OliverWyman

# Project Report

Ghodrat Rezaei

## Introduction

This report is aimed at explaining the methodology that was chosen to analyze and do the binary classification based on clients' data set for prediction of Default status of the clients. Available information is related to the clients' features including Occupation, Marital Status, Observation Date, Income, Loan_amount_requsted, Term Length, Installment/Income, Schufa credit Score, Number of Applicants.

Generally, we have 10000 number of observation and 10 features for years (2008-2018) time intervals which in each date there are set of observation falling and the class target(target_var) is binary values of 0 and 1 which shows the status of the clients' application whether is rejected or approved.

## 1. Data Visualizing and Preprocessing

### 1.1. Data Quality Assurance

As the first step in the data processing and preparation, the time "00:00:00" was dopped from ('OBS_DATE'), because there is no meaning and time "00:00:00" is same for all data, and format of the date in OBS_DATE variable was changed to standard date format.

Then we check if the target values are distributed equally or no, which is shown is figure1.

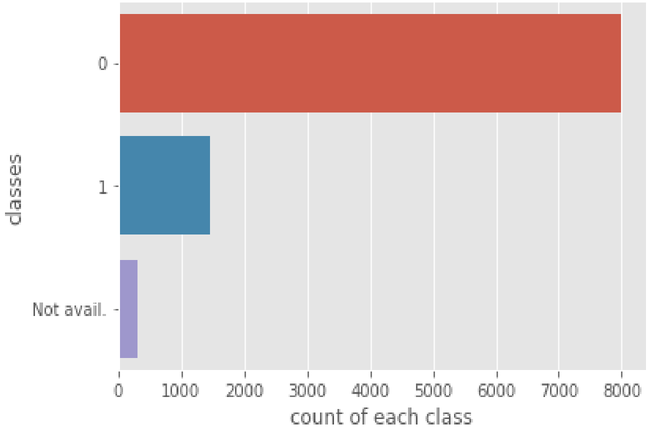


*Figure 1*

As can be Seen in above figure, there is huge discrepancy between target 0 and 1. Targets with "Not avail." values are removed in the next steps of preprocessing which is called Missing Values Removal and Imputation. This difference will be balanced before modeling process. balancing target values is called Sampling.

Then we checked whether there are missing to remove them. Total Missing values for data frame is 2968 and all data we have is 10000. Having such high proportion of missing values, we cannot easily delete corresponding rows, because we would lose one-third of the available data. Source of missing values can be Missing Completely at Random (MCAR), Missing at Random (MAR), Missing Not at Random (MNAR). In this case, it seems to be that Missing values are of type MCAR or MAR, they can be deleted. To be mentioned that its missing values type is MNAR, then it should not be deleted. Deleting all corresponding rows of missing values, evaluation index of above 90% for ROC criteria (Logistic Regression Classifier model) is obtained which will be discussed at model selection part. Despite the previous mentioned scenario deleting missing values, Missing Value Imputation can also be implemented over missing values. Following this Scenario leads to a very poor performance of the models (about 50%-40%), showing that data are very personalized and sensitive.

Next step is to check if we have duplicated value, and there is not any duplicated value.

## 1.2. Data Visualization

Having two kinds of features which are Categorical Features which has been already one hot encoded (occup, marital, OBS_DATE) and Numerical attributes which have integer or float values (, Income, Loan_amount_requsted, Term Length, Installment/Income, Schufa credit Score, Number of Applicants), we separated them into two group, categorical and numerical attributes.

### 1.2.1. Categorical Features

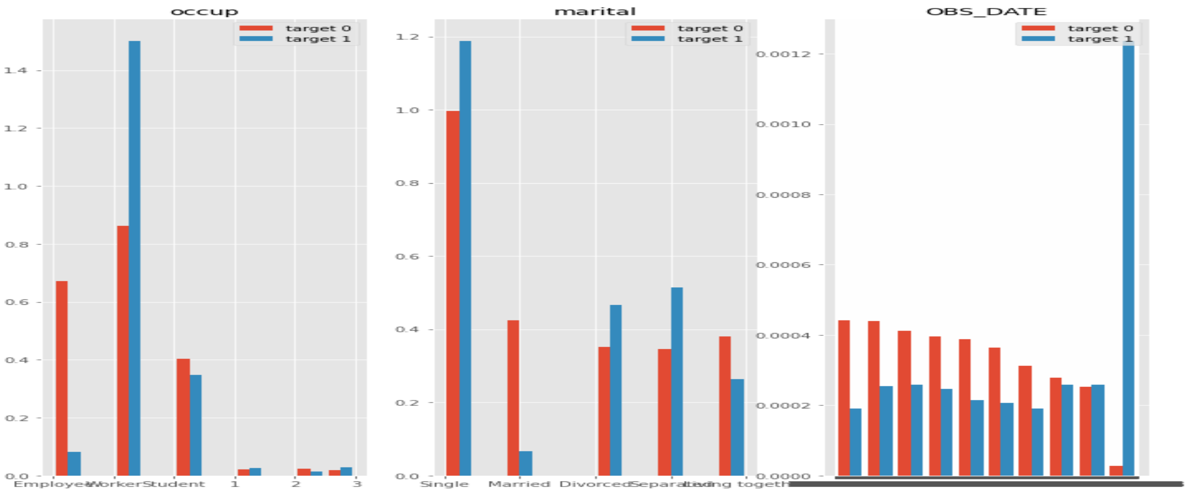


Figure 2

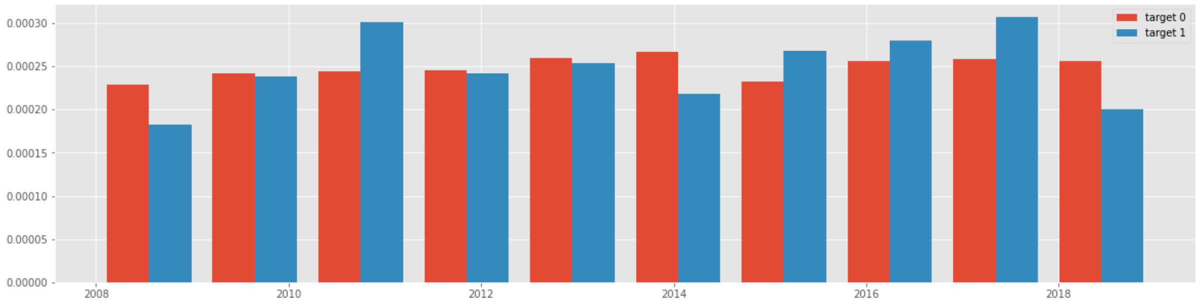


*Figure 3*

Diagram above (figure 2) shows quit the same distribution of the target over the train years (2008-2016) and test years (2016-2018).

### 1.2.2. Numerical Features

As shown in figure 4, Our original continuous data do not follow the bell curve, we can apply logarithmic or gamma transformation to this data to make them as “normal” as possible so that the statistical analysis results from this data become more valid. Depending on the distribution shape of the variables, different transformation can be applied to them. The log and gamma transformation reduces or removes the skewness of our original data.

Gamma Transformation has one constant called lambda, showing the skewness intensity of the distribution. There is a special library for the gamma transformation called boxcox, which is used in cell below.

In the histogram figure(figure 3) above, install_to_inc variable distribution has logarithmic behavior with positive Skewness (left-directed tail), and Schufa, income, loan_amount, term_length have gamma distribution with positive Skewness (left-directed tail), enabling us to apply logarithmic and gamma transformation equivalently.

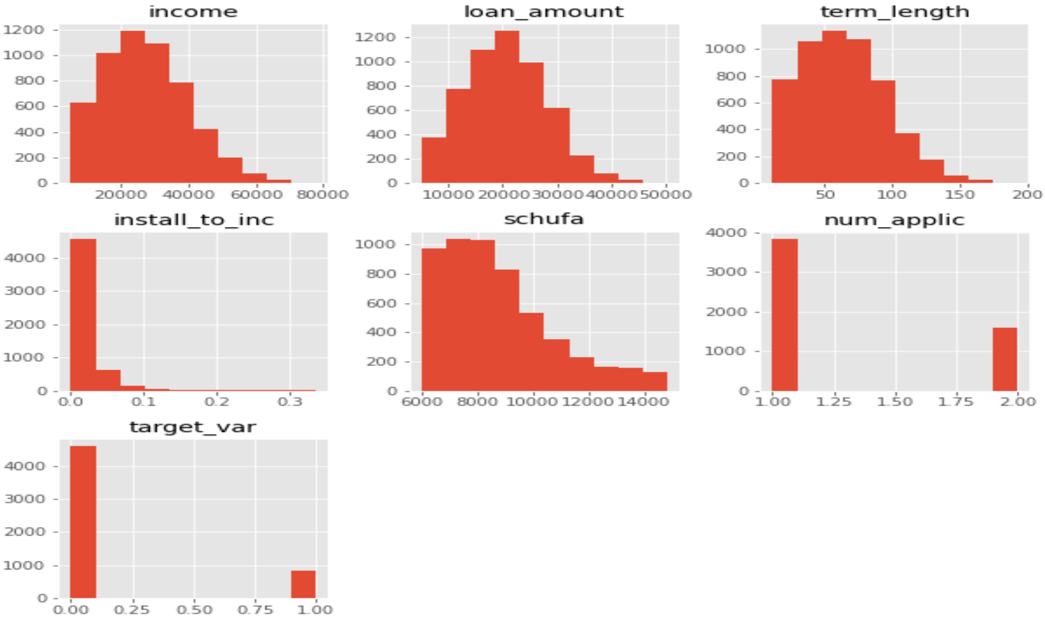


*Figure 4*

.

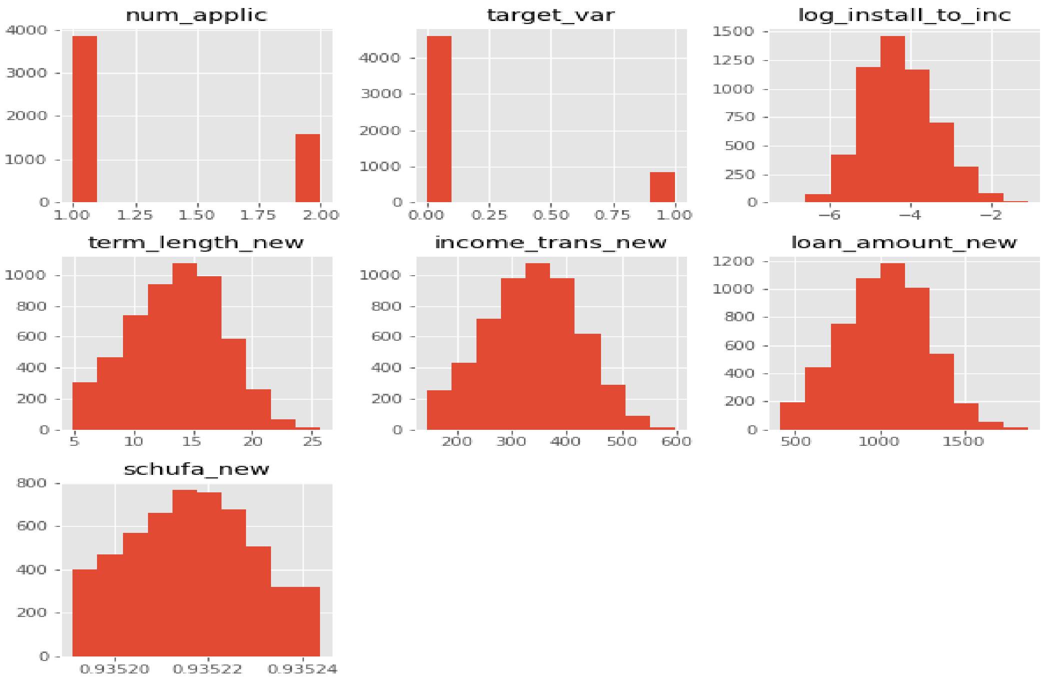


*Figure 5*

Transformed distribution shown above (figure 5) are close to normal distribution and **do not show any outlier**. Therefore, anomaly detection is not applied in this case. To be mentioned that these features are in correlated with target, meaning that high values of the features help to have a more rigid estimation of the target values.

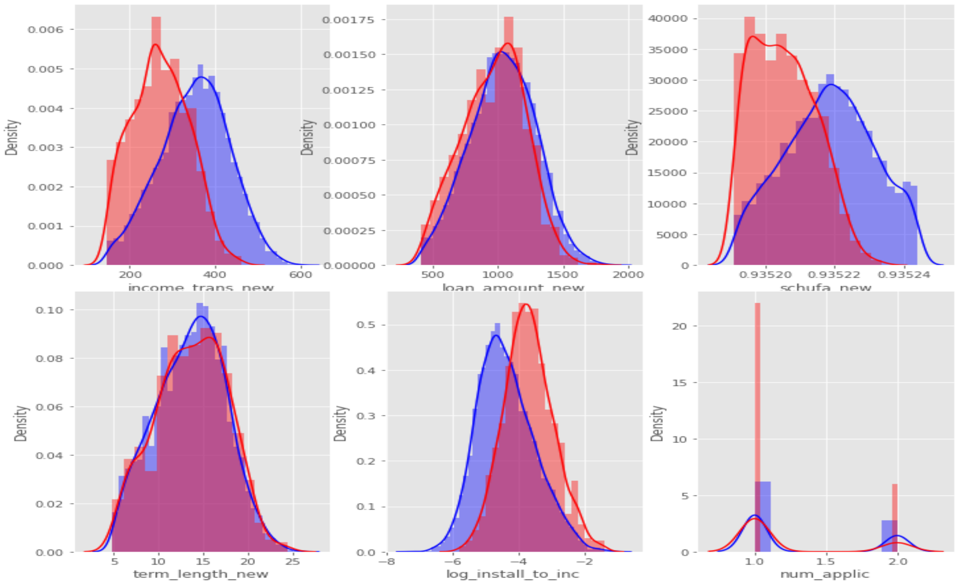


*Figure 6 - Univariate Distribution*

As can be seen above (figure 6): schufa new, log_install_to_inc and income_trans_new have two different diagrams of 0 and 1 classes, meaning that these features are more representative and important to be used in model. more features' targets are separatable, more informative and useful it is for the modeling process. On the contrary, num_applic, term_length and new amount have target diagrams of 0 and 1 which are completely overlapped, explaining that doing classification using these features will not be useful very much. However, in the feature Selection part, this concept will be more investigated and also will be shown that schufa_new, log_install_to_inc and income_trans_new are sequentially most important features as previously was discussed.

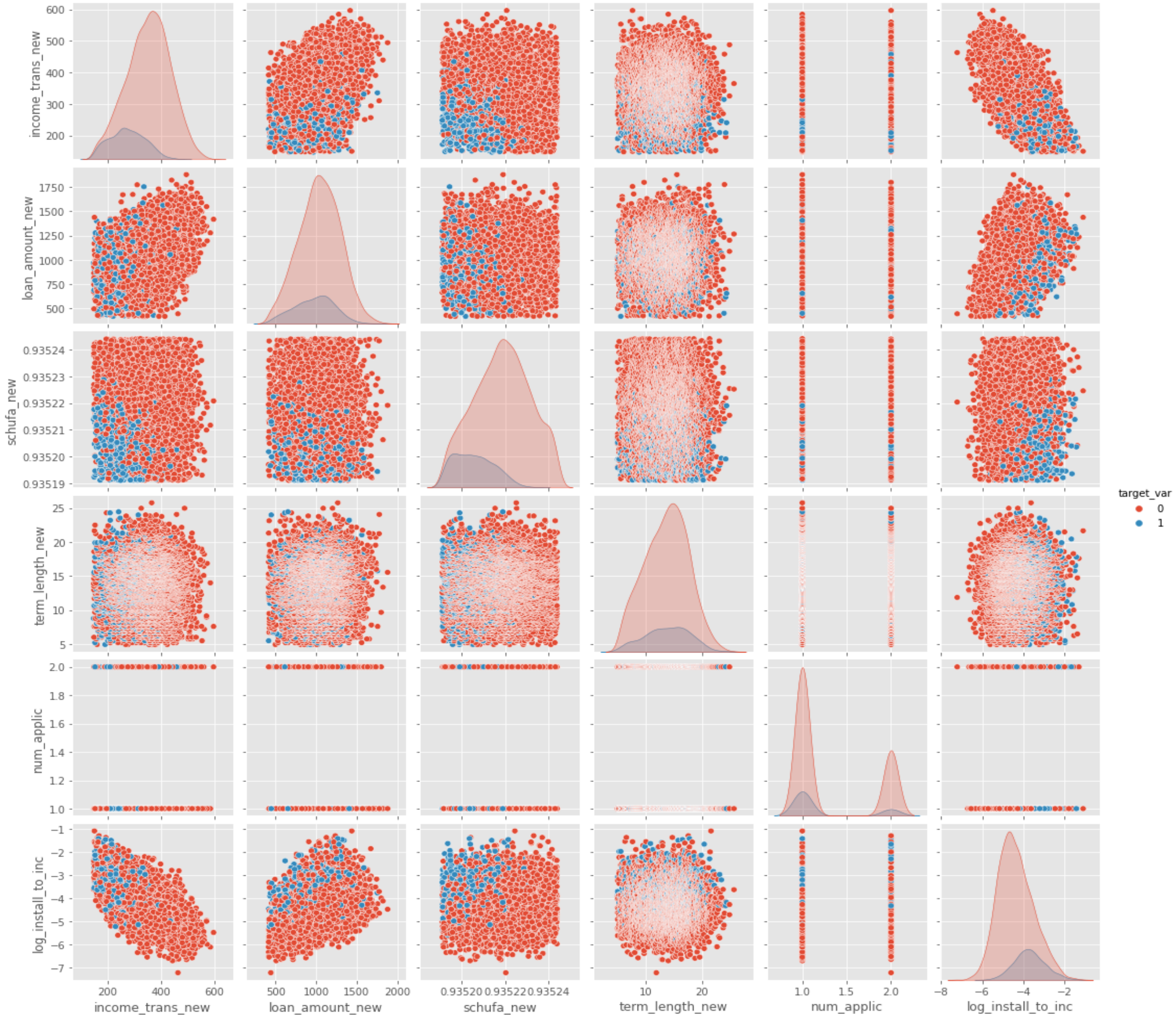
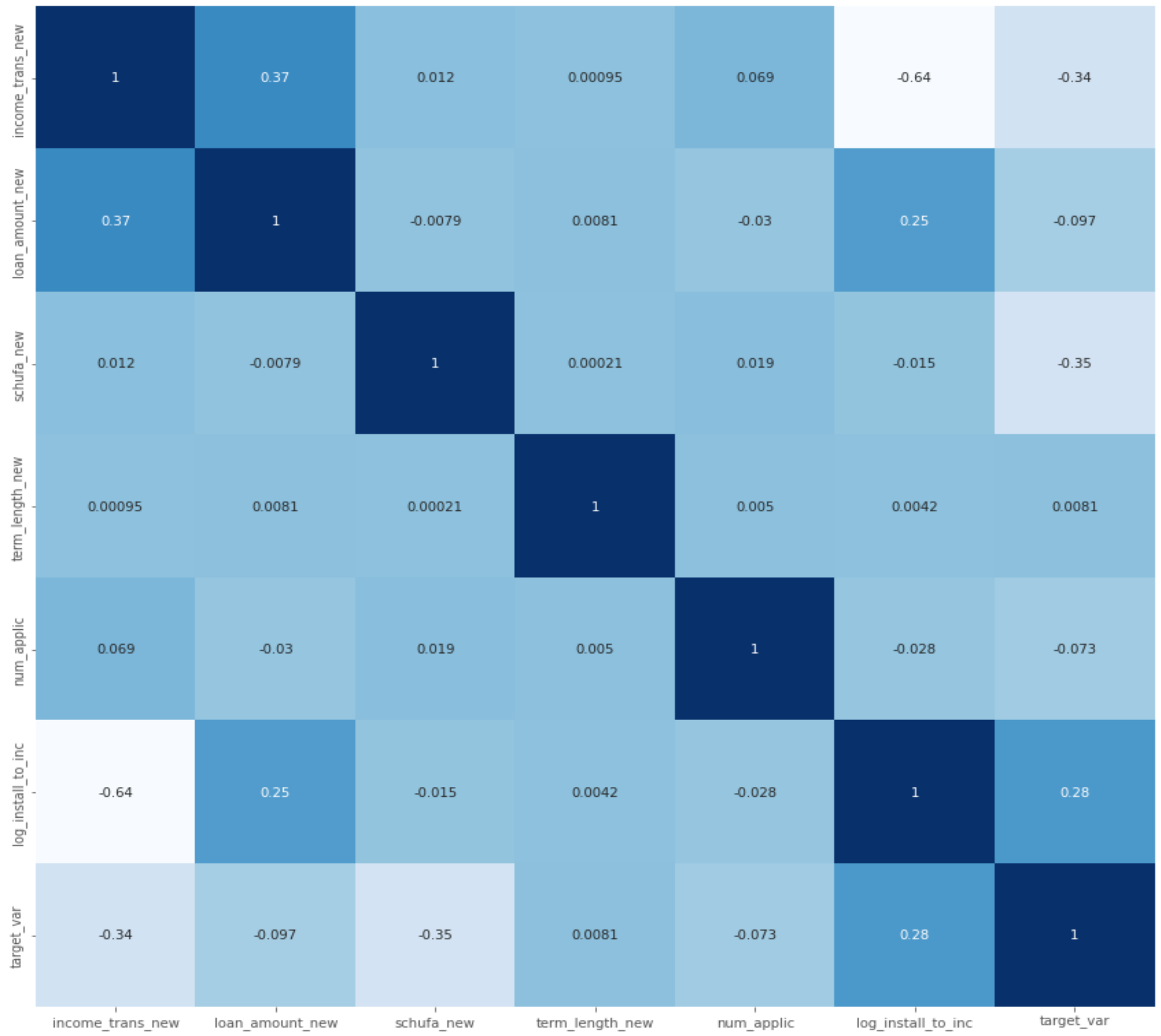


*Figure 7 - Bivariate Distribution*



| | income_trans_new | loan_amount_new | schufa_new | term_length_new | num_applic | log_install_to_inc | target_var |
|---|---|---|---|---|---|---|---|
| income_trans_new | 1 | 0.37 | 0.012 | 0.00095 | 0.069 | -0.64 | -0.34 |
| loan_amount_new | 0.37 | 1 | -0.0079 | 0.0081 | -0.03 | 0.25 | -0.097 |
| schufa_new | 0.012 | -0.0079 | 1 | 0.00021 | 0.019 | -0.015 | -0.35 |
| term_length_new | 0.00095 | 0.0081 | 0.00021 | 1 | 0.005 | 0.0042 | 0.0081 |
| num_applic | 0.069 | -0.03 | 0.019 | 0.005 | 1 | -0.028 | -0.073 |
| log_install_to_inc | -0.64 | 0.25 | -0.015 | 0.0042 | -0.028 | 1 | 0.28 |
| target_var | -0.34 | -0.097 | -0.35 | 0.0081 | -0.073 | 0.28 | 1 |

*Figure 8 - Numerical variables correlation*

Numerical Independent Features are not correlated with each other. Maximum Correlation between numerical features is 0.35, meaning that they are internally independent. So, it is not allowed to remove any of the feature. To be also mentioned that schufa and income features have the highest correlation (negative correlation) with the target. This is also a confirmation of what previously was discussed about feature importance and predictive capability.

## 1.3. Data Sampling and Splitting

After concatenating one-hot-encoded Categorical (dummies) and Numerical features, sampling and splitting was performed.

### 1.3.1. Data Sampling

Due to huge discrepancy between target 0 and 1, discussed previously, I use oversampling to increase the minority class (target 1) up to 6000. Under sampling method is not used, because we may lose information. Over-Sampling is also for preventing overfitting. Having less data for the more complex model leads to poor performance and overfitting, especially when we are using models like ANN (Deep Learning).

### 1.3.2. Data Splitting

To split the data based on the date, data should be sorted based on their date.

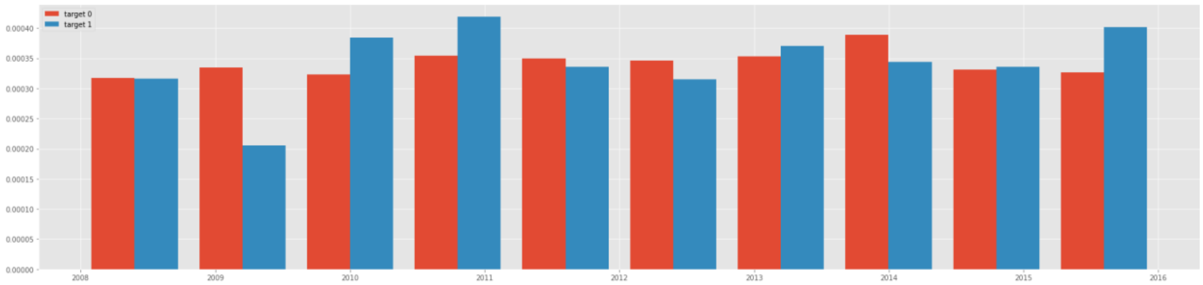


*Figure 9 - Target Distribution over Training Dataset (2008-2016)*

As can be seen in above picture (figure 9), the distributions of targets 0 and 1 are approximately the same over the years, justifying that model used in next steps, do not have bias behavior toward the majority class. In other words, future model will not use the majority voting approach.

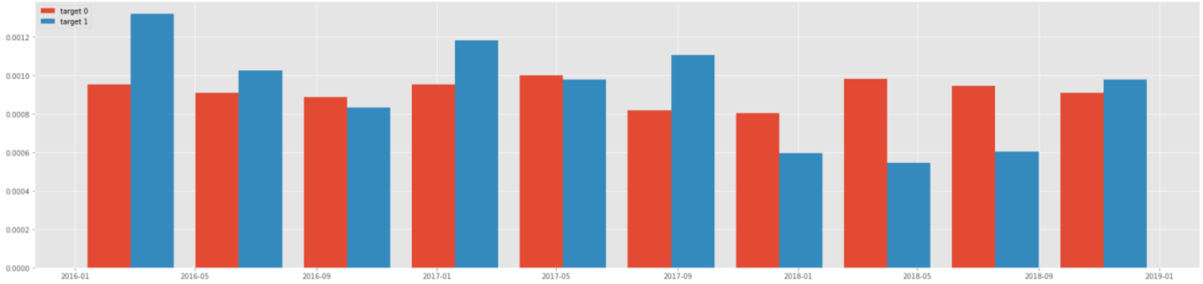


*Figure 10 - Target Distribution over test data set (2016-2018)*

As shown in Figure 10, the distributions of targets 0 and 1 are also approximately the same over the years, explaining that for that evaluation process, model is not using majority voting approach. Also, same consistency between training and test dataset shows the quality of data prepared.

# 2. Feature Selection

## 2.1. Random Forest Feature Selection

Since we have Numerical features and one-hot-encoded features which can also be classified as numerical variables, random forest would give weight equally to numerical and one-hot-encoded variables. Random Forests and decision trees, in general, give preference to features with high cardinality (Trees are biased to these type of variables). Categorical variables are said to possess high cardinality, because there are too many of their unique values, but when they are one-hot encoded their cardinality behavior will decrease. Therefore, using Random Forest for feature selection would be a good choice, and it will give us the feature relative importance (figure 11), because as mentioned it is performing like Decision Tree.

A common approach toward feature selection is to do it first on training dataset and then implement accordingly on test dataset. however, this is not a hard rule. In some cases, feature selection could happen before splitting data set.

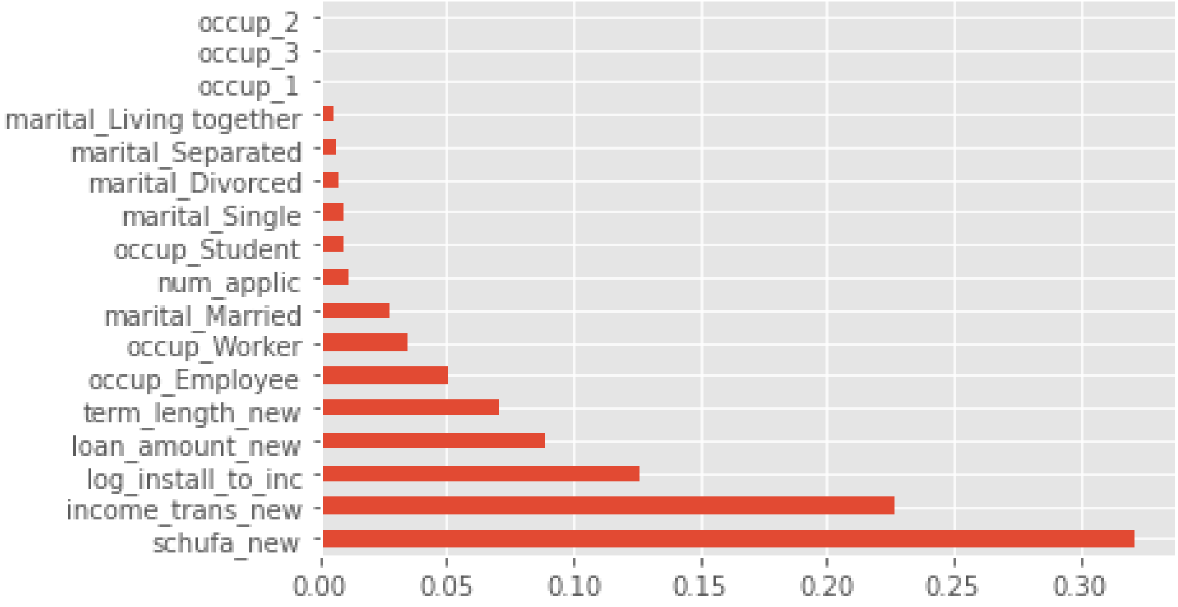


*Figure 11 - Random Forest feature importance*

As shown in above picture, sum of importance of all these features is 1, and schufa, income and install_to_inc have highest importance sequentially.

## 2.1. Feature Selection using Heatmap

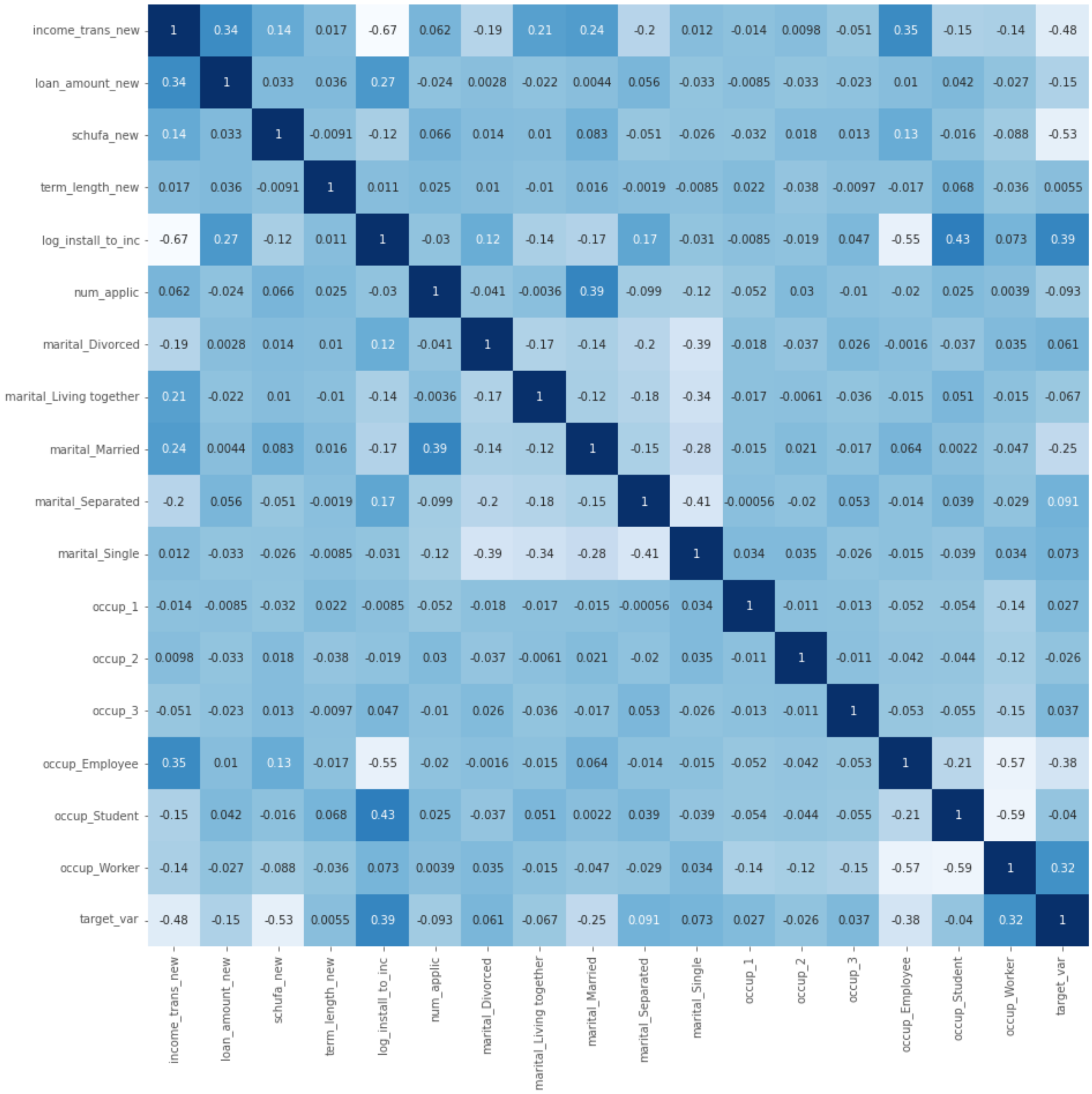


*Figure 12 - Heatmap Correlation*

As previously discussed about the concept of heatmap, schufa, income and occup_Employee have sequentially -0.53 and -0.48, -0.38 correlation with target feature, serving the interpretation of the random forest feature selection that showed schufa and income are the highest important variables. Negative values means that for example if income of the client is high, he or she is less likely to get rejected.

# 3. Modeling (Processing)

In modeling process, we use all features. However, if we use different set of the features based on their importance which was discussed in feature selection part (2), we will confront with an interesting phenomenon which will be discussed.

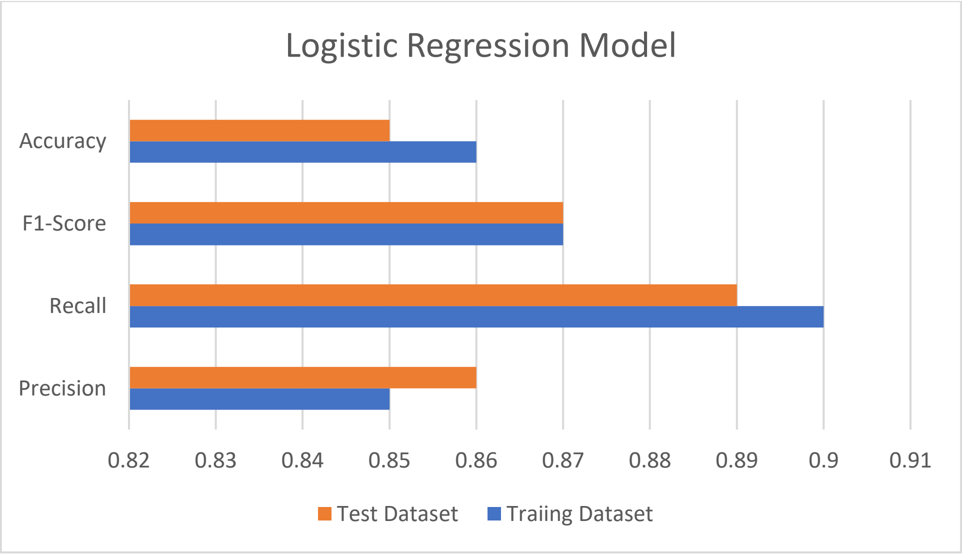
I used the Python library Sklearn to get all our models. Different models were implemented including Logistic Regression, KNN, Decision Tree, Naïve Bayes, Support Vector Machine, Adaboost (Ensemble Methods), Random Forest (Ensemble Method) and MLP (Deep Learning).

## 3.1. Logistic Regression Model

Logistic Regression is a "Supervised machine learning" algorithm that can be used to model the probability of a certain class or event. It is used when the data is linearly separable, and the outcome is binary or dichotomous in nature. That means Logistic regression is usually used for Binary classification problems.

As base Scenario, I used all features. Performances which are shown in table1, figures 13 and 14 were obtained using all features. However, the analysis of performance using different sets of features is discussed in next page.

*Table 1 - Logistic Model Performance*



The mean difference between model performances over training dataset and test dataset, which is shown in table1, is about 1 percent for all criteria, meaning that the model is not overfitted over training dataset and hence can be considered as a robust model.

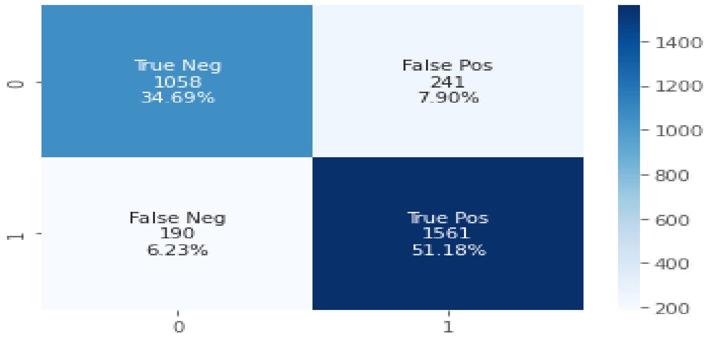


*Figure 13*

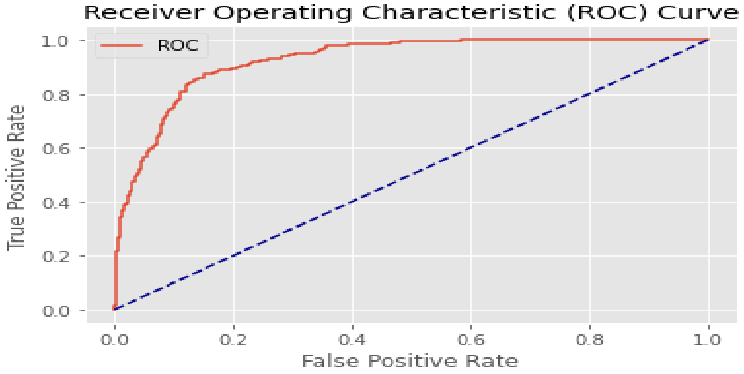


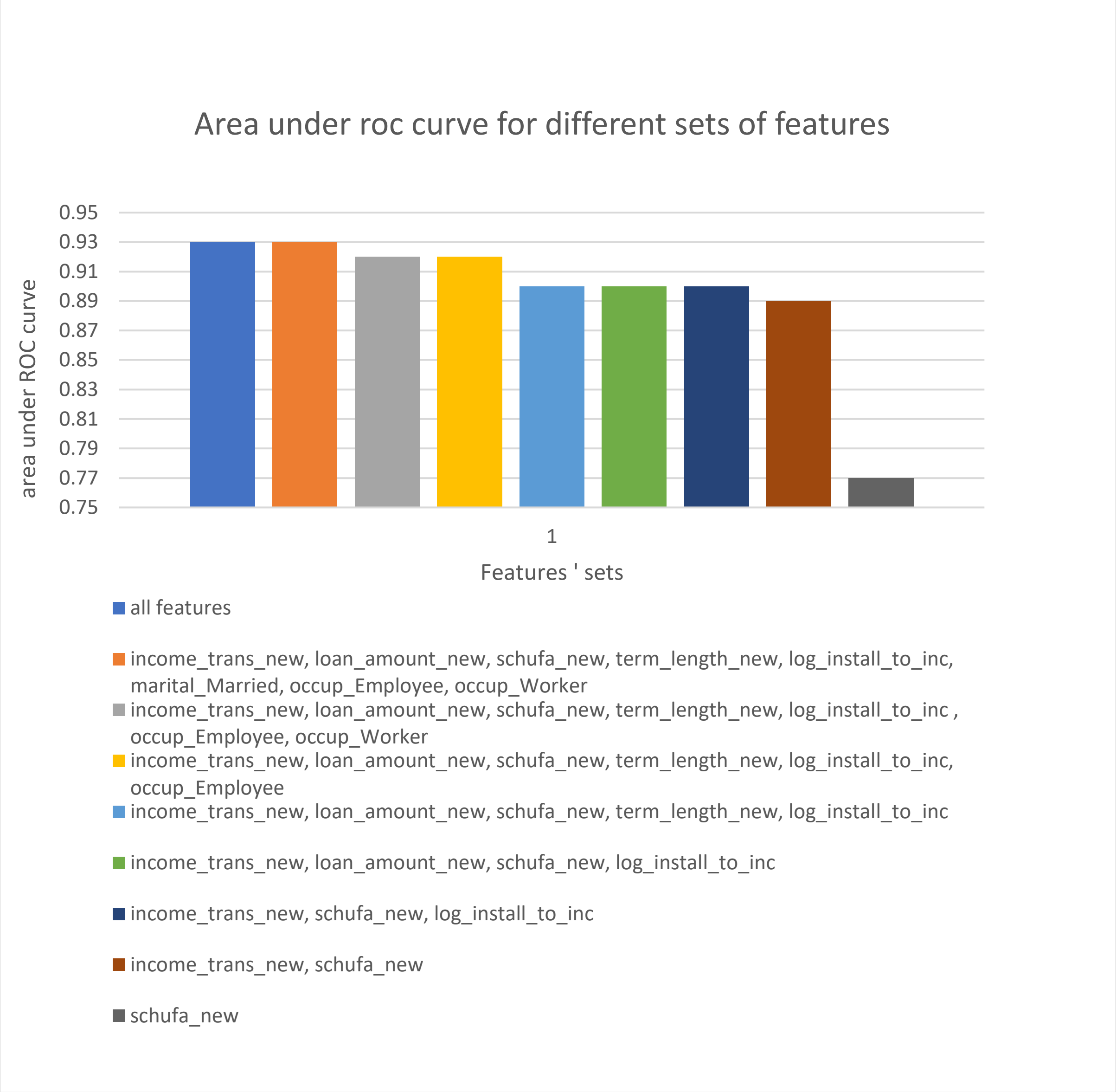
*Figure 14*

If we use all features, area under roc curve for logistic regression model is 0.93. If some features are dropped the performance changes are compatible with the feature selection analysis(part2).

In table 2 different sets of features are used, and their according performance of the model show that the “schufa”, “income” and “occup-Employee” (dummy variable; whether the client is employee or no) are sequentially most features.

Using only the variable "schufa_new" give us a performance of 0.77 for the area under roc curve, meaning that schufa (SCHUFA credit check (aka SCHUFA-Bonitätscheck) provides information about a customer's ability to pay their bills regularly, also known as 'creditworthiness') is very important and predictive. To be mentioned that, in this case we do not have a large dataset and is okay to use all variables for the modeling, but for big data it is better to use most important variables like: schufa, income, install_to_inc and loan_amount. Big data needs more time to be processed. Therefore, removing some features which are not very predictive helps us to run the algorithm faster, although there might be a slight decrease in performance which was discussed.

*Table 2 – Performance of different Features' sets*



## 3.2. Business Analysis

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between bad customers and good customers.

In our policy, we want to avoid giving loan to those who are not capable of return it, and also giving loan to those who can return it back. To decrease the risk of not losing from money from bad customers (False Positive Rate) and not losing good customer (True Positive Rate or Recall), ROC curve model is used to establish a balance between two criteria. However, in different case, different criteria can be used. As an example, for conservative policy, only decreasing FPR (false positive rate) would be a acceptable approach. In this case False Positive Rate is (FP/(FP+TN) = 241/ (1058+241) = 0.18) and True Positive Rate (Recall) is 0.8914.

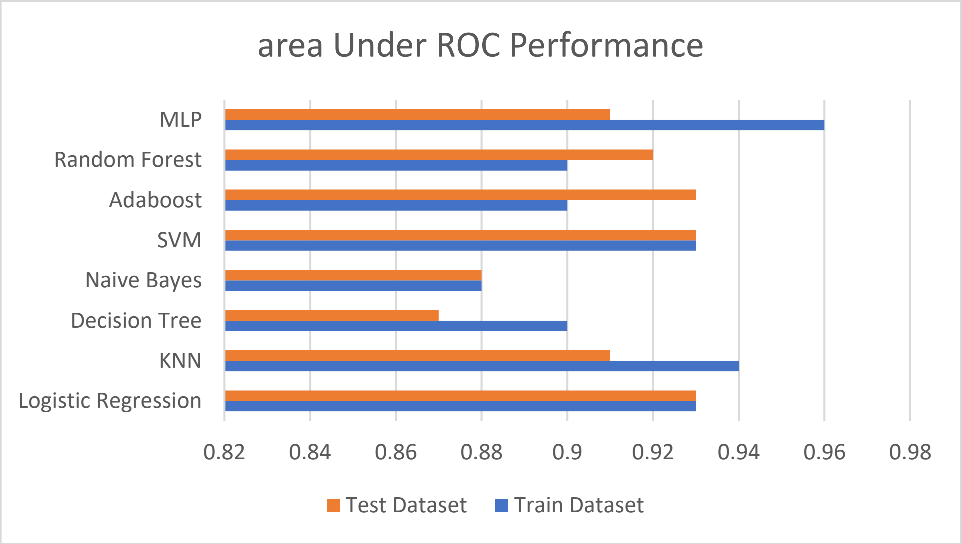
## 3.3. Other Models' Performances

As shown in table 3, SVM, Random Forest and MLP have overfitting behavior of about 4 percent that could be a very complicated structure of the model during training. Maximum overfitting is for MLP (Deep Learning) which is an issue. To be mentioned that training a custom Deep Learning model with custom activation function, layers, optimization, recall (to stop overfitting) would have the best performance. As can be seen in the table the maximum performance in training set is for MLP with 0.96, since I use MLP from the Sklearn it is hard to solve this issue. Using TensorFlow and adapting Custom model would probably be the best solution. It also allows us to capture nonlinearity behavior of the data using different activation function.

In addition, SVM like Logistic Regression has area under ROC curve of 0.93 for both test and training set, meaning that it is one the best model for binary classification (this task). An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. Since the feature mostly linearly dependent on target, this model could be very rigid

For such task models including Decision tree, SVM, KNN, Logistic Regression and ensemble methods like Adaboost and random forest could be useful.

*Table 3 – Other models' Performances*


area Under ROC Performance
MLP
Random Forest
Adaboost
SVM
Naive Bayes
Decision Tree
KNN
Logistic Regression
Test Dataset
Train Dataset