UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEMS

FINAL PROJECT REPORT

DATA ANALYTICS WITH R/PYTHON COURSE

Topic:

APPLY A MACHINE LEARNING MODEL TO PREDICT CUSTOMER CHURN IN BANKING SERVICES

Lecturer: **Nguyen Phat Dat**

Assistant: **Tran Le Tan Thinh**

Group: 2

Ho Chi Minh, April 07, 2023

Member of Group 2

No.	Name	Student ID
1	Nguyen Thị Bao Ha	K204061394
2	Cu Thi My Duy	K204061392
3	Nguyen Thi Cam Giang	K204060282
4	Nguyen My Dung	K204061389
5	Nguyen Thi My Dung	K204061390

Acknowledgments

We are grateful for the invaluable knowledge and expertise that we have gained from the Data Analysis with R/Python course lecturers, which were instrumental to the successful completion of our project. We would like to express our sincere appreciation to the lecturers for their guidance and unwavering support throughout the course.

We would also like to extend our heartfelt gratitude to Mr. Nguyen Phat Dat, whose assistance was crucial in helping us acquire the necessary skills and knowledge, as well as providing invaluable suggestions and solutions to ensure the success of our project.

Furthermore, we would like to express our profound appreciation to Mr. Tran Le Tan Thinh, who provided additional support to Mr. Dat during his teaching process. His input on specific aspects of the material and insights on the business were extremely valuable to us.

Despite our best efforts, the limited specialized knowledge and experience of our group made it challenging to address flaws in the problem's presentation, implementation, and evaluation. Therefore, we look forward to receiving feedback and evaluation from the subject lecturers to enhance our understanding and improve the quality of our work.

Commitment

During the development of this project, Group 2 confirms that the final output of the Data Analysis with R/Python project was produced by all members of the group under the guidance of Mr. Nguyen Phat Dat. The information, data, and results presented in the project are authentic and not copied from any external sources. All references cited in the project materials for developing the theoretical framework, including texts, articles, and books, have been appropriately acknowledged.

The group takes full responsibility for any errors that may arise and accepts any penalties imposed as a result.

Ho Chi Minh City, April 07, 2023

Contents

List of Figures

CHAPTER 1: PROJECT OVERVIEW

1.1 Introduction

According to a study by Harvard Business Review, companies that lead in customer loyalty grow their revenue 2.5 times faster than their peers. Especially, in the current context, businesses are facing a slumping economy, and the problem of customer retention is a huge challenge. Thus, how to make existing customers love and stick with it is the key to improving the bank's competitiveness.

Along with the wave of digital transformation, banks are constantly transforming and reinventing themselves by changing business thinking, restructuring organizations, forming business departments, Technology centers, and especially focusing on the core key that makes the difference: focus on improving the customer journey experience. The continuous development and increase in quantity and quality of commercial banks have made the market of financial and banking services more fierce than ever. The race to retain customers among commercial banks is also increasingly tense, each bank has introduced policies to attract customers. To be able to concretize those policies to customers who intend to stop using the service is very difficult. If banks just wait until customers have left the service, it is very difficult to keep them with their services. So it is necessary to predict in advance which customers will leave the service in the future. The development of data science in recent years has helped solve many different problems and this is the right choice for banks to solve the problem of leaving services.

1.2 Reason for choosing the topic

Research done by Frederick Rechained of Bain & Company (inventor of satisfaction metrics) shows that when customer retention increases by 5%, profits increase by 25% to 95%. The cost of acquiring a new customer is usually 5 to 25 times higher than maintaining an existing one. Businesses not only need to spend time and resources to find new customers outside but also need to take good care of existing customers and make them satisfied.

The development of the economy has helped customers have more choices, making the competition for customers more and more fierce. If customers feel unsatisfied with product quality, service quality, interest, or needs, their complaints are not resolved quickly, benefits are not guaranteed which can turn them away. business, choose another supplier. Customer decisions can affect the survival or loss, prosperity or decline of a business. Therefore, developing, caring for, maintaining, and retaining customers is always the top concern of every business.

Stemming from that fact, we choose the topic "Apply a machine learning model to predict customer churn in banking services" as the research objective to solve the above problem.

1.3 Project objectives

In this research, we aim to accomplish the following for this study:

We focus on identifying and visualizing which factors contribute to customer churn in the banking context. For this project, we Classify if a customer is going to churn or not, Then we use machine learning algorithms including (Regression logistics, ROC, Logistic regression, Classificant, Decision tree, and Support Vector Machine) to make predictions about the probability to churn.

From there, find valuable insights for businesses through customer evaluation factors and predictive machine learning. Besides, businesses will find out the unreasonable points in their services to improve or promote if they find it good, to come up with a smart, timely, and quick business strategy.

1.4 Project structure

For this project, we use a dataset of 10000 lines containing details of a bank's, Age, customers: CustomerId, Surname, CreditScore, Gender, and Tenure,.. to choose and come up with a machine learning perspective analysis model to help businesses save a lot of time to understand their current service quality situation. At the same time, it also helps to improve and develop Customer Relationship Management (CRM) strategies at the right time combined with reasonable marketing into each customer group to help businesses attract more potential customers to your bank.

1.5 Project structure

Chapter 1: Overview of the project, General introduction to the topic, Reason for choosing the topic, Object, and scope of the project.

Chapter 2: Introduction to algorithms: Regression logistics, ROC, Logistic regression, Classificant, Decision tree, and Support Vector Machine, and machine learning model evaluation methods.

Chapter 3: Data analysis and data preprocessing: Overview of data, finding outliers to eliminate, normalizing data to limit confounding values when entering the machine learning model.

Chapter 4: Testing and evaluating machine learning models to choose the most suitable algorithm.

Chapter 5: Conclusions and future solutions: Achievements, limitations, and proposed solutions.

1.6 Methodology and proposed model

1.6.1 Methodology

Google collab

1.6.2 Proposed model

The research model proposed in Figure 1.1, starts by identifying the research problems of the project, then proceeds to collect data. The data set will be preprocessed, removing outliers, and normalizing the data to limit the confounding values when included in the machine learning model. Then, use the Correlation Matrix to find out the strengths and weaknesses between the observed variables and select the important variables needed to be included in the model. Then the data set is divided into 2 parts: the training dataset and the testing dataset. The training data set is used to establish the machine learning methods and the test dataset is used to evaluate the machine learning method, through two methods of model evaluation (Confusion Matrix and Lineage). curve ROC_AUC). From there, select the model that best fits the collected data set. Finally, after a model is attached to the data set, data is visualized on customer churn and suggested solutions.

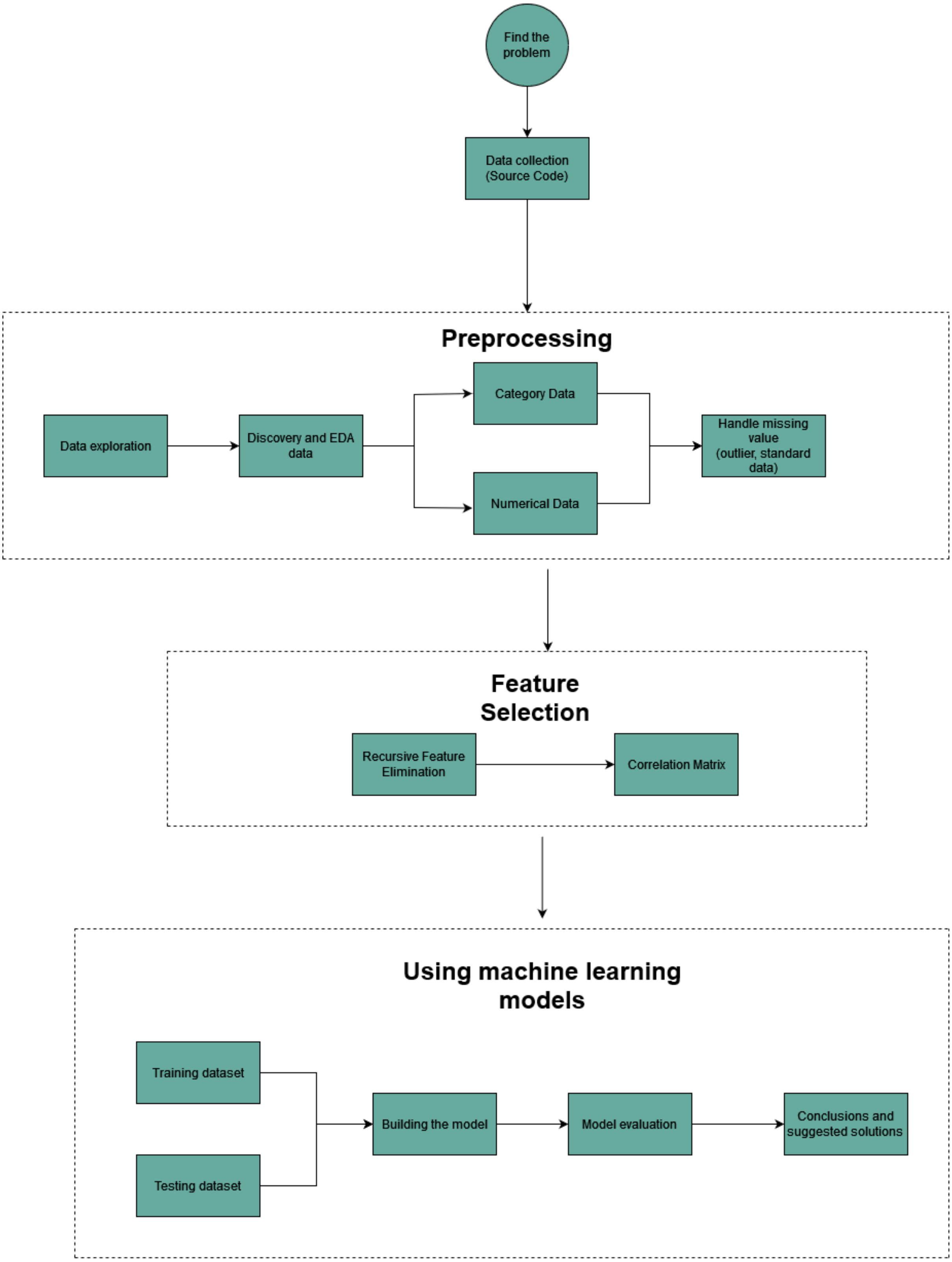


Figure 1: Model proposed(source: author)

CHAPTER 2: THEORETICAL BASIS

The foundations of machine learning models, a review of customer satisfaction theories, and some methods for evaluating model accuracy will all be covered in this chapter. From there, it provides a theoretical framework for the development and evaluation of machine learning models.

2.1 Support Vector Machine

Support Vector Machine is one of the most popular Supervised Learning algorithms, used for classification as well as regression problems. However, mainly, it is used for Classification problems in Machine Learning.

The Support Vector Machine (SVM) is a classification model that works by constructing a (n - 1) dimensional hyperplane in the n-dimensional data space such that the hyperplane classifies classes in a consistent manner. most optimal.

The task of the SVM algorithm is to determine what type the new data point is. This makes the SVM a kind of non-binary linear classifier.

The goal of the SVM algorithm is to generate the best decision line or boundary that can decompose the n-dimensional space into classes so that we can easily feature new data into the correct category in the future. This best decision boundary is called the hyperplane.

SVM algorithm can be used for Face Detection, image classification, text classification, etc.

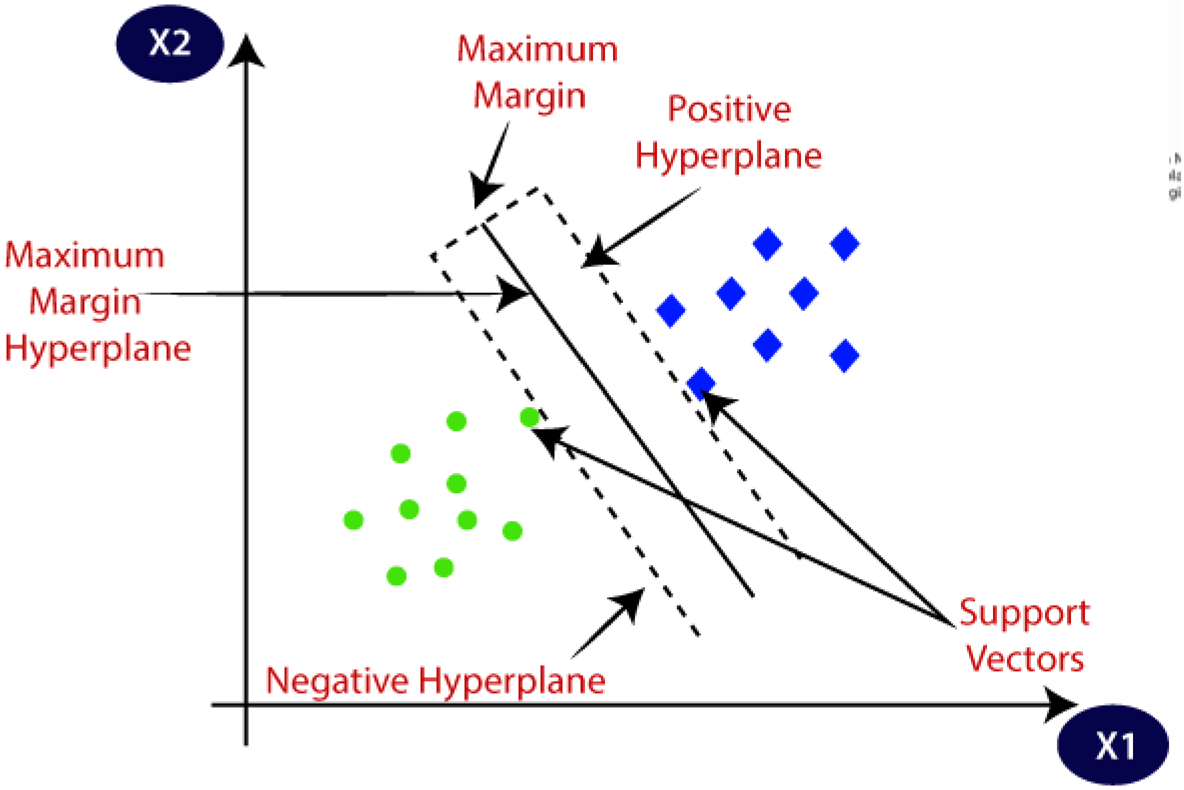


Figure 2: Support Vector Machine(Javatpoint)

2.2 Decision Tree

Decision Tree algorithms belong to the family of supervised learning algorithms. The decision tree algorithm can be used for regression and classification, but it is mainly preferred to solve classification problems. The goal of using Decision Trees is to create a training model that can be used to predict the class or value of a target variable by learning simple decision rules inferred from previous data. (training data).

It has a hierarchical, tree-like structure, consisting of a root node, branches, internal nodes, and leaf nodes, where the inner nodes represent properties. functionality of the data set, the branches represent the decision rules and each leaf node represents the outcome.

In Decision tree, there are two nodes, namely Decision Node and Leaf Node. Decision nodes are used to make any decision and have many branches, while Leaf node is the output of those decisions and does not contain any other branches. Decisions or checks are made on the basis of the features of the given data set. To build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. The decision tree only needs to ask one question and rely on it. on the answer (Yes/No), it further divides the tree into subtrees.

Decision trees are based on the type of target variable we have. It has two types:

1. Decision tree by categorical variable: Decision tree has categorical variable as target variable.

2. Decision tree of continuous variable: A decision tree with continuous variable as the target variable.

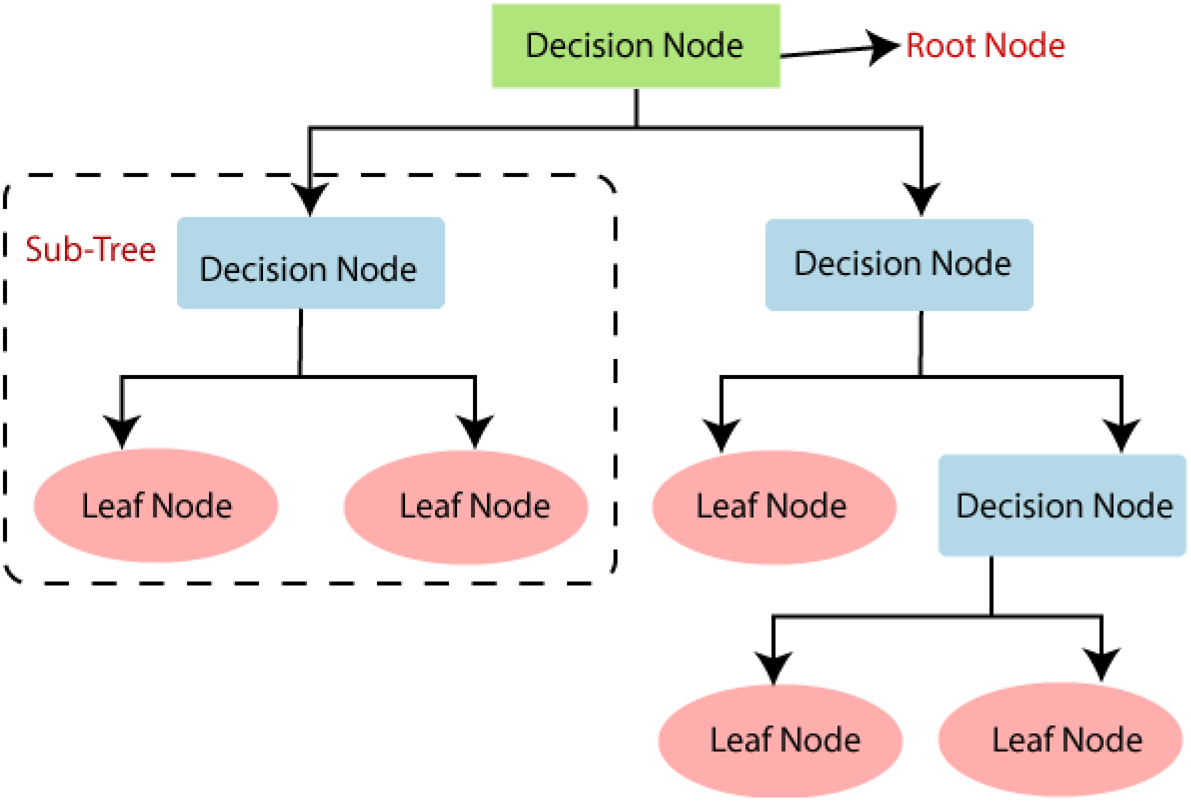


Figure 3: Decision tree(Javatpoint)

2.3 Logistic regression

Logistic regression (Edgar, T., & Manz, D. (2017) is another powerful supervised machine learning method used for binary classification problems (when the objective is classification). The best way to think of logistic regression is that it is a linear regression but intended for classification problems. Logistic regression basically uses a logistic function defined below to model the binary output variable. (Tolles & Meurer, 2016). The basic difference between linear regression and logistic regression is that the range of logistic regression is limited from 0 to 1. Also, in contrast to linear regression, regression is logistic does not require a linear relationship between input and output variables This is due to the application of a nonlinear log transform to the odds ratio (to be determined shortly).

Logistic regression uses a logistic function called a sigmoid function to map predictions and their probabilities. The sigmoid function refers to an S-shaped curve that converts any real value to a range between 0 and 1.

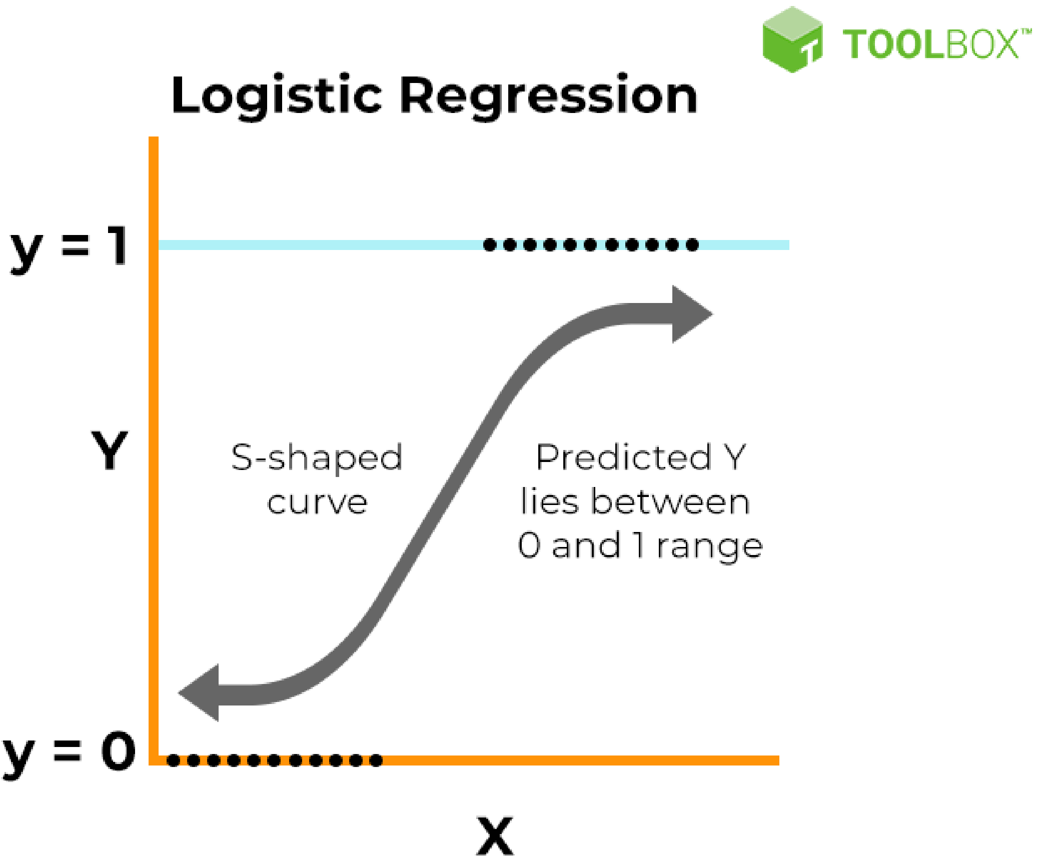


Figure 4:Logistic Regression – Sigmoid Function (source: toolbox)

The sigmoid function is defined as:

(t) = etet+1=11+e-t (1)

In the linear regression model, we have modeled the relationship between outcome and features with a linear equation:

t=0+1 x (2)

For classification, we prefer probabilities between 0 and 1, so we wrap the right side of the equation into the logistic function. This forces the output to assume only values between 0 and 1

p(x)=11+e-(0+1x) (3)

Now, when the logistic regression model comes across an outlier, it will take care of it. This transform ensures that the p stays between 0 and 1

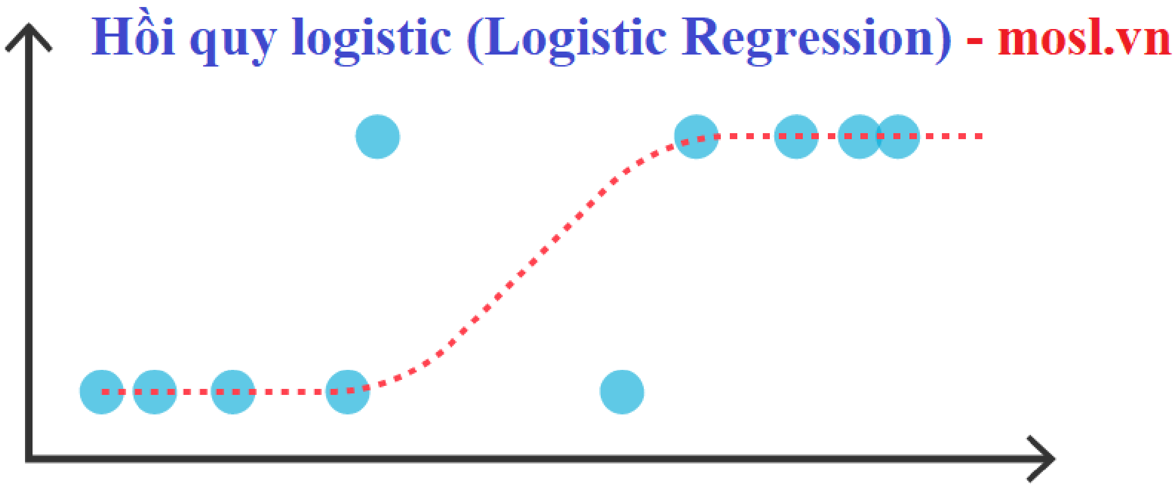


Figure 5: Logistic regression fit outliers (source: Andrew Ng provided image)

To get the exponential expression out of the denominator, we consider odds instead of probabilities. Odds, familiar to bettors everywhere, are the ratio of “successes” (1) to “nonsuccesses” (0). In terms of probabilities, odds are the probability of an event divided by the probability that the event will not occur.

Odds(Y=1) = p1-p (4)

We can obtain the probability from the odds using the inverse odds function:

p=Odds1 + Odds (5)

We combine this with the logistic response function, to get:

Odds(Y=1) = e0+1x (6)

Finally, taking the logarithm of both sides, we get an expression that involves a linear function of the predictors:

Log (Odds (Y = 1)) = 0+1x (7)

The log-odds function, also known as the logit function, maps the probability p from (0;1) to any value $(-\infty, +\infty)$. The inverse of the logit function is called the logistic function or Sigmoid function.

2.4 Classificant

GradientBoostingClassifier is a widely used machine learning algorithm in classification problems. This algorithm is based on different steps to create a better classification model.

2.4.1 Introduction

GradientBoostingClassifier is a fairly powerful algorithm and is widely used in classification problems. This algorithm is built based on the model optimization steps by enhancing decision trees. Each tree is built upon Gradient Boosting (GB) operations to improve the predictive model.

2.4.2 GradientBoostingClassifier Steps

GradientBoostingClassifier includes the following steps:

- *Model Initialization:* First, the model is initialized by calculating the mean of y. For the binary classification problem, the mean of y is calculated as the ratio of the number of labels 1 to the total number of samples.
- *Determine the loss function:* Then, the loss function is used to determine the current prediction error. The loss function commonly used in the GradientBoostingClassifier is the cross-entropy loss function.
- *Construct a decision tree:* A new decision tree is built based on the data features and the loss value calculated in the previous step. The decision tree is built by looking for the direction that minimizes the loss value.
- *Determine the coefficient of the decision tree:* After the new decision tree is built, the coefficient is calculated to determine the influence of that decision tree on the model, it can help in the calculation of the predictions. .
- *Weight Update:* Update the weights of the samples based on the distance between the prediction and the actual value. The samples that the model predicts wrong will have an increased weight, whereas the samples that correctly predict will have a decrease in weight.
- *End of model search:* The pattern search process is repeated many times until the loss value does not change much.

2.4.3 Advantages and disadvantages of GradientBoostingClassifier

The advantages of GradientBoostingClassifier is:

- *High Accuracy:* Gradient Boosting Classifier is one of the most accurate machine learning methods available today. It is very suitable for solving complex problems with big data.
- *Ability to find non-linear relationships:* Gradient Boosting Classifier is capable of finding complex relationships between input and output variables. This makes it more efficient than linear models.
- *Ability to work with many types of data:* Gradient Boosting Classifier can work with many different types of data such as qualitative data, quantitative data, continuous data, category data, sparse data.
- *Generality:* The Gradient Boosting Classifier does not require many assumptions about the data distribution, and is therefore highly general.

The disadvantages of Gradient Boosting Classifier:

- *Vulnerable to overfitting:* If not optimally tuned, the Gradient Boosting Classifier can be prone to overfitting, to the point of leading to disparities between the training data and the test data.
- *Time consuming to train:* Gradient Boosting Classifier can be time consuming to train, especially with large data sets.
- *Unstable:* Gradient Boosting Classifier is unstable when the data changes, especially when there are anomalies in the data.

2.5 XGBoost

XGBoost (Extreme Gradient Boosting) is a popular open-source software library used for supervised machine learning tasks, particularly for classification and regression problems. It uses a gradient boosting algorithm that combines multiple weak prediction models to create a strong model. XGBoost has been widely used in various fields such as finance, healthcare, and advertising due to its high accuracy, scalability, and efficiency.

Steps to apply XGBoost:

To apply XGBoost to a problem, you can follow the following steps:

1. Prepare data: collect and prepare data for the XGBoost model, making sure that the data is complete and has no missing or invalid values.
2. Choose hyperparameters: XGBoost has many different hyperparameters, such as the number of decision trees, tree depth, learning rate, and subsample rate. You need to choose appropriate hyperparameters for your problem to achieve optimal results.
3. Train the model: using the prepared data and selected hyperparameters, train an XGBoost model. During this process, the model will learn from the training data and build decision trees to make predictions.
4. Evaluate the model: test the accuracy of the model on a test dataset. If the result is not satisfactory, you can adjust the hyperparameters or find ways to improve the input data.
5. Use the model: once the model has been confirmed, you can use it to make predictions on new data.

In practice, these steps are often repeated multiple times to find optimal hyperparameters and models for your problem.

Advantages of XGBoost:

- High Accuracy: XGBoost is known for its high accuracy in classification and prediction tasks.

- Good Performance: XGBoost can handle large datasets and train faster than other machine learning algorithms.
- Ability to Address Overfitting Issues: XGBoost has features such as regularization and early stopping to address overfitting issues.
- Can Solve Regression and Classification Problems: XGBoost can be used for both predicting numerical values (regression) and classifying data (classification).
- Flexible Hyperparameter Tuning: XGBoost allows users to fine-tune hyperparameters to optimize the model.
- Cross-platform Compatibility: XGBoost can be deployed on various platforms, including Python, R, and Java.

Disadvantages of XGBoost:

- Dependence on selecting appropriate hyperparameters: If appropriate hyperparameters are not selected, the XGBoost model may not achieve optimal performance.
- Sensitive to noisy data: If there is noise or outliers in the data, the accuracy of the XGBoost model may be affected.
- Difficult to Understand: Due to the complexity of the XGBoost algorithm, interpreting and understanding the model's predictions can be challenging for inexperienced users.

Import the XGBClassifier () model from the sklearn library of python. Then, use the .fit() function to model learning from training data and labeling results used in training. And the .predict() function to predict the results of the test dataset.

```
xgboost = XGBClassifier( )
xgboost.fit(X_train, y_train)
predict_xgboost= xgboost.predict(X_test)
```

2.6 ROC

The ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classification system, plotting the True Positive Rate (TPR) on the vertical axis and the False Positive Rate (FPR) on the horizontal axis at different classification thresholds. The ROC curve helps to evaluate and compare the performance of different models or different configurations of the same model. The Area Under the Curve (AUC) of an ROC curve provides a single metric for comparing the overall performance of different models, with a higher AUC indicating better performance.

The metrics in the ROC curve include:

- True Positive Rate (TPR) or Sensitivity: The proportion of actual positive cases that are correctly identified by the model, calculated as the number of true positives divided by the total number of positives. TPR is an important metric for measuring the ability to detect positive cases accurately.
- False Positive Rate (FPR): The proportion of actual negative cases that are incorrectly identified as positive by the model, calculated as the number of false positives divided by the total number of negatives. FPR is often used to compare the performance of different models.
- Specificity (SPC): The proportion of actual negative cases that are correctly identified by the model, calculated as the number of true negatives divided by the total number of negatives. Specificity provides information about the model's ability to eliminate negative cases.
- Precision: The proportion of predicted positive cases that are actually positive, calculated as the number of true positives divided by the total number predicted as positive. Precision is an important metric for evaluating the accuracy of positive cases classified by the model.
- Accuracy: The proportion of all cases that are correctly classified by the model, calculated as the number of true positives and true negatives divided by the total number of data points. Accuracy is an overall metric for evaluating the model's performance.

For the ROC curve, the AUC (Area Under the Curve) is used to evaluate and compare the performance of different models, with a higher AUC indicating better ability to accurately classify positive and negative cases.

2.7 Data preprocessing

Data preprocessing is a very important step in solving any problem in the field of Machine Learning. Most of the datasets used in Machine Learning related problems need to be processed, cleaned and transformed before a Machine Learning algorithm can be trained on these datasets. Popular data preprocessing techniques today include: missing data, encoding categorical variables, standardizing data, and scaling data. (scaling data),… These techniques are relatively easy to understand but there will be many problems when we apply them to real data. Because the datasets that deal with real-world problems are very different, and each problem faces different data challenges.

2.7.1 Missing Data

What are missing data?

Missing value in a dataset is a very common phenomenon in the reality, yet a big problem in real-life scenarios. Missing Data can also refer to as NA(Not Available) values in pandas. In DataFrame sometimes many datasets simply arrive with missing data, either because it exists and was not collected or it never existed.

Some reasons for missing data are listed below:

- Past data might get corrupted due to improper maintenance.
- Different users may choose not to share their addresses.
- There might be a failure in recording the values due to human error.
- Due to improper maintenance, past data might get corrupted.

Why are missing data important?

Missing data are important because, depending on the type, they can sometimes bias your results. This means your results may not be generalizable outside of your study because your data come from an unrepresentative sample.

What are the types of missing data?

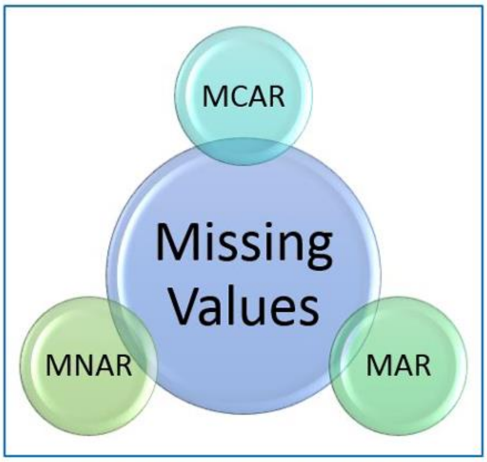


Figure 1 - Different Types of Missing Values in Datasets

Figure 6: Type of missing data (Source: analyticsvidhya)

There are three main types of missing data.

Missing completely at random (MCAR) data are randomly distributed across the variable and unrelated to other variables.

Missing at random (MAR) data are not randomly distributed but they are accounted for by other observed variables.

Missing not at random (MNAR) data systematically differ from the observed values.

How do I deal with missing data?

To tidy up your missing data, your options usually include accepting, removing, or recreating the missing data.

- Acceptance: You leave your data as is

- Listwise or pairwise deletion: You delete all cases (participants) with missing data from analyses
- Imputation: You use other data to fill in the missing data

2.7.2 Categorical data

Your dataset may contain grouping features. These features are often stored as text to represent different properties of the data. Some examples of grouping features such as color features include “Red”, “Yellow”, “Blue”, size features such as “Small”, “Medium”, “Large”, or feature. geographical location such as "Hanoi", "Ninh Binh", "Hoa Binh". Regardless of the type, we are faced with a problem of how to use these features in data analysis. A lot of machine learning algorithms can support clustered features, but there are also a lot of algorithms that can't run with this type of feature. Therefore, data analysts face the challenge of how to convert group data into digital data for further processing.

Cluster feature coding deals with the problem of transforming a clustered feature into one or more numeric features. You can use any mathematical or logical method you want to convert the clustered features because there is no limit to this. The transformations needed depend on the direction of your analysis. Together, we will learn some popular group feature encoding methods such as numeric encoding, one-hot encoding and binary encoding.

One-hot encryption

For unordered grouping features, the use of numeric encoding will change the nature of the data. This allows the model to assume that the values of the clustering feature are ordinal in nature, leading to inaccurate model results. In this case, one-hot encryption can be applied more efficiently. This method discards the clustered feature and transforms each value of that feature into a binary variable.

In the color feature example, this feature has 3 discrete values and so we will transform this feature into 3 binary features while the color feature is removed. In general, one-hot coding will need n new features to store the value for a group feature of n discrete values.

Color		Red	Yellow	Green
Red		1	0	0
Yellow		0	1	0
Green		0	0	1

2.7.3 Feature Selection

In the feature selection step, remove features that don't really contain useful information for the classification or prediction problem and improve the speed of training and prediction (when fewer features mean the model is trained). and faster forecasting) and even reduce overfitting.

In the project, we used the correlation coefficient with the target variable: The variables that are highly correlated with the target variable are those that have good explanatory power, and those that are low correlated with the target variable will be excluded. Elimination, and Correlation coefficients between variables: Removing highly correlated features, features with high correlation with each other can lead to multicollinearity in the model, leading to difficulty in interpreting the results.

Therefore, it is necessary to remove features that are highly correlated with each other to avoid this phenomenon. The importance of variables can be ranked using Pearson Correlation (Note: In addition to using the correlation coefficient with the target variable, we also have the following methods: Using the AIC index, Using the IV index, Choosing through the level of variance).

2.7.4 Scaling Data

Scaling is the transformation of a range of data values to a specific range such as 0-100 or 0-1, usually 0-1. In some Machine Learning algorithms where the distance between data points is important, like SVM or KNN, scaling the data is extremely important, because every small change in the data can lead to unpredictable results. before.

Min-max normalization (rescaling)

The goal of the method is to bring the values closer to the mean of the features. This method returns the values to a special interval, usually [0,1] [0,1] or [−1,1] [−1,1]. One of the limitations of this method is that when applied to a small range of values, we get a smaller standard deviation, which reduces the weight of the outliers in the data.

Scaling only changes the range of values, not the exact shape of the data distribution.

Mean normalization

Similar to the rescaling method, the mean rotation scaling method has a value in the range [-0.5, 0.5].

Standardization

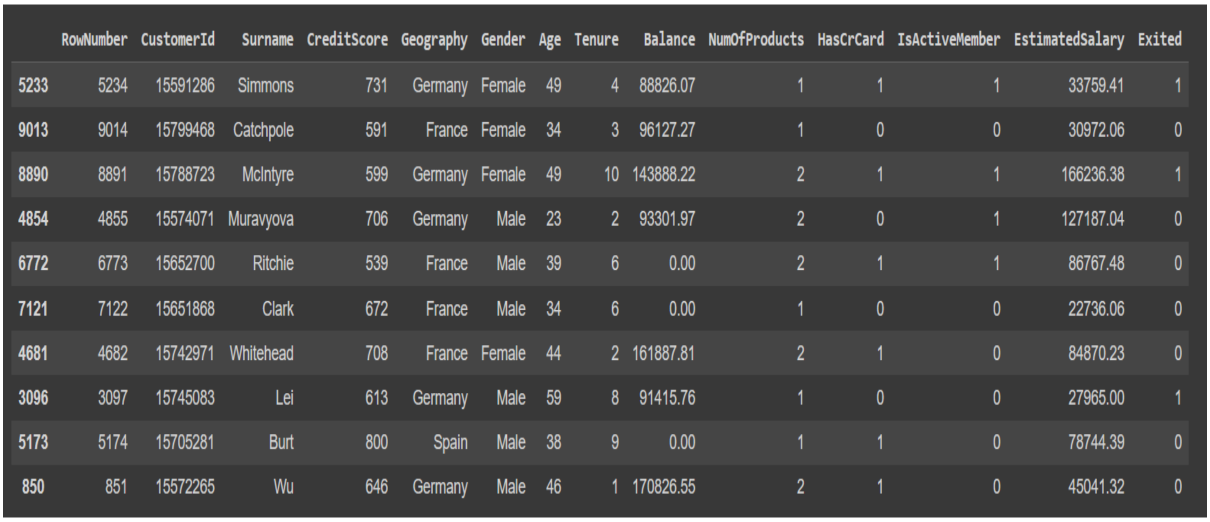
Process a large number of different data types, such as data in the form of an audio signal, pixels in an image, etc., and these data can be multidimensional data. Data regularization gives the value of each feature a mean of 0 and a variance of 1. This method is widely used in data normalization of many machine learning algorithms (SVM, logistic regression). and ANNs).

CHAPTER 3: EXPLORATORY DATA ANALYSIS AND DATA PREPROCESSING

This chapter focuses on the processes necessary for data discovery analysis, which may be used to acquire a comprehensive understanding of the data and proceed with data preprocessing. Feature selection from the postprocessing dataset improves the accuracy of the current predictive model.

3.1 Dataset Overview:

In this project, we use the Churn bank dataset from Kaggle. The original dataset contains about 10000 customer information using the service of bank. The dataset has the following:



	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
5233	5234	15591286	Simmons	731	Germany	Female	49	4	88826.07	1	1	1	33759.41	1
9013	9014	15799468	Catchpole	591	France	Female	34	3	96127.27	1	0	0	30972.06	0
8890	8891	15788723	McIntyre	599	Germany	Female	49	10	143888.22	2	1	1	166236.38	1
4854	4855	15574071	Muravyova	706	Germany	Male	23	2	93301.97	2	0	1	127187.04	0
6772	6773	15652700	Ritchie	539	France	Male	39	6	0.00	2	1	1	86767.48	0
7121	7122	15651868	Clark	672	France	Male	34	6	0.00	1	0	0	22736.06	0
4681	4682	15742971	Whitehead	708	France	Female	44	2	161887.81	2	1	0	84870.23	0
3096	3097	15745083	Lei	613	Germany	Male	59	8	91415.76	1	0	0	27965.00	1
5173	5174	15705281	Burt	800	Spain	Male	38	9	0.00	1	1	0	78744.39	0
850	851	15572265	Wu	646	Germany	Male	46	1	170826.55	2	1	0	45041.32	0

Table 3-1: The Overview of Dataset

This collected dataset consists of 10000 lines. And includes 14 columns such as RowNumber, CustomerID, SurName, CreditScore, Geography, Gender, Age, Tenure, Balance, NumofProducts, HascrCard, IsActiveMember, EstimatedSalary, Exited.

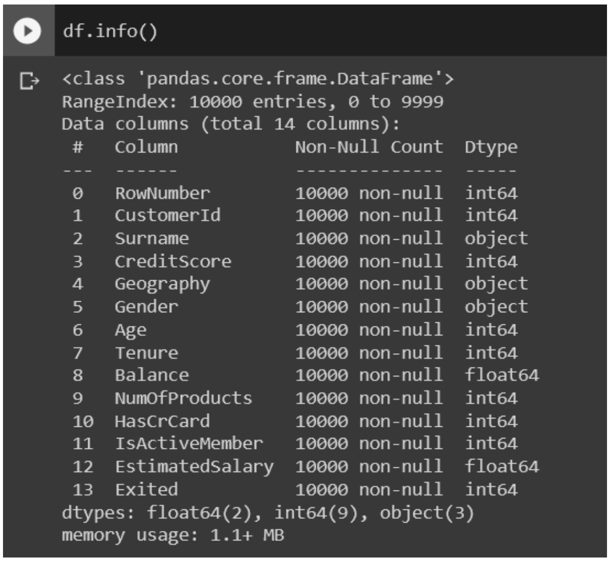
3.2 Exploratory Data Analysis (EDA)

From a concise summary as below:

__Table: Descriptions of Dataset__

Column Name	DataType	Description
RowNumber	int64	Row Numbers from 1 to 10000
CustomerId	int64	Unique Ids for bank customer identification
Surname	object	Customer’s last name
CreditScore	int64	Credit score of the customer
Geography	object	The country from which the customer belongs
Gender	object	Male or Female
Age	int64	Age of the customer
Tenure	int64	Number of years for which the customer has been with the bank
Balance	float64	Bank balance of the customer
NumOfProduct	int64	The number of banking products that the customer has used.
HasCrCard	int64	Binary Flag for whether the customer holds a credit with the bank or not
IsActiveMember	int64	Binary Flag for whether the customer is an active member with the bank or not
EstimatedSalary	float64	Estimated salary of the customer in Dollars
Exited	int64	Binary Flag 1 if the customer closed account with bank and 0 if the

		customer is retained



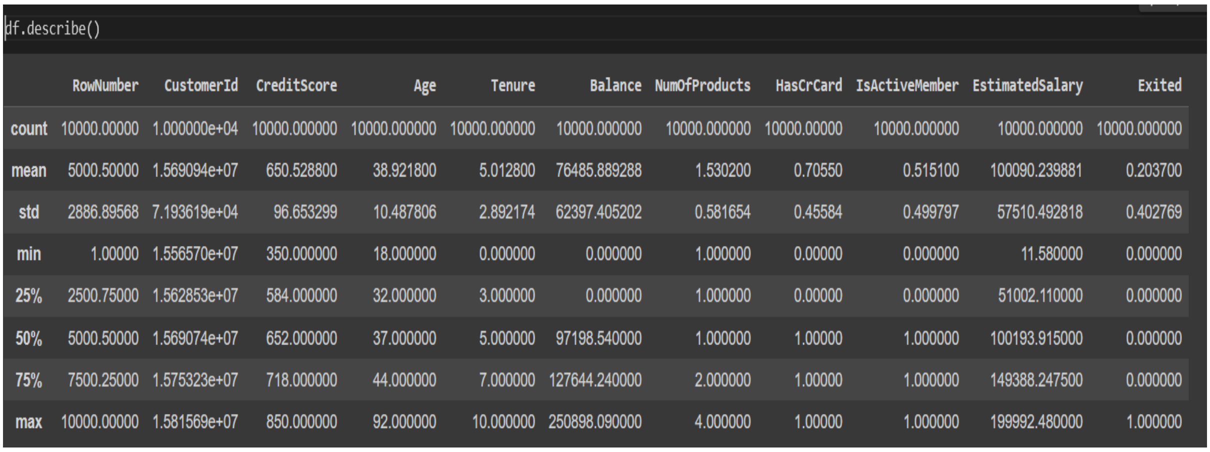
```
df.info()
```

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

The group found that the dataset has 10000 rows and 14 columns. And details of the properties are described in the table below.

When collecting data for research, it is important to know the form of data to effectively interpret and analyze them. And from the table above, consists of two types of data numeric data and categorical data.

First, we will start the statistical analysis of numerical data by Transposing a DataFrame.



```
df.describe()
```

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

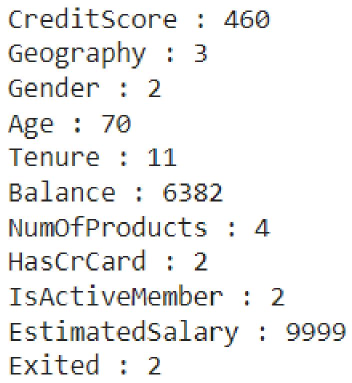
Credit score: personal credit score, between 350 - 850

Age: average age is 38.9 years old, youngest person is 18 years old, oldest person is 92 years old, 75% of people in the data set are under 44 years old.

Tenure: Customers use the service for an average of 5 years.

Balance: Average customer account balance is around $76485.

EstimatedSalary: Customers have an average salary of about $100090.

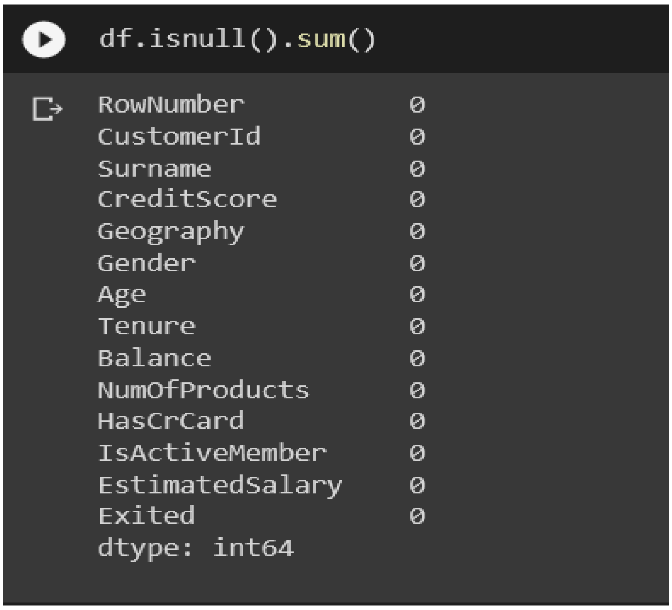


```
CreditScore : 460
Geography : 3
Gender : 2
Age : 70
Tenure : 11
Balance : 6382
NumOfProducts : 4
HasCrCard : 2
IsActiveMember : 2
EstimatedSalary : 9999
Exited : 2
```

The values of the variables also do not contain inappropriate values:

- There are 3 countries: France, Germany, Spain.
- Categorical variables: Age,HasCrCard, IsActiveMember, Exited all have 2 values.

In addition, we will determine how many null values each column has and creates a new dataframe which displays the sum of isnull values corresponding to the column name.



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

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

The dataset does not appear any Null values in the variables.

Realize that the "CustomerId, SurName, RowNumber" column contains information that is not necessary for the analysis and evaluation of churn bank. So we proceeded to drop the "CustomerId, SurName, RowNumber" column from the dataset.

```
df.drop(columns =['CustomerId','Surname','RowNumber'],inplace = True)
```

3.3 Explore variables

3.3.1 Age

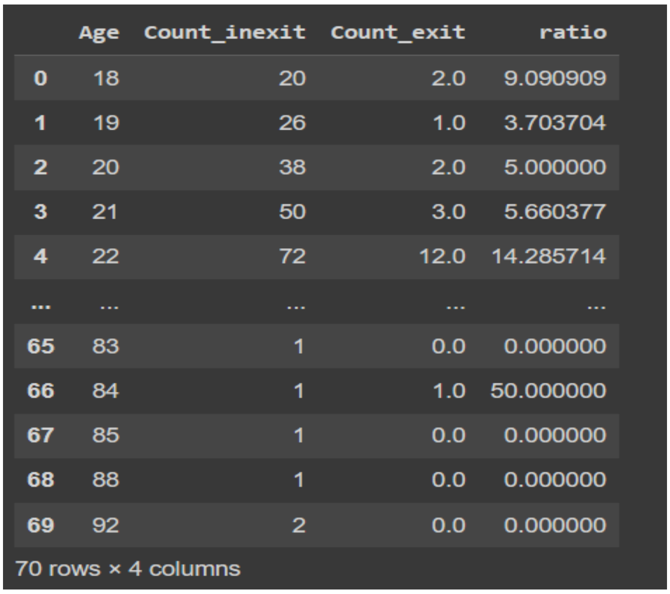
Ratio: percentage of people Exited in each age.

```
age_df = df.groupby(by=['Age','Exited']).agg('count')
age_df.reset_index(drop=False,inplace=True)
age_df_inexit = age_df[age_df['Exited']==0][['Age','CreditScore']]
age_df_inexit =age_df_inexit.rename(columns={"CreditScore":"Count_inexit"})

age_df_exit =age_df[age_df['Exited']==1][['Age','CreditScore']]
age_df_exit = age_df_exit.rename(columns={"CreditScore":"Count_exit"})

age_vs_exited =pd.merge(age_df_inexit,age_df_exit,how ='outer',left_on='Age',right_on="Age")
age_vs_exited['ratio'] =age_vs_exited.fillna(0,inplace=True)

age_vs_exited['ratio']=age_vs_exited['Count_exit']*100/(age_vs_exited['Count_inexit']+age_vs_exited['Co
unt_exit'])
age_vs_exited
```



```
     Age  Count_inexit  Count_exit      ratio
0     18            20         2.0   9.090909
1     19            26         1.0   3.703704
2     20            38         2.0   5.000000
3     21            50         3.0   5.660377
4     22            72        12.0  14.285714
...  ...           ...         ...        ...
65    83             1         0.0   0.000000
66    84             1         1.0  50.000000
67    85             1         0.0   0.000000
68    88             1         0.0   0.000000
69    92             2         0.0   0.000000
70 rows × 4 columns
```

Then, visualize the Ratio percentage of people exited in each age using a bar plot to show in the Dataset.

```
fig, ax=plt.subplots(figsize=[15,6])
sns.barplot(data=age_vs_exited,x='Age',y='ratio')
```

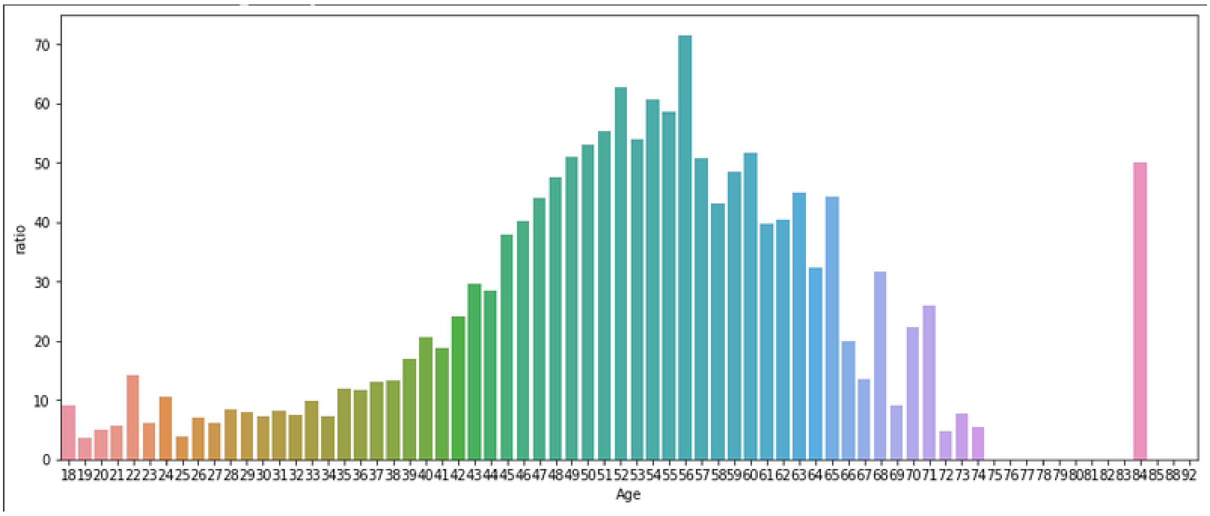
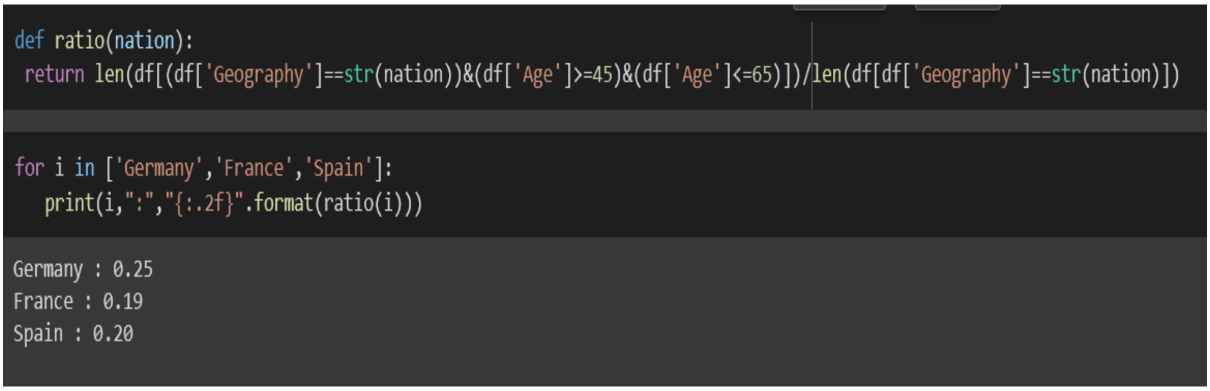


Figure 7: Ratio percentage of people exited bank(Source: authors)

From the chart above, we makes the analysis:

- The highest abandonment rate falls at the age of 56, the age group with the highest abandonment rate is 45-65
- At the age of 84 there was a spike in abandonment rates. Cause: At this age there are only 2 customers, 1 person is exited



```
def ratio(nation):
 return len(df[(df['Geography']==str(nation))&(df['Age']>=45)&(df['Age']<=65)])/len(df[df['Geography']==str(nation)])
```

```
for i in ['Germany','France','Spain']:
   print(i,":","{:.2f}".format(ratio(i)))
```

```
Germany : 0.25
France : 0.19
Spain : 0.20
```

- Germany has the highest percentage of people between the ages of 45 and 65.
- Prediction: Germany is the country with a higher rate of people in the exit group than the other two countries.

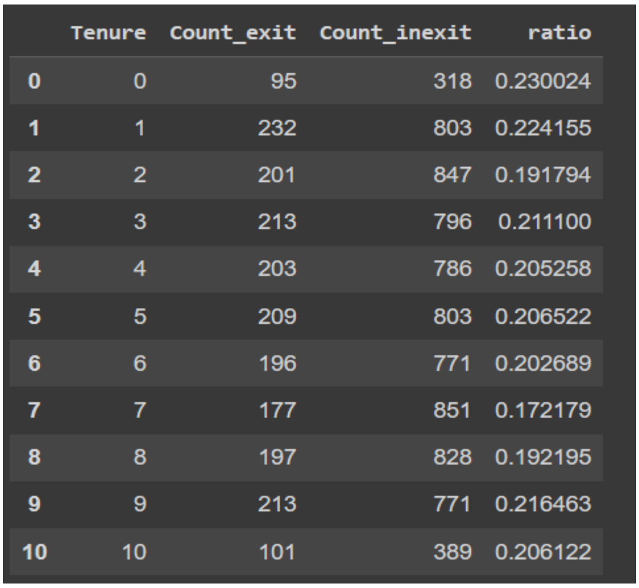
3.3.2 Tenure

Ratio: percentage of people Exited in each tenure

```
df_ten_exit = df[df.Exited ==1].groupby('Tenure').agg('count')
df_ten_exit.reset_index(drop=False,inplace=True)
df_ten_exit = df_ten_exit[['Tenure','CreditScore']]
df_ten_exit = df_ten_exit.rename(columns={"CreditScore":"Count"})

df_ten_inexited =df[df.Exited ==0].groupby('Tenure').agg('count')
df_ten_inexited.reset_index(drop=False,inplace=True)
df_ten_inexited = df_ten_inexited[['Tenure','CreditScore']]
df_ten_inexited = df_ten_inexited.rename(columns={"CreditScore":"Count"})

df_ten =pd.merge(df_ten_exit,df_ten_inexited,how ='outer',on='Tenure',suffixes=('_exit','_inexit'))
df_ten['ratio'] =df_ten['Count_exit']/(df_ten['Count_exit']+df_ten['Count_inexit'])
df_ten
```



```
    Tenure  Count_exit  Count_inexit     ratio
0        0          95           318  0.230024
1        1         232           803  0.224155
2        2         201           847  0.191794
3        3         213           796  0.211100
4        4         203           786  0.205258
5        5         209           803  0.206522
6        6         196           771  0.202689
7        7         177           851  0.172179
8        8         197           828  0.192195
9        9         213           771  0.216463
10      10         101           389  0.206122
```

Then, visualize the Ratio percentage of people exited in each tenure using a bar plot to show in the Dataset.

```
fig, ax=plt.subplots(figsize=[10,6])

ax.bar(x=df_ten.Tenure,height=df_ten['Count_inexit'],label='Count_inexit',color='Green')
ax.bar(x=df_ten.Tenure,height=df_ten['Count_exit'],label='Count_exit',color='Indigo')

ax2 =ax.twinx()
ax2.plot(df_ten.Tenure,df_ten.ratio,color='purple')
```

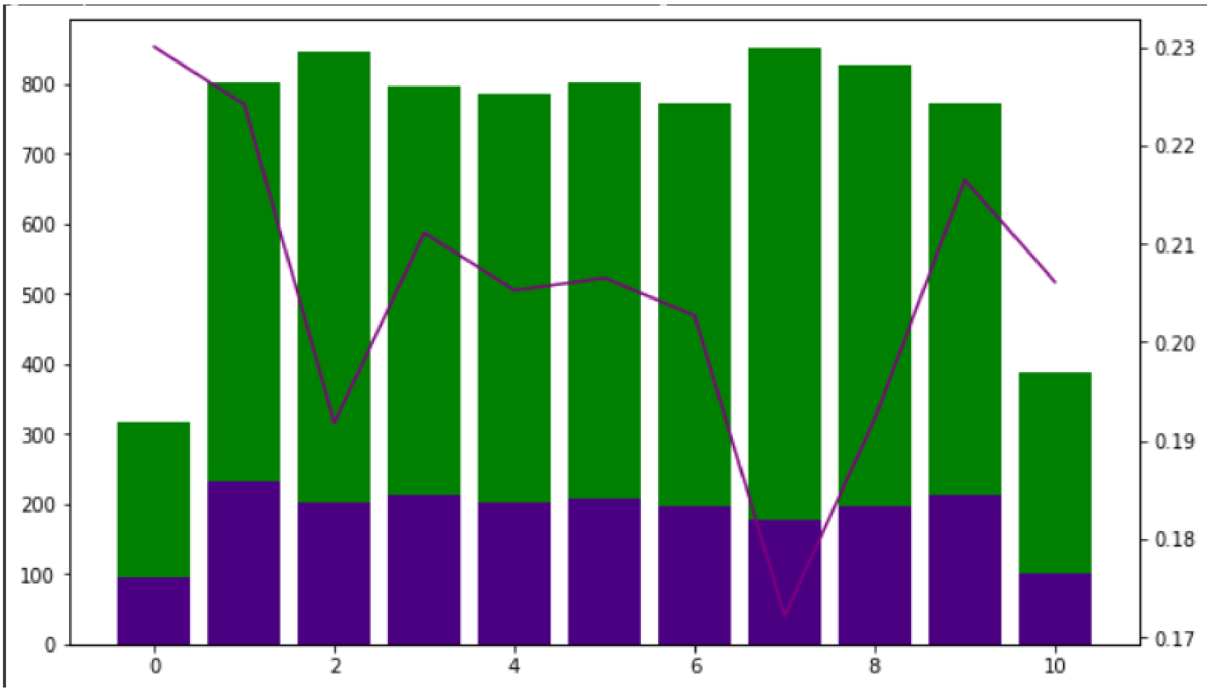


Figure 8:The Ratio percentage of people exited in each tenure(source: authors)

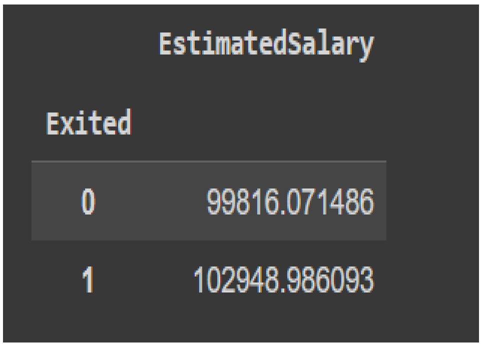
- Customers use the service from 1-9 years with the largest number, the rate of Exited/Total through each customer is basically unchanged.
- Tenure = 7, the ratio is significantly lower. We wonder if the percentage of Germans present in this Tenure = 7 is a large number.

3.3.3 Estimated Salary

The average salary of men and women in the data set is quite similar

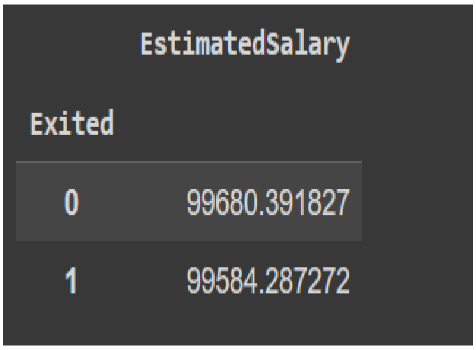
```
df_sal_exited_fe = df[df['Gender']=='Female'][['Exited','EstimatedSalary']]
df_sal_exited_ma =df[df['Gender']=='Male'][['Exited','EstimatedSalary']]

df_sal_exited_fe.groupby(by='Exited').agg("mean")
```



```
        EstimatedSalary
Exited
0          99816.071486
1         102948.986093
```

```
df_sal_exited_ma.groupby(by='Exited').agg("mean")
```

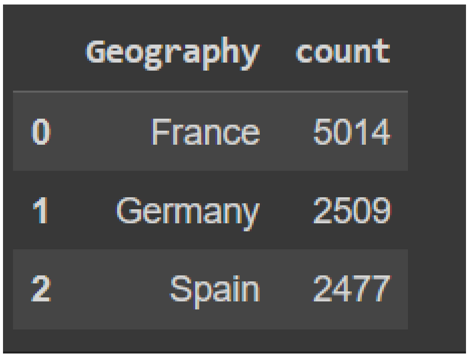


	EstimatedSalary
Exited	
0	99680.391827
1	99584.287272

3.3.4 Geography

Analysis of banking service users by geographical region.

```
df_geo =df.groupby(by=['Geography']).agg('count')
df_geo.reset_index(inplace=True,drop=False)
df_geo = df_geo[['Geography','Age']]
df_geo.rename(inplace=True,columns={'Age':'count'})
df_geo
```



	Geography	count
0	France	5014
1	Germany	2509
2	Spain	2477

From the above analysis we can see:

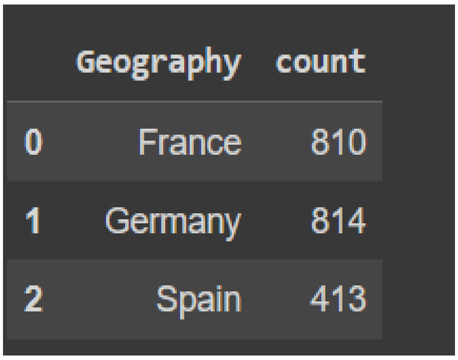
- The number of people using banking services in France is 5014 people.
- The number of people using banking services in Germany is 2509 people.
- The number of people using banking services in Spain is 2477 people.

Thus, France has the largest number of people using banking services, and the least in Spain.

```
df_geo_exit =df[df['Exited']==1].groupby(by=['Geography']).agg('count')
df_geo_exit.reset_index(inplace=True,drop=False)
df_geo_exit =df_geo_exit[['Geography','Age']]
```

```
df_geo_exit.rename(inplace=True,columns ={'Age':'count'})


df_geo_exit
```

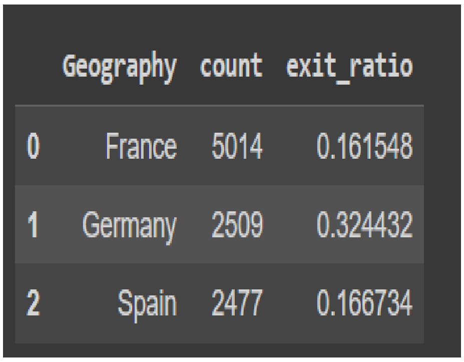


	Geography	count
0	France	810
1	Germany	814
2	Spain	413

From the above analysis we can see:

- The number of people exited banking services in France is 810 people.
- The number of people exited banking services in Germany is 814 people.
- The number of people exited banking services in Spain is 413 people.
- The highest number of people leaving was in Germany, followed by France Spain has the lowest number of people leaving.

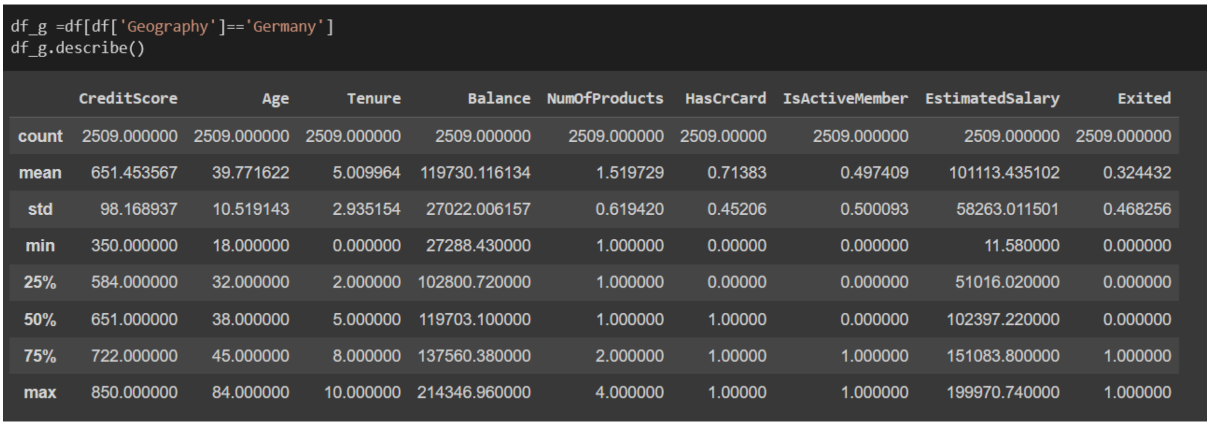
```
df_geo['exit_ratio'] = df_geo_exit['count']/df_geo['count']
df_geo
```



	Geography	count	exit_ratio
0	France	5014	0.161548
1	Germany	2509	0.324432
2	Spain	2477	0.166734

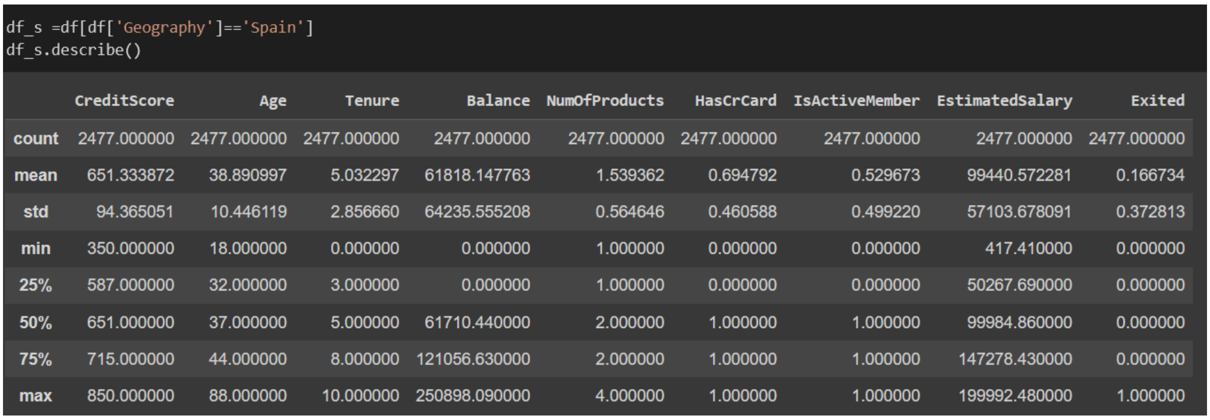
In the data, the number of French people is more than double the number of Germany and Spain, but Germany has the highest abandonment rate, up to 32%, almost double the other two countries.

It can be said that Germany is still not interested in being a loyal customer of the bank.



```
df_g =df[df['Geography']=='Germany']
df_g.describe()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	2509.000000	2509.000000	2509.000000	2509.000000	2509.000000	2509.00000	2509.000000	2509.000000	2509.000000
mean	651.453567	39.771622	5.009964	119730.116134	1.519729	0.71383	0.497409	101113.435102	0.324432
std	98.168937	10.519143	2.935154	27022.006157	0.619420	0.45206	0.500093	58263.011501	0.468256
min	350.000000	18.000000	0.000000	27288.430000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	584.000000	32.000000	2.000000	102800.720000	1.000000	0.00000	0.000000	51016.020000	0.000000
50%	651.000000	38.000000	5.000000	119703.100000	1.000000	1.00000	0.000000	102397.220000	0.000000
75%	722.000000	45.000000	8.000000	137560.380000	2.000000	1.00000	1.000000	151083.800000	1.000000
max	850.000000	84.000000	10.000000	214346.960000	4.000000	1.00000	1.000000	199970.740000	1.000000



```
df_s =df[df['Geography']=='Spain']
df_s.describe()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	2477.000000	2477.000000	2477.000000	2477.000000	2477.000000	2477.000000	2477.000000	2477.000000	2477.000000
mean	651.333872	38.890997	5.032297	61818.147763	1.539362	0.694792	0.529673	99440.572281	0.166734
std	94.365051	10.446119	2.856660	64235.555208	0.564646	0.460588	0.499220	57103.678091	0.372813
min	350.000000	18.000000	0.000000	0.000000	1.000000	0.000000	0.000000	417.410000	0.000000
25%	587.000000	32.000000	3.000000	0.000000	1.000000	0.000000	0.000000	50267.690000	0.000000
50%	651.000000	37.000000	5.000000	61710.440000	2.000000	1.000000	1.000000	99984.860000	0.000000
75%	715.000000	44.000000	8.000000	121056.630000	2.000000	1.000000	1.000000	147278.430000	0.000000
max	850.000000	88.000000	10.000000	250898.090000	4.000000	1.000000	1.000000	199992.480000	1.000000

3.3.5 Balance

```
sns.displot(df['Balance'])
```

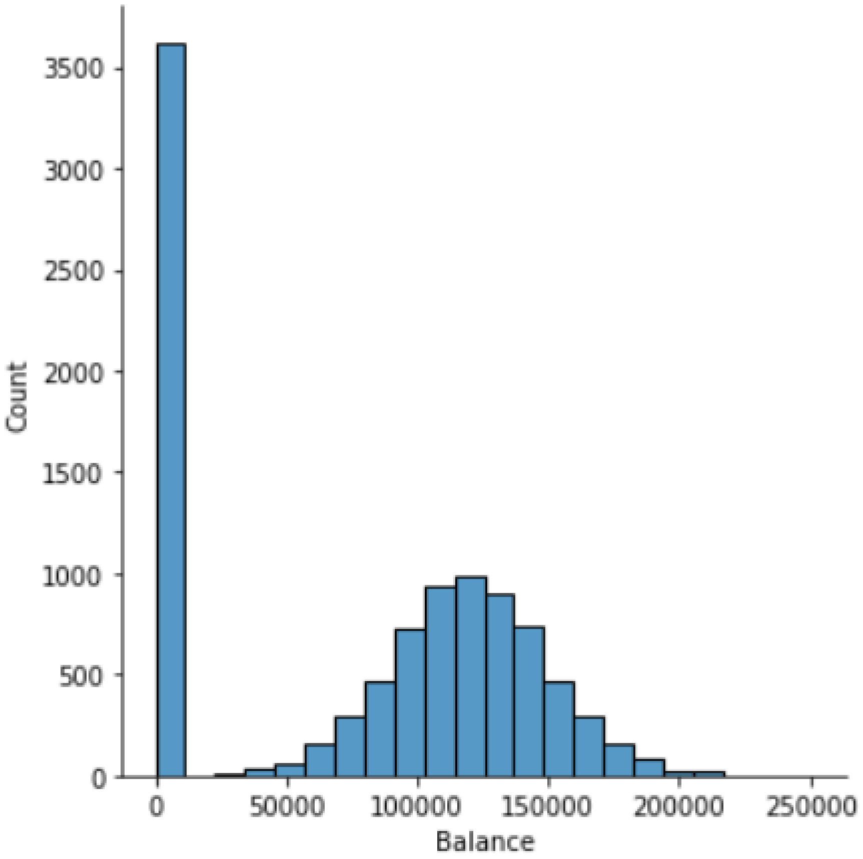


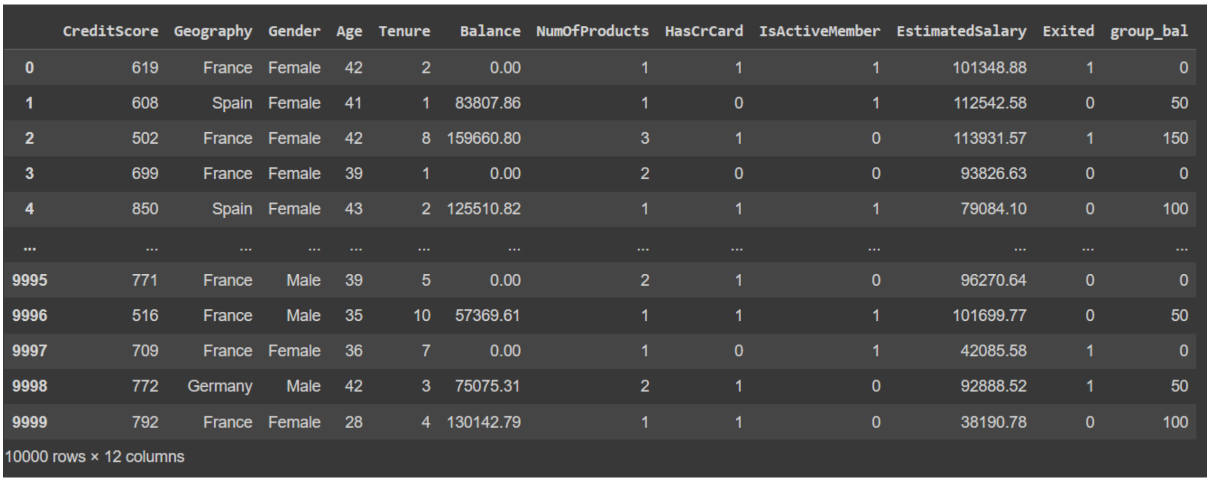
Figure 9: The number of customers who maintain account balance(source: authors)

- The number of customers who maintain account balance = 0 is relatively large We will ask a question
- Are these people contributing to the increased churn rate?

Divide into groups, group by together:

```
from ast import Return
def group_bal(bal):
 if int(bal)==0:
  return 0
 if int(bal) in range (1,50000):
  return 1
 elif int(bal) in range(50000,100000): return 50
 elif int(bal) in range(100000,150000):
   return 100
 elif int(bal) in range(150000,200000):
   return 150
 elif int(bal) in range(200000,250000):
   return 200
 else:
   return 250
```

```
df['group_bal'] =df['Balance'].map(group_bal)
df
```



	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	group_bal
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1	0
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0	50
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1	150
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0	100
...	...	...	...	...	...	...	...	...	...	...	...	...
9995	771	France	Male	39	5	0.00	2	1	0	96270.64	0	0
9996	516	France	Male	35	10	57369.61	1	1	1	101699.77	0	50
9997	709	France	Female	36	7	0.00	1	0	1	42085.58	1	0
9998	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1	50
9999	792	France	Female	28	4	130142.79	1	1	0	38190.78	0	100

10000 rows × 12 columns

```
df_group_bal = df.groupby(by='group_bal').agg('count')
df_group_bal.reset_index(inplace=True,drop=False)

labels =df_group_bal['group_bal']

figi, ax1 = plt.subplots()
ax1.pie(df_group_bal['Age'],labels=labels,autopct='%1.1f%%')
plt.show()
```

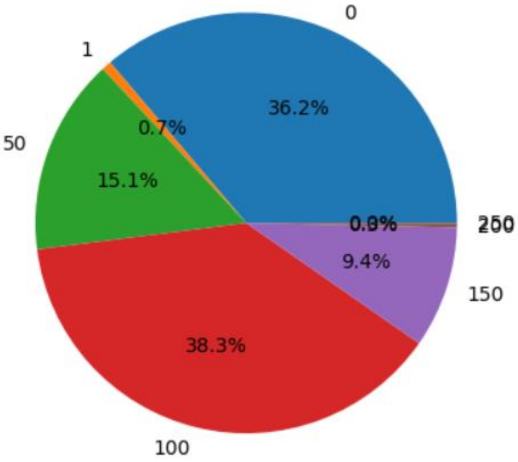
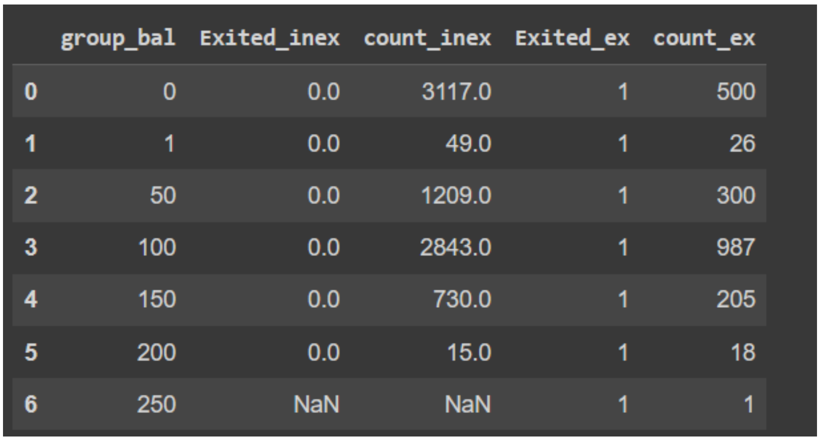


Figure 10: Account Balance(source: authors)

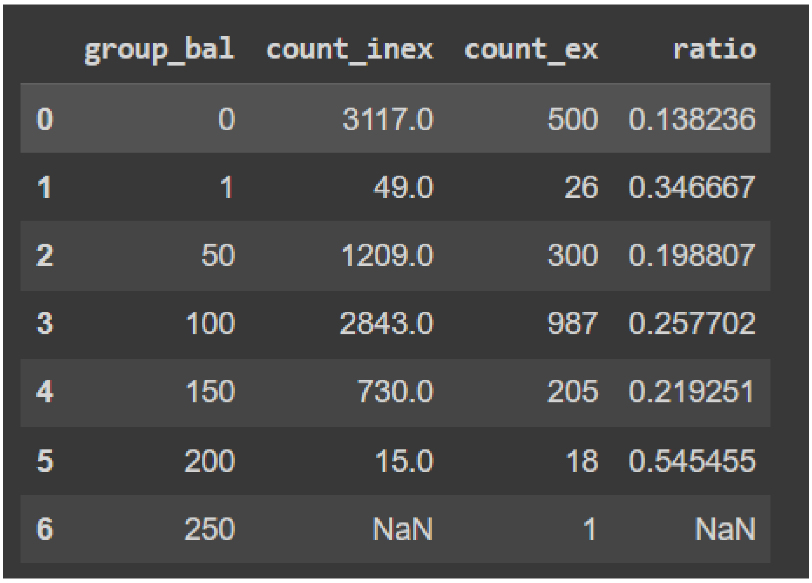
```
df_bal =df.groupby(by=['group_bal','Exited']).agg('count')
df_bal.reset_index(drop=False,inplace=True)
df_bal= df_bal.rename(columns={'Age':'count'})
df_bal =df_bal[['group_bal','Exited','count']]
df_balance_ =df_bal[df_bal['Exited']==0]
df_balance_ex =df_bal[df_bal['Exited']==1]
```

```
df_balance = pd.merge(df_balance_,df_balance_ex,how ='outer',on='group_bal',suffixes=('_inex','_ex'))
df_balance
```



	group_bal	Exited_inex	count_inex	Exited_ex	count_ex
0	0	0.0	3117.0	1	500
1	1	0.0	49.0	1	26
2	50	0.0	1209.0	1	300
3	100	0.0	2843.0	1	987
4	150	0.0	730.0	1	205
5	200	0.0	15.0	1	18
6	250	NaN	NaN	1	1

```
df_balance.drop(columns=['Exited_ex','Exited_inex'],inplace=True)
df_balance['ratio']=df_balance.count_ex/(df_balance.count_ex + df_balance.count_inex)
df_balance
```



	group_bal	count_inex	count_ex	ratio
0	0	3117.0	500	0.138236
1	1	49.0	26	0.346667
2	50	1209.0	300	0.198807
3	100	2843.0	987	0.257702
4	150	730.0	205	0.219251
5	200	15.0	18	0.545455
6	250	NaN	1	NaN

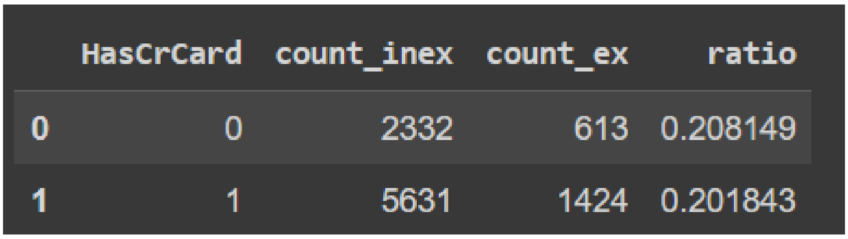
- In general, for the group of customers who maintain balance = 0, the abandonment rate is also relatively high (36.2%), but it can be seen that this index does not have a significant impact compared to the number of customers. rows in the dataset.
- It can be seen that the churn rate of the customer group with balance = 0 is not high.(not the highest).
- So, maintaining a balance = 0 does not mean that the churn rate will be high.

3.3.6 Has Credit Card

Whether or not the card has an effect on churn

```
df_creditcard =df.groupby(by=['HasCrCard','Exited']).agg('count')
df_creditcard.reset_index(drop=False,inplace=True)
df_creditcard =df_creditcard[['Exited','HasCrCard', 'Age']]
df_creditcard.rename(columns={'Age':'count'},inplace=True)

df_creditcard
=pd.merge(df_creditcard[df_creditcard['Exited']==0],df_creditcard[df_creditcard['Exited']==1],on
='HasCrCard',how ='outer',suffixes=('_inex','_ex'))
df_creditcard.drop(columns=['Exited_inex','Exited_ex'],inplace=True)
df_creditcard['ratio']=df_creditcard['count_ex']/(df_creditcard['count_ex']+df_creditcard['count_inex'])
df_creditcard
```



	HasCrCard	count_inex	count_ex	ratio
0	0	2332	613	0.208149
1	1	5631	1424	0.201843

Has Credit Card : has no effect on abandonment rate, as with or without a credit card, abandonment rates are approximately the same and are around 20%.

Having a Credit Card does not mean keeping customers. If a customer is not satisfied with the quality of the company's product or service, or the price is not consistent with the value received, they may still decide to leave whether they already have a credit card or not.

3.3.7 Credit score

```
sns.distplot(df['CreditScore'])
```

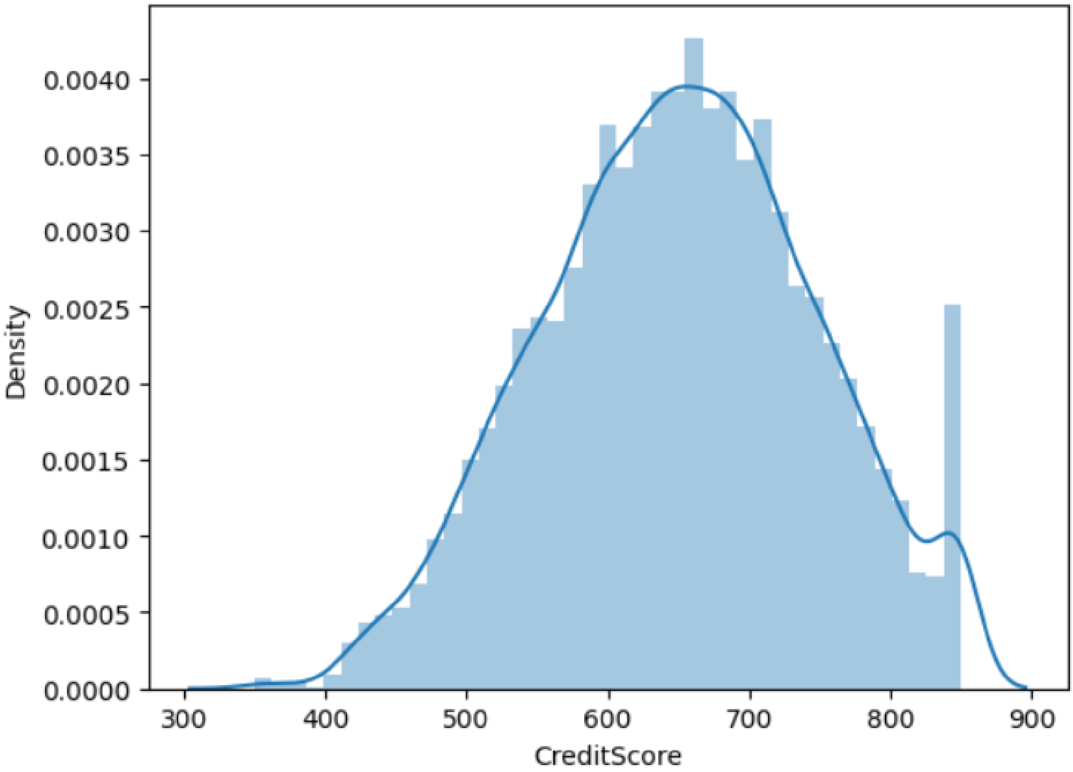


Figure 11:Impact on credit scores for customers in all 3 countries (source: authors)

```
geo =['France','Spain','Germany']
fig,axes = plt.subplots(1,3,figsize=(10,5))
for index, i in enumerate(geo):
  y = index%3
  sns.distplot(df[df['Geography']==i]['CreditScore'],ax=axes[y],axlabel=i )

plt.subplots_adjust(hspace=1,right=1.3)
```

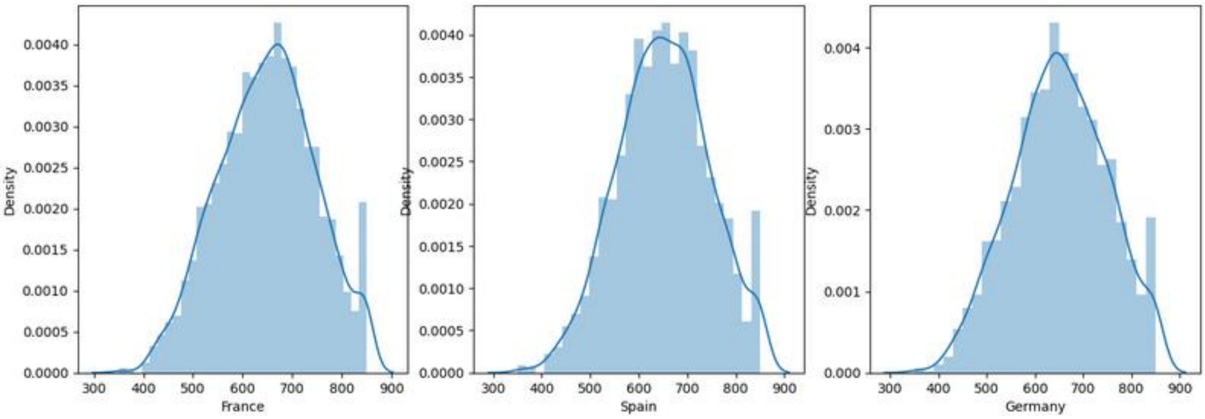


Figure 12:Impact on credit scores for customers in of France, Spain, Germany (source: authors)

All 3 countries have similar credit score distribution and similarity in common there are about 600 - 700 is the highest.

CreditScore does not influence a customer's decision to leave or not too much, because If a customer has a low credit score or a bad credit history, they may have difficulty applying for loans or other financial products, and therefore less likely to leave the current supplier. However, if the customer finds that the company does not provide enough value or the quality of the product/service does not meet their needs, they may still decide to leave, whether or not they are denied a loan.

3.3.8 Is Active Member

```
sns.countplot(data=df,x='IsActiveMember',hue='Exited')
```

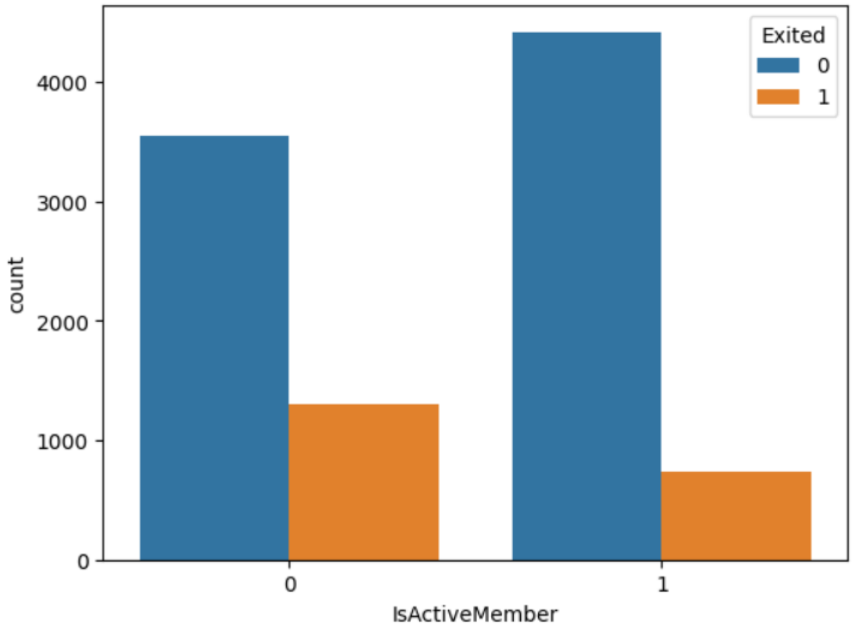


Figure 3.7: Churn rate of active member (source: authors)

Figure 3.7: Churn rate of active member (source: authors)

Active then the abandonment rate will be lower than not active. Same for countries.

Germany's Is Active Member abandonment rate:

```
sns.countplot(data=df[df['Geography']=='Germany'],x='IsActiveMember',hue='Exited')
```

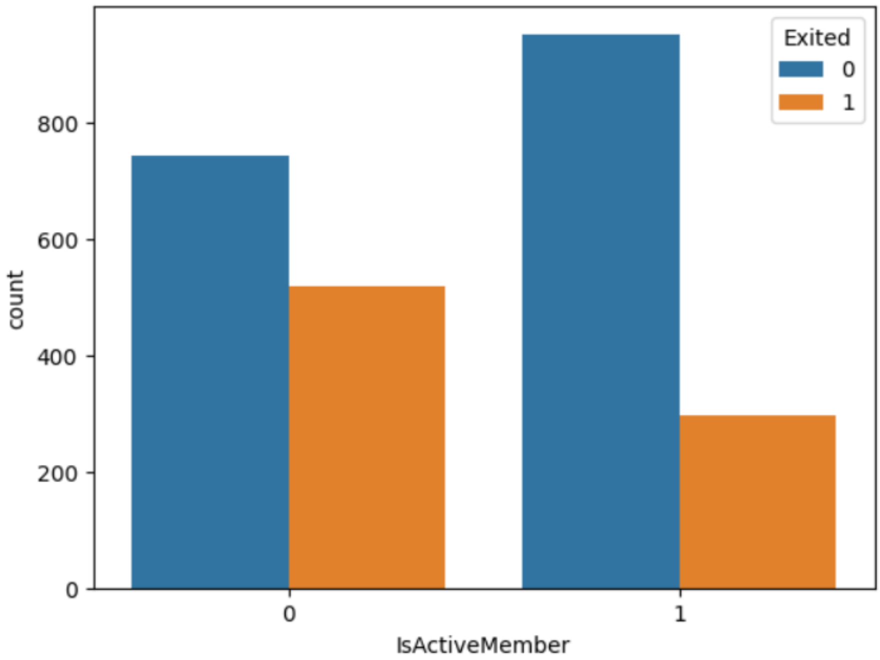


Figure 13: Churn rate of active member of Germany (source: authors)

France’s Is Active Member abandonment rate:

```
sns.countplot(data=df[df['Geography']=='France'],x='IsActiveMember',hue='Exited')
```

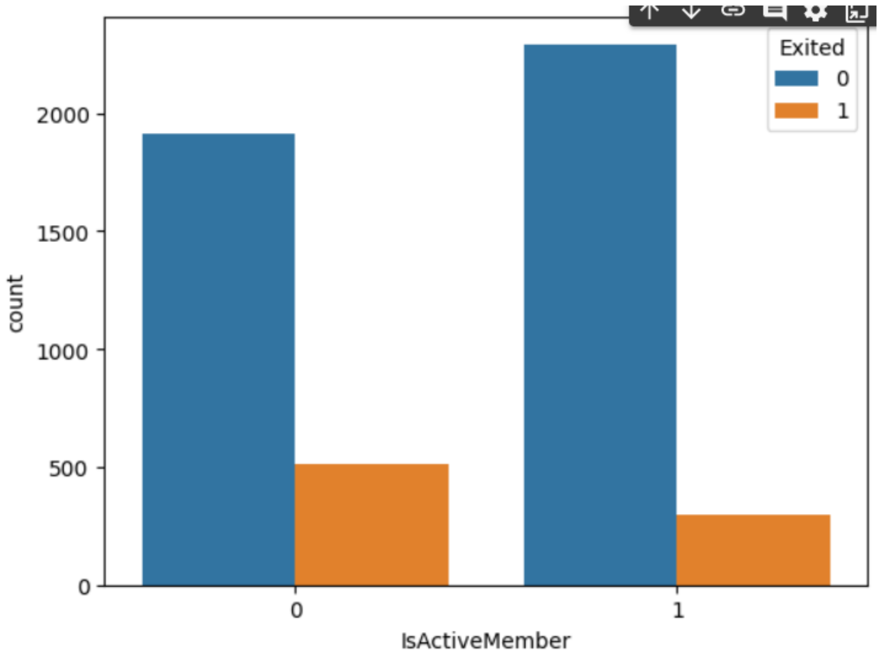


Figure 14: Churn rate of active member of France (source: authors)

Spain's Is Active Member abandonment rate:

```
sns.countplot(data=df[df['Geography']=='Spain'],x='IsActiveMember',hue='Exited')
```

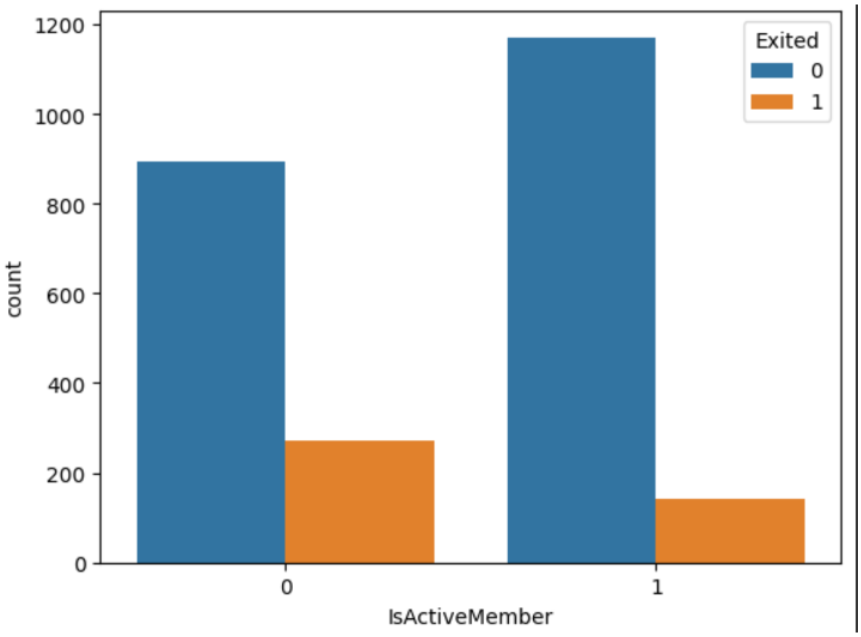


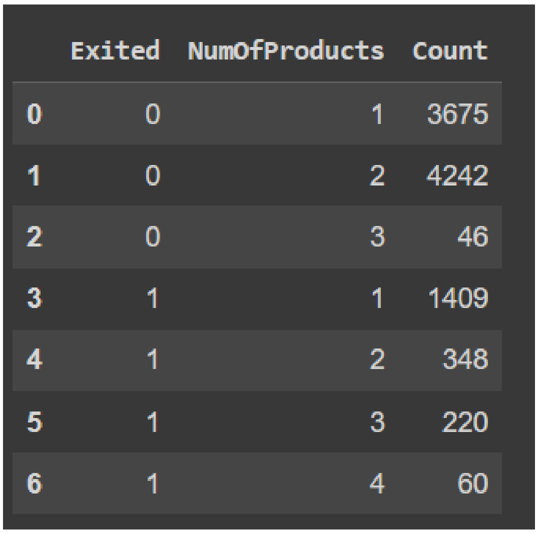
Figure 15: Churn rate of active member of Spain (source: authors)

⇒ People who are active members are less likely to leave.

Customers who are active members of a product or service are less likely to churn, i.e., they are less likely to discontinue using the product or service. Being an active member can imply different things depending on the context, but generally it refers to customers who have a higher level of engagement and usage of the product or service, such as making regular payments, logging into their account frequently, various features of the product/service, or participate in loyalty programs. The idea behind this statement is that customers who are more engaged with a product or service are more likely to see its value and benefits, and therefore are less likely to switch to a competitor or cancel their subscription.

3.3.9 Number Of Products

```
df_pro =df.groupby(by=['Exited','NumOfProducts']).agg('count')
df_pro.reset_index(drop=False,inplace=True)
df_pro =df_pro[['Exited','NumOfProducts','CreditScore']]
df_pro =df_pro.rename(columns={'CreditScore':'Count'})
df_pro
```



	Exited	NumOfProducts	Count
0	0	1	3675
1	0	2	4242
2	0	3	46
3	1	1	1409
4	1	2	348
5	1	3	220
6	1	4	60

All customers who do not leave the bank buy 1, 2, 3 products, not 4 products

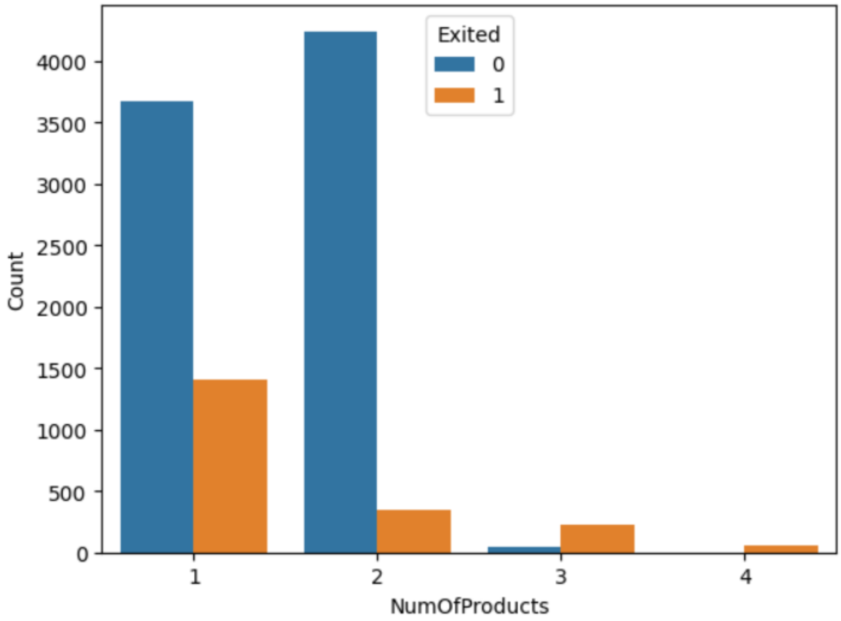


Figure 16: Churn rate of NumofProduct (source: authors)

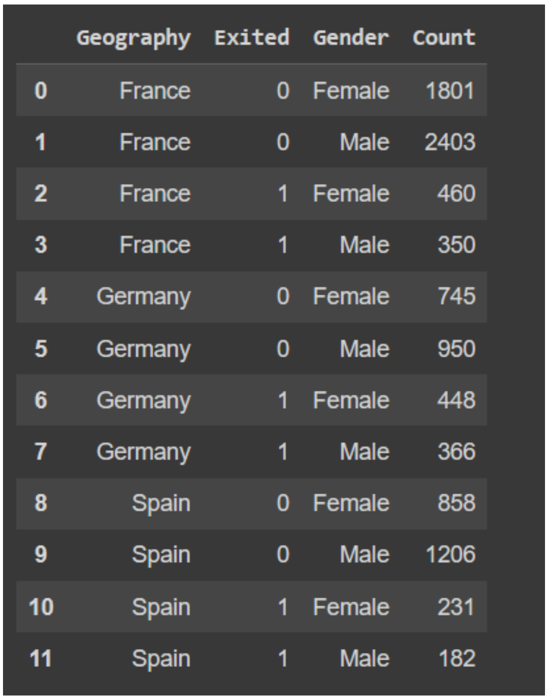
All customers who do not leave the bank buy 1, 2, 3 products, not 4 products.

There are different reasons why all customers who do not leave the bank buy only 1, 2 or 3 products but not 4 products. One reason could be that the bank does not offer a fourth product that matches the needs and preferences of the customer. Or the fourth product may not be as attractive or valuable as the other products, making customers reluctant to buy it. Alternatively, it may be a strategic decision by the bank to focus on developing and marketing a limited number of core products rather than offering too many products with limited differentiation. By doing so, banks can simplify their product portfolio, reduce costs and complexity, and can increase customer loyalty and satisfaction.

3.4 Analyze variables in correlation with each other

3.4.1 Geography and Gender

```
df_geo =df.groupby(by=['Geography','Exited','Gender']).agg('count')
df_geo =df_geo.rename(columns={"CreditScore":"Count"})
df_geo =df_geo[['Count']]
df_geo.reset_index(inplace=True,drop=False)
df_geo
```

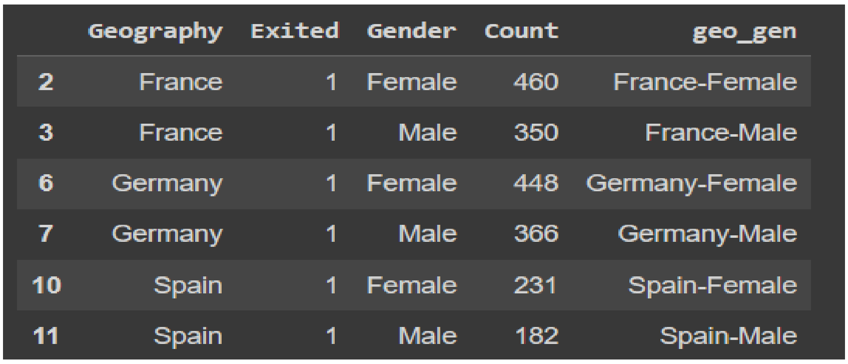


	Geography	Exited	Gender	Count
0	France	0	Female	1801
1	France	0	Male	2403
2	France	1	Female	460
3	France	1	Male	350
4	Germany	0	Female	745
5	Germany	0	Male	950
6	Germany	1	Female	448
7	Germany	1	Male	366
8	Spain	0	Female	858
9	Spain	0	Male	1206
10	Spain	1	Female	231
11	Spain	1	Male	182

Considering only customers using the service, combine 2 variables Geography and Gender together:

```
df_geo_ex =df_geo[df_geo['Exited']==1]


df_geo_ex['geo_gen'] =df_geo_ex['Geography']+'-'+df_geo_ex['Gender']
df_geo_ex
```



	Geography	Exited	Gender	Count	geo_gen
2	France	1	Female	460	France-Female
3	France	1	Male	350	France-Male
6	Germany	1	Female	448	Germany-Female
7	Germany	1	Male	366	Germany-Male
10	Spain	1	Female	231	Spain-Female
11	Spain	1	Male	182	Spain-Male

```
labels = df_geo_ex['geo_gen']
fig1, ax1 = plt.subplots()
ax1.pie(df_geo_ex['Count'],labels=labels,autopct='%1.1f%%')
plt.show()
```

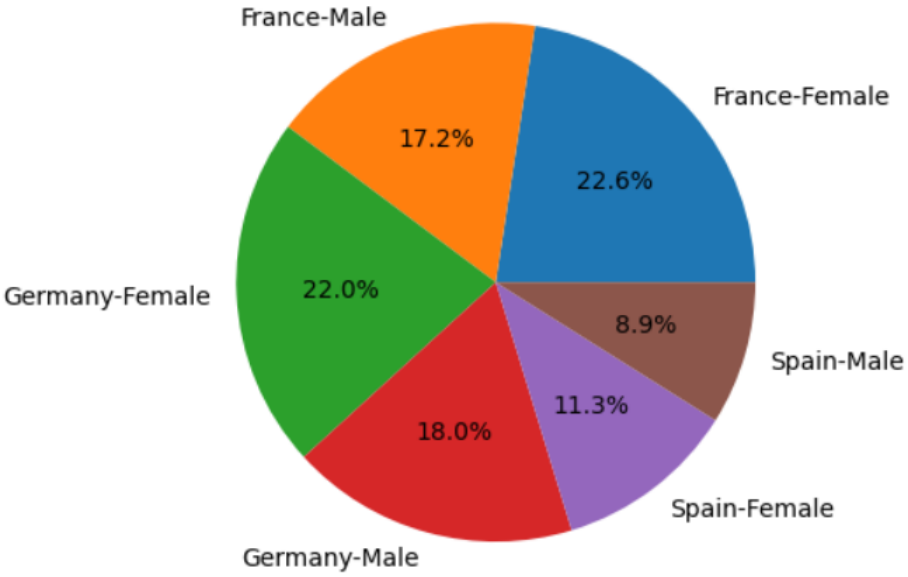


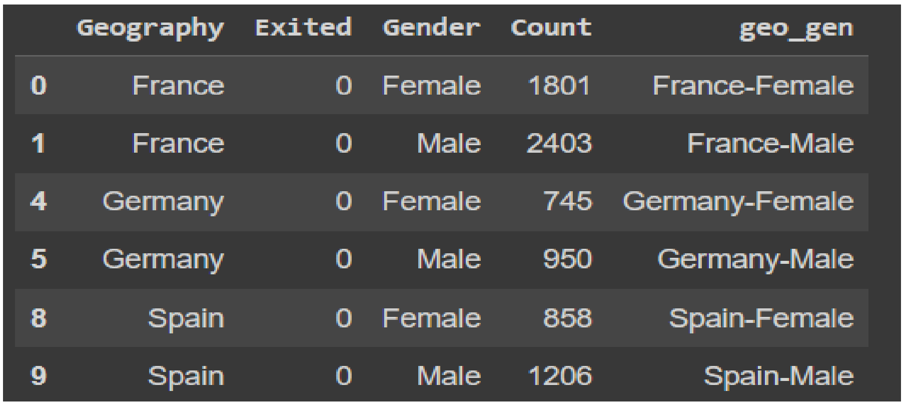
Figure 17: Gender rate of regions when customer exited (source: authors)

In which country, the rate of female leaving is still higher than that of male? Women in France and Men in Germany have the highest proportions

- For Exited = 0

```
df_geo_ex =df_geo[df_geo['Exited']==0]

df_geo_ex['geo_gen'] =df_geo_ex['Geography']+'-'+df_geo_ex['Gender']
df_geo_ex
```



	Geography	Exited	Gender	Count	geo_gen
0	France	0	Female	1801	France-Female
1	France	0	Male	2403	France-Male
4	Germany	0	Female	745	Germany-Female
5	Germany	0	Male	950	Germany-Male
8	Spain	0	Female	858	Spain-Female
9	Spain	0	Male	1206	Spain-Male

```
labels = df_geo_ex['geo_gen']
fig1, ax1 = plt.subplots()
ax1.pie(df_geo_ex['Count'],labels=labels,autopct='%1.1f%%')
plt.show()
```

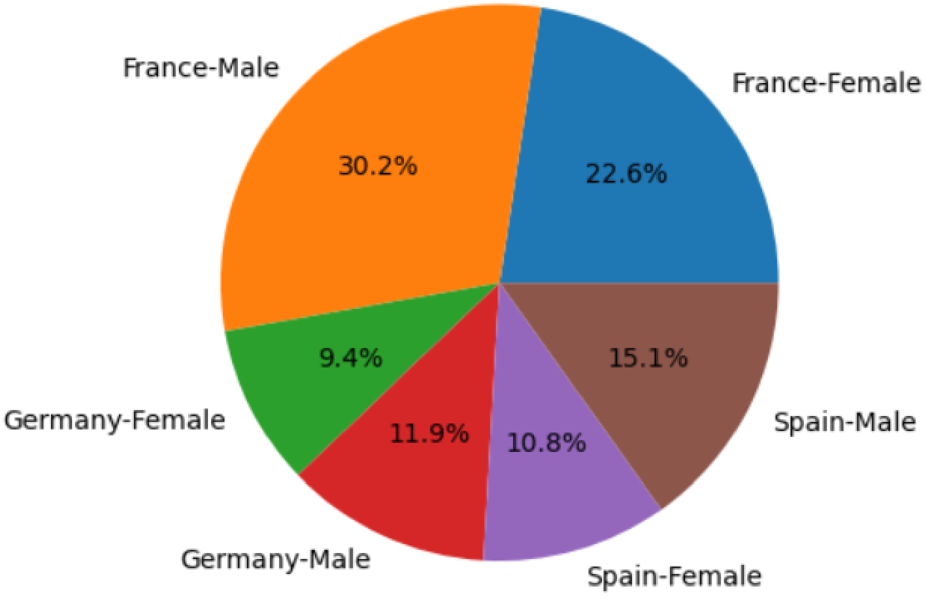


Figure 18: Gender rate of regions when exited = 0 (source: authors)

- Men in France use the service account for the highest proportion (because France has a large population, women churn more)
- Since Germany has a high churn rate, when considering Exited = 0, the rate in Germany is low, but the churn rate is still high.

3.4.2 Tenure và Gender

- Exited

```
df_ten_gen = df.groupby(by=['Tenure','Gender','Exited']).agg('count')
```

```
df_ten_gen.reset_index(inplace=True,drop=False)
df_ten_gen = df_ten_gen[['Tenure','Gender','Exited','CreditScore']]
df_ten_gen = df_ten_gen.rename(columns = {'CreditScore':"Count"})
df_ten_gen
```

Females in 3 countries all have a higher churn rate than males, so the graph looks like this:

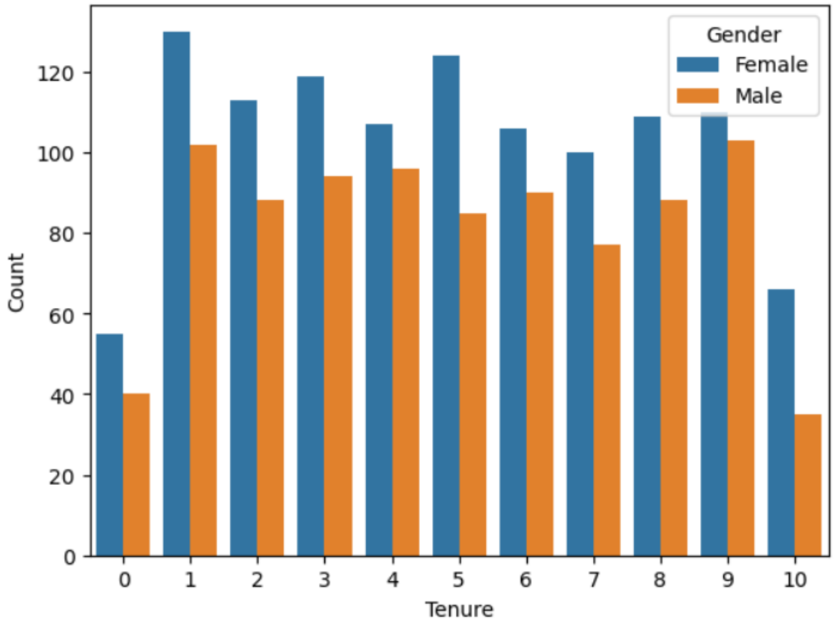


Figure 19: The churn count over incount of the year used(source: authors)

- Inexited:

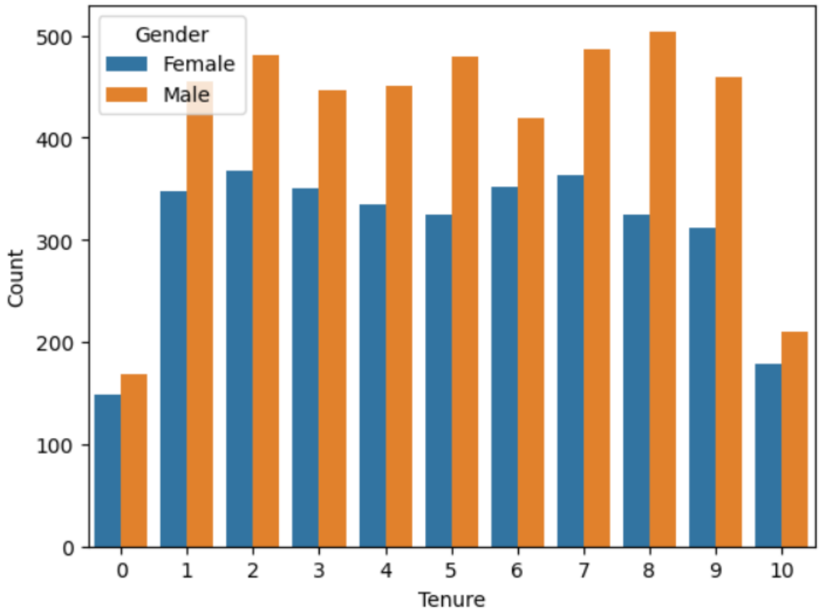


Figure 20: The non-churn count over incount of the year used(source: authors)

Higher female churn rate, spread over the number of years tenure:

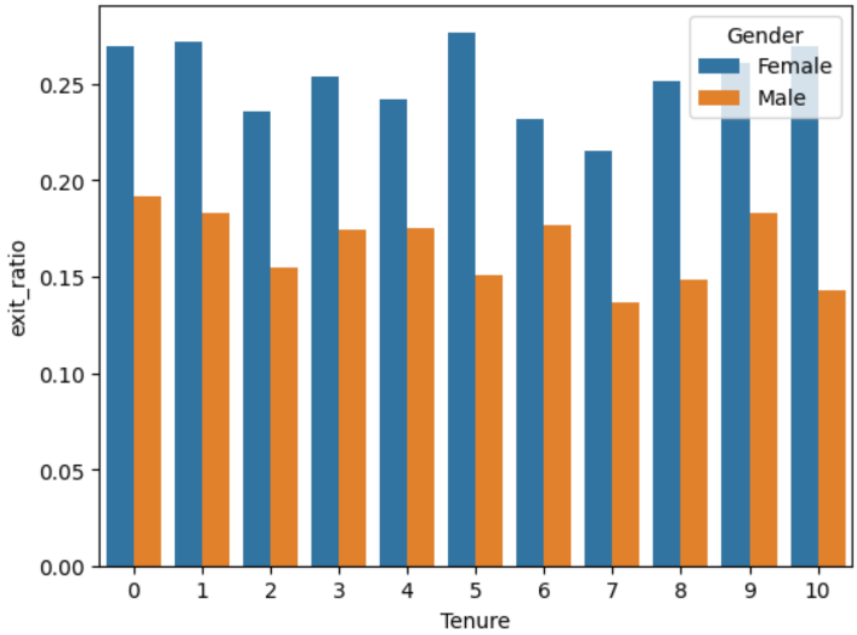
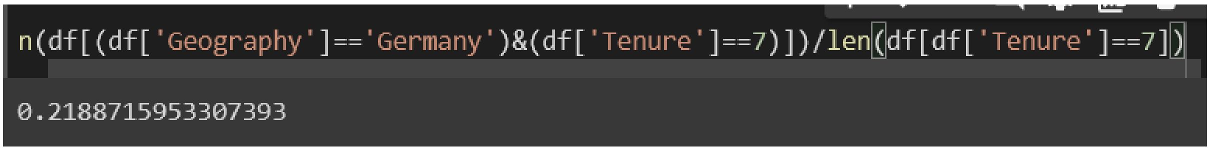


Figure 21: The churn rate over incount of the year used(source: authors)

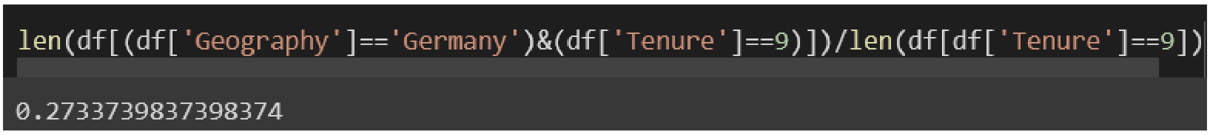
- Tenure = 7



```
n(df[(df['Geography']=='Germany')&(df['Tenure']==7)])/len(df[df['Tenure']==7])

0.2188715953307393
```

- Tenure = 9



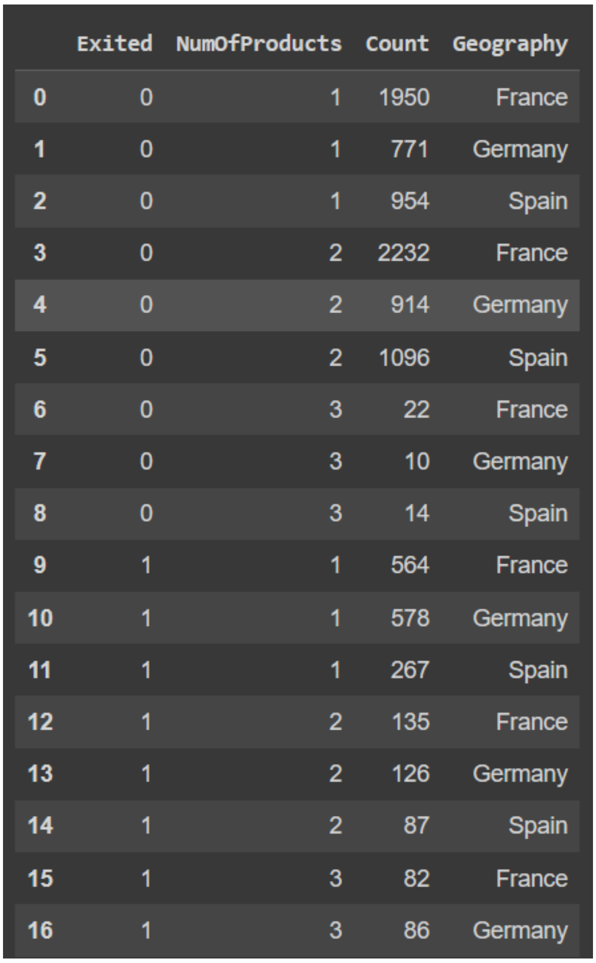
```
len(df[(df['Geography']=='Germany')&(df['Tenure']==9)])/len(df[df['Tenure']==9])

0.2733739837398374
```

⇒ The percentage of Germans (both men and women) at tenure = 7 is lower than the percentage of Germans (both men and women) at tenure = 9 => Pull the churn rate down.

3.4.3 Number of Products by Gender and Geography

```
df_pro = df.groupby(by=['Exited','NumOfProducts','Geography']).agg('count')
df_pro.reset_index(drop=False,inplace=True)
df_pro = df_pro[['Exited','NumOfProducts','CreditScore','Geography']]
df_pro = df_pro.rename(columns={'CreditScore':'Count'})
df_pro
```



	Exited	NumOfProducts	Count	Geography
0	0	1	1950	France
1	0	1	771	Germany
2	0	1	954	Spain
3	0	2	2232	France
4	0	2	914	Germany
5	0	2	1096	Spain
6	0	3	22	France
7	0	3	10	Germany
8	0	3	14	Spain
9	1	1	564	France
10	1	1	578	Germany
11	1	1	267	Spain
12	1	2	135	France
13	1	2	126	Germany
14	1	2	87	Spain
15	1	3	82	France
16	1	3	86	Germany

```
sns.barplot(data=df_pro[df_pro['Exited']==0],x='NumOfProducts', y='Count',hue='Geography')
```

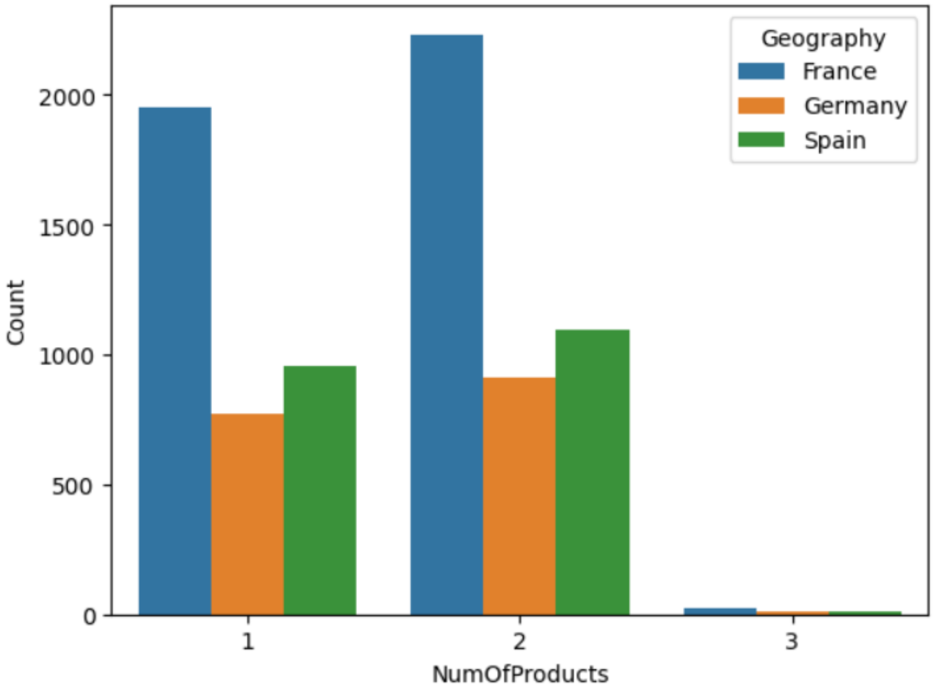


Figure 22: Number of people using 3 Banking products by country(source: authors)

In all 3 countries, the rate of using 1, 2 products is the majority.

All customers using the bank's 4 products churn:

```
sns.barplot(data=df_pro[df_pro['Exited']==1],x='NumOfProducts', y='Count',hue='Geography')
```

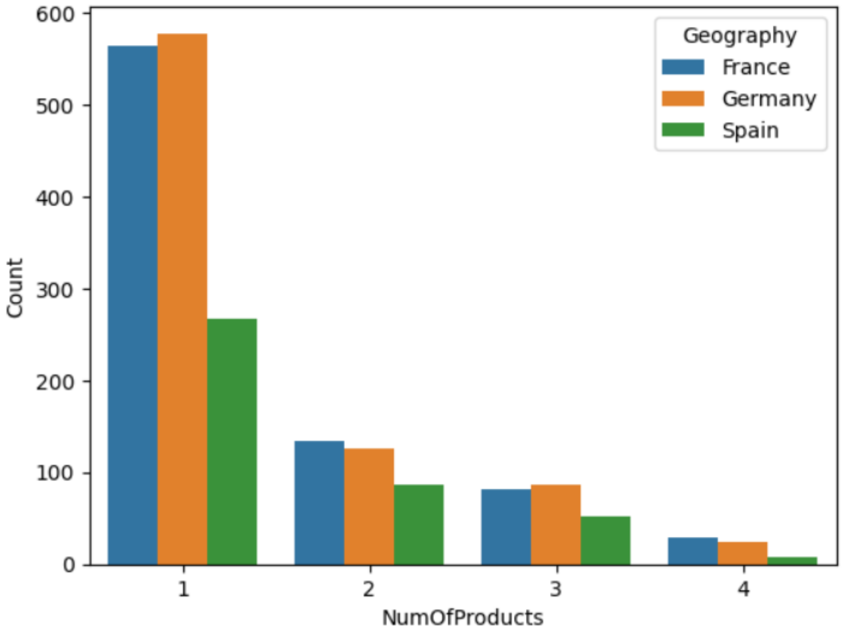


Figure 23: Number of people using 4 Banking products by country(source: authors)

The graph shows the churn rate by the number of products used:

```
df_product = df.groupby(by=['Exited','NumOfProducts']).agg('count')

df_product.reset_index(drop=False, inplace=True)

df_product = df_product.rename(columns={'Age':'count'})

df_product =
pd.merge(df_product[df_product['Exited']==0],df_product[df_product['Exited']==1],on='NumOfProducts',
how='outer', suffixes=('_inex','_ex'))


df_product.fillna(inplace=True,value=0)

df_product['ratio']= df_product['count_ex']/(df_product['count_inex']+df_product['count_ex'])

df_product['total_customer']=df_product['count_inex']+df_product['count_ex']

df_product
```

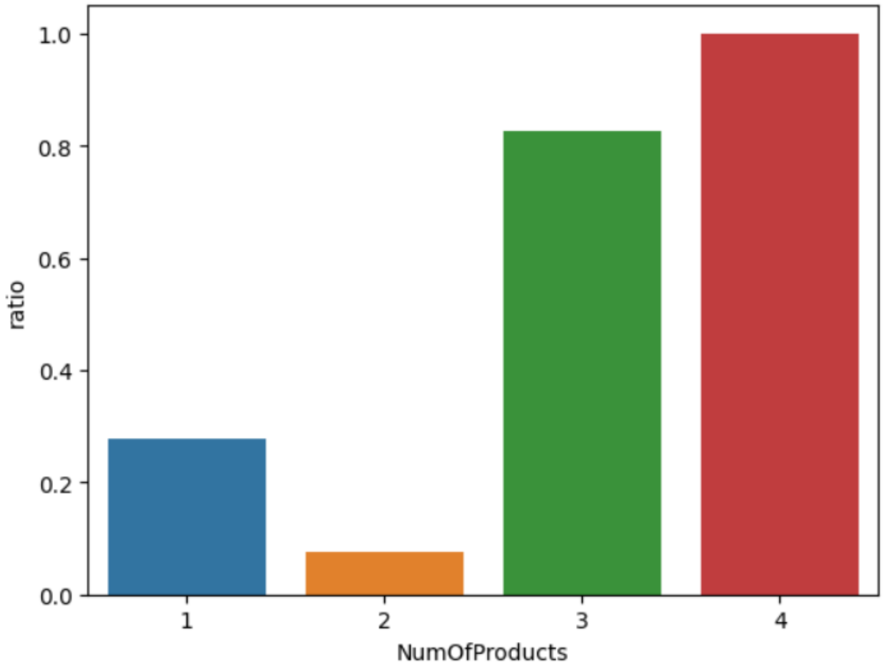


Figure 24: The churn rate by the number of products used(source: authors)

- The highest churn rate is in the group of customers who buy 4 products, but only 1 customer in this group
- Most customers are in the group using the most 1 product (5084), a little lower is the group using 2 products (4590).

```
sns.countplot(data=df,x='NumOfProducts',hue='Gender')
```

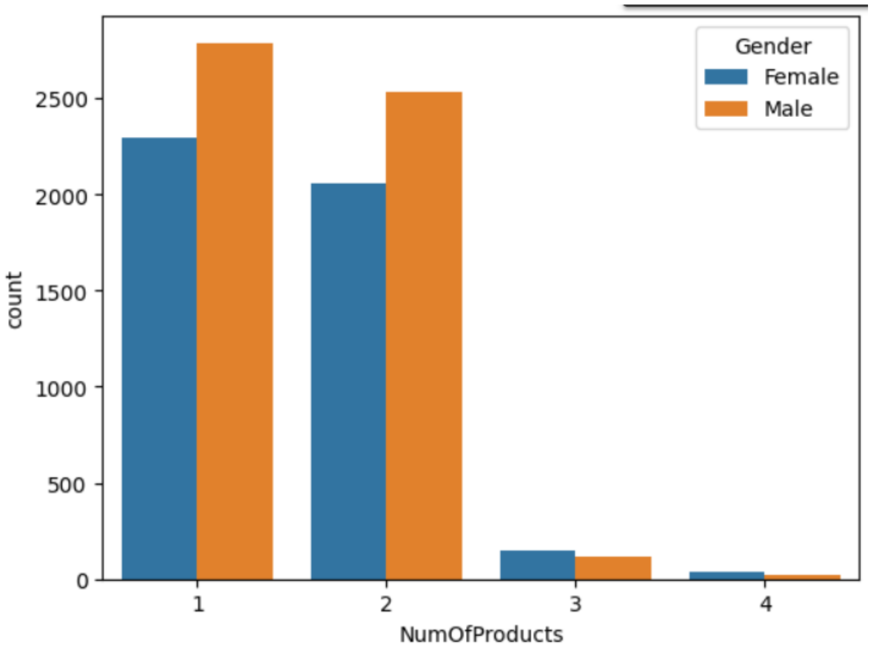


Figure 25: Number of people using 4 Banking products by gender(source: authors)

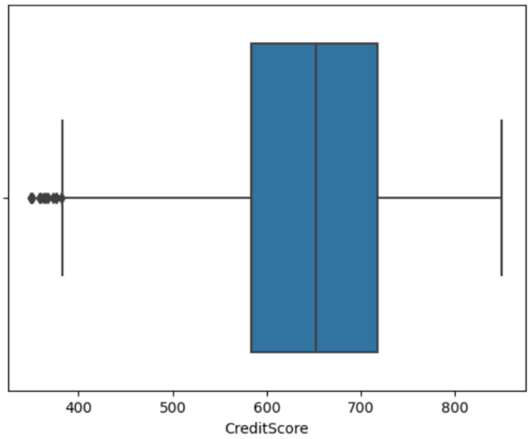
⇒ There are more women buying 1,2 bank products than women

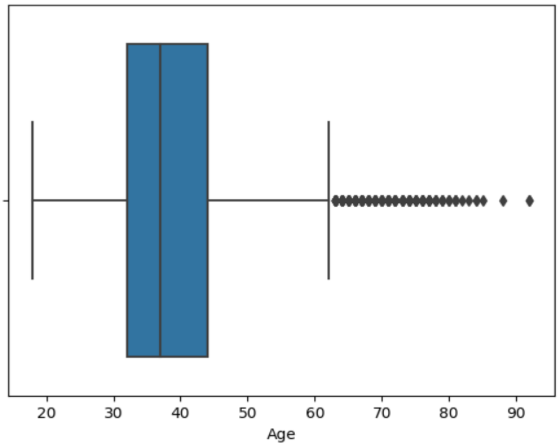
3.5. Data Preprocessing

3.5.1 Detecting and Handling Outliers

- Plot the boxplot of Numerical variables:

```
outlier_plot = ["CreditScore", "Age", "Tenure", "Balance", "NumOfProducts", "EstimatedSalary"]
for i in outlier_plot:
  sns.boxplot(x= df[i])
  plt.show()
```





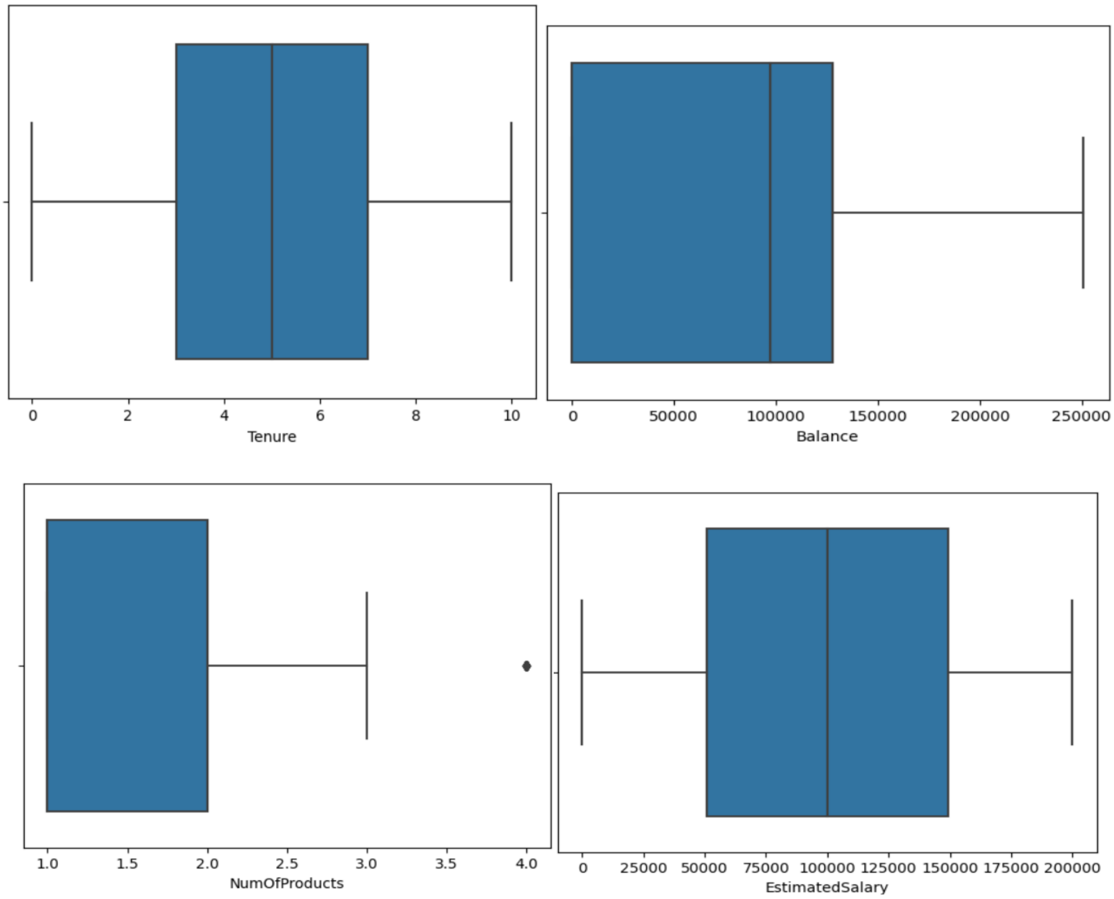


Figure 26: Plot the boxplot of Numerical variables(source: authors)

- Handling outlier:

```
df_copy = df.copy()
```

- Find outlier:

```
def detect_outlier(col):

 first_qrt = np.quantile(df_copy[col], 0.25)

 third_qrt = np.quantile(df_copy[col], 0.75)

 lower_whisker = first_qrt -(third_qrt -first_qrt)*1.5
 print('lower_whisker:',lower_whisker)
 upper_whisker = third_qrt + (third_qrt -first_qrt)*1.5
 print('upper whisker:',upper_whisker)

 outlier = df_copy.loc[(df_copy[col] < lower_whisker) (df_copy[col] > upper_whisker)]
 print('Number of outlier:',len(outlier))
 print('% outliers:',len(outlier)*100/len(df_copy),'%')
```

- Draw the boxplot and find the upper/lower bounds to prepare the outlier using scipy.stats:

```
def boxplot(col):
 plt.figure(figsize=(8,5))
 sns.boxplot(y=col, data=df_copy)
```

- Outlier handling with scipy.stats:

```
df_cleaned = df_copy.copy()
import scipy.stats
def new_df(col, lower_lim, upper_lim):
 df_cleaned[col]=scipy.stats.mstats.winsorize(df_cleaned[col], limits = [lower_lim, upper_lim])
```

- Credit Score (see variables after removing Outlier):

```
detect_outlier('CreditScore')
```

```
new_df('CreditScore', 0.0015,0)
```

```
plt.figure(figsize=(8,5))
sns.boxplot(y='CreditScore', data=df_cleaned)
```

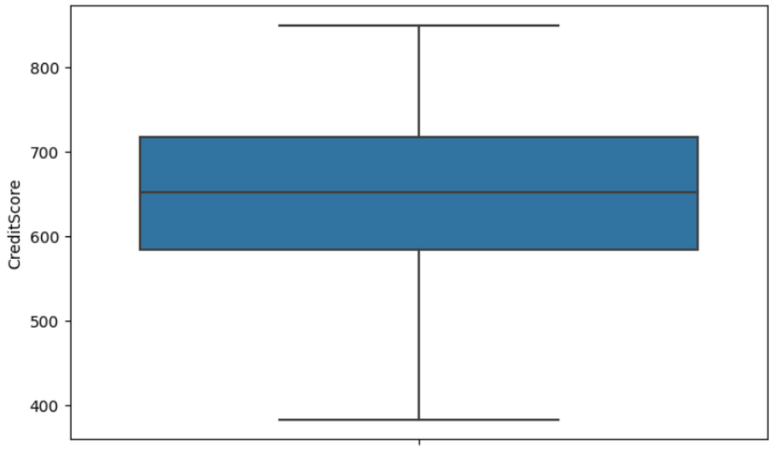


Figure 27: Plot the boxplot of Credit score variables after remove(source: authors)

- Age (see variables after removing Outlier):

```
detect_outlier('Age')
```

```
new_df('Age', 0,0.04)
```

```
plt.figure(figsize=(8,5))
sns.boxplot(y='Age', data=df_cleaned)
```

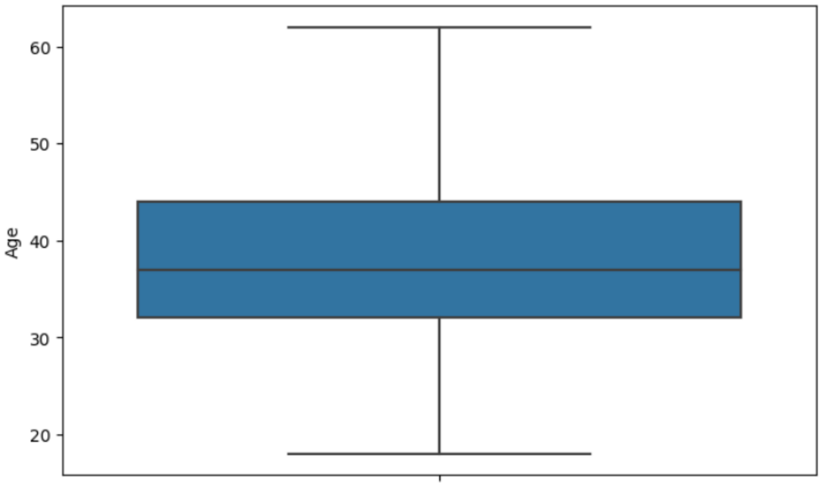


Figure 28:Plot the boxplot of Age variables after remove(source: authors)

- NumofProducts (see variables after removing Outlier):

```
detect_outlier('NumOfProducts')
```

```
new_df('NumOfProducts', 0,0.007)
```

```
plt.figure(figsize=(8,5))
sns.boxplot(y='NumOfProducts', data=df_cleaned)
```

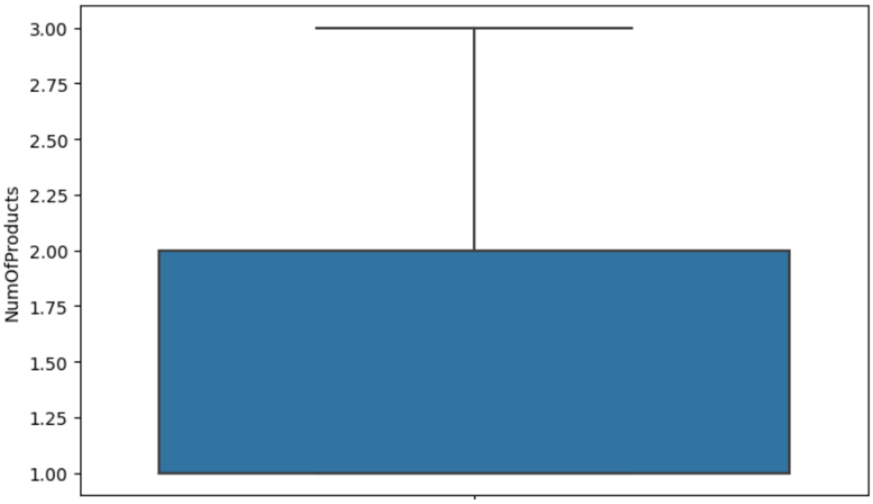
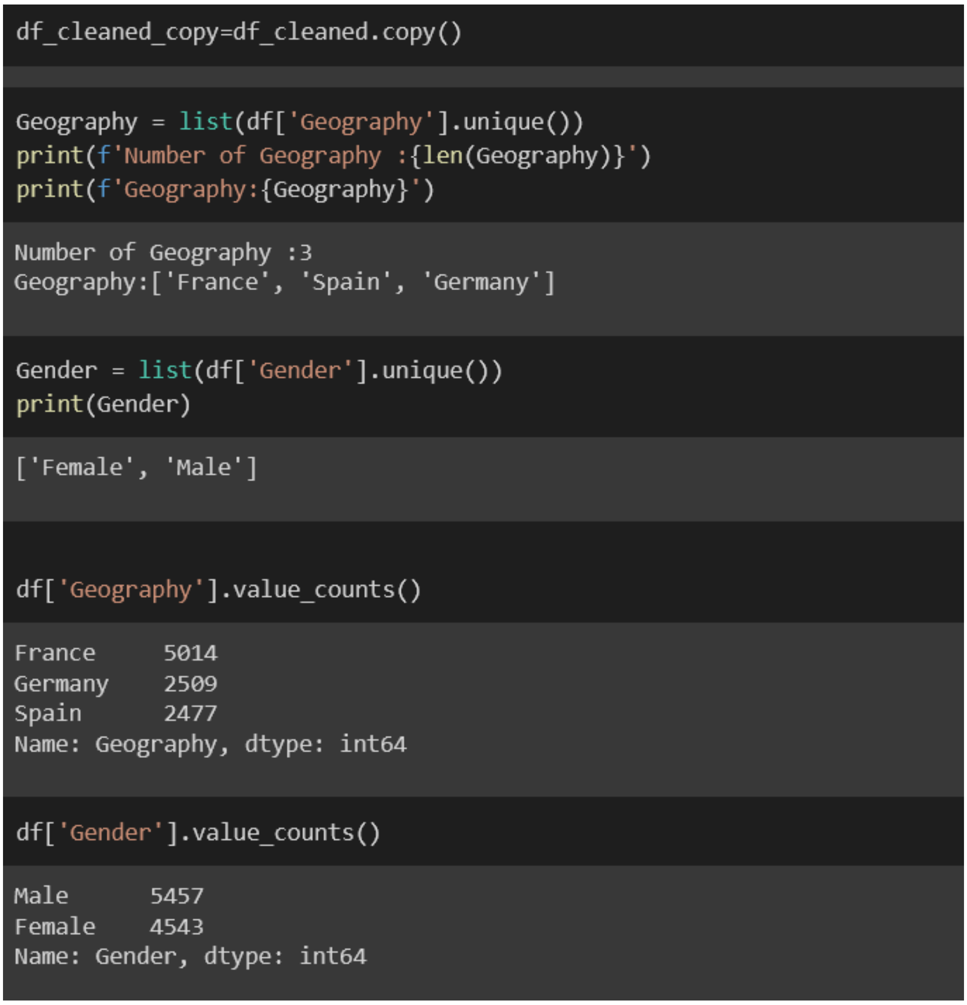


Figure 29:Plot the boxplot of NumofProduct variables after remove(source: authors)

3.5.2 Encode

Encode 2 variables: Geography and Gender*



```
df_cleaned_copy=df_cleaned.copy()
```

```
Geography = list(df['Geography'].unique())
print(f'Number of Geography :{len(Geography)}')
print(f'Geography:{Geography}')
```

```
Number of Geography :3
Geography:['France', 'Spain', 'Germany']
```

```
Gender = list(df['Gender'].unique())
print(Gender)
```

```
['Female', 'Male']
```

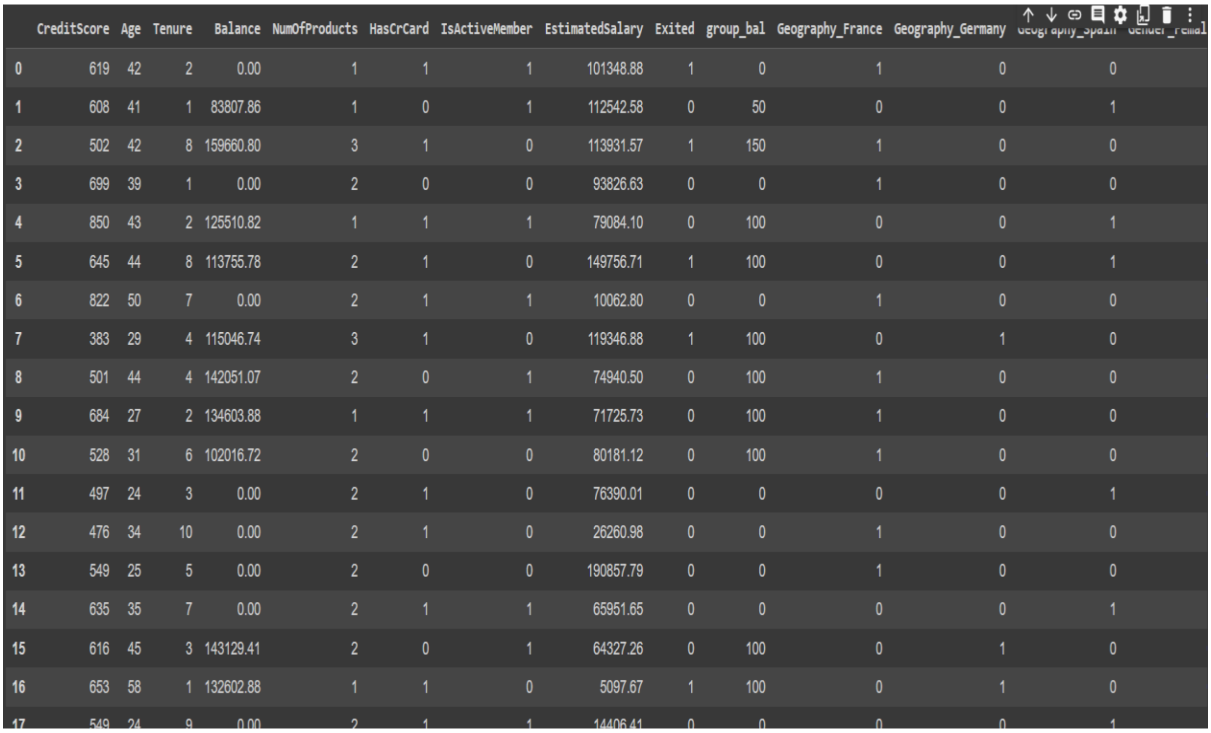
```
df['Geography'].value_counts()
```

```
France     5014
Germany    2509
Spain      2477
Name: Geography, dtype: int64
```

```
df['Gender'].value_counts()
```

```
Male      5457
Female    4543
Name: Gender, dtype: int64
```

```
one_hot_encoded_data = pd.get_dummies(df_cleaned_copy, columns = ['Geography', 'Gender'])
one_hot_encoded_data.head(20)
```



	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	group_bal	Geography_France	Geography_Germany	Geography_Spain	Gender_Femal
0	619	42	2	0.00	1	1	1	101348.88	1	0	1	0	0	
1	608	41	1	83807.86	1	0	1	112542.58	0	50	0	0	1	
2	502	42	8	159660.80	3	1	0	113931.57	1	150	1	0	0	
3	699	39	1	0.00	2	0	0	93826.63	0	0	1	0	0	
4	850	43	2	125510.82	1	1	1	79084.10	0	100	0	0	1	
5	645	44	8	113755.78	2	1	0	149756.71	1	100	0	0	1	
6	822	50	7	0.00	2	1	1	10062.80	0	0	1	0	0	
7	383	29	4	115046.74	3	1	0	119346.88	1	100	0	1	0	
8	501	44	4	142051.07	2	0	1	74940.50	0	100	1	0	0	
9	684	27	2	134603.88	1	1	1	71725.73	0	100	1	0	0	
10	528	31	6	102016.72	2	0	0	80181.12	0	100	1	0	0	
11	497	24	3	0.00	2	1	0	76390.01	0	0	0	0	1	
12	476	34	10	0.00	2	1	0	26260.98	0	0	1	0	0	
13	549	25	5	0.00	2	0	0	190857.79	0	0	1	0	0	
14	635	35	7	0.00	2	1	1	65951.65	0	0	0	0	1	
15	616	45	3	143129.41	2	0	1	64327.26	0	100	0	1	0	
16	653	58	1	132602.88	1	1	0	5097.67	1	100	0	1	0	
17	549	24	9	0.00	2	1	1	14406.41	0	0	0	0	1	

Encoding 2 columns of data: Gender and Geography, we will convert the values in these two columns into corresponding numerical values. This helps the model understand these data columns and use them to train and predict results. There are many different encoding methods such as One-Hot Encoding, Ordinal Encoding, Target Encoding, Binary Encoding, etc. Depending on the nature of the data and the purpose of the problem, programmers can use the appropriate encoding method to process data and improve the performance of the model. Here, we use One-Hot Encoding.

3.5.3 Scaling Data

```
df[['CreditScore','EstimatedSalary']].plot(kind='box')
```

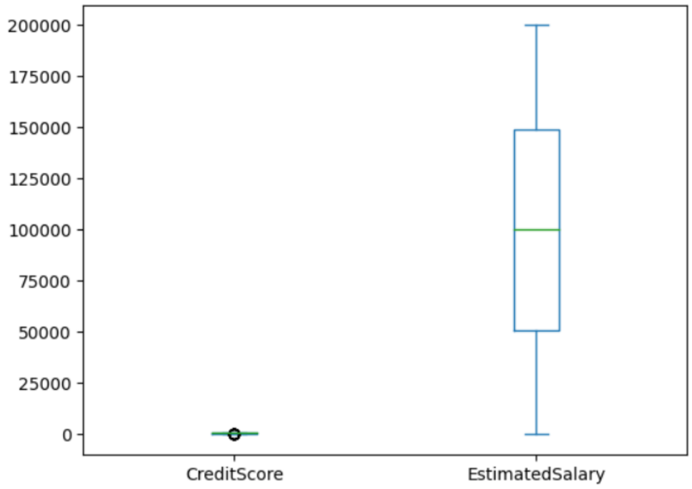


Figure 30: Display Scaling data of CreditScore and EstimatedSalary variables

```
from sklearn.preprocessing import MinMaxScaler
mms=MinMaxScaler()
mms.fit(df2)
data_mms =mms.transform(df2)
```

Convert to Dataframe:

```
data_mms =pd.DataFrame(data_mms,
columns=['CreditScore','EstimatedSalary','Age','Tenure','Balance','NumOfProducts','HasCrCard','IsActiveMember','Exited','group_bal','Geography_France','Geography_Germany','Geography_Spain','Gender_Female','Gender_Male'])
data_mms.head()
```

```
data_mms[['CreditScore','EstimatedSalary']].plot(kind='box')


plt.title('data after max_min normalization')
```

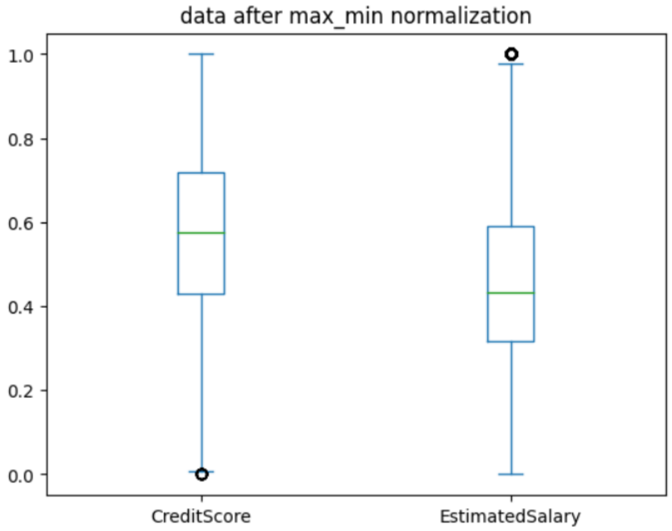


Figure 31: Data after normalization

Representing in the same interval [0,1] is easier to parse

- Correlation

```
df_cleaned =df_cleaned.drop(columns=['group_bal'])

df_cleaned
```

```
corr =df_cleaned.corr()

corr
```

```
sns.heatmap(corr)
```

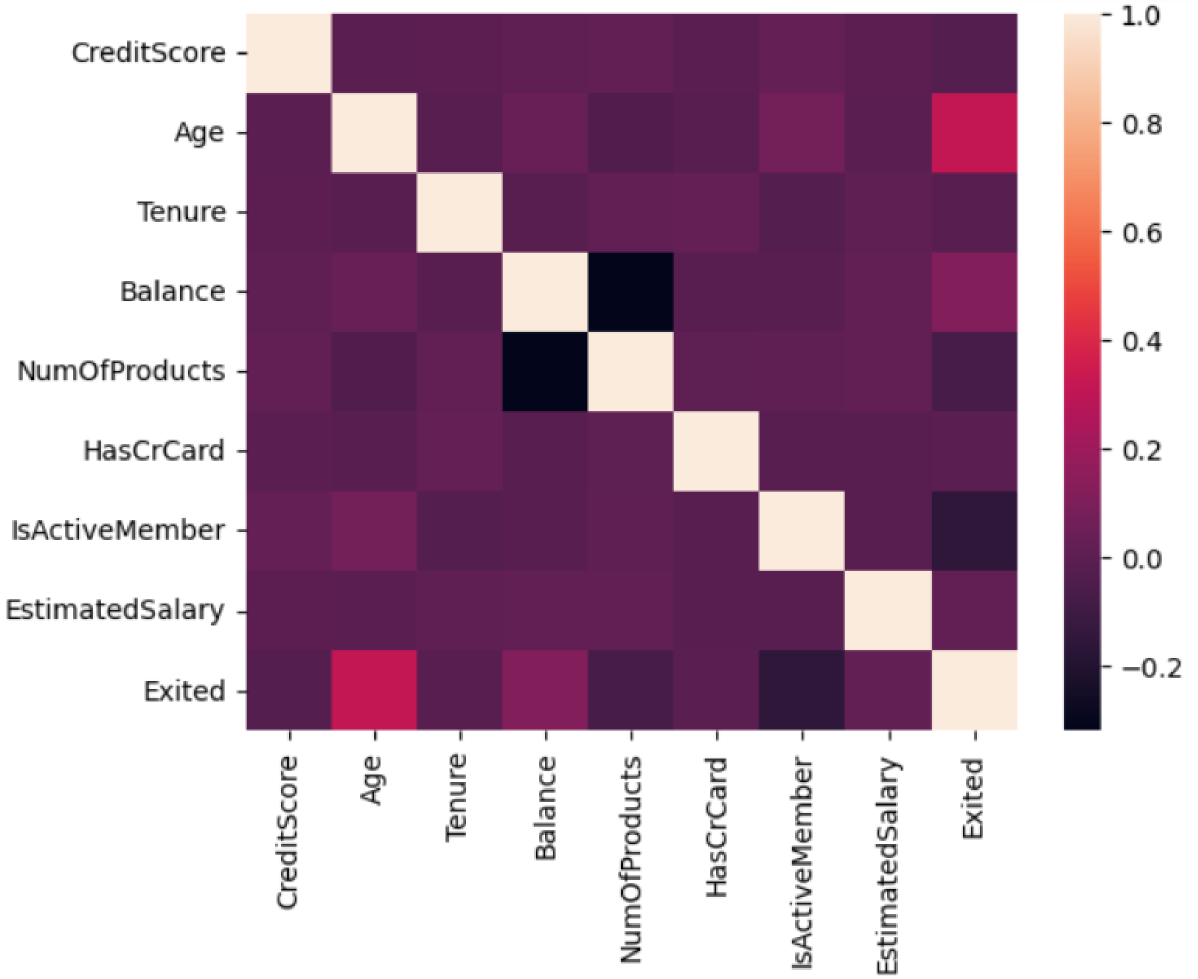


Figure 32: Correlation between variables after scaling(source: authors)

3.5.4 Feature selection

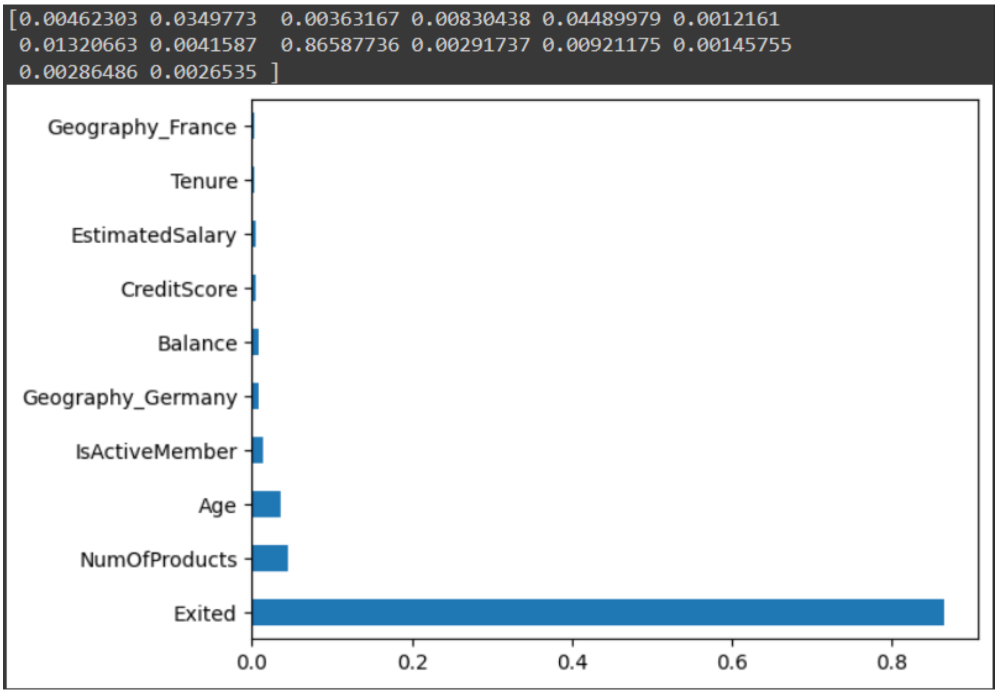
Find feature important

```
X2 = df2.iloc[:,0:14] #column data from 0 - 14
y2=df2.iloc[:,8]
from sklearn.ensemble import ExtraTreesClassifier
```

ExtraTreesClassifier: applies to algorithms of the form Tree, eg: Decision Tree, Random Forests, XGBoots

```
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X2,y2)
print(model.feature_importances_)
feat_importances = pd.Series(model.feature_importances_,index=X2.columns)
```

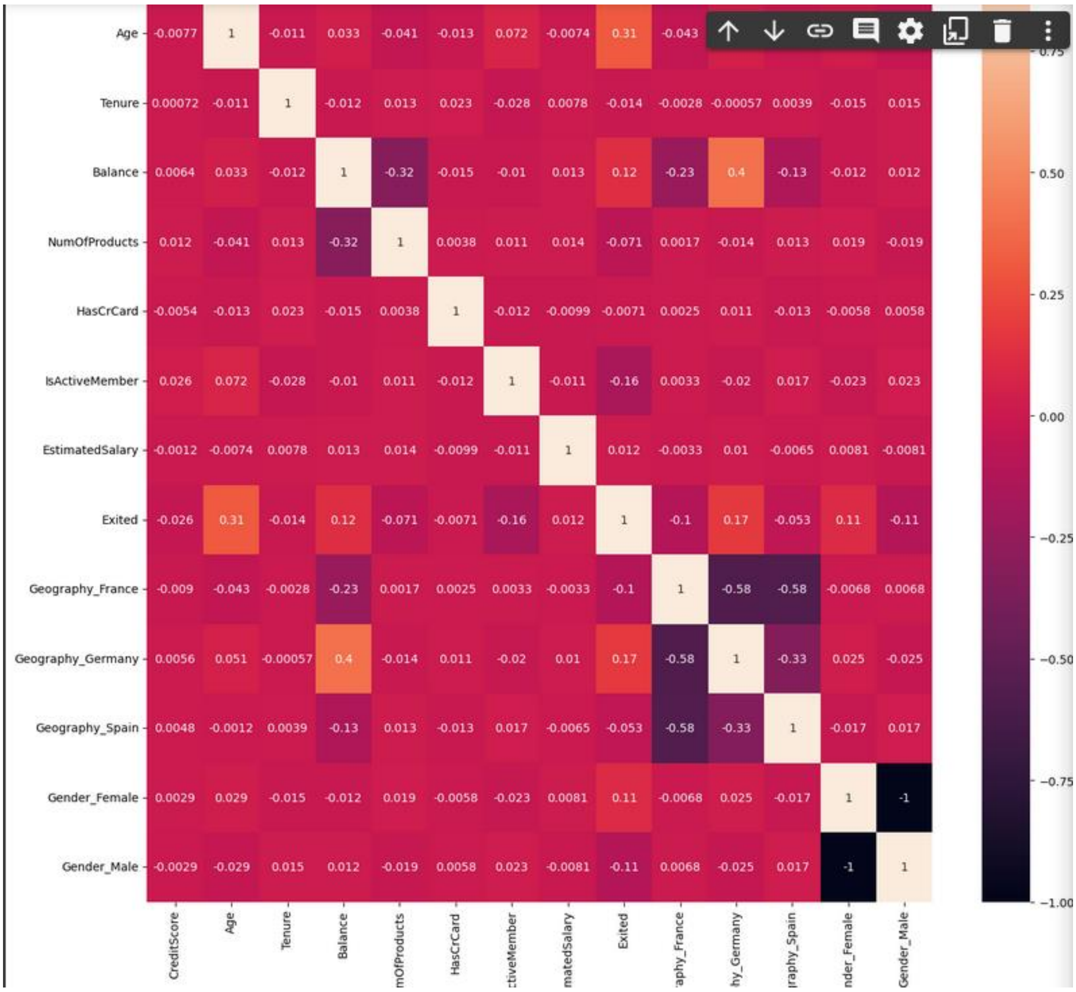
```
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
```



```
[0.00462303 0.0349773  0.00363167 0.00830438 0.04489979 0.0012161
 0.01320663 0.0041587  0.86587736 0.00291737 0.00921175 0.00145755
 0.00286486 0.0026535 ]
```

Figure 33: Correlation between variables after feature selection(source: authors)

```
import seaborn as sns
X3 = df2.iloc[:,0:14] #cột dữ liệu từ 0 - 14
y3=df2.iloc[:,8]
corrmat = df2.corr()
top_corr_features =corrmat.index
plt.figure(figsize=(15,15))
g=sns.heatmap(df2[top_corr_features].corr(),annot=True)
```



	CreditScore	Age	Tenure	Balance	mOfProducts	HasCrCard	ctiveMember	matedSalary	Exited	aphy_France	hy_Germany	jraphy_Spain	nder_Female	Gender_Male
Age	-0.0077	1	-0.011	0.033	-0.041	-0.013	0.072	-0.0074	0.31	-0.043				
Tenure	0.00072	-0.011	1	-0.012	0.013	0.023	-0.028	0.0078	-0.014	-0.0028	-0.00057	0.0039	-0.015	0.015
Balance	0.0064	0.033	-0.012	1	-0.32	-0.015	-0.01	0.013	0.12	-0.23	0.4	-0.13	-0.012	0.012
NumOfProducts	0.012	-0.041	0.013	-0.32	1	0.0038	0.011	0.014	-0.071	0.0017	-0.014	0.013	0.019	-0.019
HasCrCard	-0.0054	-0.013	0.023	-0.015	0.0038	1	-0.012	-0.0099	-0.0071	0.0025	0.011	-0.013	-0.0058	0.0058
IsActiveMember	0.026	0.072	-0.028	-0.01	0.011	-0.012	1	-0.011	-0.16	0.0033	-0.02	0.017	-0.023	0.023
EstimatedSalary	-0.0012	-0.0074	0.0078	0.013	0.014	-0.0099	-0.011	1	0.012	-0.0033	0.01	-0.0065	0.0081	-0.0081
Exited	-0.026	0.31	-0.014	0.12	-0.071	-0.0071	-0.16	0.012	1	-0.1	0.17	-0.053	0.11	-0.11
Geography_France	-0.009	-0.043	-0.0028	-0.23	0.0017	0.0025	0.0033	-0.0033	-0.1	1	-0.58	-0.58	-0.0068	0.0068
Geography_Germany	0.0056	0.051	-0.00057	0.4	-0.014	0.011	-0.02	0.01	0.17	-0.58	1	-0.33	0.025	-0.025
Geography_Spain	0.0048	-0.0012	0.0039	-0.13	0.013	-0.013	0.017	-0.0065	-0.053	-0.58	-0.33	1	-0.017	0.017
Gender_Female	0.0029	0.029	-0.015	-0.012	0.019	-0.0058	-0.023	0.0081	0.11	-0.0068	0.025	-0.017	1	-1
Gender_Male	-0.0029	-0.029	0.015	0.012	-0.019	0.0058	0.023	-0.0081	-0.11	0.0068	-0.025	0.017	-1	1

Figure 34: Correlation between variables after feature selection(source: authors)

CHAPTER 4: EXPERIMENTAL AND MODEL EVALUATION

In this chapter, machine learning approaches for experiments including Logistics Regression, Decision Trees, SVC, Random Forest, and XGBoost are presented. These methods are then compared and evaluated to determine which model is best for the given data set. The experimental findings showed that the model's performance matched the data set after preprocessing when looking at the evaluations that were visualized after the model was applied and analyzed.

4.1 Building and evaluating model

4.1.1 Logistics Regression

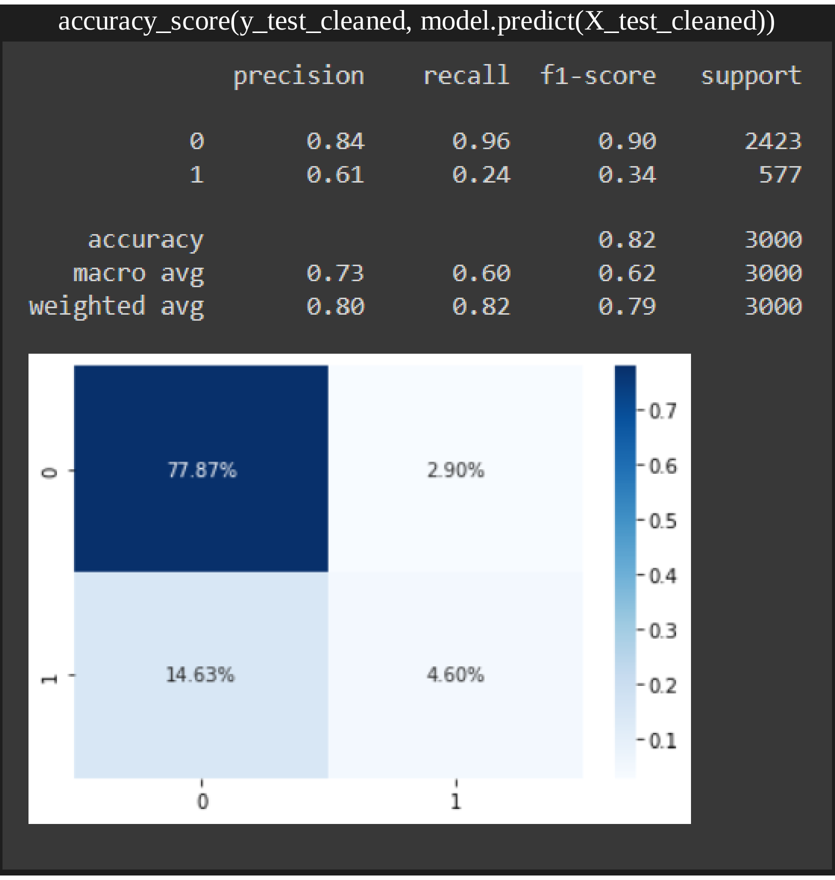
Import the LogisticRegressionCl() model from the sklearn library of python. Then, use the .fit() function to model learning from training data results used in training. And the .predict() function to predict the results of the test dataset.

```
X_train_cleaned, X_test_cleaned, y_train_cleaned, y_test_cleaned = train_test_split(X,y, test_size = 0.3,
random_state = 20)


model = LogisticRegression()


model.fit(X_train_cleaned, y_train_cleaned)
```

```
accuracy_score(y_test_cleaned, model.predict(X_test_cleaned))
```



```
              precision    recall  f1-score   support

           0       0.84      0.96      0.90      2423
           1       0.61      0.24      0.34       577

    accuracy                           0.82      3000
   macro avg       0.73      0.60      0.62      3000
weighted avg       0.80      0.82      0.79      3000
```

Figure 35: Logistics Regression Model Evaluation (source: authors)

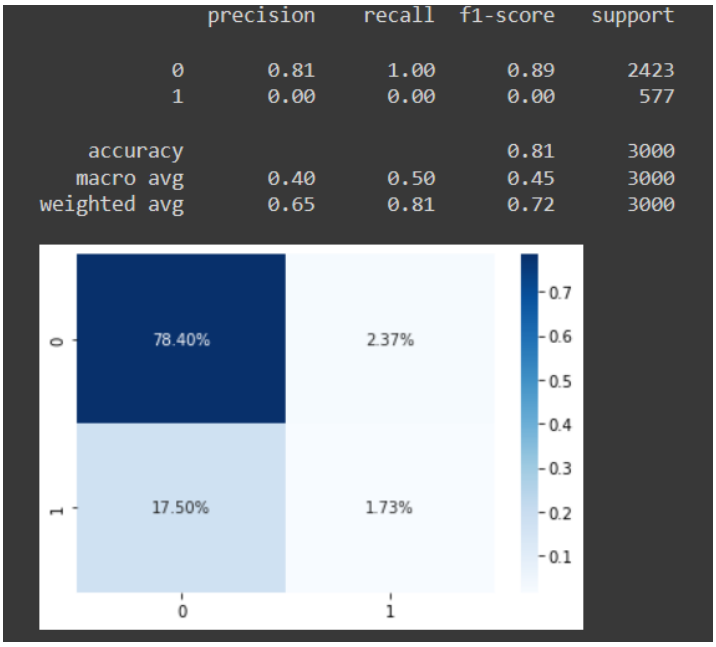
⇒ Accuracy prediction results for datasets: 82%

4.1.2 Decision tree

Import the DecisionTreeClassifier() model from the sklearn library of python. Then, use the .fit() function to model learning from training data used in training. And the .predict() function to predict the results of the test dataset.

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth=1)
clf.fit(X_train_cleaned, y_train_cleaned)
accuracy_score(y_test_cleaned, clf.predict(X_test_cleaned))
```

Then use the confusion matrix to evaluate model performance:



```
              precision    recall  f1-score   support

           0       0.81      1.00      0.89      2423
           1       0.00      0.00      0.00       577

    accuracy                           0.81      3000
   macro avg       0.40      0.50      0.45      3000
weighted avg       0.65      0.81      0.72      3000
```

Figure 36: Decision Tree Model Evaluation (source: authors)

⇒ Accuracy prediction results for datasets: 81%

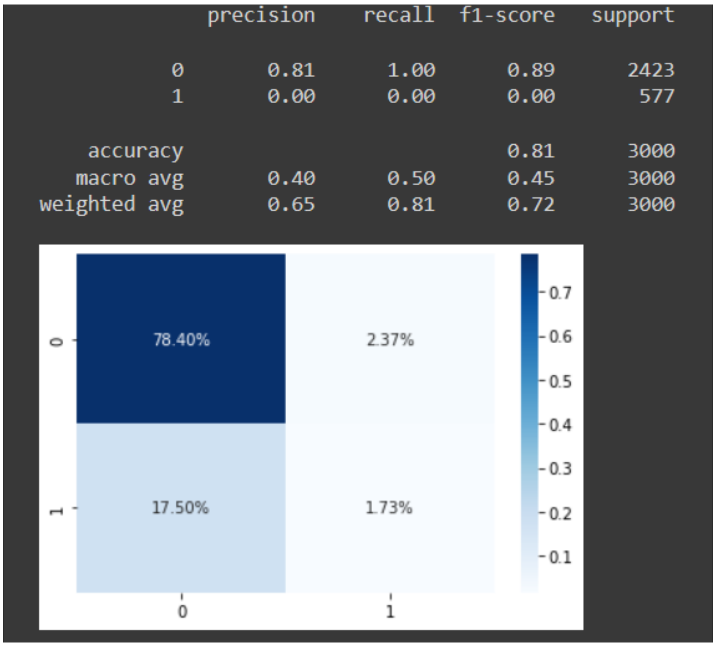
4.1.3 SVC

Import the DecisionTreeClassifier() model from the sklearn library of python. Then, use the .fit() function to model learning from training data used in training. And the .predict() function to predict the results of the test dataset.

```
svm = SVC()
svm.fit(X_train_cleaned,y_train_cleaned)

accuracy_score(y_test_cleaned, svm.predict(X_test_cleaned))
```

Then use the confusion matrix to evaluate model performance:



```
              precision    recall  f1-score   support

           0       0.81      1.00      0.89      2423
           1       0.00      0.00      0.00       577

    accuracy                           0.81      3000
   macro avg       0.40      0.50      0.45      3000
weighted avg       0.65      0.81      0.72      3000
```

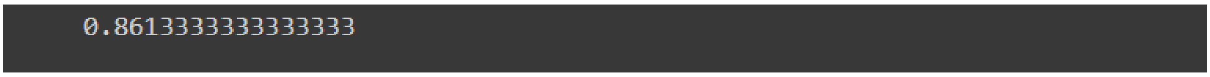
Figure 37: Support vector machine evaluation (source: authors)

⇒ Accuracy prediction results for datasets: 81%

4.1.4 Random Fores

Import the RandomForest Classifier() model from the sklearn library of python. Then, use the .fit() function to model learning from training data used in training. And the .predict() function to predict the results of the test dataset.

```
rd_fr =RandomForestClassifier()
rd_fr.fit(X_train_cleaned,y_train_cleaned)
accuracy_score(y_test_cleaned,rd_fr.predict(X_test_cleaned))
```



```
0.8613333333333333
```

⇒ Accuracy prediction results for datasets: 86%

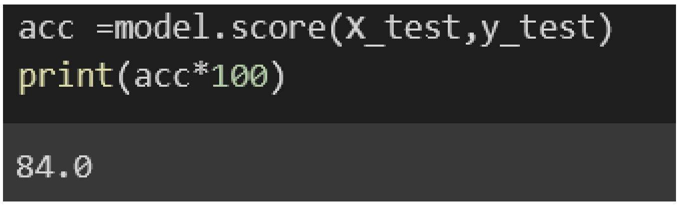
4.1.5 XGboost

Import the**XGboost**Classifier() model from the sklearn library of python. Then, use the .fit() function to model learning from training data used in training. And the .predict() function to predict the results of the test dataset.

```
import xgboost as xgb


model_xgb = xgb.XGBClassifier(random_state=42,n_estimators=100)
model_xgb.fit(X_train,y_train)
```

⇒ Accuracy prediction results for datasets: 81%



```
acc =model.score(X_test,y_test)
print(acc*100)
```

```
84.0
```

4.1.6 GradientBoosting

Import the **GradientBoosting**Classifier() model from the sklearn library of python. Then, use the .fit() function to model learning from training data used in training. And the .predict() function to predict the results of the test dataset.

```
model = GradientBoostingClassifier(learning_rate=0.07,n_estimators=200,max_depth=6)


model.fit(x_train, y_train)


y_pred = model.predict(x_train)
y_pred2 = model.predict(x_val)
```

Then use the confusion matrix, roc_auc curve to evaluate model performance:

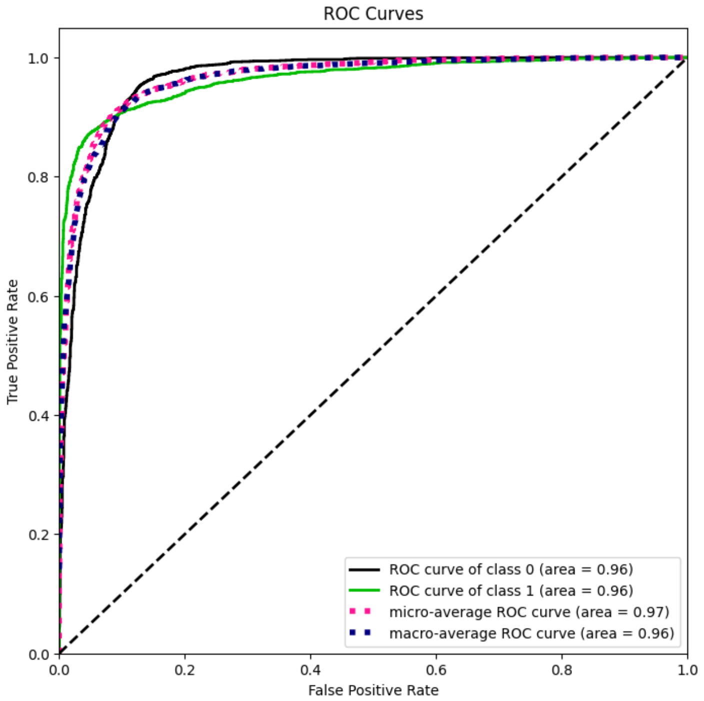
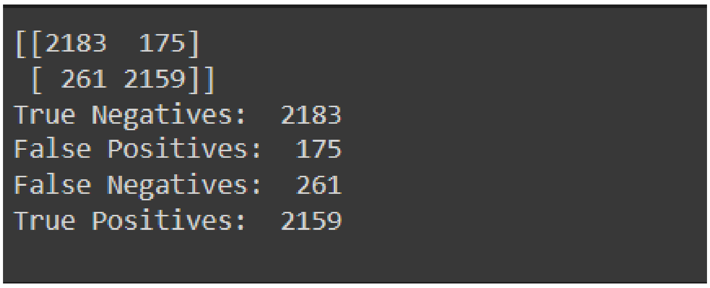
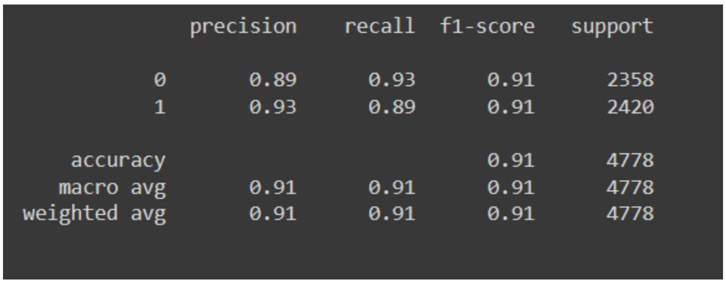


Figure 38: Roc_auc curve to evaluate model performance(source: authors)



```
[[2183  175]
 [ 261 2159]]
True Negatives:  2183
False Positives:  175
False Negatives:  261
True Positives:  2159
```



```
              precision    recall  f1-score   support

           0       0.89      0.93      0.91      2358
           1       0.93      0.89      0.91      2420

    accuracy                           0.91      4778
   macro avg       0.91      0.91      0.91      4778
weighted avg       0.91      0.91      0.91      4778
```

⇒ Accuracy prediction results for datasets: 91%

CHAPTER 5: CONCLUSION AND FUTURE WORKS

Describe the outcomes obtained, the difficulties faced during project execution, and propose some suggestions for future development.

5.1 Results

After the research process, the abandonment rate prediction model is proposed and tested on the data set of a bank and the evaluation results are given with a high degree of accurate (91%). The data set is collected from nearly 10,000 customers with 14 observed variables, the assessment is done on a large scale with a fairly high rate generalization level.

Test and compare 06 machine learning methods ((Logistic Regression, Gradient Boosting Classifier, SVC, Decision Tree, Random Forest, XGBoos) reflect the properties and characteristics of each machine learning method with Research data set through the index of confusion matrix such as Accuracy, Recall...

For the collected data set, GradientBoostingClassifier is the most efficient method with the highest accuracy (approximate 91%) and out of 06 machine learning methods,the recall indexes (0.93; 0.89) and f1-score (0.91; 0.91) give quite positive values.

Using ROC index, the evaluation result is good (0.96).

The test results on the train set, the test gives relatively good results with the Gradient Boosting Classifier method: (0.91 for the train set and 0.9 for the test set), the possibility of overfitting is very low.

For the collected data set, GradientBoostingClassifier is the most efficient method with the highest accuracy (approximate 91%) and out of 06 machine learning methods.

Predictive analysis of customer abandonment on each object group will help analysts find trends and predictions, thereby offering appropriate solutions, contributing to retaining potential customers.

Furthermore, through data visualization charts and reports in EDA process, enterprises can understand and measure influencing factors from different angles. this one has a significant impact on the business and can be :

- Understand and develop customer strategies.
- Optimize costs and resources.
- Improve customer experience.
- Increase reliability and effectiveness of the model.

5.2 Limitations

Not suitable for new situations: The churn rate prediction model is built based on historical data and current customer characteristics. If there are changes in business situations or markets, the model may not fully reflect new factors.

Ineffective in individual forecasting: The churn rate prediction model is often built to predict at the group level of customers, rather than individuals in detail. This could reduce the effectiveness of the model in retaining important customers.

Model is affected by noise: Data may contain noise or inaccurate information, which can affect the effectiveness of the churn rate prediction model.

The duration of the project is limited under the large volume of machine learning,knowledge of members has many gaps as well as not much practical experience.

5.3 Future Works

Handling imbalanced data: Using techniques such as oversampling or undersampling to balance the data and improve the accuracy of the prediction model.

Adding important factors: Adding relevant factors to the dataset to improve the accuracy of the model. For example, incorporating information about customer's business activities to better understand their needs.

Updating the model regularly: Updating and checking the model regularly to adapt to new changes in business situations or markets.

Using individual prediction models: Using individual prediction models to focus on each customer to retain important customers.

Handling noise: Using noise handling techniques to remove noisy values from the data and improve the accuracy of the model.

Using various performance evaluation methods: Using multiple performance evaluation methods to gain a comprehensive and accurate understanding of the performance of the model.

Last but not least, continue to learn, analyze and collect factors affecting customer churn rate when using services of different banks, increasing the accuracy of the model

References

Bhandari, P. (2022) *Missing data: Types, explanation, & imputation*, *Scribbr*. Available at: https://www.scribbr.com/statistics/missing-data/ (Accessed: April 7, 2023).

(no date) *Discover Colleges, Courses & exams for Higher Education in India*. Available at: https://www.shiksha.com/online-courses/articles/handling-missing-values-beginners-tutorial/ (Accessed: April 7, 2023).

Tiền xử LÝ dữ Liệu Trong Machine Learning, ví DỤ cụ thể. (2021) *Web888 chia sẻ kiến thức lập trình, kinh doanh, mmo*. Available at: https://web888.vn/tien-xu-ly-du-lieu-trong-machine-learning-vi-du-cu-the/ (Accessed: April 7, 2023).

Welcome to vimentor! (no date) *Chi tiết bài học Tiền xử lý dữ liệu trong lĩnh vực học máy (Phần 3)*. Available at: https://vimentor.com/en/lesson/tien-xu-ly-du-lieu-trong-linh-vuc-hoc-may-phan-3 (Accessed: April 7, 2023).

Deep Ai Khanhblog (no date) *11.1. Feature Engineering - Deep AI KhanhBlog*. Available at: https://phamdinhkhanh.github.io/deepai-book/ch_ml/FeatureEngineering.html (Accessed: April 7, 2023).

Regression trees (2015) *solver*. Available at: https://www.solver.com/regression-trees (Accessed: April 7, 2023).

Vu, T. (2018) *Bài 34: Decision trees (1): Iterative Dichotomiser 3*, *Tiep Vu's blog*. Available at: https://machinelearningcoban.com/2018/01/14/id3/ (Accessed: April 7, 2023).

Vu, T. (2018) *Bài 34: Decision trees (1): Iterative Dichotomiser 3*, *Tiep Vu's blog*. Available at: https://machinelearningcoban.com/2018/01/14/id3/ (Accessed: April 7, 2023).

Baodientuvtv (2020) *Quản Trị và Giữ Chân khách Hàng Thời KỲ Khủng Hoảng với công Nghệ Blockchain*, *BAO DIEN TU VTV*. vtv.vn. Available at: https://vtv.vn/cong-nghe/quan-tri-va-giu-chan-khach-hang-thoi-ky-khung-hoang-voi-cong-nghe-blockchain-20200713153053068.htm (Accessed: April 7, 2023).