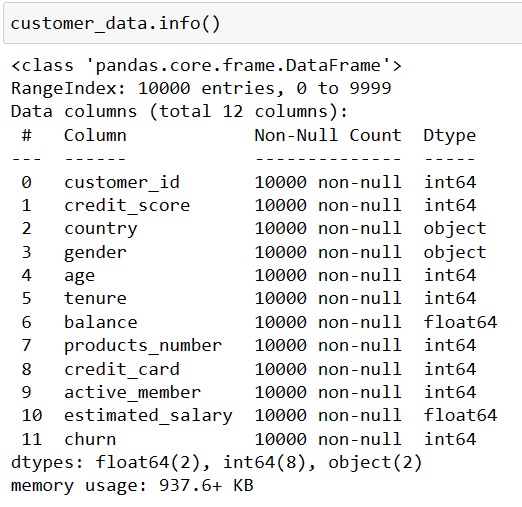
**Bank Customer Churn Prediction Model**

1. **Data set description and preparation**

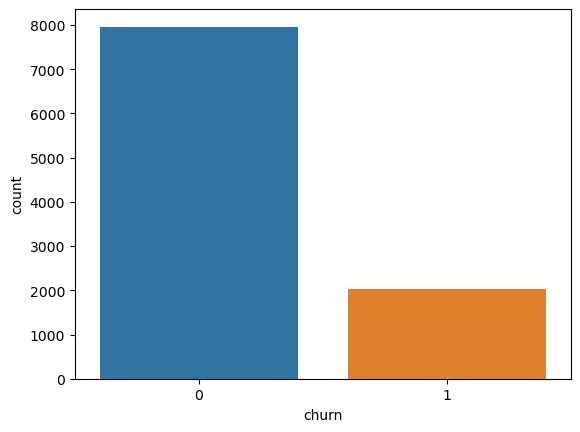
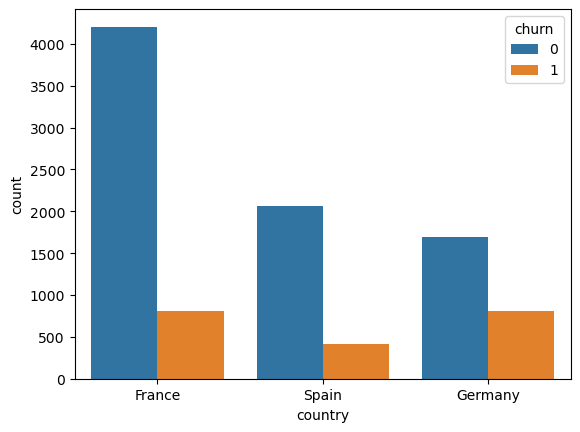
The data set for prediction model contains 10000 rows and 12 following columns in which the “churn” column is the target label for prediction.

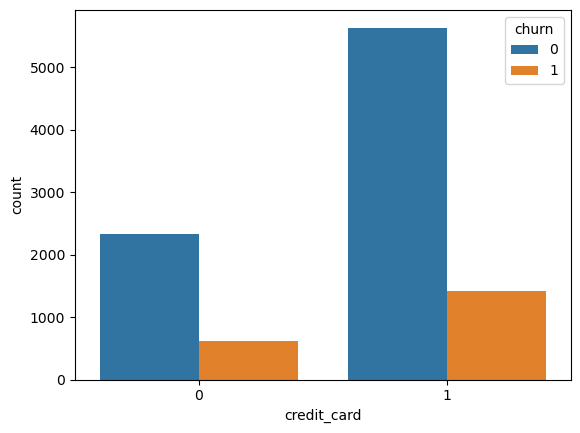
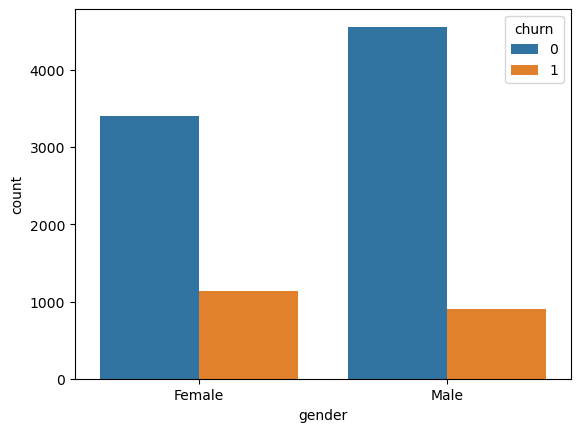
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| customer\_id | credit\_score | country | gender | age | tenure |
| balance | products\_number | credit\_card | active\_member | estimated\_salary | churn |

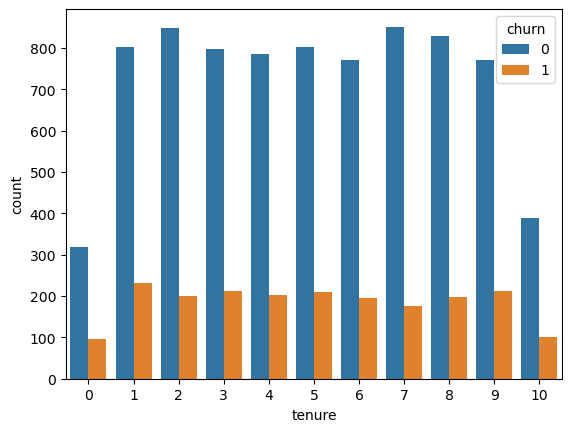
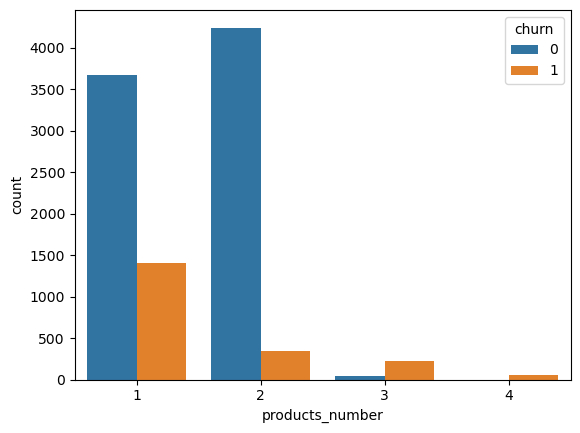
Before starting data preprocessing, it is necessary to understand the general information of the data set. The Pandas libray would be imported to read the data file. The data set includes 2 nominal attributes which are “gender” and “geography”, 3 binary attributes which are “credit\_card”, “active\_member” and “churn”. The other features are numeric attribute types. The general information of all the features and data types could be seen as in following picture.

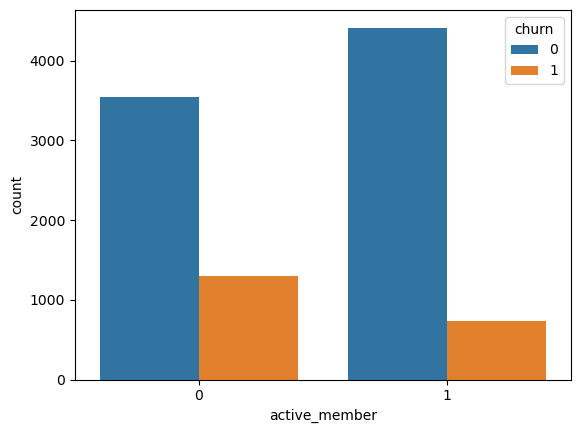
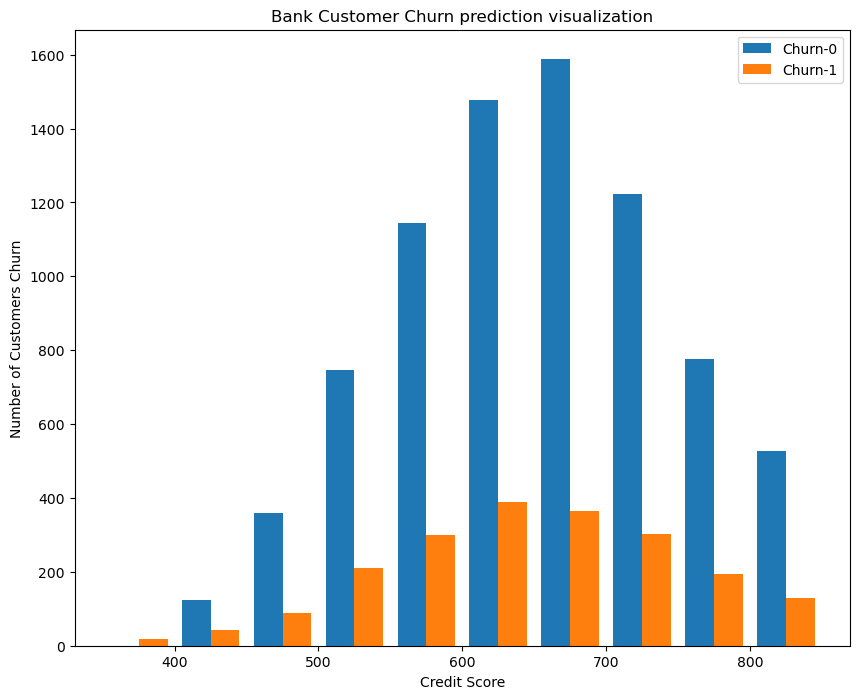


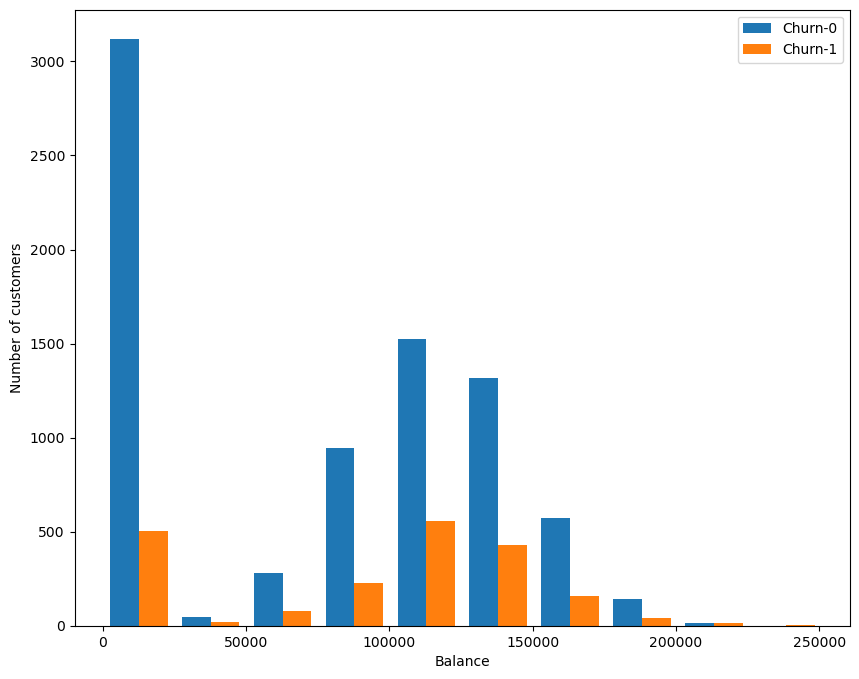
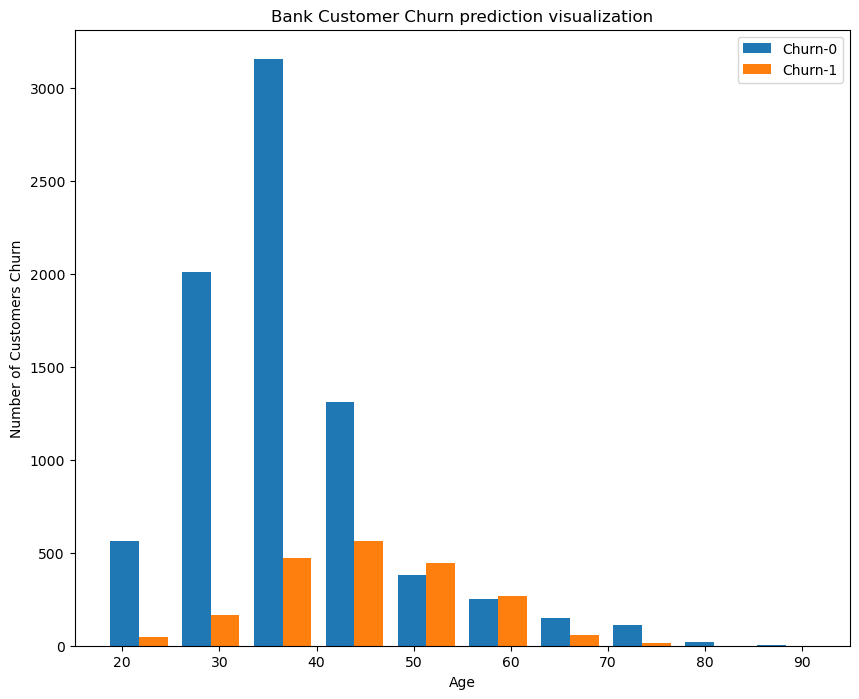
After considering carefully, the “customer\_id” columns would be drop from the dataset, because it has 10000 unique values, which indicate that it is specific for each customer and do not contribute to the classification process. Examining the “churn” features alone, it is binary attribute, with the value “1” means that the client has left the bank and the value “0” imply that the client has not. It contains 10,000 records of which 7,963 (79.63%) are non-churners and 2,037 (20.37%) are churners. A balanced dataset usually contains an equal or or almost equal number of samples of predicted class. Therefore, the dataset is highly unbalanced in terms of the proportion of churners and non-churners. A further data analysis of each feature would be conducted to determine the percentages between each feature and churn feature. The matplotlib.pyplot library and seaborn library would be imported to implement visualize content as follow.

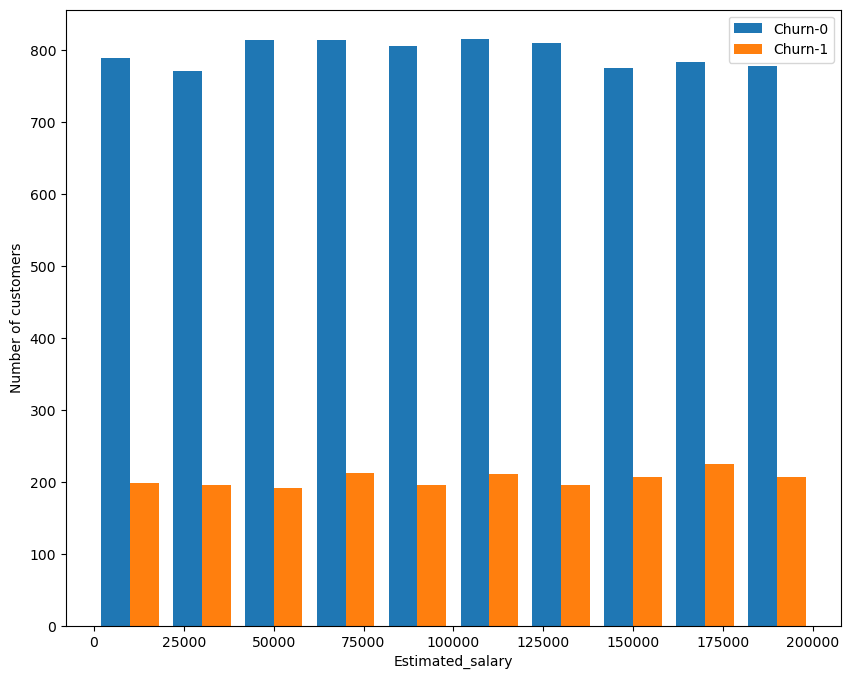
 





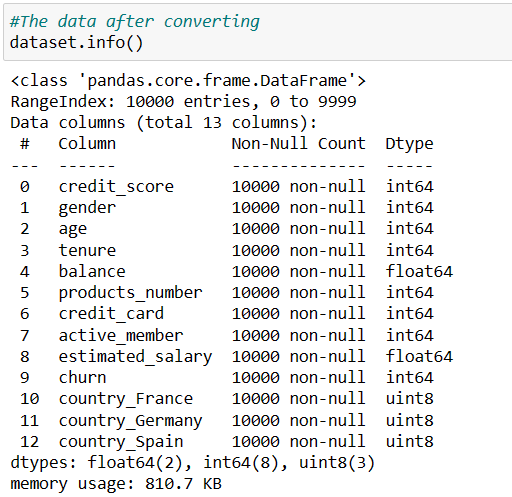


According to the visualized data, there are obvious gaps between the number of non-churners and churners because the churn value count is already unbalanced at first. The feature trends can also be witnessed from the data as followed:

* The majority of the data points are collected from French people, and the Spain, Germany, respectively. Nevertheless, the ratio of customers churned is inversely related to the number of customers from countries, considering Germany has the highest number of churners.
* The percentage of churners are female is also greater than male customers.
* Suprisingly, the factor of having a credit card does not seem to have much impact on customers’ decision to churn.
* Unsurprisingly, the inactive members are more likely to churn.
* Regarding age, the younger customers are less likely to change to other banks the older ones.
* There is no significant difference in the credit score distribution, estimated salary range, balance and number of products used in the likelihood to churn of customer.

Therefore, undersampling and oversampling techniques would be used to achieve balancing. In order for comparing results, two classifiers which are decision tree and support vector machine (SVM) would be conducted with the original data first, and then with the undersampling and oversampling data, finally 10-fold cross-validation would be applied to evaluate the models.

The data would be preprocessed before introducing it to the proposed classifiers. The nominal attributes which are “gender” and “country” need to be converted. The “gender” feature has only two unique data point therefore it would be converted into 0 and 1, with “Female” equal to “0” and “Male” equal to “1”. With the “country” columns, it has 3 three unique data points, therefore it will be split into three different columns which are “country\_France”, “country\_Germany”, and “country\_Spain”. The value would be change to “1” if the customer from that country and “0” in other columns.



1. **Decision tree classifier**

The decision tree classifier is defined as a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes. A decision tree starts with a root node, which does not have any incoming branches. The outgoing branches from the root node then feed into the internal nodes, also known as decision nodes. Based on the available features, both node types conduct evaluations to form homogenous subsets, which are denoted by leaf nodes, or terminal nodes. The leaf nodes represent all the possible outcomes within the dataset. [1]

The conduct of decision tree for the bank customer dataset would be as follow:

* All feature values would be separated as data set X, and “churn” as target label set y.
* The train\_test\_split module would split the data set into a training set and a test set, in this case the training set would be 80% of the dataset and the test set would be 20%.
* Use Gini as the standard to build a decision tree model, and set the depth between 3 and 7, and observe when the depth is better for the model. The Gini measure is the default parameter of scikit-learn library.

The results from decision tree classifier with different depths from 3 to 7 is:

Depth: 3

precision recall f1-score support

0 0.87 0.95 0.91 1607

1 0.68 0.43 0.53 393

accuracy 0.85 2000

macro avg 0.78 0.69 0.72 2000

weighted avg 0.83 0.85 0.83 2000

Accuracy score: 0.8485

Depth: 4

precision recall f1-score support

0 0.86 0.97 0.91 1607

1 0.77 0.37 0.50 393

accuracy 0.85 2000

macro avg 0.81 0.67 0.70 2000

weighted avg 0.84 0.85 0.83 2000

Accuracy score: 0.8535

Depth: 5

precision recall f1-score support

0 0.87 0.97 0.92 1607

1 0.77 0.40 0.52 393

accuracy 0.86 2000

macro avg 0.82 0.68 0.72 2000

weighted avg 0.85 0.86 0.84 2000

Accuracy score: 0.858

Depth: 6

precision recall f1-score support

0 0.88 0.96 0.92 1607

1 0.73 0.45 0.56 393

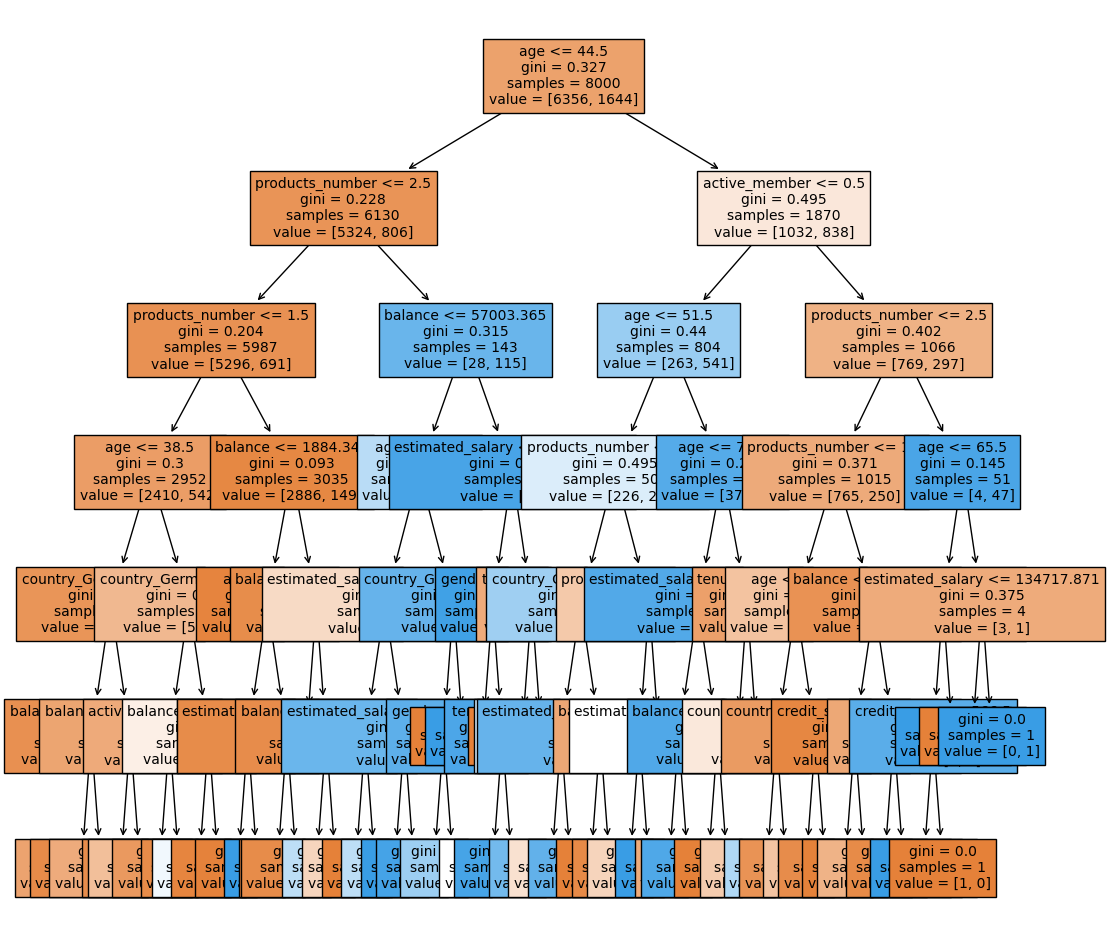
accuracy 0.86 2000

macro avg 0.80 0.71 0.74 2000

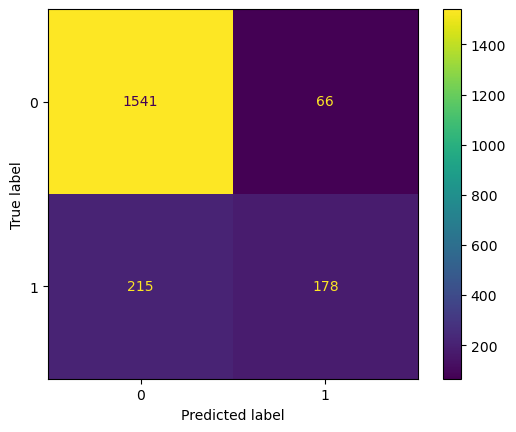
weighted avg 0.85 0.86 0.85 2000

Accuracy score: 0.8595

The highest accuracy score is 0.8595 when the depth is 6. However, as mentioned before, because the dataset is highly unbalanced, the recall and precision rate is not highly ideal. The most important feature of the dataset is “age”, which is consistent with the previous analyse from visualization data.



Despite the high accuracy score, the result from the confusion matrix below is not saticfactory. There are 1,541 values of “0” is correctly classified and only 215 values is misclassified. Unfortunately, there are only 178 values of “1” is correctly classified, which account for a small amount of testing data.



1. **Support vector machine – SVM**

A support vector machine is a supervised learning algorithm that sorts data into two categories. It is trained with a series of data already classified into two categories, building the model as it is initially trained. The task of an SVM algorithm is to determine which category a new data point belongs in. This makes SVM a kind of non-binary linear classifier. [2]

One of the key hyperparameters of SVM model is C. In brief, the C value is used to control the width of the vector margin, the higher the C values, the model accuracy increases. The GridSearchCV would be applied to find the best value of C parameters. The best value of C found in this bank customer prediction equal to 1. After finding the best C value, the features of training data, except “churn”, would be standardized to apply the SVM model. The classification report for SVM model is below.

precision recall f1-score support

0 0.87 0.97 0.92 1607

1 0.77 0.38 0.51 393

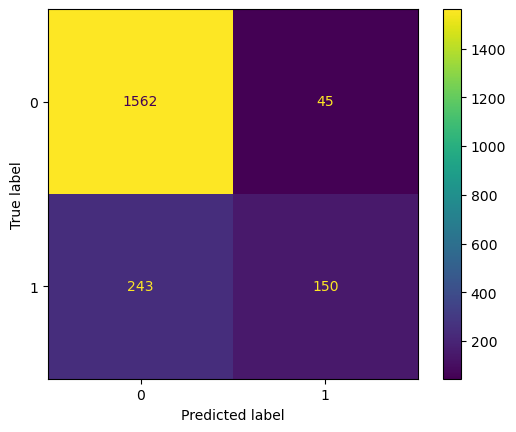
accuracy 0.86 2000

macro avg 0.82 0.68 0.71 2000

weighted avg 0.85 0.86 0.84 2000

Accuracy score: 0.856

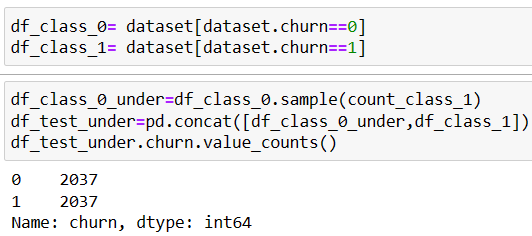
The accuracy score is approximately equal to the result from the decision tree. As mentioned before, because the data is unbalanced, the recall and f1 score is not ideal. The unbalance is much more clearer in the confusion matrix. It also demonstrate large number of value “0” is classified correctly while only 150 values of “1” is classified correctly out 2,000 testing data points.



Due to the unbalanced dataset, which has bias toward customer who are non-churners, both decision tree and SVM have low Recall and F1 score of the value “1” which mean that the possibility of customers who leave the bank would less likely to predict correctly. Therefore, the unbalanced data should be managed first before applying classification model. In some paper, there are many techniques to solve the unbalanced data issue. This experiment would apply two techniques which are undersampling and oversampling, and then evaluate the models with 10-fold cross-validation to find out which technique and model give the best result.

1. **Undersampling technique**

The undersampling technique would take all the classes to the equal amount as the minority class. In this case, the undersampling dataset would reduce the amount of values “0” to the same amount of values “1” of predicted label.



The dataset at this point would have the same amount of customers who are churners and non-churners. Then, it will be divided into training set of 80% and 20% is the test set.

**Decision Tree Classifier**

The accuracy report for decision tree classifier is:

precision recall f1-score support

0 0.78 0.79 0.78 443

1 0.75 0.73 0.74 372

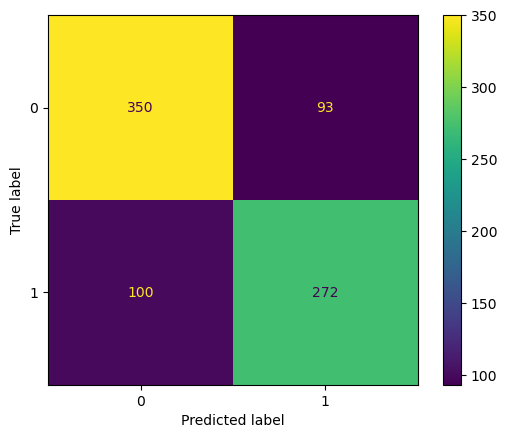
accuracy 0.76 815

macro avg 0.76 0.76 0.76 815

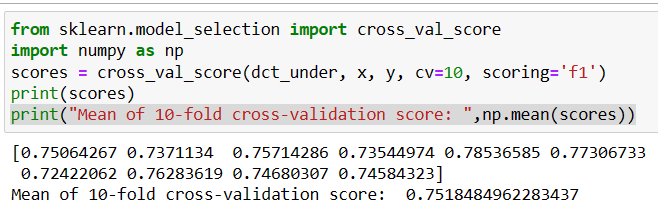
weighted avg 0.76 0.76 0.76 815

Accuracy score: 0.7631901840490798

Compared with the result from the decision tree classifier of original data above, it is obvious that all the measures are more consistent despite lower accuracy score. The Recall and F1 score of are improved to over 70%, much better than that of orginal data. The confusion matrix also demonstrate more satisfactory with higher number of value “1” is classified correctly.



The 10-fold cross-validation for model evaluation also has a consistent result with mostly over 70%.



**SVM classifier**

The results come from SVM model are not as ideal as decision tree but they are also more consistent between two values “0” and “1” than those of original data. However, the 10-fold cross-valiadation is quite poorly performed. The accuracy report, confusion matrix and 10-fold cross-validation of undersampling date with SVM classifier is expressed as below.

Accurracy report:

precision recall f1-score support

0 0.75 0.81 0.78 443

1 0.75 0.69 0.72 372

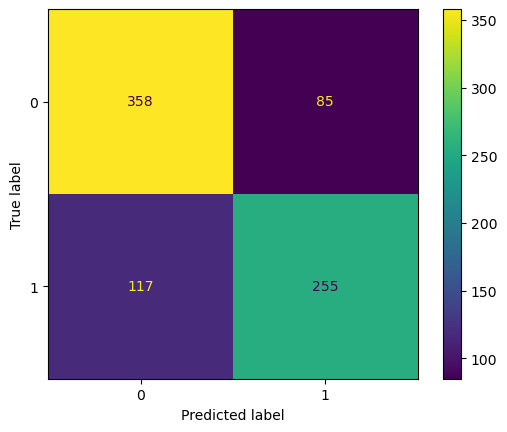
accuracy 0.75 815

macro avg 0.75 0.75 0.75 815

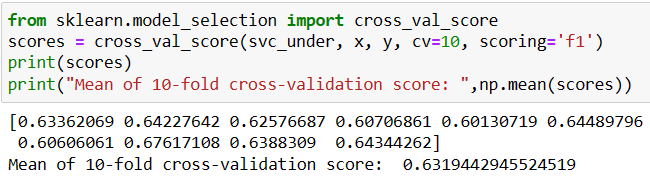
weighted avg 0.75 0.75 0.75 815

Accuracy score: 0.7521472392638037

Confusion matrix:

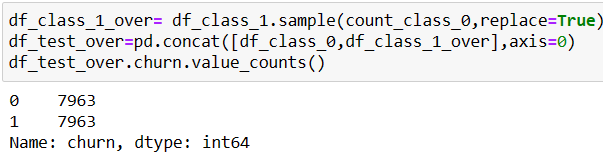


10-fold cross-validation:



1. **Oversampling technique**

The oversampling techniques in some ways is similar to the undersampling techniques, which means that the dataset will have equal amount of churners and non-churners. However, the amount of churners would be duplicate randomly to equal the amount of non-churners. The dataset would look like below.



Then, the oversampling data would be applied to the classifiers similar to the previous process. Results from decision tree classifier would be shown before the SVM.

**Decision Tree Classifier**

Accuracy report from the decision tree classifier is:

precision recall f1-score support

0 0.78 0.80 0.79 1605

1 0.79 0.77 0.78 1581

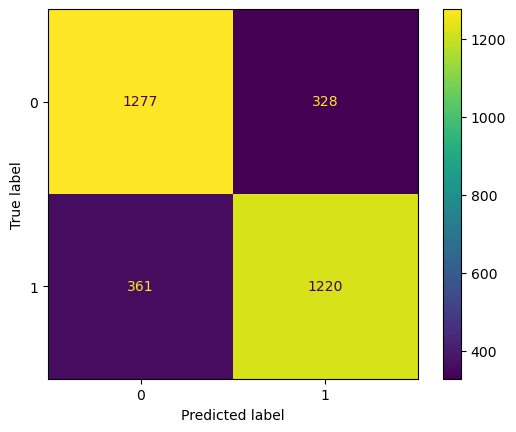
accuracy 0.78 3186

macro avg 0.78 0.78 0.78 3186

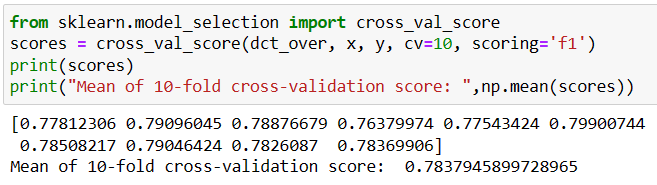
weighted avg 0.78 0.78 0.78 3186

Accuracy score: 0.7837413684871312

Confusion matrix:



The 10-fold cross-validation score:



**SVM classifier**

In general, the accuracy report and the confusion matrix of SVM classifier seems ideal, but the the 10-fold cross-validation does not give positive result.

The accuracy report is:

precision recall f1-score support

0 0.80 0.81 0.81 1605

1 0.81 0.79 0.80 1581

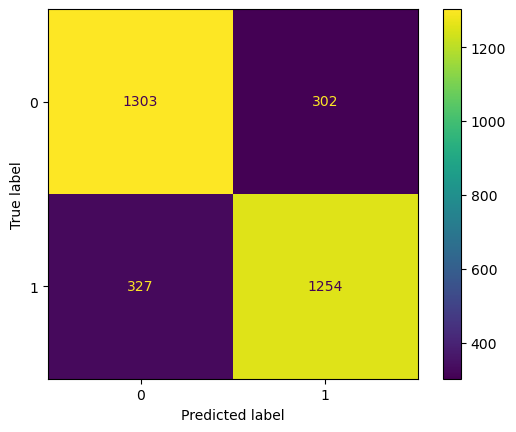
accuracy 0.80 3186

macro avg 0.80 0.80 0.80 3186

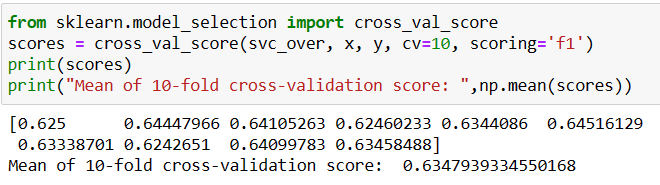
weighted avg 0.80 0.80 0.80 3186

Accuracy score: 0.8025737602008789

The confusion matrix:



The 10-fold cross-validation result:



1. **Discussion and comments**

In this experiment, “churn” is the predicted field, and it only has two unique values, which is a binary classification problem, therefore the main criteria to decide whether the model is succesful or not is accuracy. The accuracy over 0.5 could be considered a successful model. However, with highly unbalanced data, the accuracy score is not enough, it should be combined with the F1 score, which allows combining both precisions and recall into a single measure.

The majority of the datset is labeled “0”, which accounted for nearly 80% customers who retained at the bank and below 20% of cutomers left the bank. Therefore, many of the predictions that the customer retained at the bank would be correct, while customers who left would be misclassified. This could bring trouble to the learning of the model. After the dataset being resampled with undersampling and oversampling techniques, the results improved better. Nevertheless, these techniques also have some weaknesses. In undersampling, this includes removing many random records from the majority class, which could lead to lack of information. Meanwhile random oversampling duplicates the minority can cause overfitting for some models.

The results of oversampling data and predicted with decision tree classifier is the most consistent out of all the methods implemented above. For the SVM model combined with undersampling and oversampling data, it could be due to the weaknesses of resampling methods that cause conflict in the accuracy report and poor performance from model evaluation with 10-fold cross-validation. The mean of 10-fold cross validation of F1 score from the conducted classifier with resampling techniques are presented in the table below:

|  |  |  |
| --- | --- | --- |
|  | Undersampling technique | Oversampling technique |
| Decision Tree Classifier | 0.752 | 0.784 |
| SVM | 0.632 | 0.635 |

SVM model has an disadvantage that is it does not suffer the condition of overfitting, which is the weakness of resampling methods. While one of the advantages of decision tree is data exploration, which means that it allows the exploration of the changing variables. [4] Therefore, results from model evaluation of decision tree is much more ideal than SVM.

**Reference:**

[1] IBM article “What is a Decision Tree?” <https://www.ibm.com/topics/decision-trees>

[2] Techopedia article “Support Vector Machine (SVM)” <https://www.techopedia.com/definition/30364/support-vector-machine-svm>

[3] OpenGenus IQ article “Advantages of Support Vector Machines (SVM)” <https://iq.opengenus.org/advantages-of-svm/#:~:text=The%20other%20important%20advantage%20of%20SVM%20Algorithm%20is,its%20usage%20and%20application%20in%20Machine%20learning%20field>.

[4] CBSE Library article “Top 5 Advantages and Disadvantages of Decision Tree | Types, Pros and Cons, Benefits and Drawbacks” by Prasanna (June 1, 2022) https://cbselibrary.com/advantages-and-disadvantages-of-decision-tree/#:~:text=Advantages%20of%20a%20Decision%20Tree%201%20Easy%20to,so%20unnecessary%20data%20collection%20is%20avoided.%20...%20%E6%9B%B4%E5%A4%9A%E9%A0%85%E7%9B%AE