat=0.285, p=0.000
Age There is a correlation
stat=0.119, p=0.000
Balance There is a correlation
stat=0.012, p=0.212
EstimatedSalary There is no correlation
```

Observations:

There is **correlation** between the following variables and the Target binary variable:

1. Age
2. EstimatedSalary

There is **no correlation** between the following variables and the Target binary variable:

1. Balance

iv. Chi sqaure test of independence

The objective is to determine whether the association between two qualitative variables is statistically significant. The formulation of the hypotheses for this statistical analysis is something like this.

Null Hypothesis (H0): There is no substantial relationship between the two variables (in case of independence test), or there is no difference.

Alternative Hypothesis (H1): There is a substantial relationship between the two variables (in case of independence test), or there is a difference.

Read more at: https://analyticsindiamag.com/ai-mysteries/how-to-use-the-chi-square-test-for-two-categorical-variables/

a) Variable, Geography

The important assumption: No more than 20% of the cells have and expected cell count < 5. This can be checked by looking at the expected frequency table.

```
Chi2ContingencyResult(statistic=300.6264011211942, value=5.245736109572763e-66, dof=2, expected_freq=array([[3992.1468, 1021.8532],  [1997.6658,  511.3342],
[1972.1874,  504.8126]]))
```

```
Percentage of cells with expected counts less than 5: 0.00%
```

```
ChiSq Stat: 300.6264011211942, P value: 5.245736109572763e-66
```

Observations:

Independent Variable, Geography and Target variable are dependent

b) Variable, Gender

The important assumption: No more than 20% of the cells have and expected cell count < 5. This can be checked by looking at the expected frequency table.

```
Chi2ContingencyResult(statistic=112.39655374778587, pvalue=2.9253677618642e-26, dof=1, expected_freq=array([[3617.1366,  925.8634], [4344.8634, 1112.1366]]))
```

```
Percentage of cells with expected counts less than 5: 0.00%
```

ChiSq Stat: 112.39655374778587, P value: 2.9253677618642e-26

Observations:

Independent Variable, Gender and Target variable are dependent.

c) Variable, Card Type

The important assumption: No more than 20% of the cells have and expected cell count < 5. This can be checked by looking at the expected frequency table.

```
Chi2ContingencyResult(statistic=5.053223027060927,pvalue=0.16794112067810177,
dof=3, expected_freq=array([[1996.0734, 510.9266],[1992.0924, 509.9076],
[1986.519 ,  508.481 ],[1987.3152,  508.6848]]))

Percentage of cells with expected counts less than 5: 0.00%

ChiSq Stat: 5.053223027060927, P value: 0.16794112067810177
```

Observations:

Independent Variable and Target variable are independent

Inferences:

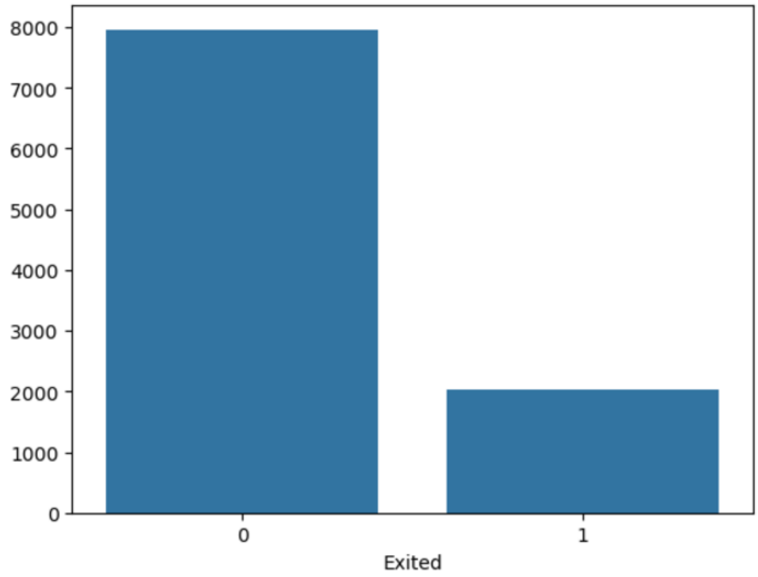
1) There is a **substantial relationship** between the two variables Geography and Target since chi2_stat = 300.63, p = 5.246 e-66

2) There is a **substantial relationship** between the two variables Gender and Target since chi2_stat = 112.40, p = 2.9254 e-26

3) There is **NO substantial relationship** between the two variables Card Type and Target since chi2_stat = 5.05, p = 0.167941

3.9 Class imbalance and its treatment

Bar chart for classes in the Target variable



Observations:

1. Dataset is **highly imbalanced** as the minority class, ** 1** denoting Exited is having around 20.38% of observations while the other class, **'0'** denoting "Not Exited" is having the balance of 79.62% of the observations.
2. So, we need to choose the model performance measure carefully to avoid bias to the majority class.
3. Precision and recall are common metrics used when evaluating classification models for detection of a certain important class. Recall represents how many samples of the important. class was discovered by the model of all the samples in the class, while precision represents the accuracy of predictions for that certain class.

In this project, the important class is "Exited". True positives is the number of correctly identified data points of the important class, False positives is the number of data points incorrectly identified as important and False negatives is the number of data points incorrectly identified as not important. So, we shall use **Recall of the minority class** as our measure of model performance.

4. Precision and recall are common metrics used when evaluating classification models for detection of a certain important class. Recall represents how many samples are important. class was discovered by the model of all the samples in the class, while precision represents the accuracy of predictions for that certain class.

SMOTE

The problem with imbalanced classification is that there are too few examples of the minority class for a model to effectively learn the decision boundary.

One way to solve this problem is to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model.

An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective.

Perhaps the most widely used approach to synthesizing new examples is called the Synthetic Minority Oversampling Technique, or SMOTE for short. This technique was described by Nitesh Chawla, et al. in their 2002 paper named for the technique titled “SMOTE: Synthetic Minority Over-sampling Technique.”

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space.
We shall apply SMOTE on training data.

3.10 Feature Engineering

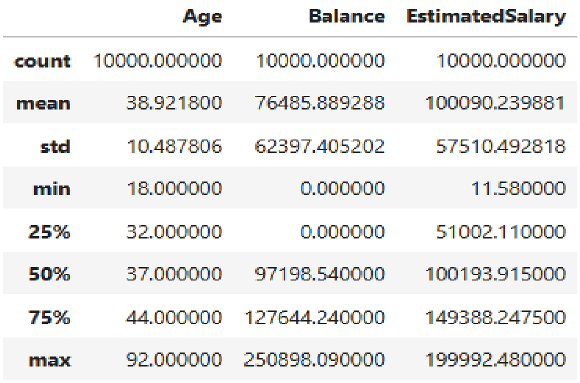
3.10.1. Whether any transformations required

Data transformation is used when data needs to be converted to match that of the destination system. We have performed label encoding to make our data suitable for model building.

3.10.2. Scaling the data

Data scaling is applied to numeric columns. In our dataset we have three continuous numerical columns:

1. Age
2. Balance
3. EstimatedSalary



	Age	Balance	EstimatedSalary
count	10000.000000	10000.000000	10000.000000
mean	38.921800	76485.889288	100090.239881
std	10.487806	62397.405202	57510.492818
min	18.000000	0.000000	11.580000
25%	32.000000	0.000000	51002.110000
50%	37.000000	97198.540000	100193.915000
75%	44.000000	127644.240000	149388.247500
max	92.000000	250898.090000	199992.480000

Robust scaling

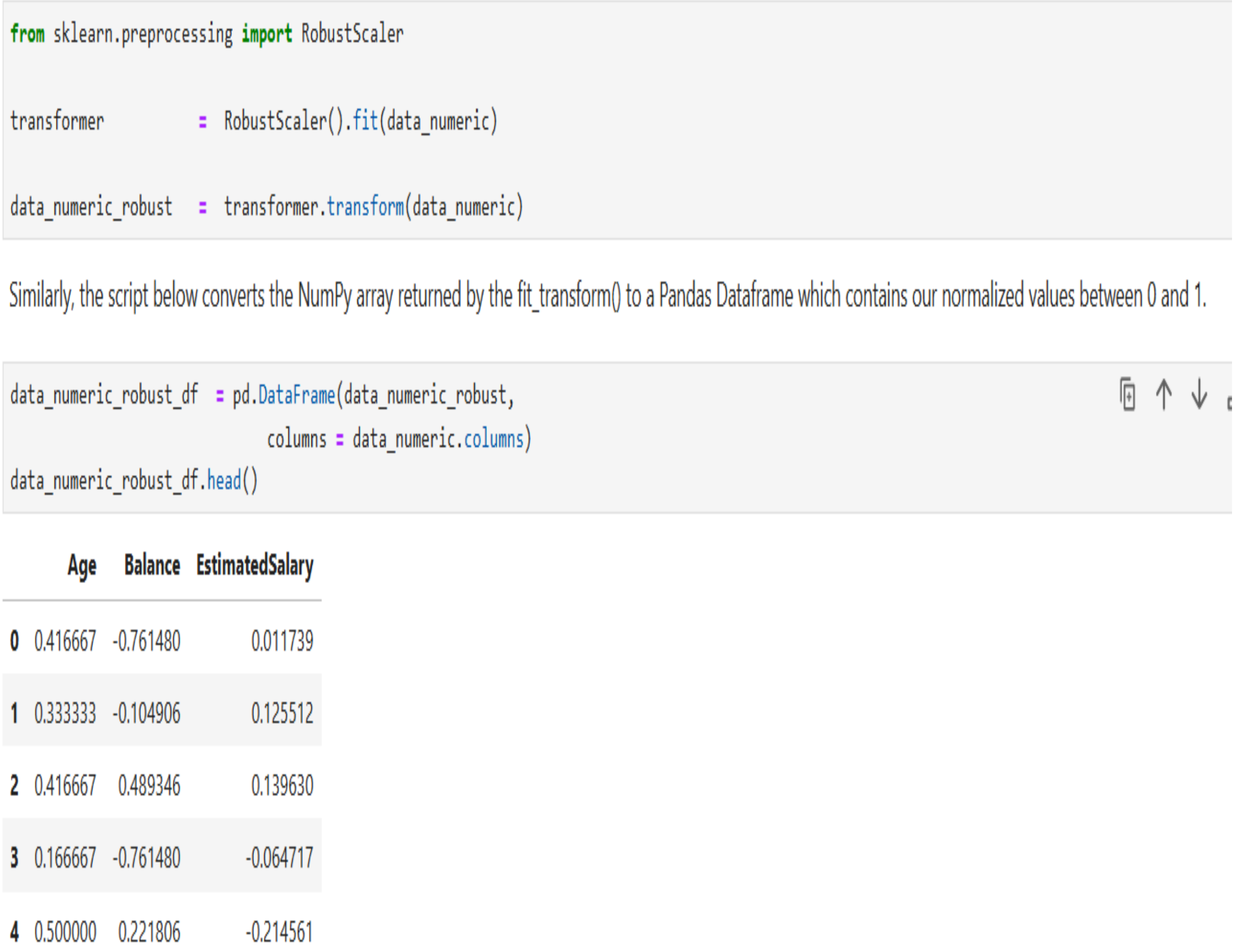
Goal: To scale features using statistics that are robust to outliers.

This Scaler removes the median and scales the data according to the quantile range (defaults to IQR: Interquartile Range). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile).

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Median and interquartile range are then stored to be used on later data using the transform method.

Standardization of a dataset is a common preprocessing for many machine learning estimators. Typically this is done by removing the mean and scaling to unit variance. However, outliers can often influence the sample mean / variance in a negative way. In such cases, using the median and the interquartile range often give better results. For an example visualization and comparison to other scalers, refer to Compare RobustScaler with other scalers. Robust scaling answers a simple question. How far is each data point from the input's median?

The fact that robust scaling uses median and IQR makes it resistant to outliers.



```
from sklearn.preprocessing import RobustScaler

transformer         = RobustScaler().fit(data_numeric)

data_numeric_robust  = transformer.transform(data_numeric)
```

Similarly, the script below converts the NumPy array returned by the fit_transform() to a Pandas Dataframe which contains our normalized values between 0 and 1.

```
data_numeric_robust_df  = pd.DataFrame(data_numeric_robust,
                                       columns = data_numeric.columns)
data_numeric_robust_df.head()
```

	Age	Balance	EstimatedSalary
0	0.416667	-0.761480	0.011739
1	0.333333	-0.104906	0.125512
2	0.416667	0.489346	0.139630
3	0.166667	-0.761480	-0.064717
4	0.500000	0.221806	-0.214561

3.11 Feature selection

Automatically select the number of features

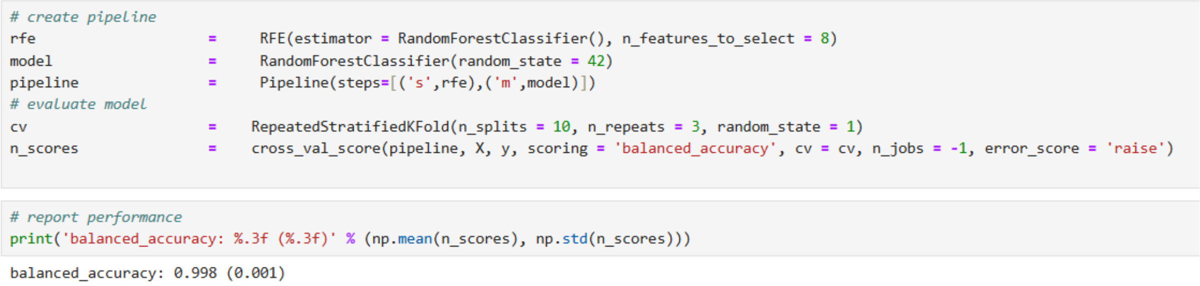
The RFE method is available via the RFE class in scikit-learn.

RFE is a transform. To use it, first the class is configured with the chosen algorithm specified via the “estimator” argument and the number of features to select via the “n_features_to_select” argument.

The algorithm must provide a way to calculate important scores, such as a decision tree. The algorithm used in RFE does not have to be the algorithm that is fit on the selected features; different algorithms can be used.

Once configured, the class must be fit on a training dataset to select the features by calling the fit() function. After the class is fit, the choice of input variables can be seen via the “support_” attribute that provides a True or False for each input variable.

Once configured, the class must be fit on a training dataset to select the features by calling the fit() function. After the class is fit, the choice of input variables can be seen via the “support_” attribute that provides a True or False for each input variable.



```
# create pipeline
rfe                  =    RFE(estimator = RandomForestClassifier(), n_features_to_select = 8)
model                =    RandomForestClassifier(random_state = 42)
pipeline             =    Pipeline(steps=[('s',rfe),('m',model)])
# evaluate model
cv                   =   RepeatedStratifiedKFold(n_splits = 10, n_repeats = 3, random_state = 1)
n_scores             =   cross_val_score(pipeline, X, y, scoring = 'balanced_accuracy', cv = cv, n_jobs = -1, error_score = 'raise')
```

```
# report performance
print('balanced_accuracy: %.3f (%.3f)' % (np.mean(n_scores), np.std(n_scores)))
```

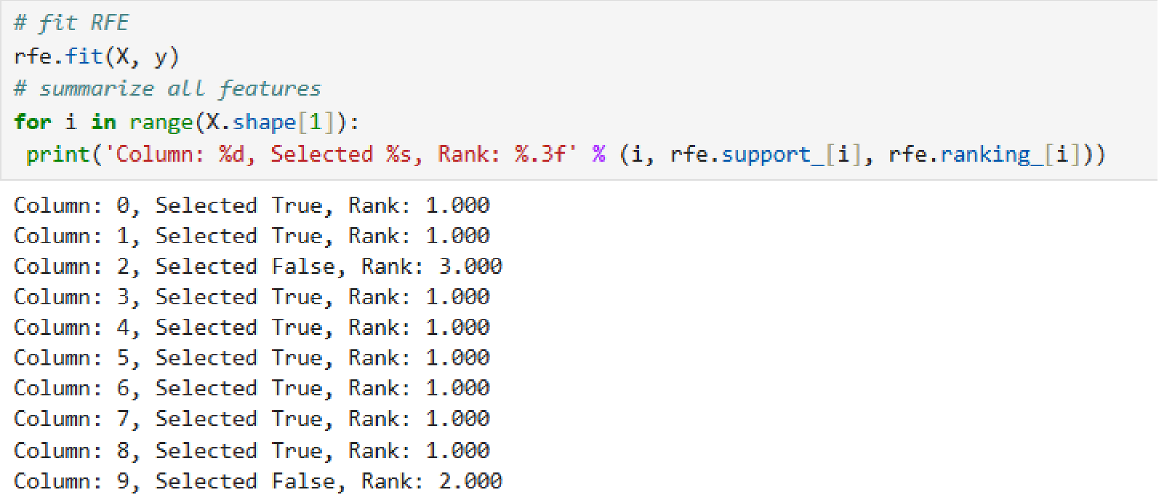
```
balanced_accuracy: 0.998 (0.001)
```

We observe that the RFE that uses a Random Forest and selects 8 features and then fits a model on the selected features achieves a balanced accuracy of about 99.8 %.

Balanced accuracy in binary and multiclass classification problems to deal with imbalanced datasets. It is defined as the average of recall obtained on each class.

Fit an RFE model on the whole dataset and selects five features, then reports each feature column index (0 to 9), whether it was selected or not (True or False), and the relative feature ranking.

The “support_” attribute reports true or false as to which features in order of column index were included and the “ranking_” attribute reports the relative ranking of features in the same order.



```
# fit RFE
rfe.fit(X, y)
# summarize all features
for i in range(X.shape[1]):
 print('Column: %d, Selected %s, Rank: %.3f' % (i, rfe.support_[i], rfe.ranking_[i]))
```

```
Column: 0, Selected True, Rank: 1.000
Column: 1, Selected True, Rank: 1.000
Column: 2, Selected False, Rank: 3.000
Column: 3, Selected True, Rank: 1.000
Column: 4, Selected True, Rank: 1.000
Column: 5, Selected True, Rank: 1.000
Column: 6, Selected True, Rank: 1.000
Column: 7, Selected True, Rank: 1.000
Column: 8, Selected True, Rank: 1.000
Column: 9, Selected False, Rank: 2.000
```

Selected 8 important features to predict the target variable:
['Tenure', 'Balance', 'IsActiveMember', 'EstimatedSalary', 'Complain', 'Satisfaction Score', 'Geography_code', 'Gender_code']

We shall use these features in our model building.

3.12 Dimensionality reduction

Since we have selected top 8 variables affecting the dependent variable, our dataset is Not Huge. **We are not going to apply dimensionality reduction such as Principal Component Analysis or Factor Analysis etc**. in our project.

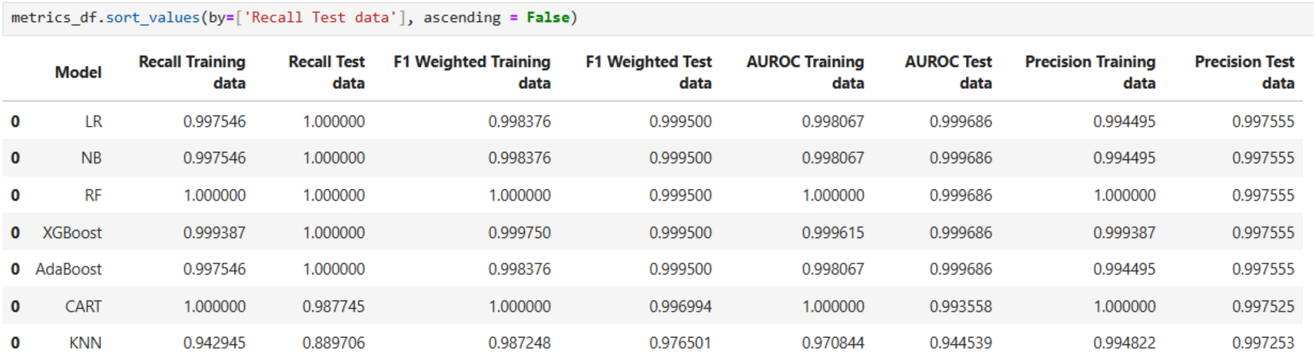
We use PCA when you have high-dimensional data to reduce its dimensionality while preserving most of the variance, simplifying analysis and visualization.

4. Model evaluation

We shall build several models such as Logistic Regression, Decision Tree (CART), KNN, Naïve Bayes, Random Forest, XGBoost and AdaBoost using both dataset without applying SMOTE and dataset after applying SMOTE.

We split the dataset into training and test datasets in the ratio of 80:20 using Stratified random sampling. Various measures of model performance using both training and test datasets are given below:

Before applying SMOTE:



```
metrics_df.sort_values(by=['Recall Test data'], ascending = False)
```

	Model	Recall Training data	Recall Test data	F1 Weighted Training data	F1 Weighted Test data	AUROC Training data	AUROC Test data	Precision Training data	Precision Test data
0	LR	0.997546	1.000000	0.998376	0.999500	0.998067	0.999686	0.994495	0.997555
0	NB	0.997546	1.000000	0.998376	0.999500	0.998067	0.999686	0.994495	0.997555
0	RF	1.000000	1.000000	1.000000	0.999500	1.000000	0.999686	1.000000	0.997555
0	XGBoost	0.999387	1.000000	0.999750	0.999500	0.999615	0.999686	0.999387	0.997555
0	AdaBoost	0.997546	1.000000	0.998376	0.999500	0.998067	0.999686	0.994495	0.997555
0	CART	1.000000	0.987745	1.000000	0.996994	1.000000	0.993558	1.000000	0.997525
0	KNN	0.942945	0.889706	0.987248	0.976501	0.970844	0.944539	0.994822	0.997253

Observations:

We observe the performance of all the models are good and there is no model overfit.
Performance measures of the model, Logistic Regression are given below:
Recall: 99.75% for Training data, 100% for test data
F1-Weighted: 99.84% for Training data, 99.95% for test data
AuROC: 99.81% for Training data, 99.97% for test data
Precision: 99.45% for Training data, 99.76% for test data

Let us check the model performance improves after applying SMOTE.

Synthetic Minority Oversampling TEchnique (SMOTE)

A problem with imbalanced classification is that there are too few examples of the minority class for a model to effectively learn the decision boundary.

One way to solve this problem is to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model.
An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective.
Perhaps the most widely used approach to synthesizing new examples is called the Synthetic Minority Oversampling TEchnique, or SMOTE for short. This technique was described by Nitesh Chawla, et al. in their 2002 paper named for the technique titled "SMOTE: Synthetic Minority Over-sampling Technique."

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

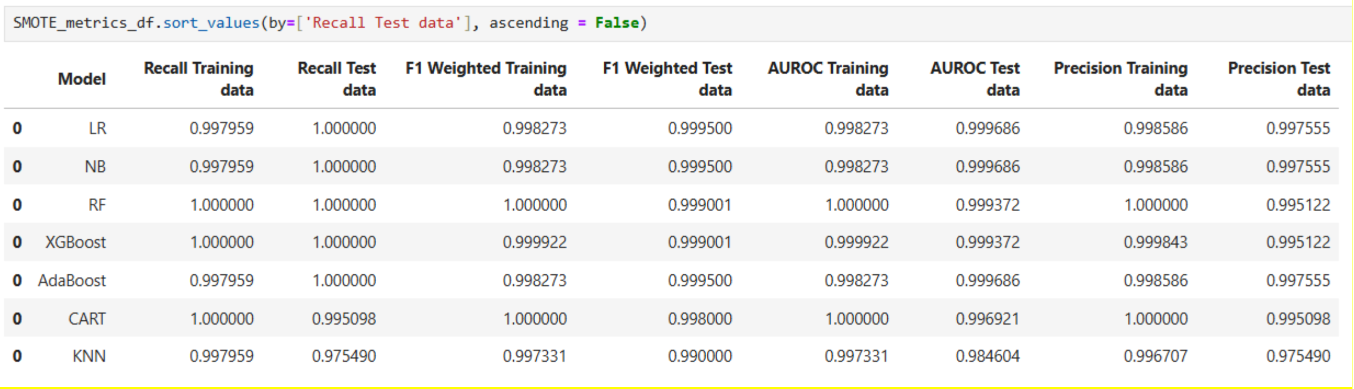
Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space.

Undersampling is a technique to balance uneven datasets by keeping all of the data in the minority class and decreasing the size of the majority class. It is one of several techniques data scientists can use to extract more accurate information from originally imbalanced datasets.

Undersampling can result in the loss of relevant information by removing valuable and significant patterns. Undersampling is appropriate when there is plenty of data for an accurate analysis. The data scientist uses all the rare events but reduces the number of abundant events to create two equally sized classes.

We have very less observations for the minority class and very high observations for the majority class. So, it was decided to go for Oversampling method.

After applying SMOTE:



```
SMOTE_metrics_df.sort_values(by=['Recall Test data'], ascending = False)
```

	Model	Recall Training data	Recall Test data	F1 Weighted Training data	F1 Weighted Test data	AUROC Training data	AUROC Test data	Precision Training data	Precision Test data
0	LR	0.997959	1.000000	0.998273	0.999500	0.998273	0.999686	0.998586	0.997555
0	NB	0.997959	1.000000	0.998273	0.999500	0.998273	0.999686	0.998586	0.997555
0	RF	1.000000	1.000000	1.000000	0.999001	1.000000	0.999372	1.000000	0.995122
0	XGBoost	1.000000	1.000000	0.999922	0.999001	0.999922	0.999372	0.999843	0.995122
0	AdaBoost	0.997959	1.000000	0.998273	0.999500	0.998273	0.999686	0.998586	0.997555
0	CART	1.000000	0.995098	1.000000	0.998000	1.000000	0.996921	1.000000	0.995098
0	KNN	0.997959	0.975490	0.997331	0.990000	0.997331	0.984604	0.996707	0.975490

Observations

Without applying SMOTE technique to treat the data imbalance, all the models are performing well with the recall values on both training and test datasets are 88% or above.

After applying SMOTE technique to treat the data imbalance, all the models are performing well with the recall values on both training and test datasets are 97% or above. There is no significant increase in the performance measures of the models after applying SMOTE.

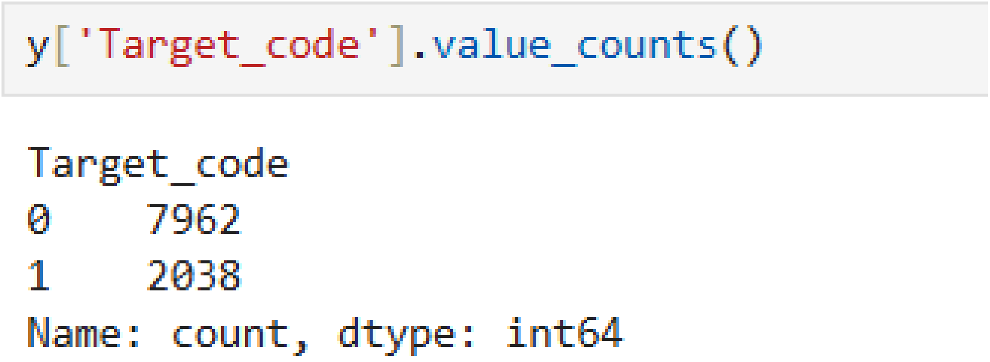
Final Model:

After evaluating various predictive models, the Logistic Regression has been selected as the final approach due to its superior performance and reliability and explainability. The Logistic Regression model offers a competitive alternative to benchmark models in customer churn prediction by effectively handling multicollinearity through regularization and offering straightforward interpretability. Its ability to provide clear probabilistic outputs allows businesses to make data-driven decisions with confidence. Moreover, its simplicity in model training and prediction enhances operational efficiency. By accurately identifying at-risk customers, Logistic Regression supports the development of more precise retention strategies, minimizing churn and increasing customer lifetime value.

Let us check if the assumptions of the Logistic Regression model are satisfied.

Assumption 1 - Binary logistic regression requires the target / dependent variable to be binary.

For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome (such as Success etc..).



```
y['Target_code'].value_counts()
```

```
Target_code
0    7962
1    2038
Name: count, dtype: int64
```

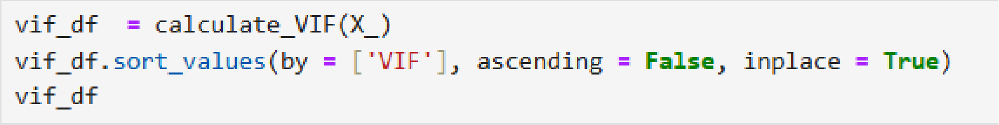
Observations:

The target variable is categorical having 0 (representing No Churn) and 1 (representing Churn) binary. This assumption is satisfied.

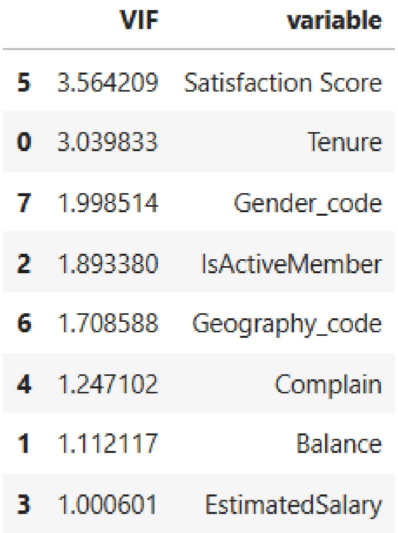
Assumption 2 - Only the meaningful variables should be included.

We have ensured that there are no unwanted variables selected for model building. This assumption is satisfied.

Assumption 3 -The predictor variables should not be correlated to each other meaning the model should have little or no multicollinearity.
We calculate Variance Inflation Factor (VIF) for the independent variables to check for presence or absence of multi-collinearity.



```
vif_df  = calculate_VIF(X_)
vif_df.sort_values(by = ['VIF'], ascending = False, inplace = True)
vif_df
```



	VIF	variable
5	3.564209	Satisfaction Score
0	3.039833	Tenure
7	1.998514	Gender_code
2	1.893380	IsActiveMember
6	1.708588	Geography_code
4	1.247102	Complain
1	1.112117	Balance
3	1.000601	EstimatedSalary

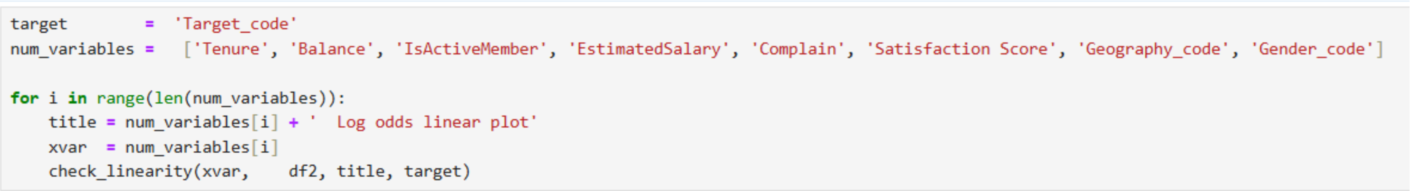
Observations:

All the variables are non-collinear as their VIF values are less than the threshold value of 5:
1. Satisfaction Score (3.564209)
2. Tenure(3.039833)
3. Gender_code(1.998514)
4. IsActiveMember(1.893380)
5. Geography_code(1.708588)
6. Complain(1.247102)
7. Balance(1.112117)
8. EstimatedSalary(1.000601)

This assumption is satisfied.

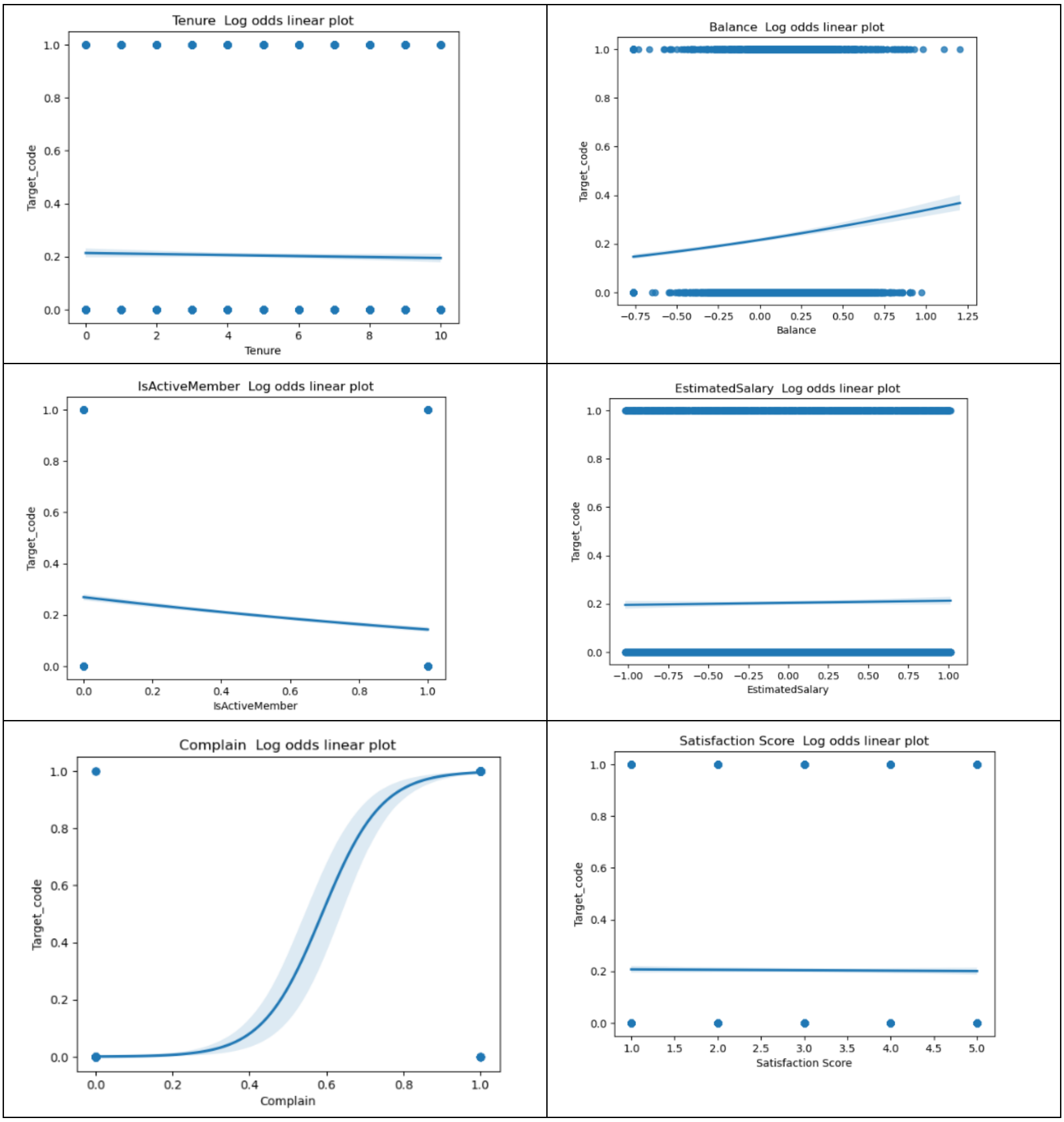
Assumption 4 - The independent variables are linearly related to the log odds.

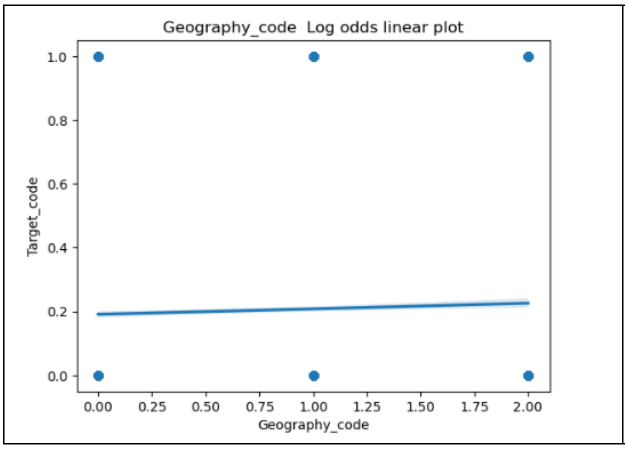
We need to check the assumption of Independent variables are linearly related to the log odds. One way to checking this is to plot the Independent variables in question and look for an S-shaped curve.

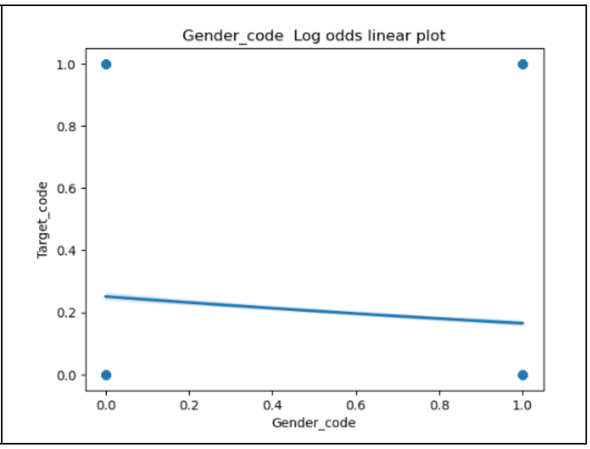


```
target        =  'Target_code'
num_variables =   ['Tenure', 'Balance', 'IsActiveMember', 'EstimatedSalary', 'Complain', 'Satisfaction Score', 'Geography_code', 'Gender_code']

for i in range(len(num_variables)):
    title = num_variables[i] + '  Log odds linear plot'
    xvar  = num_variables[i]
    check_linearity(xvar,    df2, title, target)
```

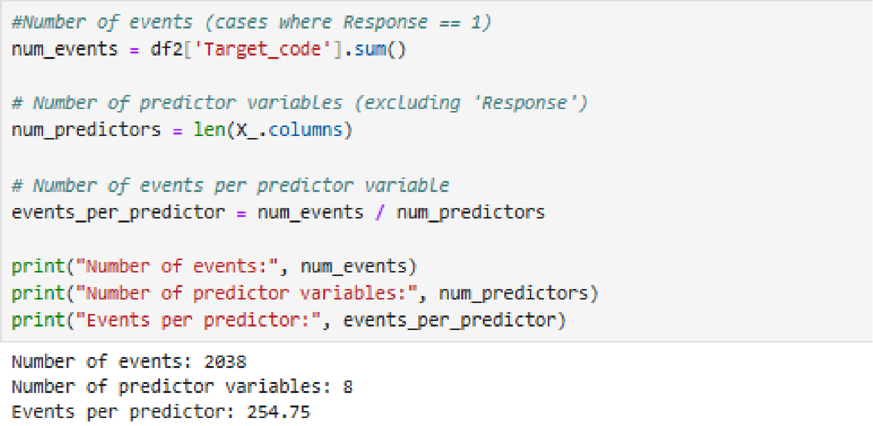






This assumption is satisfied.

Assumption 5 - Logistic regression requires quite a large number of observations.



```
#Number of events (cases where Response == 1)
num_events = df2['Target_code'].sum()

# Number of predictor variables (excluding 'Response')
num_predictors = len(X_.columns)

# Number of events per predictor variable
events_per_predictor = num_events / num_predictors

print("Number of events:", num_events)
print("Number of predictor variables:", num_predictors)
print("Events per predictor:", events_per_predictor)
```

```
Number of events: 2038
Number of predictor variables: 8
Events per predictor: 254.75
```

We calculate the number of events by summing the 'Target_code' column, which represents the cases where the outcome of interest occurs.

We calculate the number of predictor variables by counting the number of columns in the DataFrame and excluding the outcome variable.

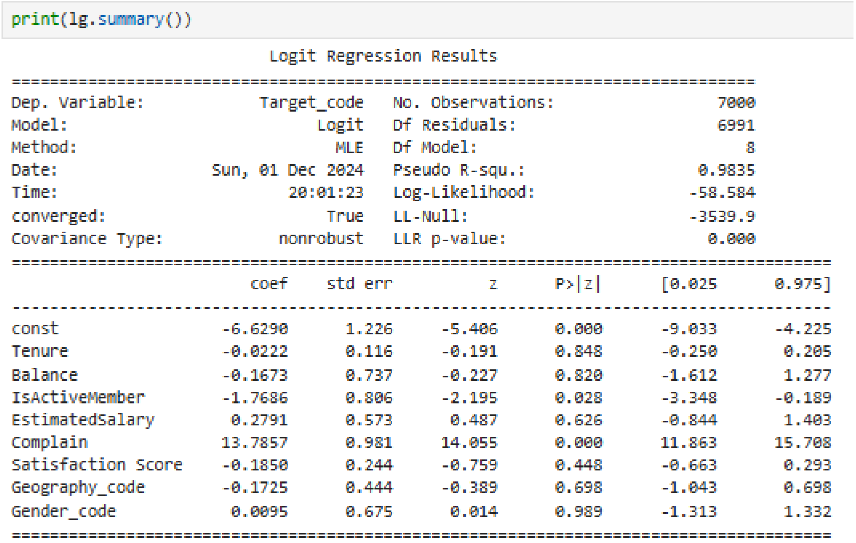
We divide the number of events by the number of predictor variables to get the events per predictor. We can then compare the calculated events per predictor with the recommended guideline of 10-20. If the ratio is below this guideline, it may indicate a potential violation of the assumption of a sufficiently large sample size.

With 2038 events and 8 predictor variables, the calculated number of events per predictor is approximately 254.75. This exceeds the commonly recommended guideline of having at least 10-20 events per predictor variable.

Inference: The dataset appears to meet the assumption of having a sufficiently large sample size for logistic regression.

Having a high number of events per predictor variable suggests that there should be adequate statistical power and precision in estimating the model parameters, enhancing the reliability of the logistic regression analysis. Therefore, the dataset likely provides a robust basis for fitting a logistic regression model and conducting statistical inference. This assumption is satisfied.

McFadden's R-squared is a measure of goodness-of-fit for statistical models, particularly in the context of logistic regression.



```
print(lg.summary())
```

```
                           Logit Regression Results                           
==============================================================================
Dep. Variable:            Target_code   No. Observations:                 7000
Model:                          Logit   Df Residuals:                     6991
Method:                           MLE   Df Model:                            8
Date:                Sun, 01 Dec 2024   Pseudo R-squ.:                  0.9835
Time:                        20:01:23   Log-Likelihood:                -58.584
converged:                       True   LL-Null:                       -3539.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
======================================================================================
                         coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------
const                 -6.6290      1.226     -5.406      0.000      -9.033      -4.225
Tenure                -0.0222      0.116     -0.191      0.848      -0.250       0.205
Balance               -0.1673      0.737     -0.227      0.820      -1.612       1.277
IsActiveMember        -1.7686      0.806     -2.195      0.028      -3.348      -0.189
EstimatedSalary        0.2791      0.573      0.487      0.626      -0.844       1.403
Complain              13.7857      0.981     14.055      0.000      11.863      15.708
Satisfaction Score    -0.1850      0.244     -0.759      0.448      -0.663       0.293
Geography_code        -0.1725      0.444     -0.389      0.698      -1.043       0.698
Gender_code            0.0095      0.675      0.014      0.989      -1.313       1.332
======================================================================================
```

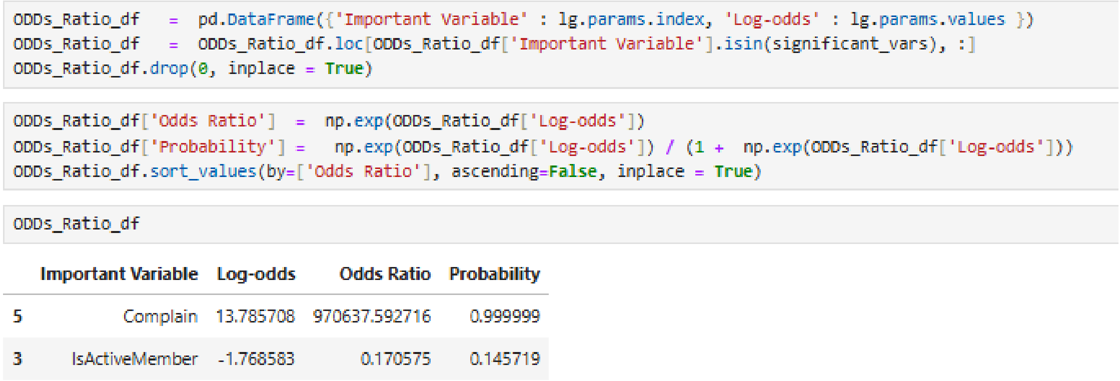
We observe that the McFadden R square (Pseudo R square) is 98 % and the model fitness is very good. This McFadden approach is one minus the ratio of two log likelihoods. The numerator is the log likelihood of the logit model selected and the denominator is the log likelihood if the model just had an intercept.

A goodness of fit using McFadden‟s pseudo r square (ρ^2) is used for fitting the overall model. McFadden suggested ρ^2 values of between 0.2 and 0.4 should be taken to represent a very good fit of the model (Louviere et al.,2000).
Ref: http://www.lifesciencesite.com/lsj/life1002/286_B01288life1002_2028_2036.pdf

Significant independent variables

From the above summary of Logistic Regression results, we observe that the variables, IsActiveMember and Complain are significant at 5% level of significance.

Calculation of Odd’s ratio and probability



```
ODDs_Ratio_df   =  pd.DataFrame({'Important Variable' : lg.params.index, 'Log-odds' : lg.params.values })
ODDs_Ratio_df   =  ODDs_Ratio_df.loc[ODDs_Ratio_df['Important Variable'].isin(significant_vars), :]
ODDs_Ratio_df.drop(0, inplace = True)
```

```
ODDs_Ratio_df['Odds Ratio']  =  np.exp(ODDs_Ratio_df['Log-odds'])
ODDs_Ratio_df['Probability'] =   np.exp(ODDs_Ratio_df['Log-odds']) / (1 +  np.exp(ODDs_Ratio_df['Log-odds']))
ODDs_Ratio_df.sort_values(by=['Odds Ratio'], ascending=False, inplace = True)
```

```
ODDs_Ratio_df
```

	Important Variable	Log-odds	Odds Ratio	Probability
5	Complain	13.785708	970637.592716	0.999999
3	IsActiveMember	-1.768583	0.170575	0.145719

We observe that the probability of customers who churn is 99.99 % if they have complaints.

5. COMPARISON WITH BENCHMARK

At the outset of the project, the benchmark was set based on the performance of the base model and other models (before SMOTE), which served as a reference point for evaluating subsequent models. The key metrics for this benchmark model were:

We had chosen Logistic Regression as our base model and the recall after K-Fold cross validation is 99.80%. Now our best model is the same Logistic Regression with the recall for training data is 100% and test data is 99.51%.

6. Visualizations

All relevant visualizations that support the ideas/insights that gleaned from the data are covered in the section 3.4.2 of this report.

7. Implications

The solution developed for predicting customer churn in a European multinational bank operating in France, Germany, and Spain leverages Logistic Regression, a well-established and interpretable machine learning model. Logistic Regression excels in binary classification tasks like churn prediction by assigning probabilities to each customer's likelihood of churning, making it easier to identify at-risk customers and implement proactive retention strategies. This enables the bank to optimize customer engagement and improve long-term profitability.

Accurate churn predictions empower banks to tailor interventions based on customer behavior and risk profiles. For instance, personalized offers, loyalty rewards, or targeted communication can be designed for high-risk customers. Such strategies reduce churn, foster stronger customer relationships, and increase customer lifetime value. Additionally, understanding churn patterns helps segment customers more effectively, allowing the bank to focus resources on high-value segments and improve overall customer satisfaction.

While the primary goal is churn prediction, insights from Logistic Regression extend to broader business strategies. Identifying key factors such as complaints, age, and transaction frequency can guide product development, enhance customer support, and inform marketing campaigns. Furthermore, proactive churn management improves brand reputation, fostering trust and loyalty in competitive markets.

To further enhance the model, the bank could integrate additional features that influence churn, such as economic indicators, regional market trends, and customer sentiment from feedback channels. Expanding the dataset with external sources like social media trends and financial news can improve predictive accuracy. Given Logistic Regression's proven success in similar contexts and its ability to handle large, complex datasets when combined with regularization techniques, there is high confidence in these recommendations. Iterative improvements through continuous data collection and analysis will further refine the model's performance, supporting effective customer retention strategies.

8. Limitations

In predicting customer churn for a European multinational bank operating in France, Germany, and Spain, several limitations have been identified that highlight areas for improvement and future enhancement:

Model Performance: While the model shows promise, its performance metrics (e.g., recall, precision, or F1-score) may indicate room for improvement in accurately predicting churn across diverse customer segments and regions. This suggests the need for further refinement to ensure reliable predictions in real-world scenarios.

Data Quality and Completeness: The quality and completeness of the dataset significantly impact the model's accuracy. Ensuring rigorous data cleaning, validation, and preprocessing is crucial to enhancing model performance. Additionally, customer data from different regions may have varying levels of completeness due to different banking regulations and data collection practices.

Feature Selection and Engineering: The current feature set may not fully capture all factors influencing churn, such as customer satisfaction scores, interaction frequency, or external economic indicators. Expanding the feature set to include regional economic conditions, competitive offerings, and customer sentiment (e.g., from surveys or complaints) could improve the model's predictive power and relevance.

Model Complexity and Overfitting: Balancing model complexity is essential to prevent overfitting, where the model performs well on training data but poorly on unseen data. Conversely, a model that is too simplistic may fail to capture important patterns. Techniques such as cross-validation, regularization, and hyperparameter tuning can help optimize model complexity and generalization.

Dynamic and External Factors: As per the Logistic Regression model, probability of the two significant independent variables at 5% level of significance:

Complain: 99.99%

IsActiveMember: 45.719%

Hence, we need more relevant independent variables affecting customer churn in order to reduce it. The model may not fully account for dynamic factors like sudden economic shifts, regulatory changes, or market competition. These factors can significantly influence customer behavior and churn rates. Incorporating real-time data and external indicators, such as economic forecasts or competitor actions, can enhance the model's responsiveness to changing conditions.

Model Interpretability: Logistic Regression is inherently interpretable, making it easier for stakeholders to understand why certain customers are predicted to churn. Unlike more complex models, Logistic Regression provides clear, straightforward coefficients that indicate the direction and magnitude of each feature's impact on the likelihood of churn. This transparency helps build stakeholder trust and facilitates more informed decision-making.

Enhancements: To address these limitations, the following improvements are recommended:

- Enhance data quality through rigorous cleaning and validation processes.
- Expand feature engineering to include additional relevant variables and external data sources.
- Regularly update the model to incorporate dynamic factors and real-time data.
- Use interpretability techniques to improve stakeholder understanding of the model’s predictions.

9. Closing Reflections

From the Logistic Regression model, it is evident that complain – indicating customer complaints among all the independent variables affects the customer churn more. It is clear that ***Ignoring customer complaints often lead to increased dissatisfaction and customer churn, as customers have numerous alternatives and may turn to competitors after negative experiences***.
Ref: https://www.omniconvert.com/blog/how-customer-complaints-affect-customer-satisfaction/

Throughout the process of developing a model for customer churn prediction for a European multinational bank, several valuable insights were gained, alongside key lessons that would shape future approaches.

Data Quality is Crucial: The importance of clean, accurate, and comprehensive data became evident. Missing values, inconsistencies, or biases in the data can significantly affect model accuracy and lead to unreliable predictions. Effective data preprocessing and validation are critical to building a robust churn prediction model.

Feature Engineering is Key to Success: Identifying the right features and understanding their relationship with churn is paramount. Features like complaints, age, and transaction frequency were found to be essential, but additional features (such as customer satisfaction, service usage patterns, or sentiment from interactions) could further enhance model performance.

Model Complexity and Overfitting: While Logistic Regression is a simpler and more interpretable model compared to complex models like Random Forest, it is still susceptible to overfitting, especially when dealing with high-dimensional data or multicollinearity. Balancing model complexity and generalization is crucial for optimal performance. Techniques such as cross-validation and regularization (L1 for feature selection and L2 for coefficient shrinkage) help mitigate overfitting by controlling model complexity and ensuring that the model generalizes well to unseen data.

By fine-tuning hyperparameters like the regularization strength (C) and employing cross-validation, Logistic Regression can deliver robust predictions while maintaining interpretability and computational efficiency.

The Importance of Interpretability: In banking, model interpretability is crucial for gaining stakeholder trust. Being able to explain why certain customers are predicted to churn—and how key features influence this prediction—helps in making actionable business decisions. Using interpretability techniques, like SHAP values or feature importance charts, proved to be valuable.

Dynamic and External Factors Matter: Customer behaviour is influenced by many dynamic factors—such as economic conditions, competitive landscape, and regulatory changes—that may not be captured by static historical data. Incorporating external data sources and real-time information can significantly improve model responsiveness.

Invest More in Data Collection and Enrichment: A larger and more diverse dataset would improve model accuracy and robustness. In the future, I would aim to collect more granular data, especially from feedback channels (e.g., customer surveys or complaints), and explore external datasets like economic indicators or market trends to capture broader factors influencing churn.

Focus More on Customer Segmentation: Customer churn is not homogeneous across all segments. More effort would be placed on segmenting customers by demographics, usage patterns, or service types. Tailoring churn prediction models to different segments could lead to more targeted and effective retention strategies.

Experiment with Additional Models: While Logistic Regression performed well, I would experiment with other models like Gradient Boosting, XGBoost, or even deep learning techniques (e.g., neural networks for sequential data) to compare performance. Additionally, combining models in an ensemble method could improve predictions.

Incorporate Real-Time Data and External Factors: For a more dynamic and adaptable churn prediction model, I would integrate real-time data streams and external indicators (e.g., macroeconomic trends, news, and competitor activities). This would help capture sudden shifts in customer behavior and adapt predictions accordingly.

Enhance Interpretability: Greater focus would be placed on improving model transparency, not just for internal stakeholders but also for customers when applicable. Providing clear, understandable reasons for why customers might churn can help in building customer trust and loyalty.

In summary, while the model has demonstrated a good recall score, highlighting its effectiveness in identifying churners, the process has provided valuable lessons in data cleaning, feature engineering, and model complexity. Future efforts should focus on further enhancing precision to balance the recall and avoid excessive false positives. Additionally, expanding the feature set and integrating external factors like economic conditions or customer sentiment could improve the model's overall accuracy. A systematic and iterative approach, with regular model updates and refinements, will help maintain its effectiveness in accurately predicting churn and optimizing retention strategies.

APPENDIX:

DATA DICTIONARY:

FIELDS	DESCRIPTION
RowNumber	Corresponds to the record (row) number
CustomerId	contains random values
Surname	the surname of a customer
CreditScore	Credit Score of a Customer
Geography	Customer's Location
Gender	Gender of a Customer
Age	Age of a Customer
Tenure	the number of years that the customer has been a client of the bank
Balance	Balance of a Customer
NumOfProducts	the number of products that a customer has purchased through the bank
HasCrCard	denotes whether a customer has a credit card
IsActiveMember	denotes whether a customer is Active
EstimatedSalary	Estimated Salary of a Customer
Exited	whether or not the customer left the bank. (Target variable)
Complain	customer has complaint or not
Satisfaction Score	Score provided by the customer for their complaint resolution.
Card Type	type of card holds by the customer
Points Earned	the points earned by the customer for using credit card