**Churn Prediction for Bank Customers**

Group 65

Shutong Fan

Xinyu Yang

617-543-0468

857-348-9290

[fan.shut@northeastern.edu](mailto:fan.shut@northeastern.edu)

[yang.xinyu2@northeastern.edu](mailto:yang.xinyu2@northeastern.edu)

**Percentage of Effort Contributed by Student 1: 50% 1**

**Percentage of Effort Contributed by Student 2: 50% 1**

**Signature of Student 1: Shutong Fan 1**

**Signature of Student 2: Xinyu Yang 1**

**Submission Date: 2022/4/22 1**

# 1 Intro

## 1.1 Problem Setting:

As we all know, it is imperative for companies to keep the customer retention in order to develop their business. If we are able to know who would abandon the company, they may provide some loyalty retention in advance. This dataset is aimed to predict whether a bank customer will exit the current bank or not. It contains personal information of the customers and information about the bank products owned by each customer to help to train the models.

## 1.2 Problem Definition:

1. Build a predictive model of bank customer churn prediction.
2. Find which variables lead a customer to make a decision of leaving the company.
3. Distinguish the customers who are loyal to the company.
4. Find out which algorithm has the best performance on this dataset and why.

## 1.3 Data Description:

* Name of dataset: churn.csv
* Sample of variable names: RowNumber, CustomerId, Surname, CreditScore, Geography, Gender, Age, Tenure, etc.
* Number of rows: 10000
* Number of columns: 14

# 2 Data collection and processing

## 2.1 Check available predictors

First of all, by checking our data contains no missing values, we can skip the step for filling NA. The attribute 'Surname', ‘CustomerId’ and 'RowNumber' were removed since they are unnecessary columns as they contain the identifier value for each customer. Then, we can classify all variables as categorical or numerical by type of each variable.

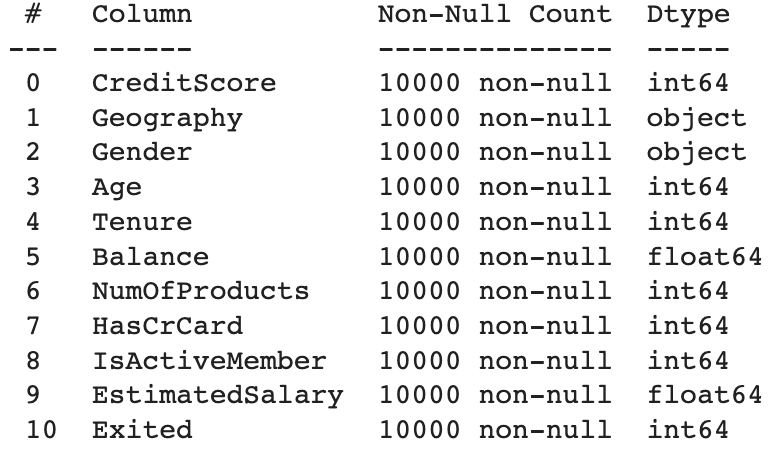


Fig. 2.1 information of predictors

Basically, the data has 10 variables and 1 response (‘Exited’). Variables include 6 numeric variables and 4 categorical variables.

Tab. 2.1 Variables classifications

| **Categorical Variables** | **Numeric Variables** |
| --- | --- |
| 1. Geography  2. Gender  3. HasCrCard  4. IsActiveMember  5. Tenure  6. NumOfProducts | 1. Credit Score  2. Age  3. Balance  4. EstimatedSalary |

## 2.2 Encode categorical predictors

For those categorical variables which need to be encoded, we map those features, such as ‘Geography’ and ‘Gender’, to integers 1-3 as labels to represent the values. Also, the statistical information is as follows:

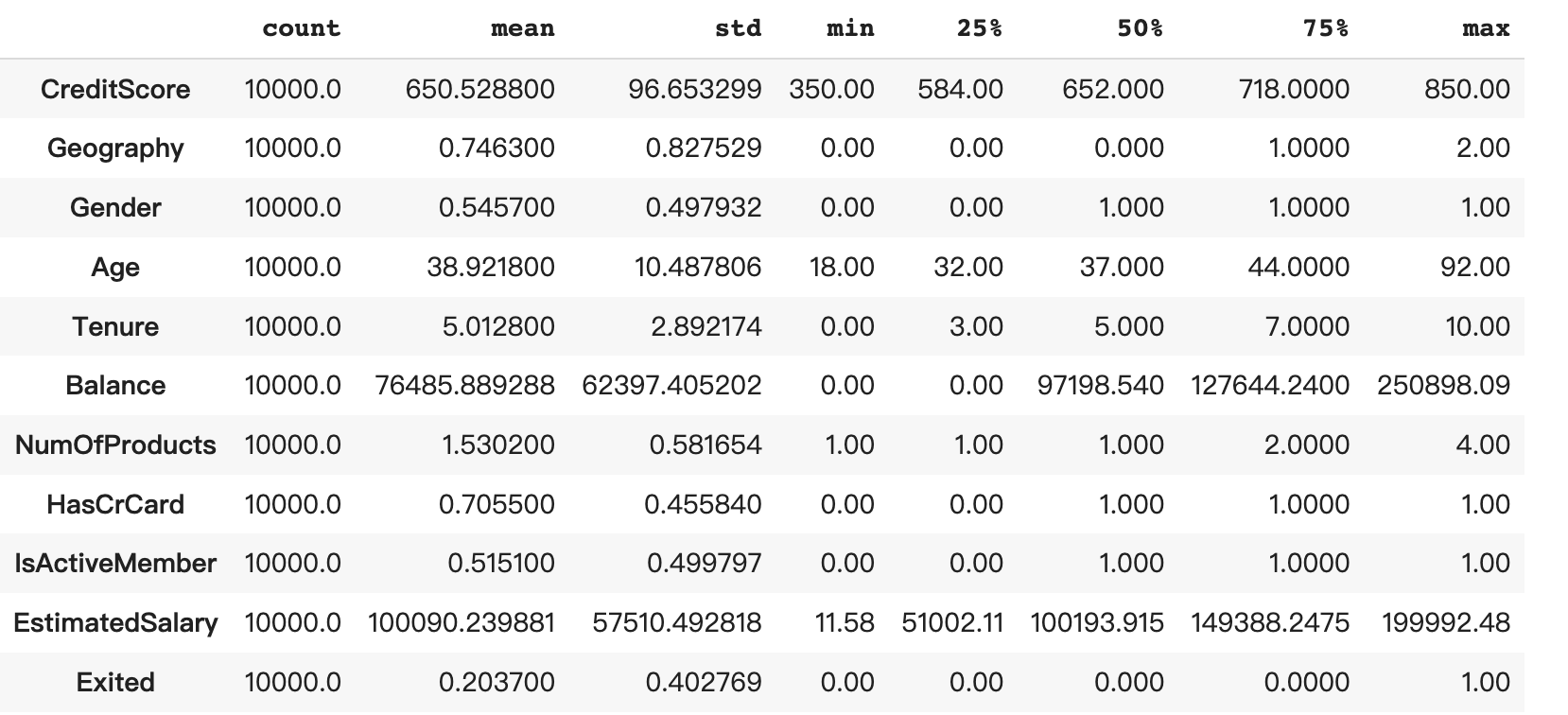


Fig. 2.2 Statistical information of predictors

## 2.3 Data partitioning

With the purpose of training model and model performance evaluation, we partition data into two parts: 75% as training data and 25% as testing data.

## 2.4 Oversampling

Because the interest of class is 1 and it is extremely rare than the other class, we need to oversample data to keep data balance and apply the oversample data to those models that do not automatically adjust data balance.

Tab. 2.4 Oversampling data comparison

| **Class** | **Before** | **After** |
| --- | --- | --- |
| 0 | 5972 | 5972 |
| 1 | 1528 | 5972 |

## 2.5 Scaling

Scaling is one of the most important steps in the data mining process. We use the formula to scale the data.

# 3 Data exploration and visualization

## 3.1 Proportion of target value

Firstly, the pie chart of target value conveys that the exited users are closely 20% in total.

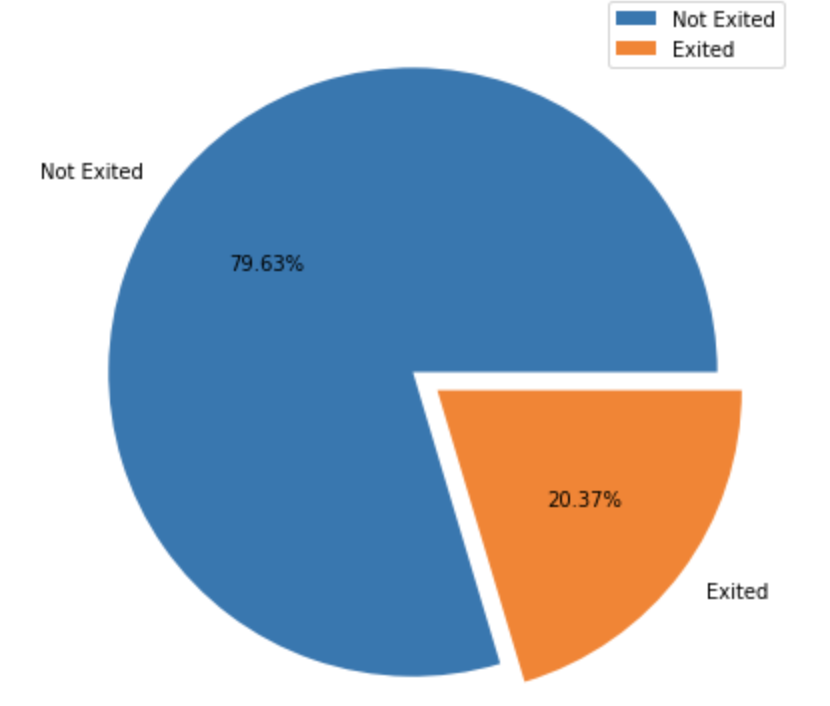


Fig. 3.1.1 Proportion of Customer Churn

## 3.2 Categorical predictors

### 3.2.1 Distribution exploration

It is imperative to separate categorical and numeric variables in the exploratory data analysis part. For all 6 categorical variables, we plot the count plot in order to observe the data originally. We can conclude that most of the categories have more ‘Not Exited’ than ‘Exited’ except when the ‘NumOfProducts’ are 3 and 4. We ought to explore deeply why ‘NumOfProducts’ equals 1 has the lowest exited ratio?

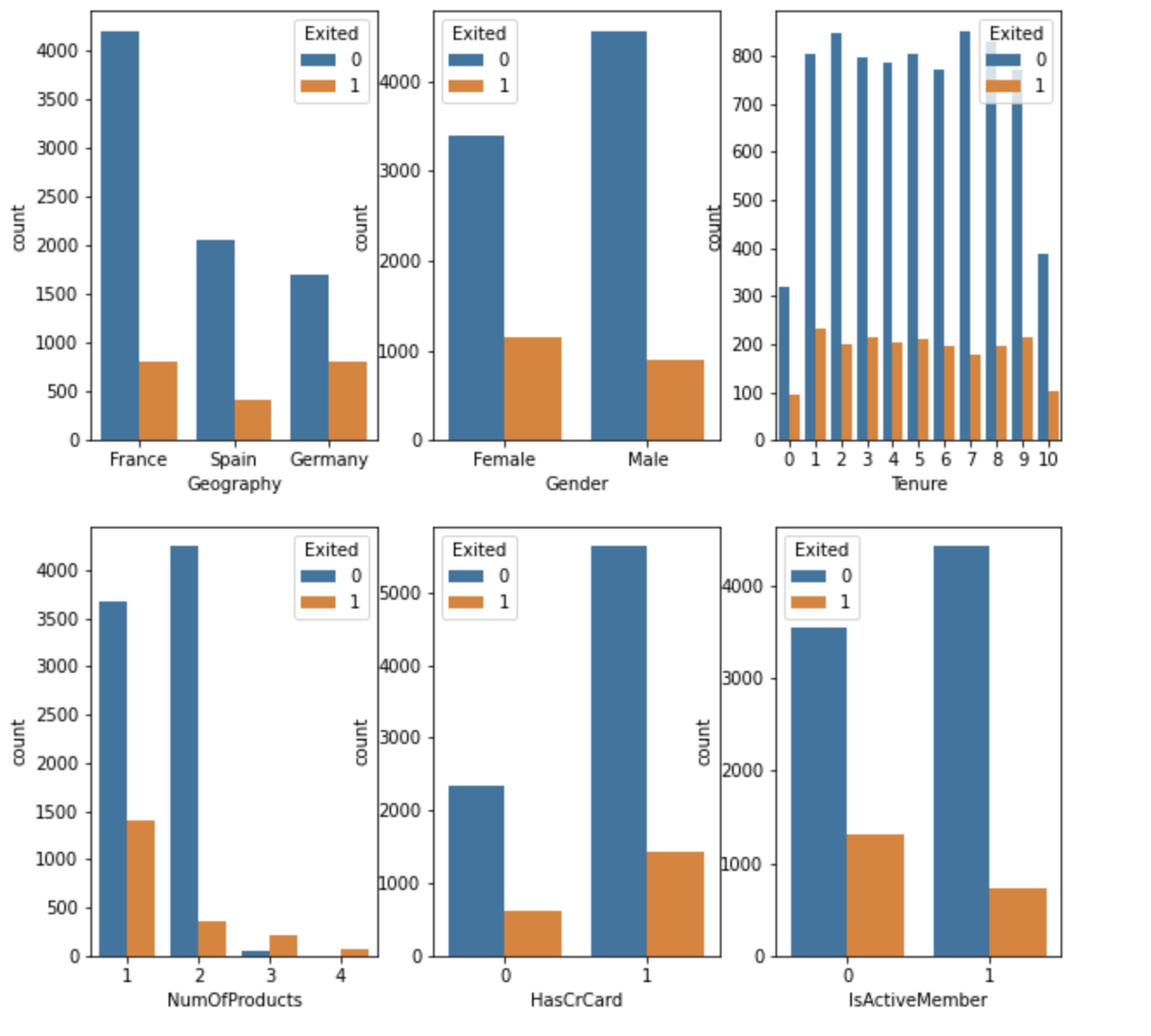


Fig. 3.2.1 Distribution of categorical variables

### 3.2.2 Statistical test

With the purpose of validating the statistical significance between categorical variables and target, the chi-square test method can be applied. The two hypotheses of the statistical test are defined as follows:

* H0: the predictor has no significant correlation with target
* H1: the predictor has significant correlation with target

The output indicates that two predictors are significantly uncorrelated with target (alpha = 0.05), which are ‘Tenure’ and ‘HasCrCard’. They have a large p value so that they cannot reject H0.

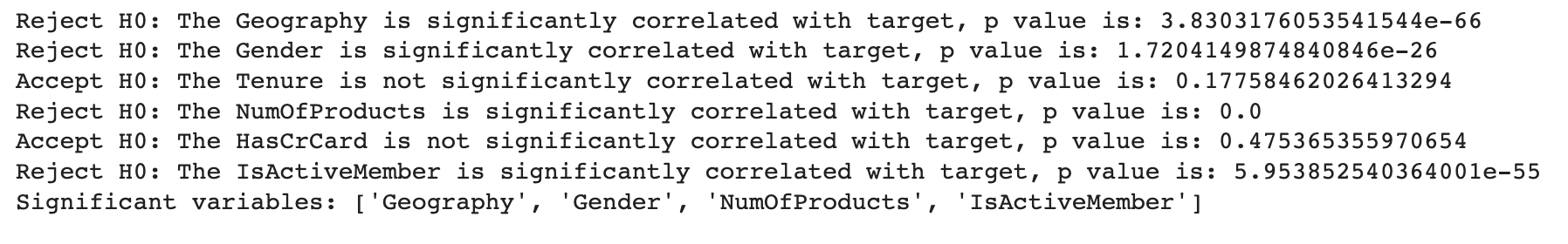


Fig. 3.2.2 Chi square result

## 3.3 Numeric predictors

### 3.3.1 Distribution exploration

* Boxplot

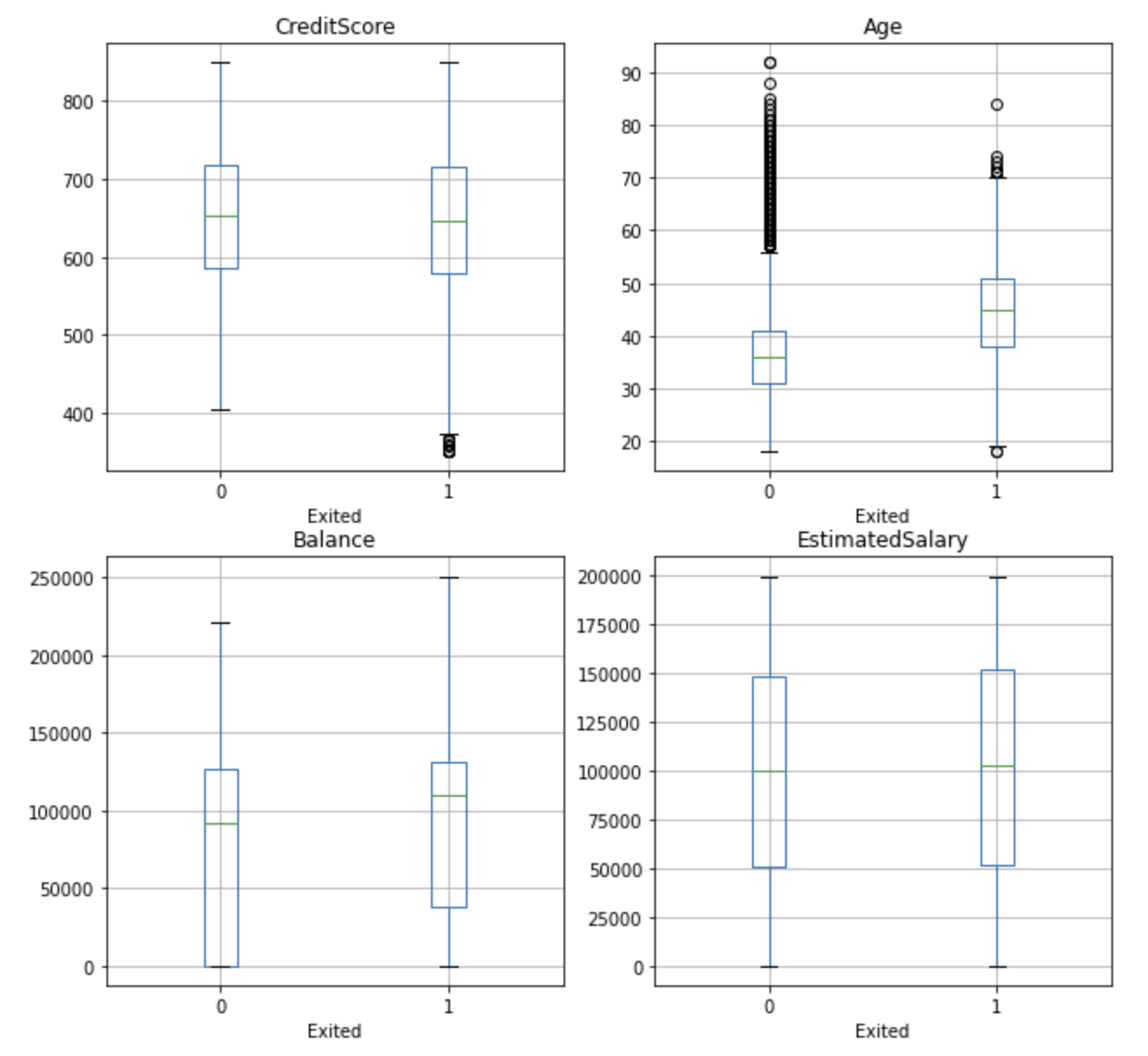


Fig. 3.3.1 Boxplot of numeric predictors

As can be seen from the above boxplot figure, there are some outliers in ‘CreditScore’ and ‘Age’ in the boxplot definition. And then we check outliers in the statistical definition which is not in the 95% area in the normal distribution by calculating mean 土 3 \* std. There are two numeric variables: ‘Age’ and ‘CreditScore’ have outliers. However, if we delete those outliers we would lose too much information. As we can see, the data is imbalanced in target, the ‘Exited’ is fewer.Therefore, we would like to keep them to observe in the model.

* Density plot

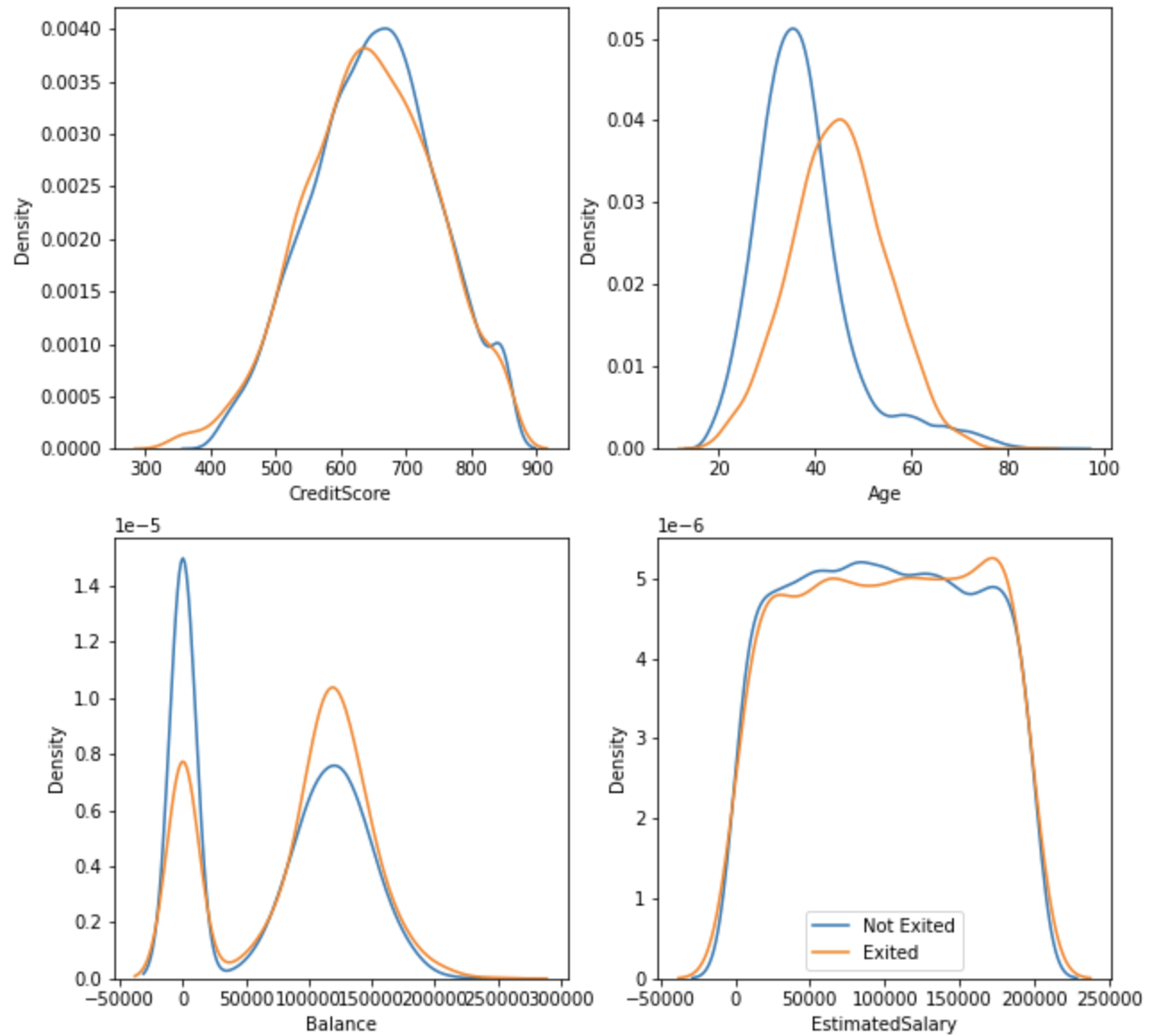


Fig. 3.3.2 Density plot of numeric predictors

### 3.3.2 Statistical test

With the purpose of validating the statistical significance between numeric variables and target, the point biserial test method can be applied.

* H0: the predictor has no significant correlation with target
* H1: the predictor has significant correlation with target

Only one variable ‘EstimatedSalary’ has a big p value so that it cannot reject H0 (alpha = 0.05). We are not sure whether this variable is authentic or not. Probably, the information of estimation has huge bias.

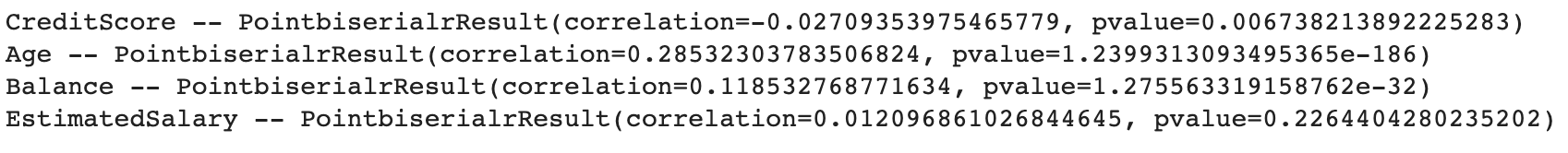


Fig. 3.3.3 Point biserial result

## 3.4 Correlation Heatmap

The Pearson correlation heatmap is shown to analyze if there is any relationship among predictors. The conclusion is there is no strong relationship among predictors.

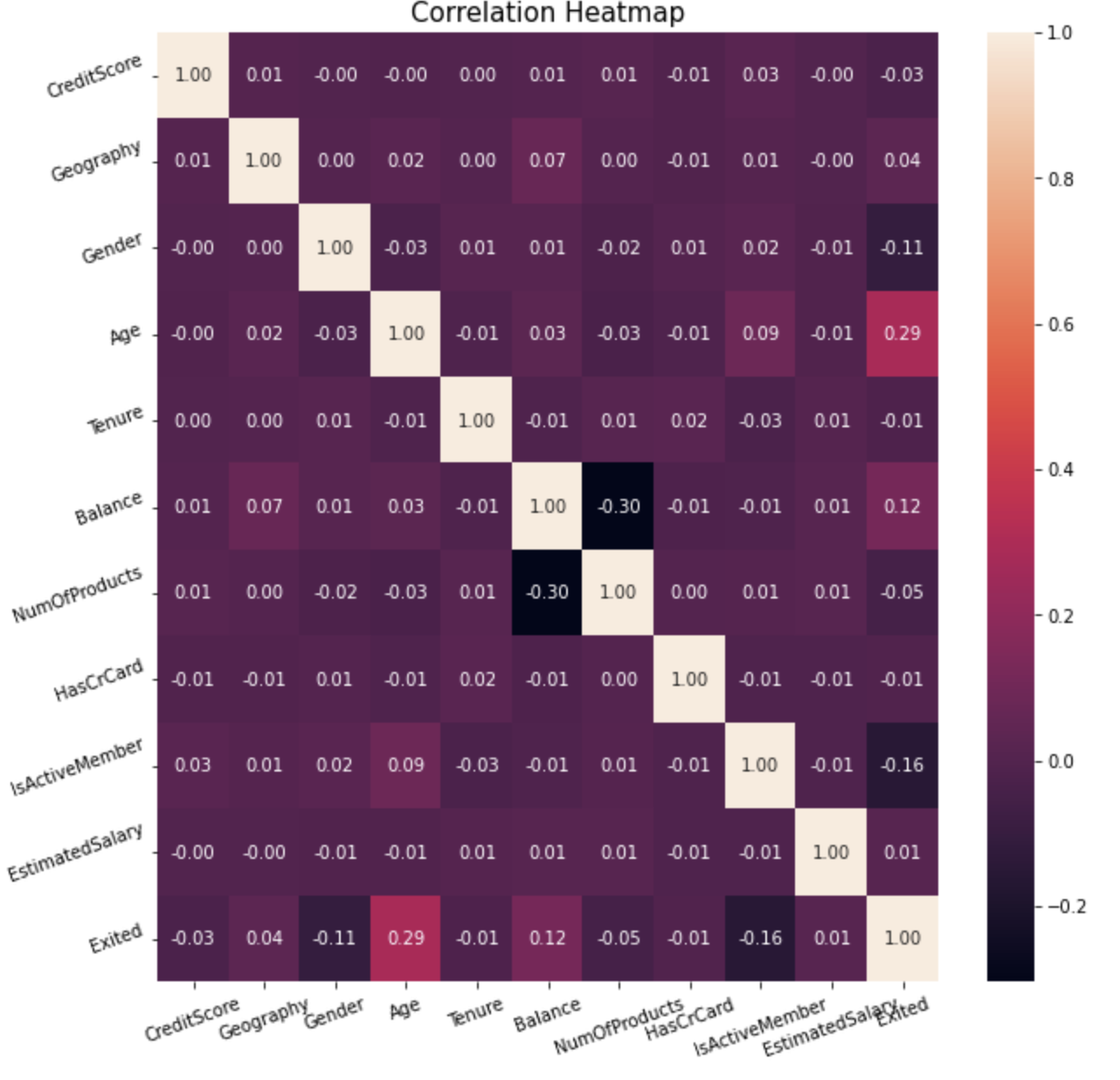


Fig. 3.4.1 Pearson correlation heatmap of predictors

# 4 Model Exploration and Model Selection

Tab. 4.1 Models comparison

| **Algorithm** | **Advantages** | **Disadvantages** |
| --- | --- | --- |
| Logistic Regression | * No parametric assumptions * Easy implementation * Automatically adapt while adding new data | * Time consuming * Sensitive to noisy data * Curse of dimensionality |
| Naive Bayes | * Easy and fast * Highly scalable * Robust to noisy data | * Require much more data * Inaccurate propensity |
| Decision Tree | * Interpretable rules for small-sized trees * Tolerance for missing value * Robust to outliers | * Greedy search algorithm * Decision boundaries are orthogonal |
| Random Forest | * No feature selection and dimension reduction * Tolerance for large missing value * Friendly to imbalance data | * Not friendly to small-sized data * Overfitting on noisy data |
| Neural Network | * Good predictive capability * Can capture complex relationship * No need to specify a model | * Black box algorithm * Need to select variables * Heavy computational requirements |

# 5 Implementation of selected models

## 5.1 Logistic Regression

Logistic Regression is considered as one of the fastest and cheapest classification models. Rather than using y directly as the outcome variable, we use a function, which is called the logit function.

The logit function is able to be modeled as a linear function of the predictors. Once the logit has been predicted, it can be mapped back to a probability. And also, the probability of logit belonging to [0, 1].

Since logistic regression classifiers have no restrictions on predictors, we just use the cross validation method to find the optimal parameters, which produce the highest accuracy rating on the dataset. After fitting the 10 folds of subsets of data, we conclude when c=0.8, the score of the validation data score is 0.809.

## 5.2 Naive Bayes

Naive Bayes is a generative model that uses the prior and likelihood of data into bayesian theorem. To avoid having no records restricted to match the new data as well as reduce the amount of data, Naive Bayes assumes each predictors are independent. Under this circumstance, we only care about one predictor every time.

But it provides an accurate class of output but inaccurate propensity compared with the exact bayesian.

Because Naive Bayes requires all predictors of categorical variables. First, we convert all the numeric variables to categorical variables before implementing the Naive Bayes model. And also, Naive Bayes do not need to tune parameters.

## 5.3 SVM

The goal of the support vector machine is to find a linear hyperplane that can classify correctly and maximize the distance between two categories. We can find only one optimal hyperplane at last.

We used the cross validation method to find all optimal parameters, including c, kernel and degree. Finally, we got the optimal results: c=1.2, kernel is rbf and degree=3. The score in the validation dataset is 0.86.

## 5.4 Decision Tree

Among the data-driven data mining methods, decision trees are the most transparent and easy to interpret models. Based on separating records into subgroups by creating splits on predictors, decision trees try to find the nodes who have the lowest impurity information. Besides, avoiding overfitting and pruning are essential as well.

For the decision tree, we used the cross validation method to find optimal parameters, so we tested different criteria, different depths from 1 to 10 and different minimum numbers of leaves from 1 to 50. As a result, we got the optimal settings which is that the criteria is gini, 8 layers deep and 26 minimum leaves and the score of validation data is 0.86.

## 5.5 Random Forest

A random forest is a classifier that contains multiple decision trees, and its output category is determined by the mode of the categories output by each individual tree.

We used GridSearchCV to find the optimal parameters for the Random Forest model which are {n\_estimators = 90, max\_depth = 10, min\_samples\_split = 50}.

## 5.6 XGBoosting

Xgboost is pretty much the same as Gradient Boosting but Xgboost is an efficient implementation of Gradient Boosting. There are a lot of improvements in xgboost than Gradient Boosting. For example, it added a regularization term in the objective function and a second derivative in the loss function.

We used RandomizedSearchCV to find the optimal parameters as well and the parameters are shown below:

Optimal parameter: {'colsample\_bytree': 0.5, 'gamma': 0.2, 'learning\_rate': 0.05,'max\_depth': 5, 'min\_child\_weight': 1}.

## 5.7 Neural Network

Neural network is a flexible data-driven (albeit blackbox) method that can be used for classification, prediction, and is the basis for deep learning—a powerful technique that lies behind many artificial intelligence applications. Neural network puts many neurons together in each layer in order to transmit information to each other. And then according to the backpropagation algorithm, fix mistakes.

As we know, the neural network is a black box, so no one can know how the information would be delivered among these neurons. And also they need to be given accurate layer numbers and neuron numbers per layer. There are so many parameters that we cannot use gridsearch to find the optimal one. Therefore, we randomly try and tune the parameters and get the relative optimal model. Finally, the model chooses three layers and eight neurons per layer, and alpha=0.005, solver='lbfgs',activation='relu'.

# 6 Performance Evaluation and Interpretation

In most instances, the data miners may use accuracy, F1 score, and so on, to evaluate the model performance. However, the most important thing in the project is which class is the class of interest. In bank churn prediction, the churning customers or the customers who are likely to churn is the interest of class (in this dataset, response = 1). Because the purpose of the bank is to recognize these bunch of users and have some strategies on them in order to avoid losing them. Therefore, we put much attention on the performance of class 1 rather than blindly pursue the effect of the overall model.

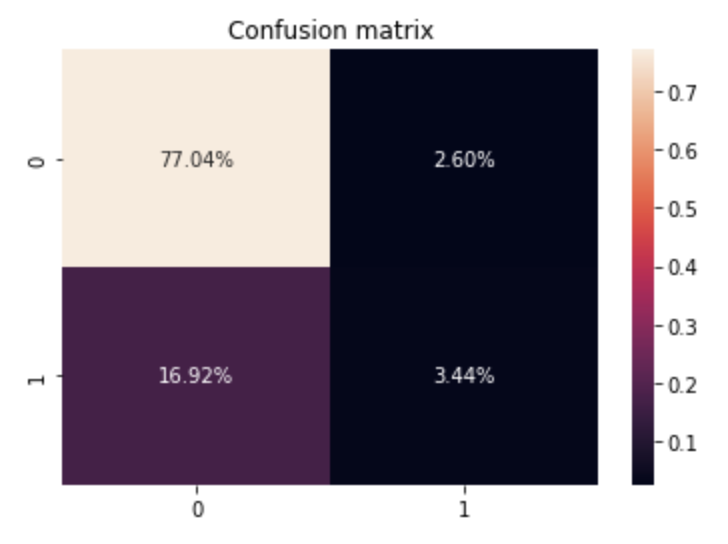
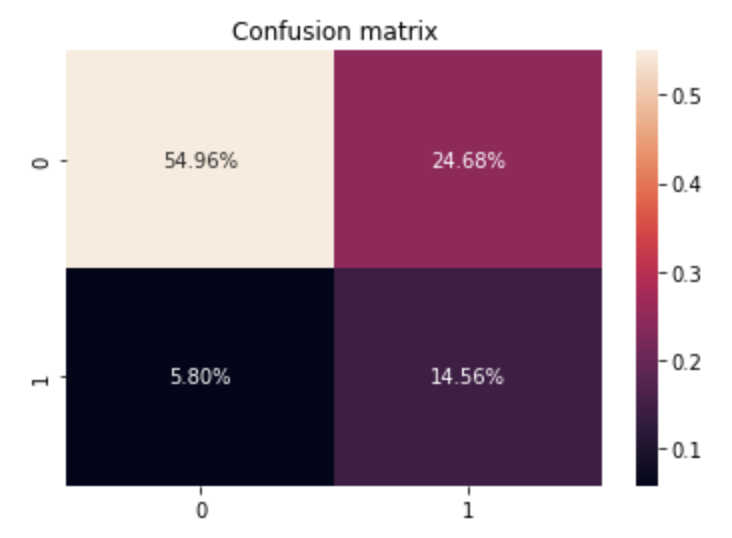
In this project, we implement 2 models in each algorithm to tackle 2 scenarios in a real business. With consideration of the different costs of churning strategies, we would like to apply the balanced model in the situation when the expense of strategies on churners is not expensive or it is more important and cheaper if the bank retains a churning customer successfully than loses a customer. Similarly, if the expense of strategies is expensive, the bank needs to consider their ratio of retaining successfully, they can use the imbalanced model.

Also, We discarded some models we implemented previously with poor performance and selected models with more representative and better actual effects for further development.

## 6.1 Logistic Regression

The confusion matrix is shown in balanced and imbalanced models. As we said before, if the bank thinks it is more important to distinguish the churning users, we can use the balanced model. The recall value of class 1 is 72% (compared with 17% in the imbalanced model), which means the balanced model can distinguish 72% churning customers successfully from actual churning customers. Even though the performance of the whole model is not as good as the other one.

For the imbalanced model, the precision value of class 1 is 57% (compared with 37% in the balanced model), which means the imbalanced model can distinguish 57% churning customers successfully from predicted churning customers. And also the total accuracy is better than the balanced one because there is a higher recall of class 0, which is the majority in data.



balanced imbalanced

Fig. 6.1.1 Confusion Matrix of Logistic Regression

The following tables show the evaluation of the models:

Tab. 6.1.1 Model performance of Logistic Regression(balanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.90 | 0.69 | 0.78 |
| **1(Exited)** | 0.37 | 0.72 | 0.49 |
| **accuracy** |  |  | 0.70 |

Tab. 6.1.2 Model performance of Logistic Regression(imbalanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.82 | 0.97 | 0.89 |
| **1(Exited)** | 0.57 | 0.17 | 0.26 |
| **accuracy** |  |  | 0.80 |

In addition to the above evaluation methods, we plot the ROC curve and calculate the AUC. From the following plots, we are able to conclude that the ROC curve and AUC of two models are almost the same.

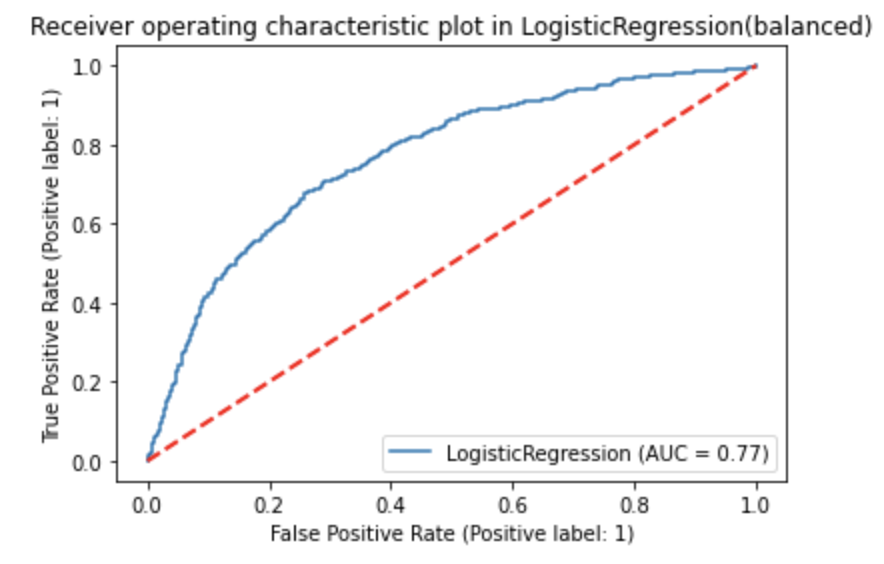


Fig. 6.1.2 ROC curve of Logistic Regression

Another way to overcome rare classes of interest is to decrease the cut-off value of the model so that there is more data that is going to be distinguished as class 1. While increasing by recall of the class of interest, it will sacrifice the accuracy of the total model. Given higher recall of the class 1 and lower accuracy, the cut-off value is the intersection of the two lines: 0.49.

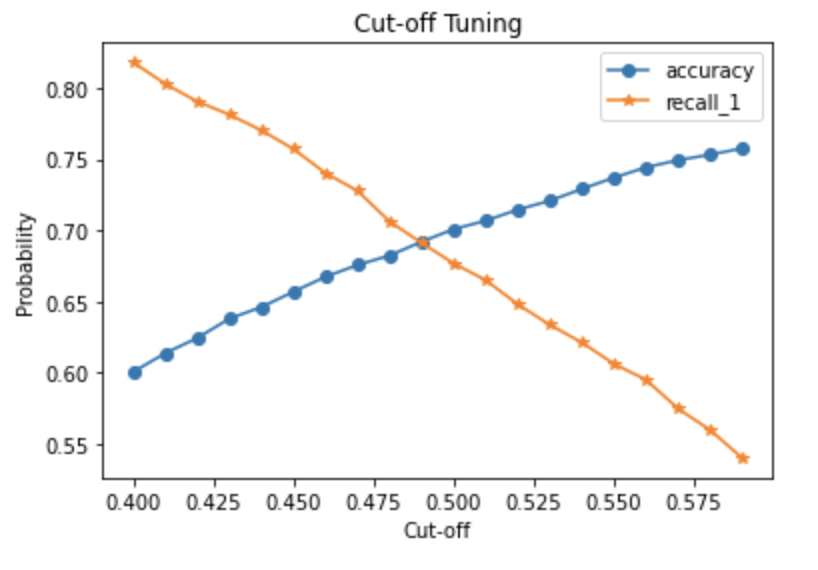


Fig. 6.1.3 Cut-off value tuning

Tab. 6.1.3 Model performance of Logistic Regression(cut-off=0.49)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.91 | 0.68 | 0.77 |
| **1(Exited)** | 0.36 | 0.723 | 0.48 |
| **accuracy** |  |  | 0.69 |

## 6.2 Naive Bayes

In the Naive Bayes algorithm, we only have 1 model since the Naive Bayes is needed to calculate the prior probability. The confusion matrix shows that the model is not suitable to apply in churning prediction because class 1 is the least one. The likely reason is the distributions of important predictors are similar between class 0 and class 1 in the data visualization section. So, there is no difference that can distinguish class 0 and class 1.

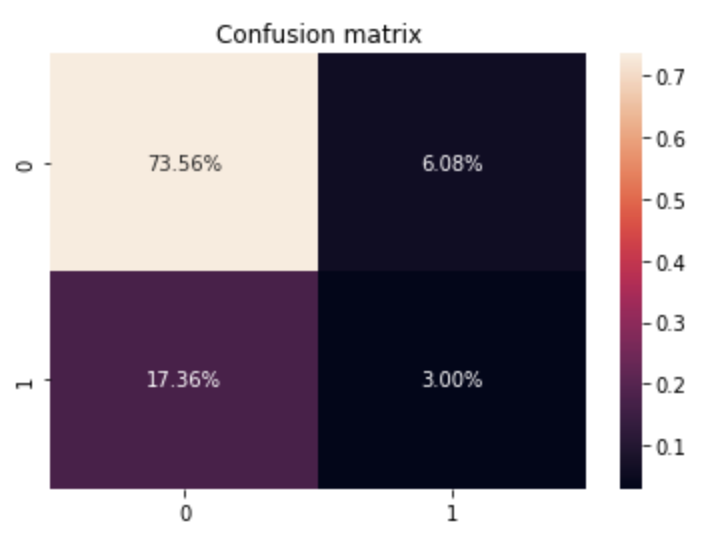


Fig. 6.2.1 Confusion Matrix of Naive Bayes

The Naive Bayes model performance is in the following table. The performance of the interest of class is not good, no matter the precision or recall value.

Tab. 6.2.1 Model performance of Naive Bayes

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.81 | 0.92 | 0.86 |
| **1(Exited)** | 0.33 | 0.15 | 0.20 |
| **accuracy** |  |  | 0.77 |

The AUC is 0.62 and the ROC curve is like:

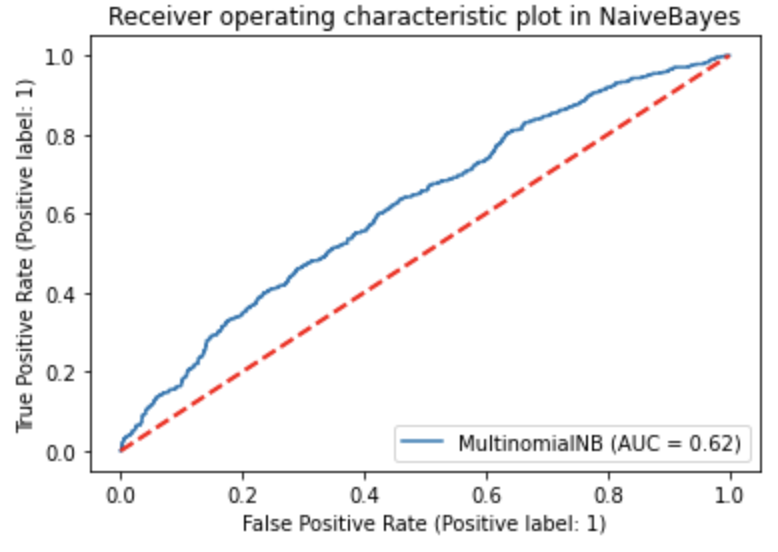
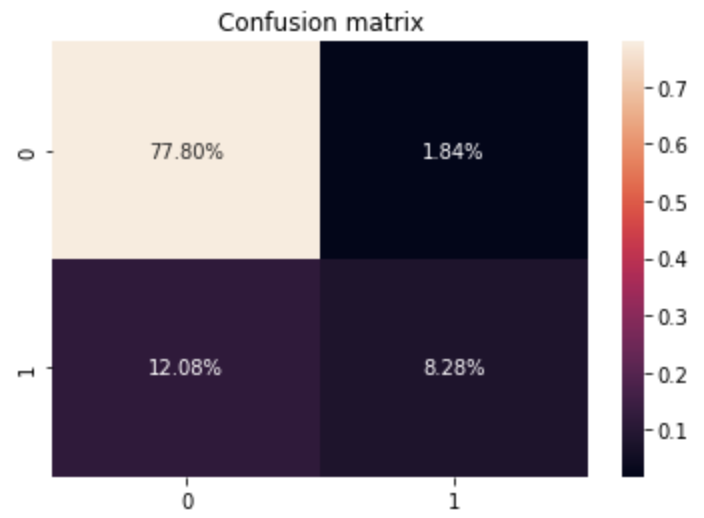
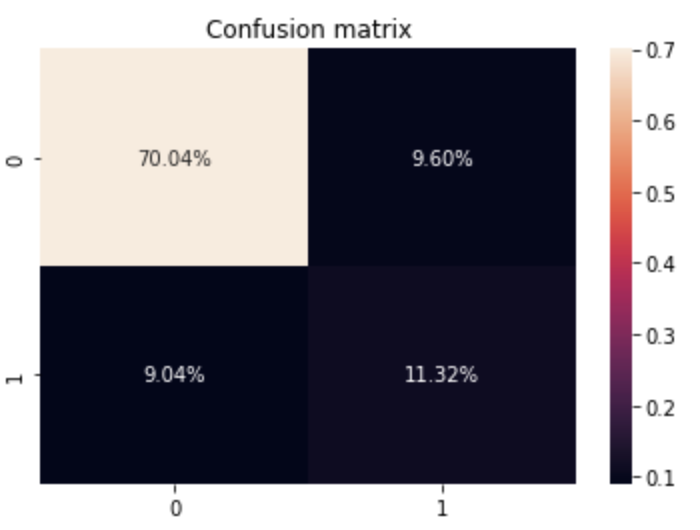


Fig. 6.2.2 ROC curve of Naive Bayes

## 6.3 SVM

Similarly, the two confusion matrices of SVM are shown as follows. For the balanced model, the precision and recall are similar, which is 54% and 56% respectively. The recall value of SVM is lower than other balanced models, even though the accuracy value 81% is a little bit higher than other models.

For the imbalanced model, however, the performance is better than other imbalanced models. Not only perform perfectly in class 0, but also in class 1. The F1 Score of class 1 0.54 is even identity with the balanced model. The precision value is 82%, so the model is able to distinguish 82% churning users successfully from predicted churning users. Furthermore, the accuracy is higher one among all models.



balanced imbalanced

Fig. 6.3.1 Confusion Matrix of SVM

The following tables show the evaluation of the model:

Tab. 6.3.1 Model performance of SVM(balanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.89 | 0.88 | 0.88 |
| **1(Exited)** | 0.54 | 0.56 | 0.55 |
| **accuracy** |  |  | 0.81 |

Tab. 6.3.1 Model performance of SVM(imbalanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.87 | 0.98 | 0.92 |
| **1(Exited)** | 0.82 | 0.41 | 0.54 |
| **accuracy** |  |  | 0.86 |

Also, the AUC 0.84 gives the answer of performance in the imbalanced model of SVM.

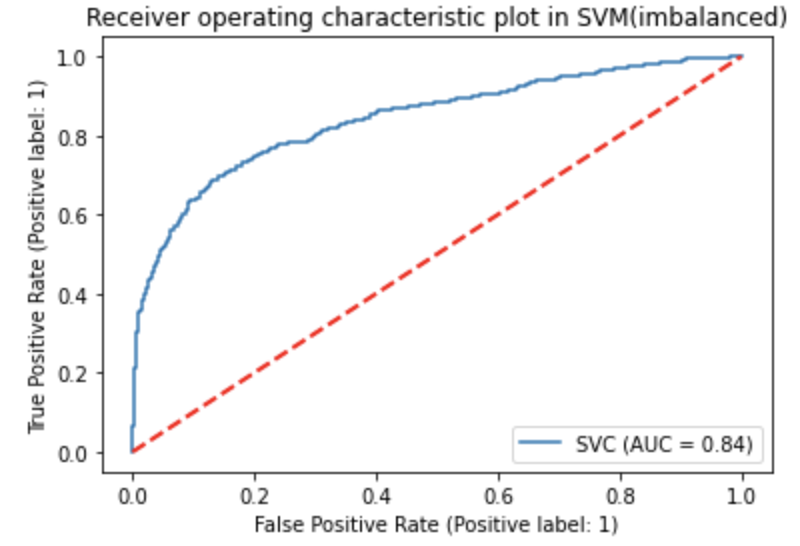
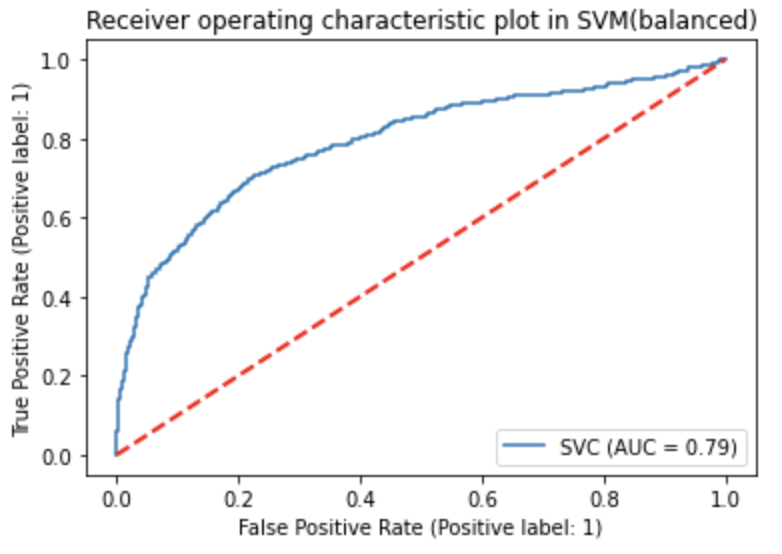
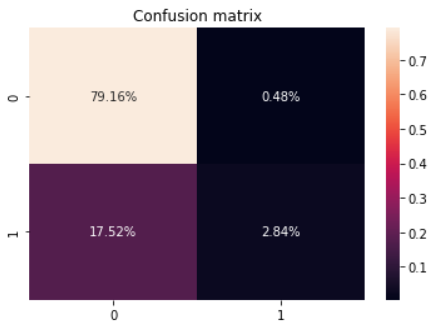
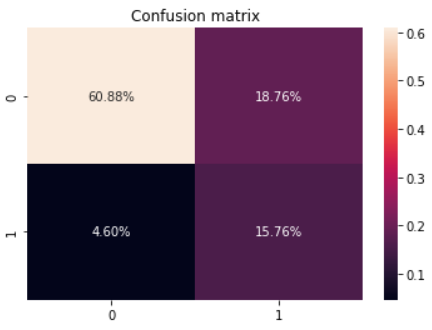


Fig. 6.3.2 ROC curve of SVM

## 6.4 Decision Tree

The following charts contain the performance evaluation of the decision tree. For the model which has the balanced samples, there is a huge improvement in selecting the exited customers. The recall value for class 1 reached 70% , which is a higher recall value among all models, and the F1 score reached 0.54. Even though the model with a balanced sample doesn't have a better overall performance than the model with an imbalanced sample, there is still a reason to use it because it has a better ability to distinguish each class.

As we can see, in the model that we didn't try to balance the samples the performance is not bad and the accuracy reached 82% although it doesn't have a good ability to recognize the exited customers (class 1). It is worth mentioning that the recall value of class 0 is 99%, the imbalanced model is close to recognizing 100% successfully from real not exited users. The bank can utilize this model to give targeted strategies to recognized exited users with the purpose of increasing their closeness.



balanced imbalanced

Fig. 6.4.1 Confusion Matrix of Decision Tree

Tab. 6.4.1 Model performance of Decision Tree(balanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.91 | 0.77 | 0.84 |
| **1(Exited)** | 0.44 | 0.70 | 0.54 |
| **accuracy** |  |  | 0.76 |

Tab. 6.4.2 Model performance of Decision Tree(imbalanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.82 | 0.99 | 0.90 |
| **1(Exited)** | 0.86 | 0.14 | 0.24 |
| **accuracy** |  |  | 0.82 |

These two roc curves don't have much difference but the model with an imbalanced sample has a slightly higher score.

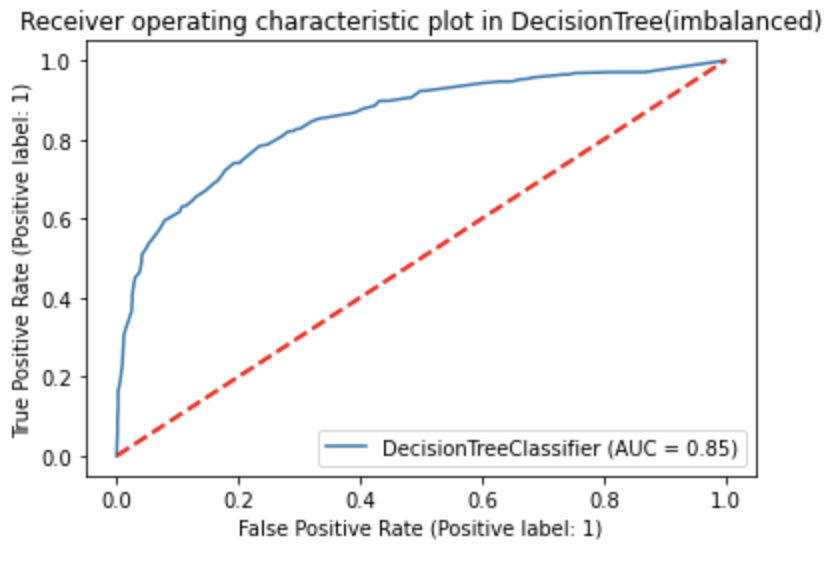
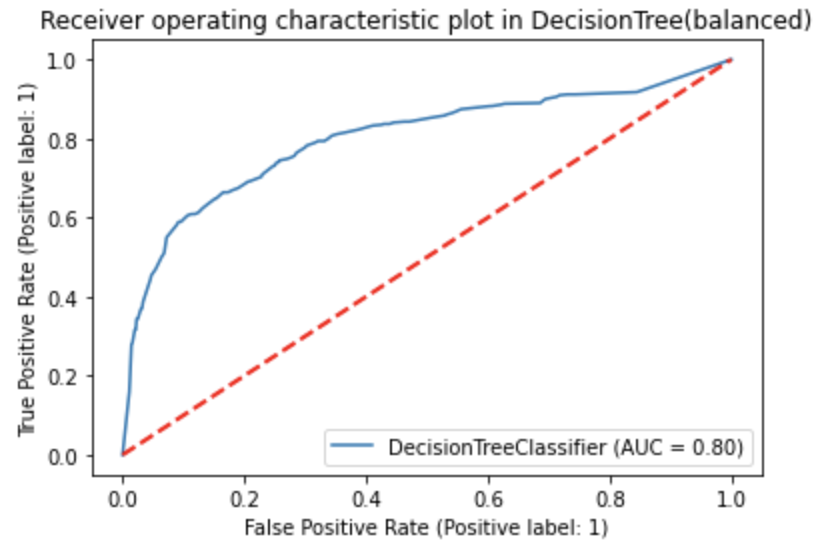
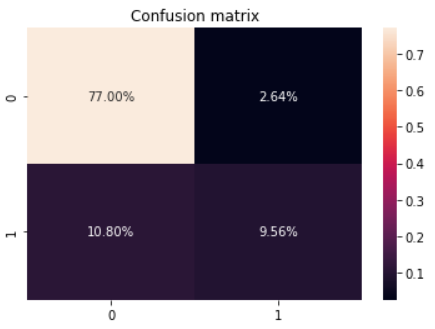
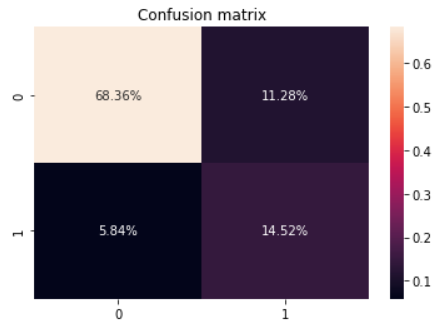


Fig. 6.4.2 ROC curve of Decision Tree

The following two graphs show the figures of the tree’s structures. We can see that the depths are the same. The only difference they have is the second model has four more leaves which will make a huge difference in the performance.

## 6.5 Random Forest

For the random forest, it has achieved a better performance than the decision tree and a similar situation to the other models. The model with balanced data has a stronger ability to predict for class 1 but a lower overall performance. The recall value of class 1 achieved 71%, which is the highest one among all balanced models. So, that means the model can identify 71% churning users successfully from all actual churning users. In addition, the accuracy of the total is 86%, the highest value among all balanced models. The imbalance model of random forest behaves perfect as well, with the highest accuracy of 87%.



balanced imbalanced

Fig. 6.5.1 Confusion Matrix of Random Forest

Tab. 6.5.1 Model performance of Random Forest(balanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.92 | 0.86 | 0.89 |
| **1(Exited)** | 0.56 | 0.71 | 0.63 |
| **accuracy** |  |  | 0.83 |

Tab. 6.5.2 Model performance of Random Forest(imbalanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.88 | 0.97 | 0.92 |
| **1(Exited)** | 0.78 | 0.47 | 0.59 |
| **accuracy** |  |  | 0.87 |

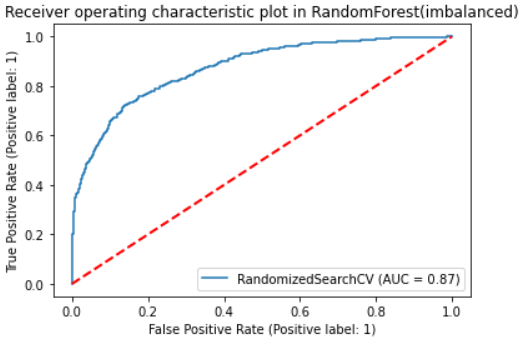
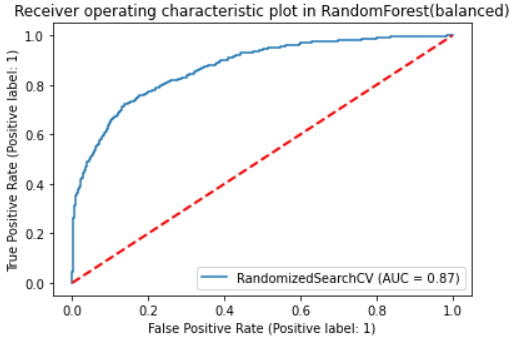
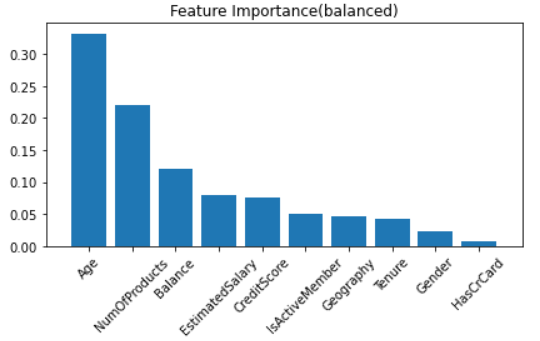


Fig. 6.5.2 ROC curve of Random Forest

In these two models, we can see that the contribution made by each parameter is basically the same, however, the model with a balanced sample has relatively lower importance on every feature than the imbalanced one. In the shown plot, predictors ‘Age’ and ‘NumOfProduct’ mainly impact a user, whether churning or not. It is reasonable in the real business, if a customer owns plenty of products in the bank, the customer would probably not choose to churn.



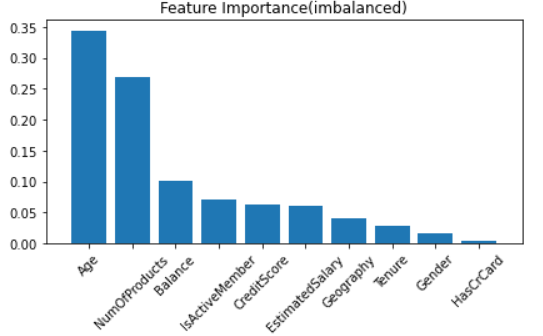
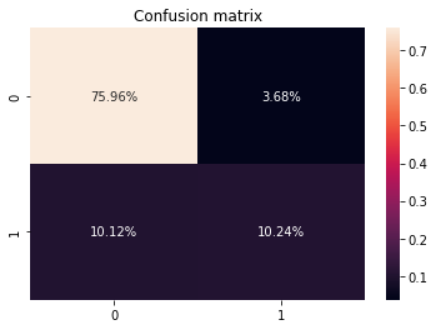
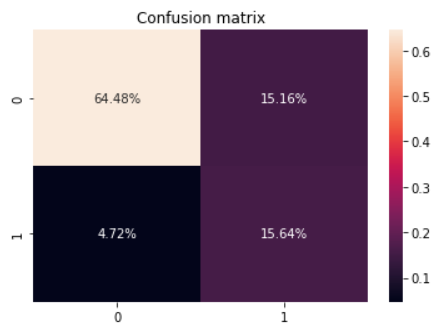


Fig. 6.5.3 Feature Importance graphs of Random Forest

## 6.6 XGBoosting

XGBoosting has a similar performance as the decision tree and random forest since they have resemblance algorithms. For the balanced model, the recall value for class 1 reached 77% , which is the highest recall value among all models. This model is the most suitable balanced model to be used in recognizing churning users. For the imbalanced model, it behaves similarly to the random forest algorithm.



balanced imbalanced

Fig. 6.6.1 Confusion Matrix of XGBoosting

Tab. 6.6.1 Model performance of XGBoosting(balanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.93 | 0.80 | 0.86 |
| **1(Exited)** | 0.49 | 0.77 | 0.60 |
| **accuracy** |  |  | 0.79 |

Tab. 6.6.2 Model performance of XGBoosting(imbalanced)

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.88 | 0.95 | 0.92 |
| **1(Exited)** | 0.74 | 0.50 | 0.60 |
| **accuracy** |  |  | 0.86 |

In addition, XGBoosting has the same AUC as the random forest.

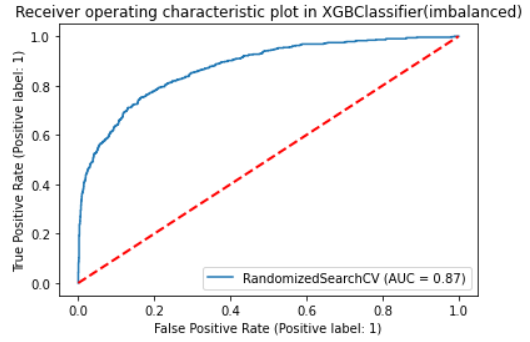
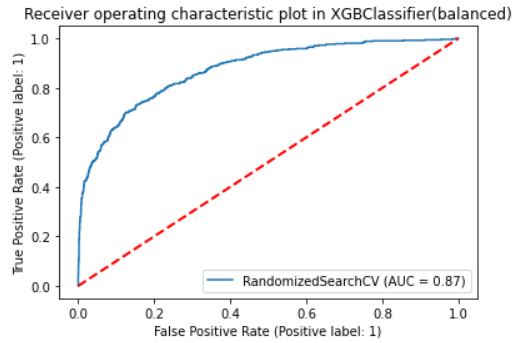
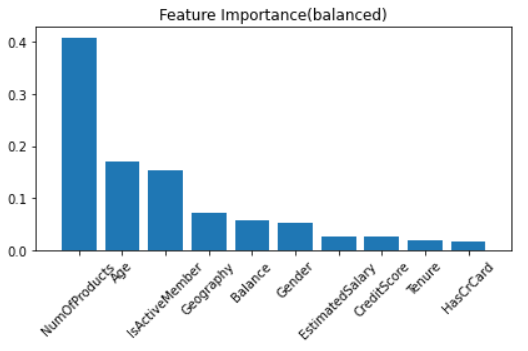


Fig. 6.6.2 ROC curve of XGBoosting

From observing the feature importance figures, we can see it has a completely different result from the random forest. Even though the first two highest importance score predictors are the same, the parameter 'NumOfProducts' takes the domination in XGBoosting, rather than ‘Age’. Besides, the third one ‘ISActiveMember’ is very close to the second one. The predictor ‘Balance’ has a lower importance score than in the random forest model.



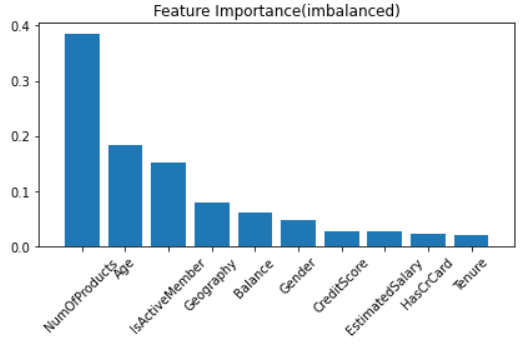


Fig. 6.6.3 Feature Importance graphs of XGBoosting

## 6.7 Neural Network

Although the balanced model is not used in neural network algorithms, the performance is not bad in the following figure. The recall value of the class 1 is 52%, which is the highest value among imbalanced models. It shows how powerful the neural network is. And also the accuracy of the total model is 86%, tied for first. Similar to the AUC value.

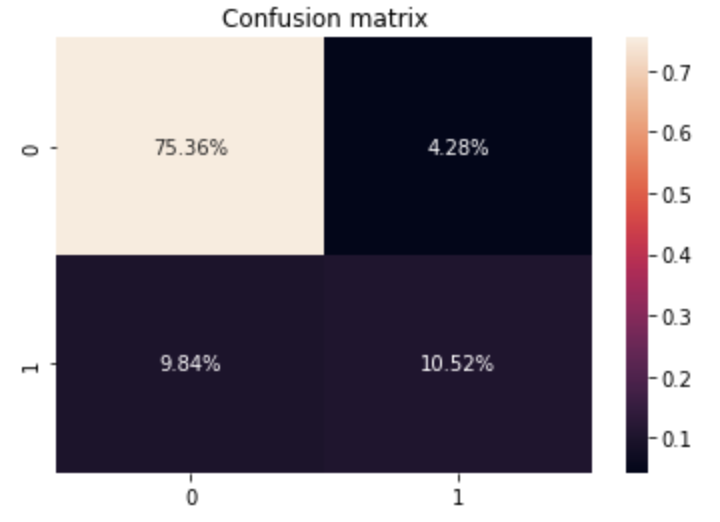


Fig. 6.7.1 Confusion Matrix of Neural network

Tab. 6.7.1 Model performance of Neural network

|  | **precision** | **recall** | **F1 score** |
| --- | --- | --- | --- |
| **0(Not Exited)** | 0.88 | 0.95 | 0.91 |
| **1(Exited)** | 0.71 | 0.52 | 0.60 |
| **accuracy** |  |  | 0.86 |

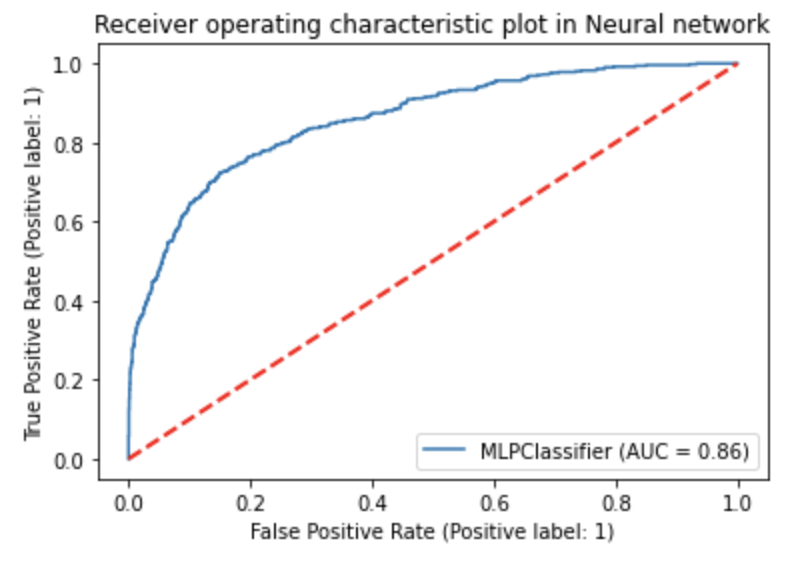


Fig. 6.7.2 ROC curve of Neural network

# 7 Summary

All in all, we developed several different models from different algorithms, however, they can be divided into two groups, one is a model developed using a balanced sample and the other is a model developed using an imbalanced sample. The imbalanced sample model has a stronger ability to try to predict customer churn from the current bank, the other model is to maintain a high accuracy of the prediction. Even though the first model has an overall lower performance score than the second one, it is still a better choice in certain circumstances. Furthermore, models with lower cut-off value can be used if the company would like to get a higher recall of the class of interest.

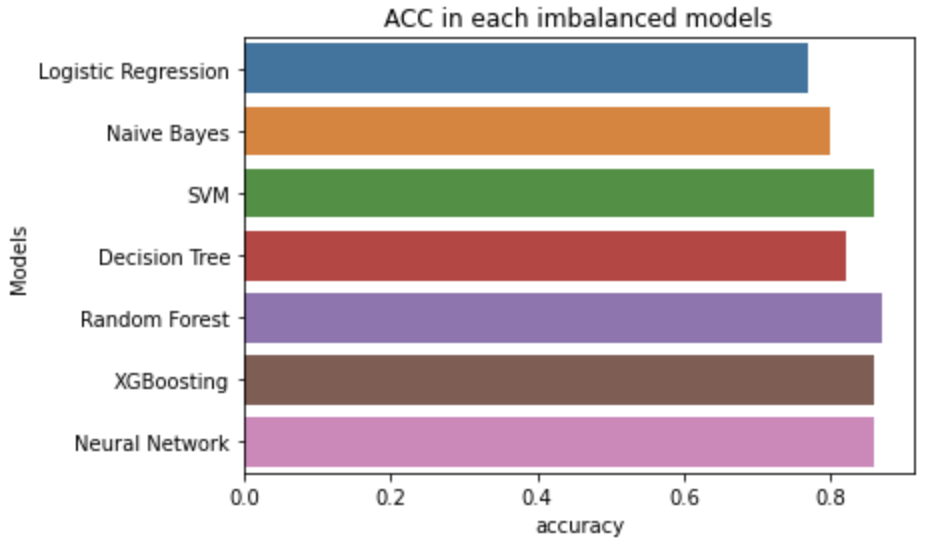
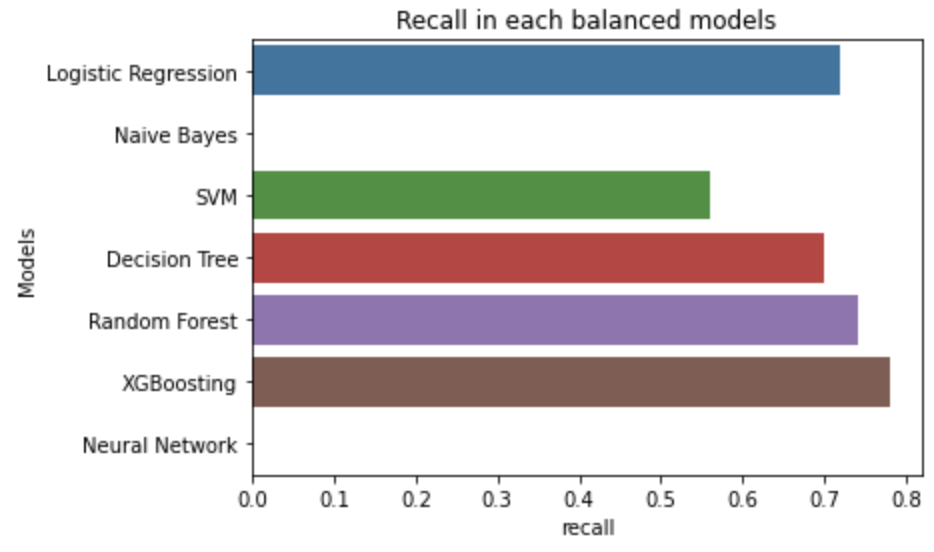


Fig. 6.8.1 Performance evaluation of all models

In conclusion, we believe that choosing which model is according to the goal of the company. If companies want to retain as many customers as possible, they should use an imbalanced data model because it is using accuracy as a trade-off for predicting more churning customers. On the other hand, if the company wants to minimize the cost, then they are looking for the second model that provides higher accuracy.

**Work Cited**

Shmueli, Galit, et al. *Data Mining for Business Analytics: Concepts, Techniques and Applications in Python*. Wiley, 2020.