Churn Modeling for Bank

Aslihan Celik
MS in Computational Science and Engineering
GT ID: 903356676
acelik8@gatech.edu

ABSTRACT

Churn modeling is important for banks to maximize their profit by maximizing the number of customers. This project explores four different machine learning classification methods and compares their performances in order to choose the best model for predicting if a customer will end using bank's services or continue to be a customer.

PVLDB Artifact Availability:
The source code:
Data : https://www.kaggle.com/shrutimechlearn/churn-modelling?select=Churn_Modelling.csv.

1 INTRODUCTION

Businesses have an objective to maximize the number of customers. This objective can be reached in two ways: attracting new customers or retaining the existing customers. The old customer is familiar with the business system as well as business having the necessary information on customers interaction with the service. On the other hand, there is little information available about the new customer that the business is aware of. Since the new customer have less information on the service it may initially require more resources to introduce the business and it can be difficult to work with them as they might be slightly interested in the business. Therefore, it is less costly to retain the old customer than to allocate resources to attract the new customer.

In consideration of this, just like many businesses it is also important for banks to maximize their customers. Therefore, it is important for the bank to predict beforehand the customers who would like to leave so that they can take action in time such as promotions or special offers in order to prevent the client from leaving the bank. This prediction criterion is called "churn" in business terms which implies a measure of how many customers stop using the service.

This project delivers a study of different classification methods applied to this dataset in order to predict the churn. Project explores the churn modeling as binary classification task by giving insights to the most important variable for this prediction and comparison of model performances via score metrics. Analysis is done in Python using jupyter notebook.

2 DATASET

The dataset used for this project, provides details of a bank's customers. There are 12 features including numerical variables of age, credit score, tenure, balance, number of products, estimated salary; categorical variables of customer ID, surname, geography, gender, has credit card, is active member. Tenure refers to how many years the person has been a client of the bank. NumOfProducts refers to the number of bank products the customer is utilising. The "Exited" column represents the target variable values in binary as 1 being customer closed his bank account and 0 being the customer continues to be a customer.

The dataset has 10000 data points. In the original dataset, there is an imbalance between classes as binary 1 class (customer exited the bank) having 2037 and 0 class having 7963 instances. The models are initially implemented on the original dataset and the score metrics can be seen in Table 7. In the imbalanced dataset case, F1 score can be considered instead of accuracy as accuracy can be misleading. The F1 scores were quite low compared to the balanced version of the dataset which the score metrics for this are discussed later in this report. Since the score metrics are improved in the balanced version of the dataset which is obtained by downsampling, the downsampled data which includes 2037 instances for each of the classes is used for the analysis in this report.

3 EXPLORATORY DATA ANALYSIS

Below, you can see the correlation matrix in Figure 1. We can say that there is little linear correlation between variables Age and IsActiveMember. There is also low correlation between variables NumOfProducts and Balance. From the correlation plot we see that none of the explanatory variables are highly correlated with one another.

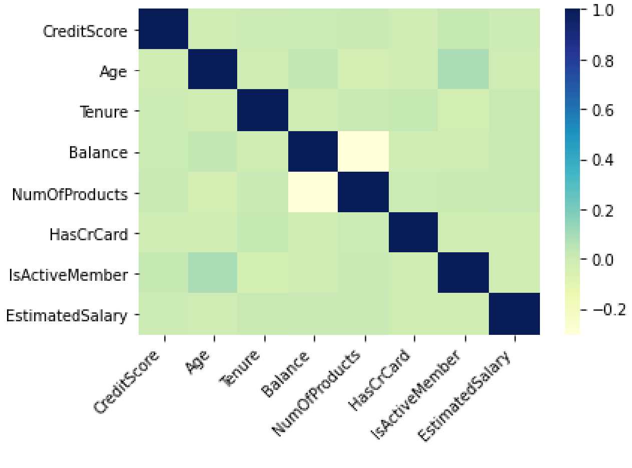

Figure 1: Correlation Matrix

Feature importance obtained via Random Forest can be seen in Figure 2. According to Figure 2, Age is the most important feature in determining if the customer will terminate their subscription to the bank services. The second most important feature is Number of Products that the customer utilizes in the bank.

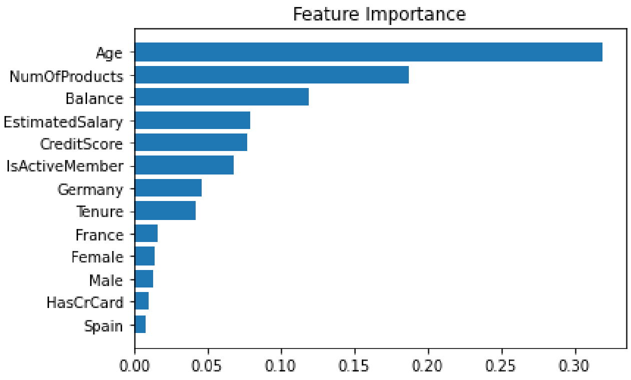

Figure 2: Feature Importance from Random Forest

4 PREPROCESSING

In order to have a balanced dataset, we downsample the majority class by randomly removing the instances that belong to this class. This way, we obtain 2037 instances in each of the classes. Moreover, we one-hot-encode the categorical variables Gender and Geography. Customer ID and Surname variable columns are also dropped since each instance is specific to the client. The dataset is shuffled and the features are standardized as part of the process. The dataset is divided into training and test set as 80% of the data randomly selected as the training set and the rest 20% as the test set.

5 CLASSIFICATION MODELS

For predicting the churn, four different classification models are trained which are SVM, Logistic Regression, Decision Tree and Random Forest. Upon training the models, the hyperparameter tuning is performed via GridSearch. For the mentioned tasks, scikit-learn library in Python is used.

5.1 Hyper-Parameter Tuning

GridSearch is performed with cross validation over a parameter grid in order to optimize the estimator parameters.

5.1.1 Cross Validation. 5-fold cross validation is performed on the training set. To perform this, the dataset is divided into 5 subsets. A subset which is 20% of the original training set is used as a the test set for one round of cross validation and the rest of the four subsets which corresponds to the 80% of training set used as the training set for cross validation in this round. Using the hyperparameter combinations we fit the model 5 times on different subsets of the original training set by using different subset for training and testing each time. Once the model is fit five times for a hyperparameter combination, the average accuracy is taken. After fitting the model 5 times for each hyperparameter combination, the best model average among the hyperparameter combinations is selected as the optimal hyperparameters.

5.2 SVM

For the GridSearch performed for SVM, the considered hyperparameters are as follows:

- kernel: linear, rbf, poly, sigmoid
- C: 0.0001, 0.001, 0.01, 0.1, 1, 10

C is the regularization parameter which is inversely proportional to the strength of the regularization. From the possible combinations for parameters, C=1 and rbf kernel are selected as a result of GridSearch to optimize the estimator parameters. Upon GridSearch, the model is fit with the optimized parameters and the score metrics can be seen in Table 1 . With the optimized parameters, SVM reaches an accuracy of 78% and F1 score of 77%.

Table 1: Score metrics for SVM

Accuracy	0.78
F1	0.77
Recall	0.73
Precision	0.81

5.3 Logistic Regression

Different than rest of the models trained for this project, logistic regression make some assumptions.Logistic regression assumes that there isn't multicollinearity among the explanatory variables. In order to check this assumption, we previously looked into the correlation matrix in section 3. According to this there isn't high correlation between any of the explanatory variables.

For the GridSearch performed for Logistic Regression (LR) , the considered hyperparameters are as follows:

- penalty : l1, l2
- C : 0.001, 0.01, 0.1, 1, 10

The parameter "penalty" is used to specify the norm used for the penalty term in loss function. L2 norm is calculated as the square root of the sum of the squared vector values. L1 norm uses absolute values instead of squared values. This selection is important to introduce bias to the model and to decrease the variance by regularization. The parameter C is the inverse regularization parameter.

C=0.1 and penalty as l2 are selected as a result of GridSearch. Upon GridSearch, the model is fit with the optimized parameters and the score metrics can be seen in Table 2 . With the optimized parameters, logistic regression reaches an accuracy of 70% and F1 score of 70%.

Table 2: Score metrics for LR

Accuracy	0.70
F1	0.70
Recall	0.68
Precision	0.71

5.4 Decision Tree

For the GridSearch performed for Decision Tree (DT) , the considered hyperparameters are as follows:

- criterion: gini , entropy
- splitter: best , random

- max_depth: 3, 5, 7, None

The parameter splitter is the strategy used to make the split decision at each node. The parameter max_depth determines the maximum depth of the tree. Criterion as entropy, splitter as best and max_depth of 7 are selected as a result of GridSearch. Upon GridSearch, the model is fit with the optimized parameters and the score metrics can be seen in Table 3. With the optimized parameters, logistic regression reaches an accuracy and F1 score of 80% both.

Table 3: Score metrics for DT

Accuracy	0.80
F1	0.80
Recall	0.78
Precision	0.82

5.5 Random Forest

Random Forest is an ensemble method made up of a large number of small decision trees. Random Forest is fit to mainly get an idea of the feature importances that is discussed in section 3.

The GridSearch is also performed for Random Forest, the considered hyperparameters are as follows:

- n_estimators: 10, 50, 100, 150
- max_depth: 5, 7, 9, None
- min_samples_split: 2, 4, 8

The parameter n_estimators represent the number of trees in the forest. Max_depth is the maximum depth of the tree. Min_samples_split is the minimum number of samples required to split an internal node according to the sklearn's definition. n_estimators as 100, max_depth as 9 and min_samples_split of 8 are selected as a result of GridSearch. Upon GridSearch, the model is fit with the optimized parameters and the performance metrics can be seen in Table 4. With the optimized parameters, random forest obtains an accuracy of 80% and F1 score of 79%.

Table 4: Score metrics for RF

Accuracy	0.80
F1	0.79
Recall	0.74
Precision	0.84

6 MODEL PERFORMANCES AND COMPARISON

Accuracy is defined as the ratio of the number of correct predictions over the total number of predictions. When this score is considered for the four different models implemented, we see that the highest accuracy is reached by both Random Forest and Decision Tree with 80% then SVM is the third highest with 0.78%. Logistic Regression have the fourth place by means of accuracy with 0.70. While the accuracies are similar for SVM, Decision Tree and Random Forest; Logistic Regression accuracy is low compared to these models. One possible reason for this is that the SVM's kernel and the decision boundary of the decision tree and random forest are not linear while the decision boundary of logistic regression is linear. In addition to this, the data is high dimensional and it may not be linearly seperable. Since the logistic regression performs well on linearly seperable data, here it may not be performing as good as other models which have nonlinear decision boundaries.

In order to check how the SVM would perform when the decision boundary is linear, SVM with linear kernel is also fit. As can be seen from the score metric table in Table 5 in Appendix, SVM performance is worse than the parameter optimized SVM using rbf kernel. When the kernel is set to be linear for SVM, the score metrics were almost the same as Logistic Regression which also have a linear decision boundary. This supports our claim that the dataset might not be linearly seperable and the models with nonlinear decision boundaries perform better when predicting the churn for this dataset.

We can also define other metrics in our context. Precision: of all the customers that labeled as exited the bank, how many actually exited. Recall: of all customers truly exited the bank how many did we actually labeled. For the objective of this project that we described at the introduction, we are interested in both of these metrics. Therefore, we can consider the F1 score as it is the weighted average of precision and recall. Referring to the Table 6 in Appendix, F1 Score for Random Forest(RF) and Decision Tree(DT) are very close. SVM's F1-score is slightly less than them but it can be considered as close. DT is the highest with 80%, RF is the second highest with 79% and SVM follows with 77%. F1 Score for Logistic Regression is comparably less than other models.

7 CONCLUSION

This project implemented different classification models as well as making a performance comparison in predicting the churn. Decision Tree is selected as the best performing method to make churn prediction. Use of the selected best performing classification method would help a bank to predict if a current customer will leave the bank or stay with the confidence coming from the mentioned score metrics. This way the bank will be able to choose to invest effort and resources in their customers to retain them as a customer. Therefore, the bank will not need to invest more money and resources in order to attract new customers compared to retaining its customers. The findings are summarized as follows:

(1) Age is the most important feature in predicting the churn for a bank which is predicting if a customer will exit the bank.
(2) Models that can have nonlinear decision boundaries outperform the models with linear decision boundaries in the case of a data set that may not be linearly seperable.
(3) Decision Tree performs the best as it outperforms all the models in predicting the churn according to the F1 score.

8 APPENDIX

Table 5: Score metrics for Linear SVM

Accuracy	0.72
F1	0.71
Recall	0.70
Precision	0.73

Table 6: F1 Score Comparison

Model	F1 Score
SVM	0.77
Logistic Regression	0.70
Decision Tree	0.80
Random Forest	0.79

Table 7: Scores for Imbalanced Dataset

Model	Accuracy	F1 Score
SVM	0.85	0.56
LR	0.81	0.29
DT	0.84	0.54
RF	0.86	0.58