Bank Churn Prediction

Abstract

When customers of a business stop making purchases or interacting with the firm, this is known as customer churn or customer attrition. Given this, the study explains how to use the information of clients provided by Kaggle to construct supervised learning algorithm models for churn prediction. The study will present the analysis of the prediction results and critical customer characteristics from the classification process.

Introduction

Problem statement: A bank's management is concerned with the increasing churning rate for their credit card services. They would like to predict clients that are leaving so they could reach out to them in advance. They would like to leverage the analysis to improve their services, and influence customers' decisions. 10,000 customers are included in this dataset, together with information about their age, income, marital status, credit card category, and limit. Nearly 18 characteristics are present.

Objective: Based on these problems, the bank would like to predict the clients that are leaving with machine learning results.

Data Source: The dataset was obtained from Kaggle

Target Variable :

0: Existing Customer

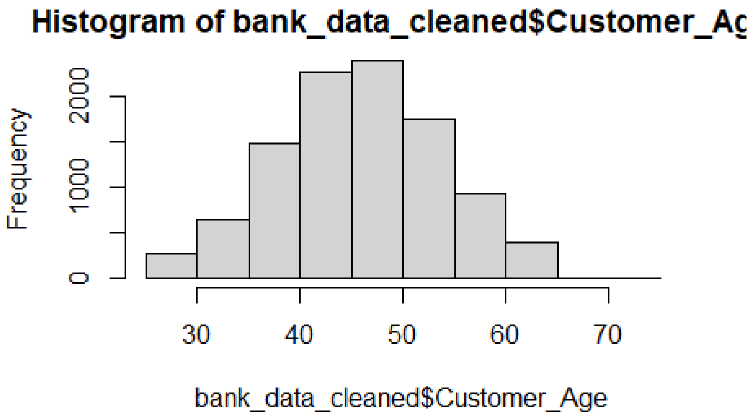
1: Attrited Customer

Chart 1-1 Dataset variables

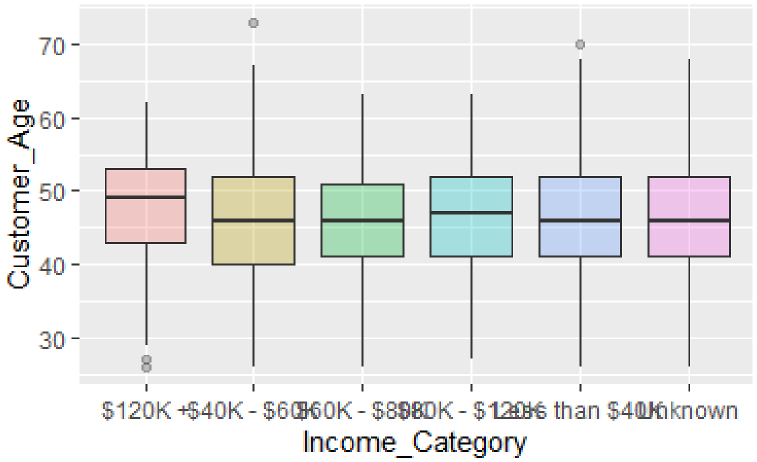
Attribute	**Data Type**	**Description**
Customer_Age	int64	Demographic variable - Customer's Age in Years
Card_Category	object	Product Variable - Type of Card (Blue, Silver, Gold, Platinum)
Months_on_book	int64	Period of relationship with the bank
Total_Trans_Amt	int64	Total Transaction Amount (Last 12 months)
Avg_Utilization_Rate	int64	Average Card Utilization Ratio

2. Exploratory Data Analysis

The ages of customers are normally distributed. Hence, the normality assumption may be used moving forward when using the age feature.

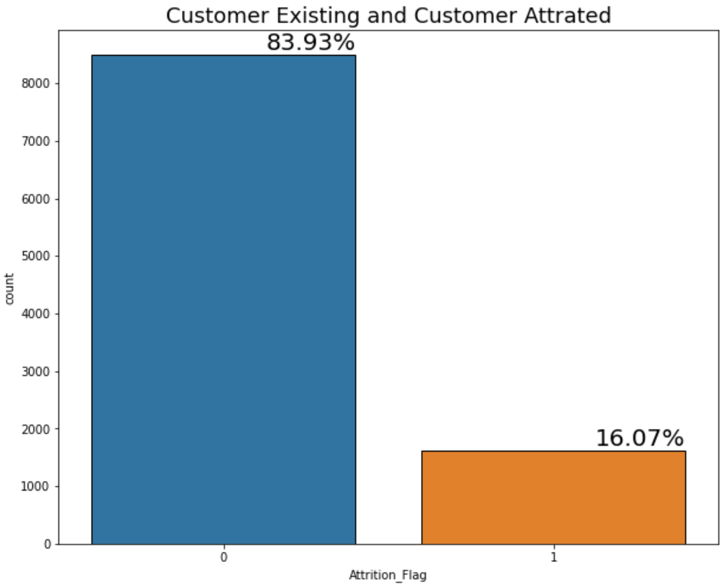


- image 2-1 Customer age distribution



- Image 2-2 Relationship between income level and customer age

As we could see from the histogram chart, the distribution of the target variable showed that the dataset was imbalanced: there were only 16.07% of customers that would leave the credit card service. Therefore, we need to fix this.



- Image 2-3 target variable distribution

The dataset does not have any missing values.

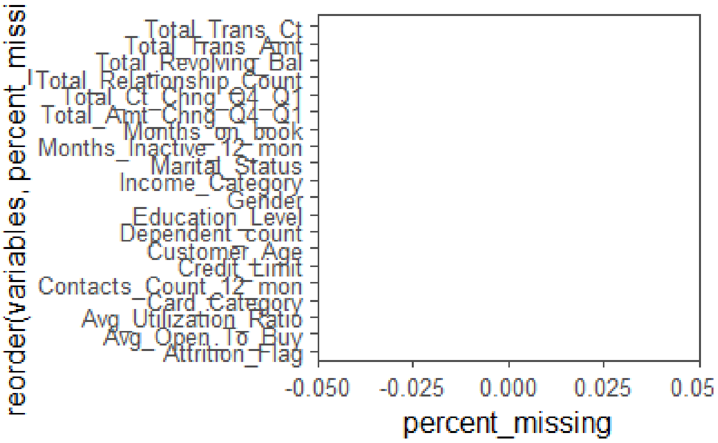
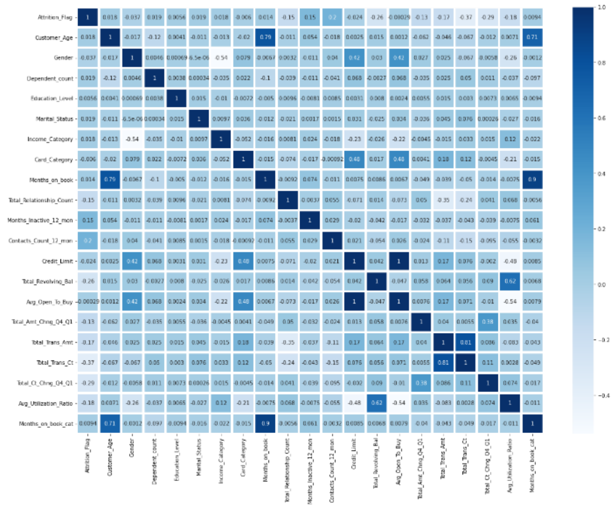


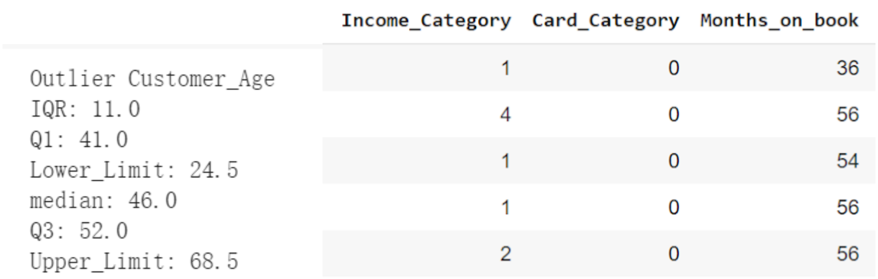
Image 2-4 missing values

The correlation between each variable is displayed in each cell of the table. In the darker blue cell below, for instance, "Total Trans Ct" and "Total Trans Amt" have a 0.81 correlation coefficient, indicating a strong positive association.



- Image 2-5 Heatmap

With the deployment of the IQR method, we could filter out five outliers. We first spot outliers outside of Q1 and Q3 using the method. Outliers are any values that lie outside of this range. We multiply the IQR by 1.5, deduct Q1 from this result, and add Q3 to the result.

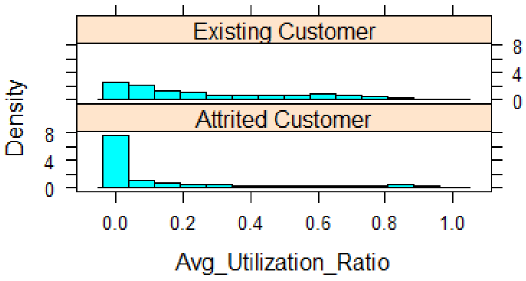


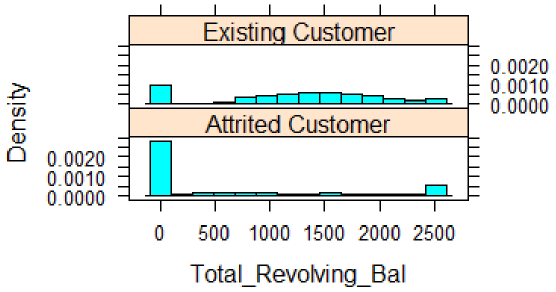
```
Outlier Customer_Age
IQR: 11.0
Q1: 41.0
Lower_Limit: 24.5
median: 46.0
Q3: 52.0
Upper_Limit: 68.5
```

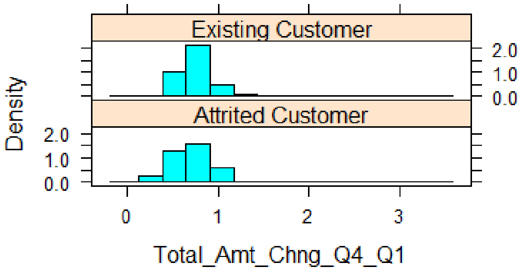
Income_Category	Card_Category	Months_on_book
1	0	36
4	0	56
1	0	54
1	0	56
2	0	56

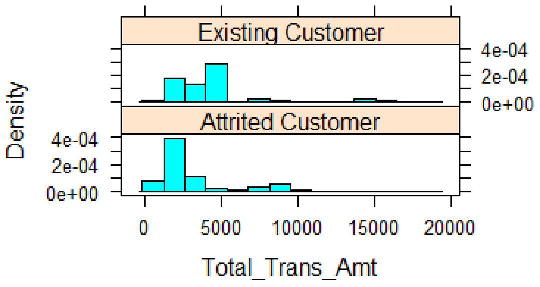
- Image 2-6 checking outliers

The distributions of variables do vary for existing customers and attrited customers. For example, attrited customers have lower average utilization ratios, total revolving balance, and total transaction amount compared to customers that are not leaving.





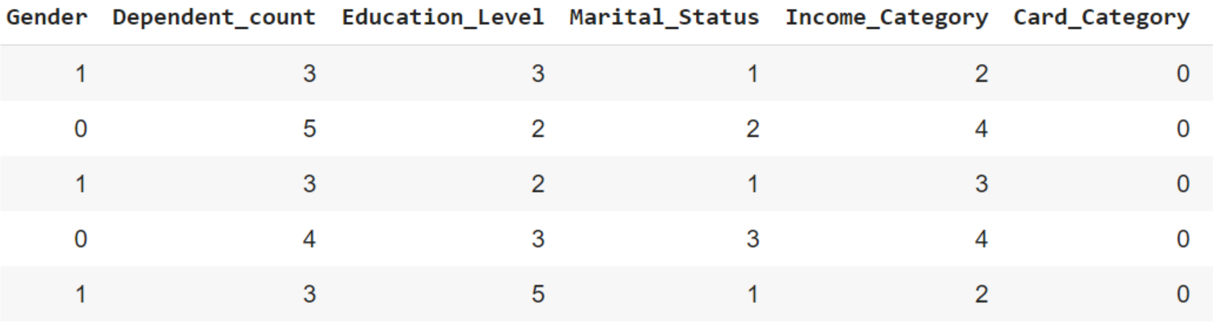




3. Data preprocessing

3-1 Label Encoding

I used the method of label encoding to handle categorical variables. Label encoding assigns numeric values for the labels so that they would be machine-readable.

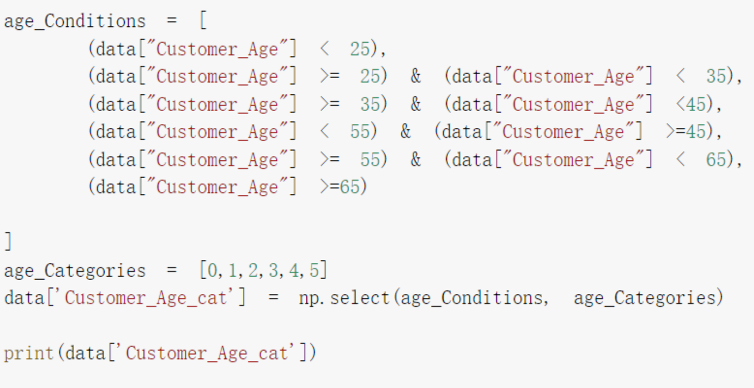


Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category
1	3	3	1	2	0
0	5	2	2	4	0
1	3	2	1	3	0
0	4	3	3	4	0
1	3	5	1	2	0

- Image 3-1-1

3-2 Converting Numerical variables to Categorical variables

“Customer_Age” and “month_of_book” are continuous variables, which are problematic when building machine learning models. Therefore, these values have to be transformed into categorical values. For example, in "Customer_Age”, binning was utilized to address the problem.

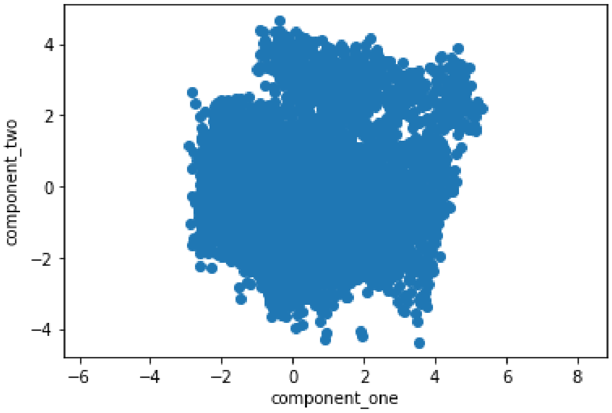


```
age_Conditions  =  [
        (data["Customer_Age"]  <  25),
        (data["Customer_Age"]  >=  25)  &  (data["Customer_Age"]  <  35),
        (data["Customer_Age"]  >=  35)  &  (data["Customer_Age"]  <45),
        (data["Customer_Age"]  <  55)  &  (data["Customer_Age"]  >=45),
        (data["Customer_Age"]  >=  55)  &  (data["Customer_Age"]  <  65),
        (data["Customer_Age"]  >=65)

]
age_Categories  =  [0,1,2,3,4,5]
data['Customer_Age_cat']  =  np.select(age_Conditions,  age_Categories)

print(data['Customer_Age_cat'])
```

- Image 3-2-1

3-3 PCA

“Principle component analysis” will be used to minimize the dimensionality of the label-encoded categorical variables, which will result in the loss of some variance. Employing a few principal components rather than a large number of encoded features will result in better model performance.



- Image 3-3-1

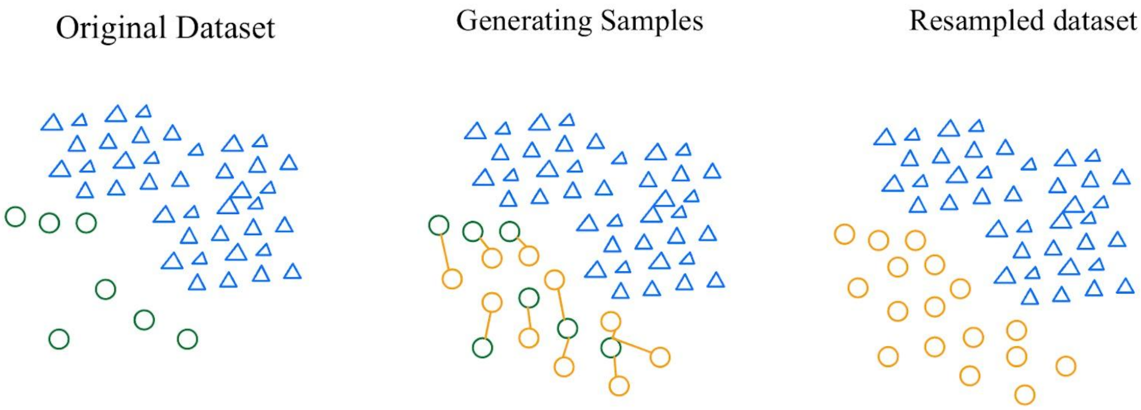
3-4 Upsampling with SMOTE

Since only 16% of samples represent attrited clients, the SMOTE method would be used to address the issue of the data set's unbalanced classes.

SMOTE:

Oversampling the minority class is one way to deal with unbalanced datasets. Duplicating examples from the minority class is the simplest method but will not provide new insights into the model. In contrast, SMOTE can create fresh samples by synthesizing the old ones.

As indicated in the image below, SMOTE selects examples in the feature space that are close to one another, draws a line, and then creates a new sample at a location along the line.



- Image 3-4-1

4. Analysis - Machine Learning using 5 different models

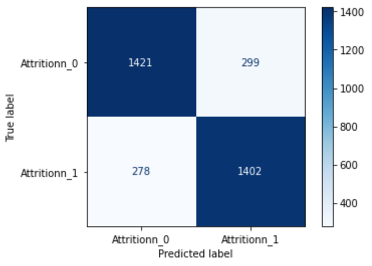
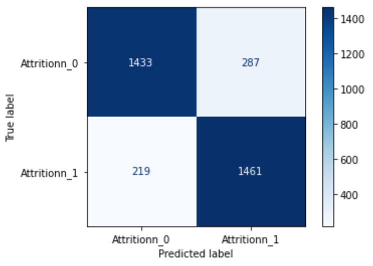
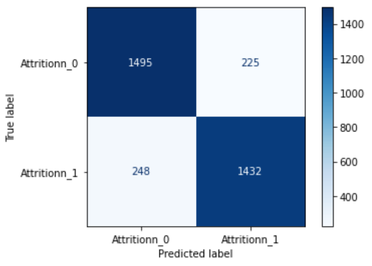
I used five machine learning models for comparison.

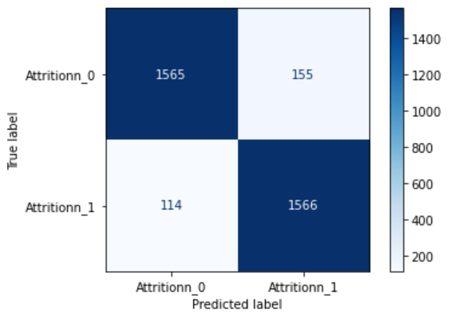
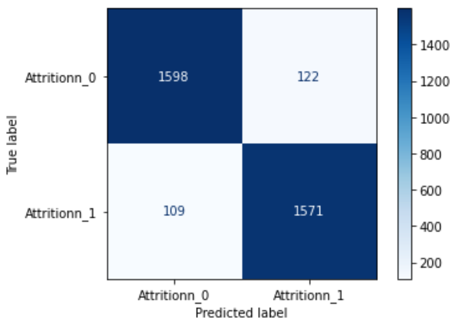
1. AdaBoost Classifier
   AdaBoost classifier is an iterative ensemble boosting method that combines weakly performing classifiers to obtain strong classifiers.
2. Decision Tree Classifier
   In Decision Tree Classifier, we begin at the tree's root when anticipating a record's class label. We check the root attribute's values with that of the attribute on the record, follow the branch that corresponds to that value, and go on to the next node based on the comparisons
3. Gradient Boosting
   Gradient Boosting is an algorithm that combines multiple weak models to have the best results as a whole.
4. Random Forest Classifier
   Random Forest Classifier is an algorithm made up of decision trees. It produces an uncorrelated forest of trees by using bagging and feature randomness when generating each tree.
5. Extra Tree Classifier
   ExtraTrees is an ensemble machine learning approach, similar to Random Forests, that trains a large number of decision trees and combines the output of the group of decision trees to produce a forecast.

Comparison:
Of all the machine learning models, Extra Tree Classifier performs the best with the highest accuracy rate, followed by Random Forest, Gradient Boosting, Decision Tree, and AdaBoost. The accuracy rate of the Extra Tree Classifier is 0.93, while the accuracy rate of AdaBoost is 0.85.
Even though the accuracy rates of models without SMOTE method do not drop much compared to models with SMOTE, the precision rates largely decrease.

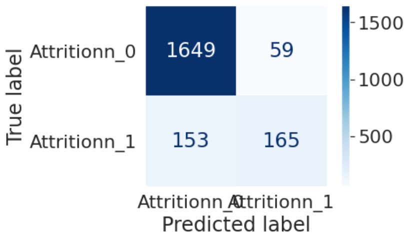
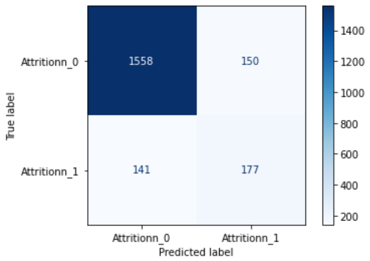
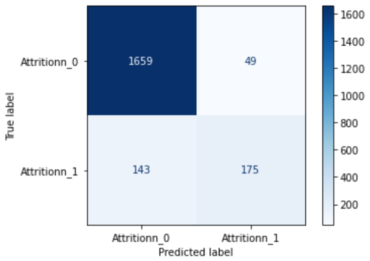
model	accuracy rate	precision rates
AdaBoost (SMOTE)	0.83	0.83
AdaBoost	0.83	0.52
Decision Tree (SMOTE)	0.86	0.87
Decision Tree	0.86	0.56
Extra Tree(SMOTE)	0.93	0.94
Extra Tree	0.93	0.48
Random Forest(SMOTE)	0.92	0.93
Random Forest	0.92	0.55
Gradient Boosting(SMOTE)	0.86	0.85
Gradient Boosting	0.86	0.55

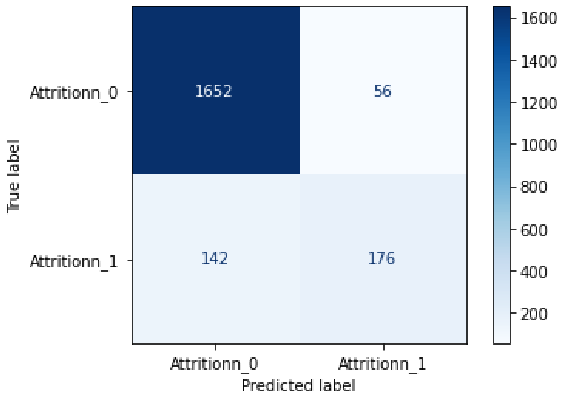
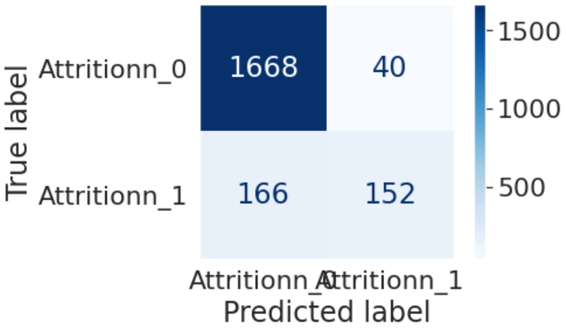
(1) With SMOTE method

AdaBoost	Decision Tree	Gradient Boosting
		

Random Forest	Extra Tree
	

(2) Without SMOTE method

AdaBoost	Decision Tree	Gradient Boosting
		

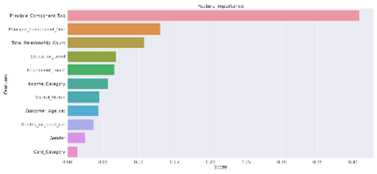
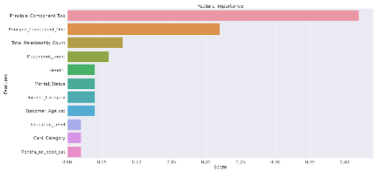
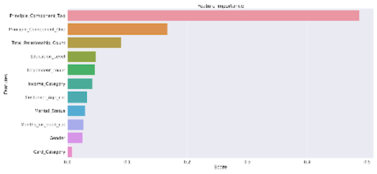
Random Forest	Extra Tree
	

Feature Importance

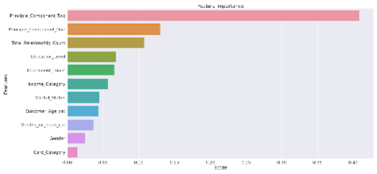
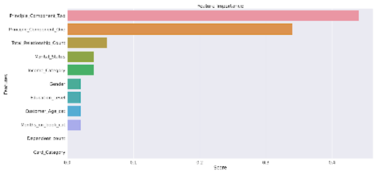
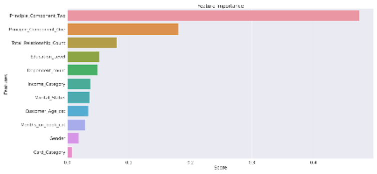
The results of models with and without SMOTE method all share similar important features. “Total relationship count”, “education level” and “dependent count” are features that ranked top on the list.

(1) With SMOTE method:

Extra tree classifier	AdaBoost classifier	Random Forest classifier

(2) Without SMOTE method:

Extra tree classifier	AdaBoost classifier	Random Forest classifier
		

5. Conclusions

- Dataset Problem: Solving the imbalance problem of the dataset can help increase the accuracy rate of models.
- Model Selection: Extra tree classifier performs better than all the other algorithms in terms of accuracy rate and recall rate.
- Important Features: “Total relationship count”, “education level” and “dependent count” are important features.
- Marketing Implications: Knowing which customers will leave allows us to offer promotions to keep them from leaving or switching banks.

6. Reference

- Saini, A. (2022, December 1). *AdaBoost Algorithm – A Complete Guide for Beginners*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/
- *Decision tree algorithm, explained*. KDnuggets. (n.d.). Retrieved December 22, 2022, from https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
- Yiu, T. (2021, December 10). *Understanding Random Forest - Towards Data Science*. Medium. https://towardsdatascience.com/understanding-random-forest-58381e0602d2