# Home Credit Default Risk

# 

# 

# Home Credit Default Risk – Group 3

# Charu Girdhar

# Harrisburg University

# 

# Author Note

We would like to thank Prof Roozbeh Sadeghian for his invaluable support and guidance

throughout the course ANLY530 Machine Learning - 1

# **Introduction**

# Globally more than 2 billion people fall into unbanked population category. These people either do not have bank accounts or are financially challenged to use it on a regular basis to maintain a decent credit score. This situation results in two-fold consequences. First, it prevents that unbanked population from fulfilling their dreams like having a home or good education due to lack of financial support (like loans). Second it prevents financial institutions from stepping into an untouched market and utilizing the potential of an important market of borrowers.

Organizations like Home Credit aim to address this situation worldwide and provide financial support in the form of loans and credits to unbanked population. It’s a global loan provider with presence in 10 countries in 3 continents. In an official statement in china in 2017, home credit stated that “70% of HCC’s customers are first-time borrowers. That means they have no record in the credit system”. The Intent of such financial organizations may be to capture and provide for underprivileged market, but these are For-Profit organizations and want to avoid and minimize risks and identify potential defaulters, which is a big challenge when most of your customers don't even have a bank account let alone credit history data. Home credit strives to broaden financial inclusion of such population by using alternative data, to predict their client’s repayment abilities.

The objective of this paper is to utilize this alternate data along with power of computing and application of complex machine learning algorithms to predict potential defaulters and risk.

# 

# 

# **Literature Review**

Financial risk assessment has taken a central focus in finance recently due to the volatile

financial market and increasing competition (Dima & Vasilache, 2016). Several methods have been proposed and practiced predicting the risk of credit over the years. However, with the recently gained computing power and accessibility to large data, application of machine learning and deep learning algorithms to predict default risk has become increasingly popular.

Dima and vasilache (2016) in their study to predict the default risk for an international bank in Romania, collected a sample that mimicked the structure of larger population of the companies in Romania and created seven classes from no risk to high risk based on their credit history and applied logit regression and artificial neural networks to predict the credit risk. The study found that neural nets classification outperforms the logistic regression.

Khemakhem and Boujelbene (2015) conducted a study to assist the bankers to predict the nonpayment risk the companies asking for a loan with the sample that consisted of 86 Tunisian companies over the period from 2005 to 2007. The study used discriminant analysis and neural networks techniques to predict the risk and neural nets (82.5%) produced more accurate results than discriminant analysis (74.4%).

In his study of comparing the nonlinear and non-parametric methods with traditional logit regression method for predicting home loan default, Bagherpour (nd) compiled a large dataset with over 20 million loan observations from Fannie Mae and Freddie Mac from 2001-2016 at the quarterly frequency and applied different machine learning algorithms (KNN, RF, SVM) to predict in sample and to forecast out-of-sample. The results suggested that the forecast performance of nonlinear and non-parametric algorithms (KNN = 0.90, RF = 0.88, SVM = 0.88) are substantially better than the traditional logit model (0.85).

Like Bagherpour (nd)’s approach, in the present study, we tried different machine learning algorithms to predict the home loan default. The algorithms includes decision trees, random forest, naive bayes, and extreme gradient boosting.

**Data**

The Data for this paper has been extracted from a competition running on kaggle.com.

<https://www.kaggle.com/c/home-credit-default-risk/data> . The relational dataset provided by home credit has 7 tables with loan application and previous credit data. Our aim is to utilize all the data tables for predicting target value.

1. Application\_train.csv: This is the primary table for home credit loan application and has a number of features (personal and profession) of the applicant. One row represents one loan in our data sample. It also has target values 0 and 1. 1 being client with payment difficulties and 0 for all other cases. 307511 obs. of 122 variables.
2. Bureau.csv : All client's previous credits provided by other financial institutions that were reported to Credit Bureau (for clients who have a loan in our sample).For every loan in our sample, there are as many rows as number of credits the client had in Credit Bureau before the application date.
3. Bureau\_balance.csv: Monthly balances of previous credits in Credit Bureau. This table has one row for each month of history of every previous credit reported to Credit Bureau.
4. POS\_CASH\_balance.csv: Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit. This table has one row for each month of history of every previous credit in Home Credit.
5. Credit\_card\_balance.csv: Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample
6. Previous\_application.csv: All previous applications for Home Credit loans of clients who have loans in our sample. There is one row for each previous application related to loans in our data sample.
7. Installments\_payments.csv: Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample. There is one row for every payment that was made plus one row each for missed payment.

The relational aspect of the dataset is depicted in Fig -1 (provided by the data source provider). List of columns and their description is available in appendix A table 2.

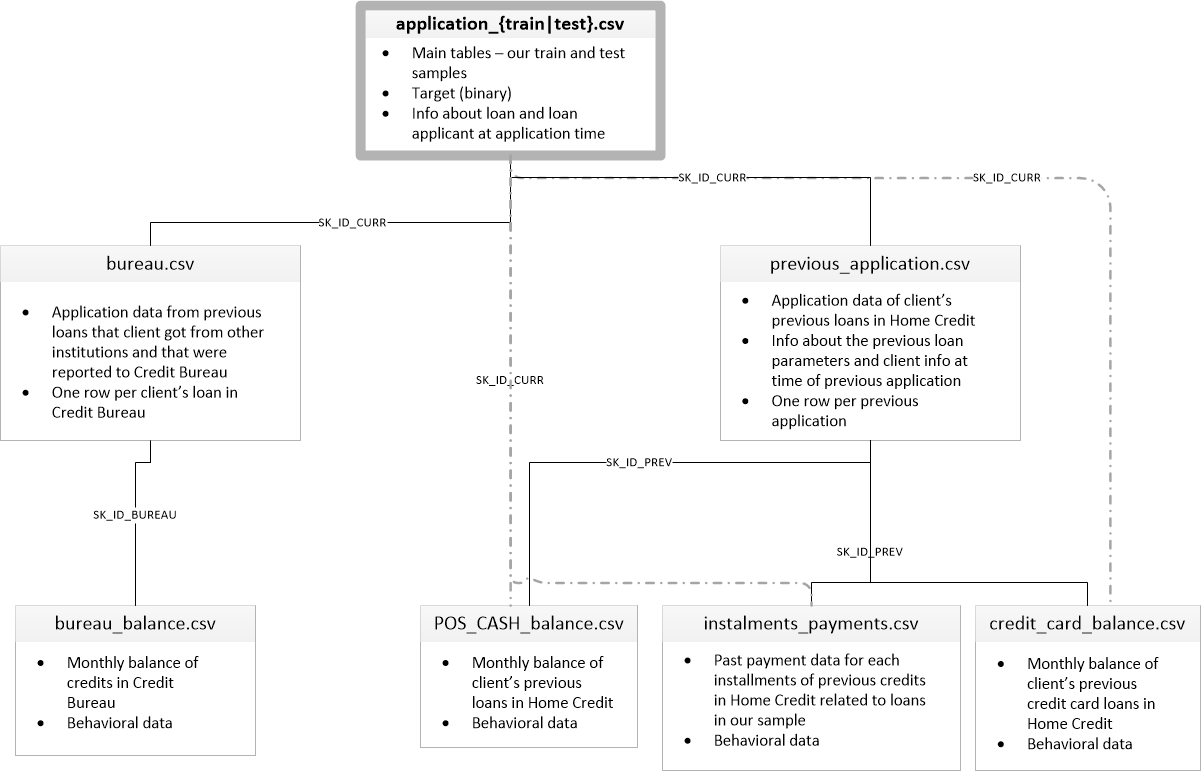


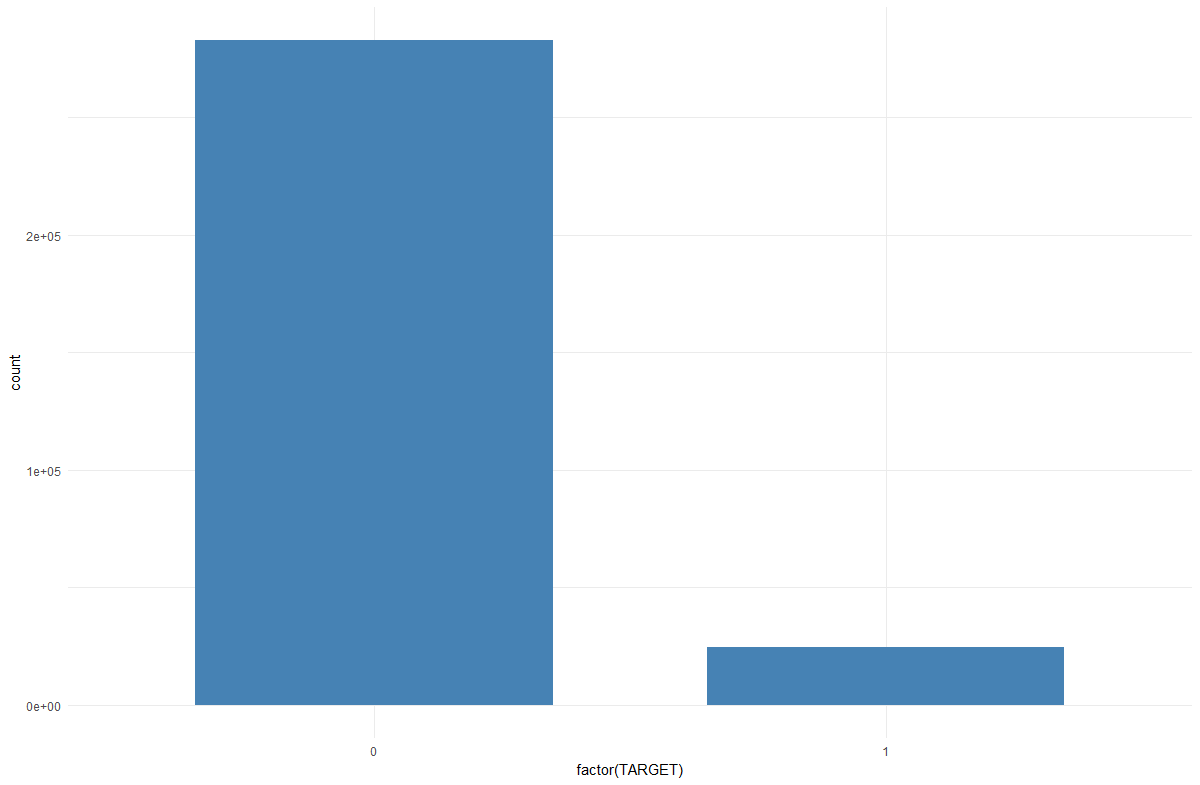
Fig -1

**Exploratory data analysis**

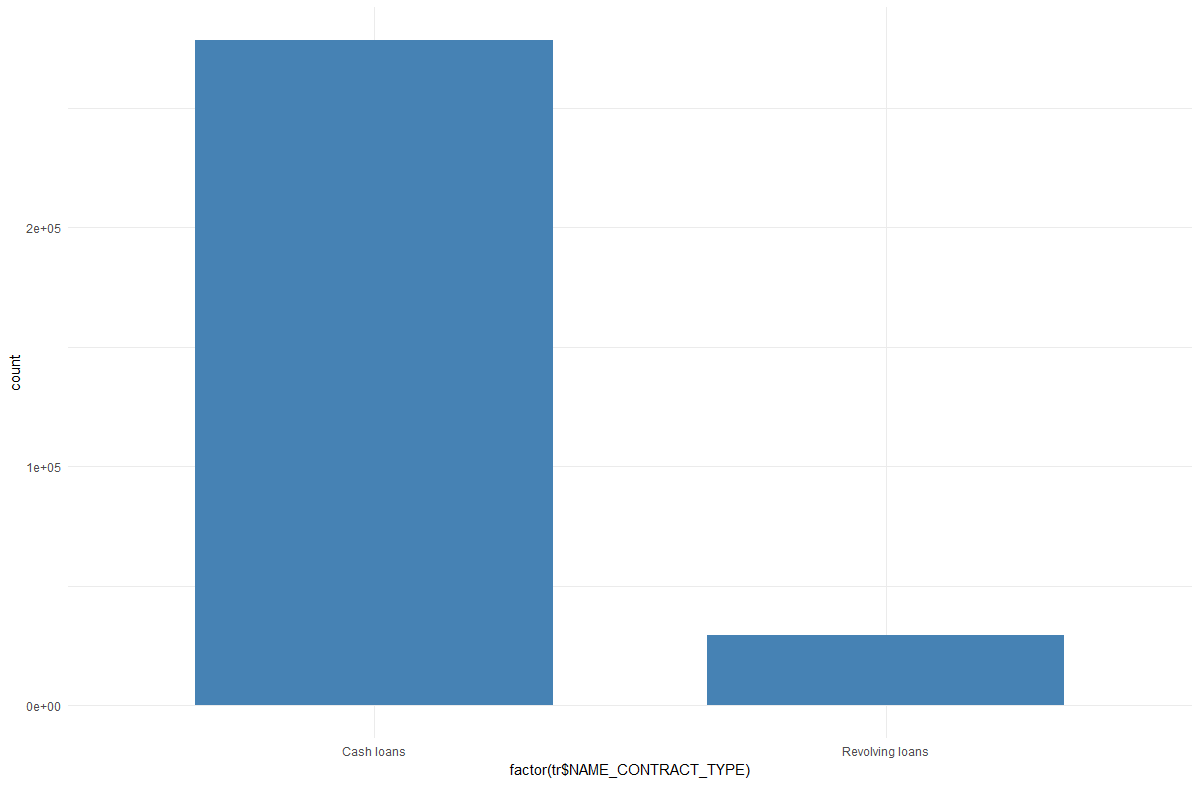
Below we have details of exploratory analysis

1. distribution in few categorical columns and target column.

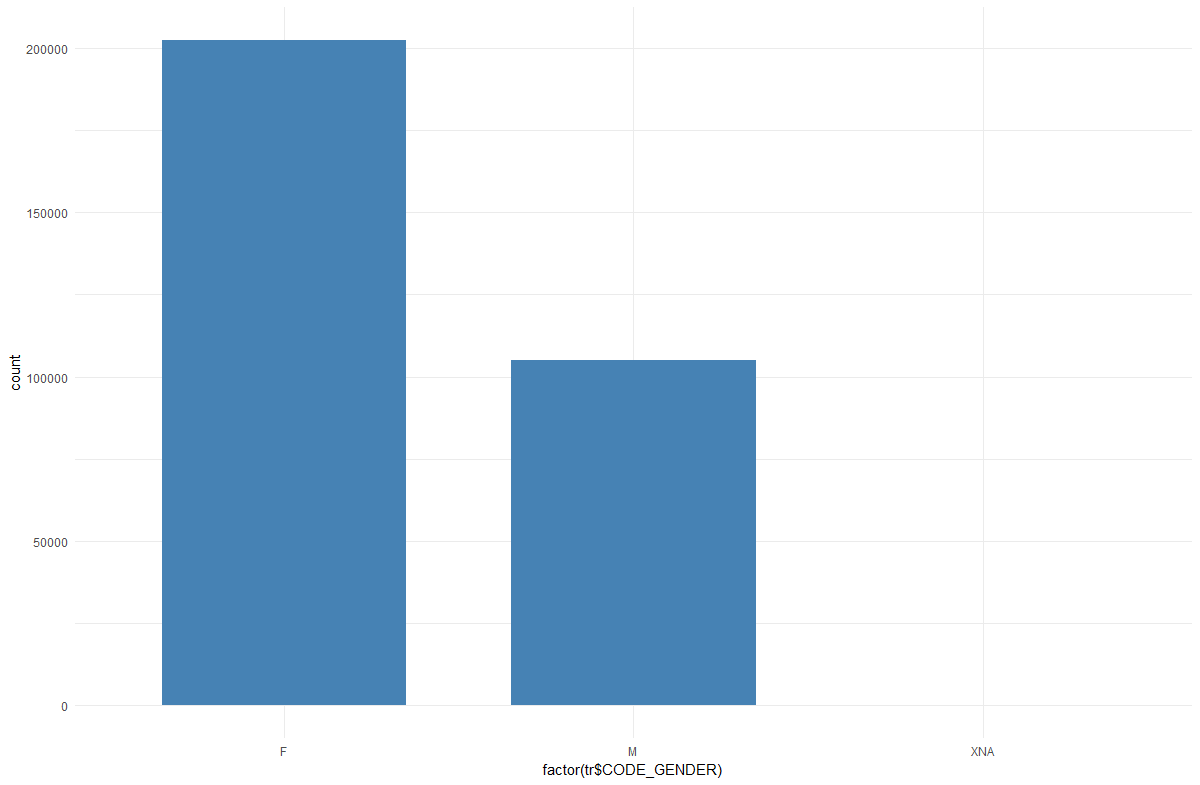
Target data: Highly imbalanced target data.



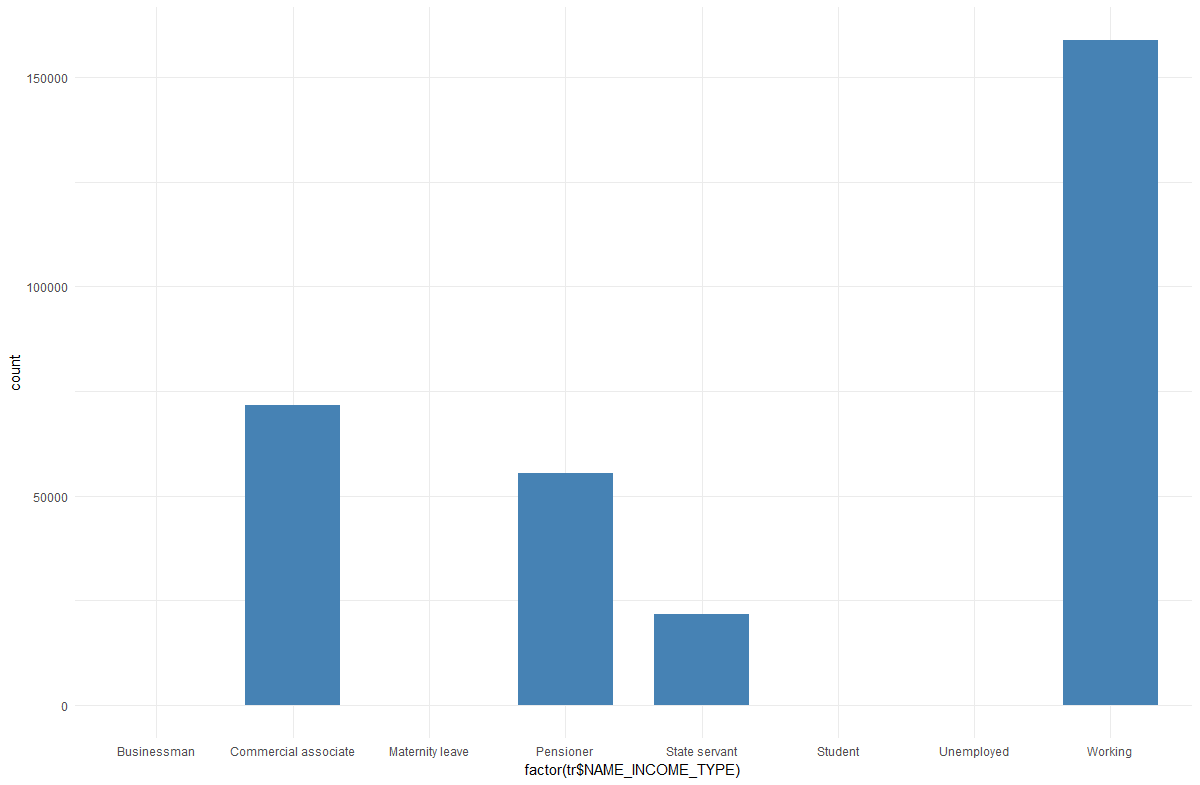
Type of loans



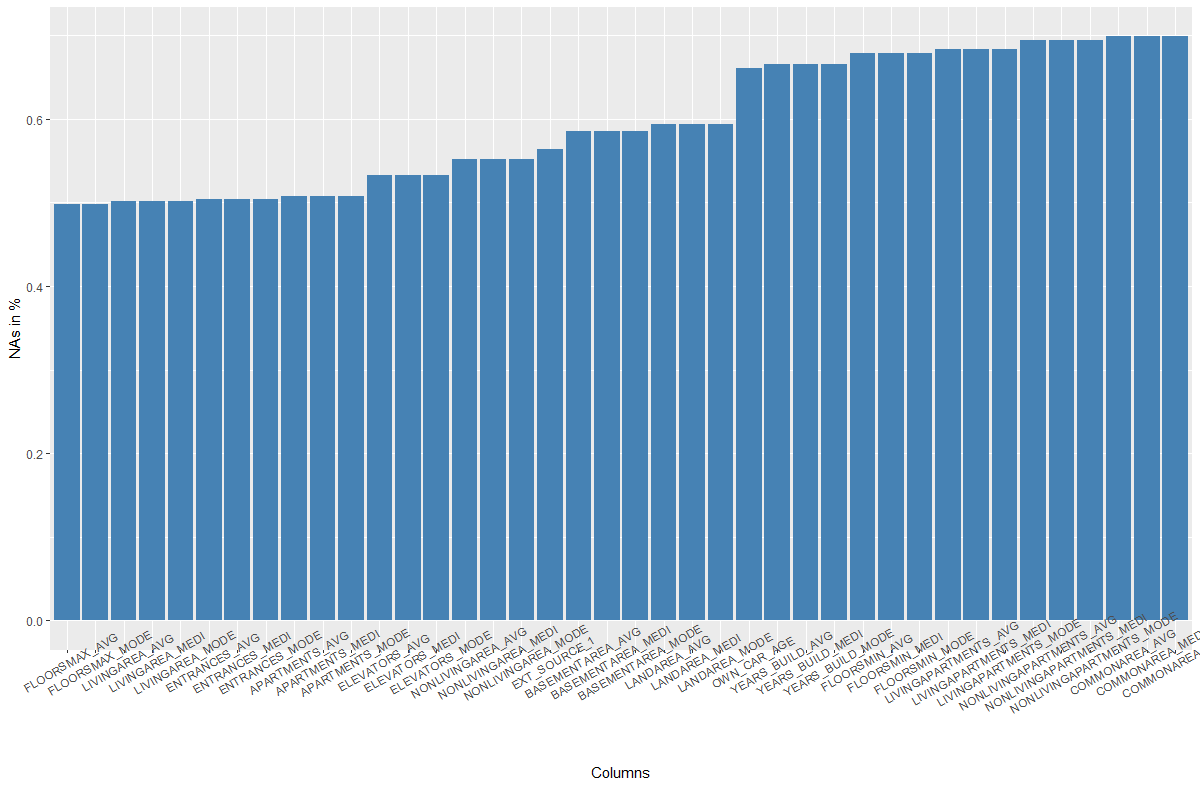
Gender disparity



Source of income



2) Number of NA’s in non categorical columns: Some columns have really high number of missing data



With a quick EDA, the data looks pretty unbalanced and incomplete. With high number of NA’s and a lot of disparity in data distribution for categorical data, we may have to perform a lot of preprocessing on this relational dataset.

**Data Preprocessing**

Since the dataset is huge with several relational tables and large number of columns, We used a number of different approaches to clean and merge data into single csv file. All approaches are mentioned below:

In one approach we concentrated on merging bureau.csv and previous\_application.csv into application\_train. Since for each row in application\_train.csv (unique sk\_id\_curr) we may have 2 or more rows in these 2 tables(multiple sk\_id\_curr), for merging we used one hot encoding categorical columns and summation of numeric values to create a single row for each sk\_id\_curr.

We used previous\_application.csv file to extract information related to previous application for credit. The variable we found could be useful was how many times the previous credit applications were approved or rejected. To calculate this variable, we used aggregation function on Name\_Contract\_Status column. In the final output file, we have two columns, one for approval% and another one for rejection% of previous applications.

In another approach, we took average of numerical columns based on same sk\_id\_curr (avoiding categorical columns) and created single summarised row for each sk\_id\_curr in bureau, pos\_cash\_balance, credit\_card\_balance and previous\_application. Then we left joined these tables to application train (and test as well).

We noticed a highly imbalanced class distribution in the response variable for our train and test. To avoid the risk of overfitting, we decided to use down sampling. In the process of down-sampling, we first filtered the positive class (1) and then randomly selected a near equal sample of negative class and prepared a new training set with near equal distribution of positive and negative classes.

**Technical Approach**

**Naive Bayes - All tables merged**

We started our analysis on a consolidated dataset, created by left joining application\_train file with bureau, pos\_cash\_balance, credit\_card\_balance and previous\_application tables (after taking mean of numerical values for each sk\_id\_curr). We created a train-validation split (75:25) from our newly formed data with all the column and all the observations and ran naive bayes classifier on this. Please note that this had no categorical but only numeric columns from merged tables. We then tested the naive bayes fit object on validation set and created a confusion matrix and other performance parameters listed in Table 1 (Appendix A).

**Rpart model**

In one of our approach, we used rpart model. First we divided the merged application\_train data into 80:20 train and validation data. For these two sub datasets, we find highly imbalanced class distribution as shown below

> prop.table(table(data\_train$TARGET))

0 1

0.91922214 0.08077786

To balance the class distribution in the train data, we took the down sampling approach as mentioned above in data preprocessing section. In the process of down-sampling, we first filtered the positive class (1) and then randomly selected a near equal sample of negative class and prepared a new training set with near equal distribution of positive and negative classes. After down-sampling, the class distribution of the train data looks as below

prop.table(table(data\_train\_new$TARGET))

0 1

0.4601976 0.5398024

We used variables same as under another model to develop our rpart model. We tested the model on our validation set and results are as shown in Table 1. We also did trial with adding more variables and adjusting the down sampling methodology but we got similar results and final test data accuracy of 64.3% result on Kaggle.

**Naive Bayes**

We selected features from a random forest important matrix from a kaggle study on the same data to avoid redundant tasks. We created a train-test split (75:25) from our newly formed data base on feature importance (26\* 307511). We then converted all integer features to numeric type and all categorical data into factor type.

We noticed a highly imbalanced class distribution in the response variable for our train and test. To avoid the risk of overfitting, we decided to use the technique called down sampling. In the process of down-sampling, we first filtered the positive class (1) and then randomly selected a near equal sample of negative class and prepared a new training set with near equal distribution of positive and negative classes. The final train set had 38455 observations and 23 variables.

|  |
| --- |
| EXT\_SOURCE\_2 |
| EXT\_SOURCE\_3 |
| EXT\_SOURCE\_1 |
| DAYS\_BIRTH |
| CODE\_GENDER |
| NAME\_EDUCATION\_TYPE |
| DAYS\_EMPLOYED |
| ORGANIZATION\_TYPE |
| AMT\_GOODS\_PRICE |
| FLAG\_DOCUMENT\_3 |
| FLAG\_EMP\_PHONE |
| DAYS\_LAST\_PHONE\_CHANGE |
| AMT\_CREDIT |
| DAYS\_ID\_PUBLISH |
| AMT\_ANNUITY |
| REGION\_RATING\_CLIENT\_W\_CITY |
| REG\_CITY\_NOT\_WORK\_CITY |
| OWN\_CAR\_AGE |
| FLAG\_OWN\_CAR |
| Approveddper |
| Refusedper |
| New |

We used naive bayes method available caret package along with 10-fold cross validation on the train set. We then tested the naive bayes fit object on test set and created a confusion matrix and other performance parameters listed in Table 1.

We performed the same data transformation and screening techniques for validation data set. Finally, we ran the fit object against the validation set (Table 1). R script and output are listed in Appendix B (Part A).

**Random Forest**

We repeated the same data transformations and screening techniques for random forest algorithm with the above listed feature. R script and output are listed in Appendix B (Part B).

We used random forest method available ranger and caret package along with 10-fold cross validation on the train set. We then tested the random forest fit object on test set and created a confusion matrix and other performance parameters listed in Table 1.

We performed the same data transformation and screening techniques for validation data set. Finally, we ran the fit object against the validation set (Table 1).

**XgBoost**

Lastly**,** we decided to run XgBoost. For XgBoost we selected features from important matrix from a kaggle study on the same data to avoid redundant tasks. We created a train-test split (75:25) from our newly formed data base on feature importance (26\* 307511).

|  |
| --- |
| EXT\_SOURCE\_3 |
| EXT\_SOURCE\_2 |
| EXT\_SOURCE\_1 |
| DAYS\_BIRTH |
| AMT\_CREDIT |
| AMT\_ANNUITY |
| DAYS\_EMPLOYED |
| AMT\_GOODS\_PRICE |
| DAYS\_ID\_PUBLISH |
| CODE\_GENDER |
| DAYS\_REGISTRATION |
| DAYS\_LAST\_PHONE\_CHANGE |
| NAME\_EDUCATION\_TYPE |
| OWN\_CAR\_AGE |
| OCCUPATION\_TYPE |
| AMT\_INCOME\_TOTAL |
| NAME\_CONTRACT\_TYPE |
| ION\_RATING\_CLIENT\_W\_CITY |
| NAME\_FAMILY\_STATUS |
| GION\_POPULATION\_RELATIVE |
| Approvedper |
| Refusedper |
| New |

We noticed a highly imbalanced class distribution in the response variable for our train and test. To avoid the risk of overfitting, we decided to use the technique called down sampling. In the process of down-sampling, we first filtered the positive class (1) and then randomly selected a near equal sample of negative class and prepared a new training set with near equal distribution of positive and negative classes. The final train set had 38455 observations and 23 variables.

As XgBoost can only take numeric matrix we converted the test and train dataset into sparse matrix. We then created XgBoost parameters and tuned the model to make it more conservative. We also removed the missing values at this step. We then generated an test/train error log graph to see the overfitting and produced a confusion matrix (Table 1, Appendix A). R script and output are listed in Appendix B (Part C).

**Test and Evaluation**

We decided to split our merged train\_application data into train and test before using the validation set. For Rpart, we created a split of 60:40 ratio whereas for rest of the models, we used of 75:25 split. For evaluating our results, we have focused on precision and recall more than accuracy. However, irrespective of low or high accuracy, we ran our predictions against validation set. We also calculated the F-score for easy comparison among the models and have also listed a kaggle score in Table 1 along with other performance measures.

Having said that, we noticed certain limitations that we would like to overcome in future. We noticed that imbalanced data was the main hurdle. We tried two methods (down sampling and K-fold cross validation) to deal with the issue of imbalanced data. However, we would like to try other methods like over sampling and improving on our performance matrix measures. We also look forward of using the hyper parameter tuning techniques like grid search for optimizing the model performance. We could not find to resolve the error at the validation run of xgBoost. We aim to fine tune random forest and resolve the errors for XgBoost to get higher score. Major

Table 1: Results and Evaluation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Precision | Recall | F-Score | Accuracy | Kaggle Score |
| Naive Bayes all tables merged | 0.32 | 0.84 | 0.46 | 0.36 | 0.565 |
| Naive Bayes | 0:0.72  1:0.59 | 0:0.95  1:0.16 | 0:0.82  1:0.25 | 0.72 | 0.655 |
| Random Forest | 0:0.74  1:0.58 | 0:0.95  1:0.16 | 0:0.83  1:0.26 | 0.73 | 0.664 |
| XgBoost | 0:1  1:1 | 0:1  1:1 | 0:1  1:1 | 100 |  |
| Rpart | 0.12 | 0.71 | 0.21 | 0.58 | 0.643 |

Kaggle Team- **“MachLearnGroup530Team3”**.

**Reference**

Dima, A. M., Vasilache, S. (2016). Credit Risk Modeling for Companies Default Prediction Using Neural Networks. *Romanian Journal of Economic Forecasting*, 3, 127-143.

Khemakhem, S., Boujelbene, Y. (2015). Credit Risk Prediction: A comparative study between discriminant analysis and the neural network approach. *Accounting and Management Information System*, 14, 60-78.

Bagherpour, A. (nd). Predicting Mortgage Loan Default with Machine Learning Methods. *University of California, Riverside*.

**Appendix A**

Table : Columns and description:

|  |  |  |  |
| --- | --- | --- | --- |
| no | Table | Row | Description |
| 1 | application\_train.csv | SK\_ID\_CURR | ID of loan in our sample |
| 2 | application\_train.csv | TARGET | Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases) |
| 5 | application\_train.csv | NAME\_CONTRACT\_TYPE | Identification if loan is cash or revolving |
| 6 | application\_train.csv | CODE\_GENDER | Gender of the client |
| 7 | application\_train.csv | FLAG\_OWN\_CAR | Flag if the client owns a car |
| 8 | application\_train.csv | FLAG\_OWN\_REALTY | Flag if client owns a house or flat |
| 9 | application\_train.csv | CNT\_CHILDREN | Number of children the client has |
| 10 | application\_train.csv | AMT\_INCOME\_TOTAL | Income of the client |
| 11 | application\_train.csv | AMT\_CREDIT | Credit amount of the loan |
| 12 | application\_train.csv | AMT\_ANNUITY | Loan annuity |
| 13 | application\_train.csv | AMT\_GOODS\_PRICE | For consumer loans it is the price of the goods for which the loan is given |
| 14 | application\_train.csv | NAME\_TYPE\_SUITE | Who was accompanying client when he was applying for the loan |
| 15 | application\_train.csv | NAME\_INCOME\_TYPE | Clients income type (businessman, working, maternity leave,…) |
| 16 | application\_train.csv | NAME\_EDUCATION\_TYPE | Level of highest education the client achieved |
| 17 | application\_train.csv | NAME\_FAMILY\_STATUS | Family status of the client |
| 18 | application\_train.csv | NAME\_HOUSING\_TYPE | What is the housing situation of the client (renting, living with parents, ...) |
| 19 | application\_train.csv | REGION\_POPULATION\_RELATIVE | Normalized population of region where client lives (higher number means the client lives in more populated region) |
| 20 | application\_train.csv | DAYS\_BIRTH | Client's age in days at the time of application |
| 21 | application\_train.csv | DAYS\_EMPLOYED | How many days before the application the person started current employment |
| 22 | application\_train.csv | DAYS\_REGISTRATION | How many days before the application did client change his registration |
| 23 | application\_train.csv | DAYS\_ID\_PUBLISH | How many days before the application did client change the identity document with which he applied for the loan |
| 24 | application\_train.csv | OWN\_CAR\_AGE | Age of client's car |
| 25 | application\_train.csv | FLAG\_MOBIL | Did client provide mobile phone (1=YES, 0=NO) |
| 26 | application\_train.csv | FLAG\_EMP\_PHONE | Did client provide work phone (1=YES, 0=NO) |
| 27 | application\_train.csv | FLAG\_WORK\_PHONE | Did client provide home phone (1=YES, 0=NO) |
| 28 | application\_train.csv | FLAG\_CONT\_MOBILE | Was mobile phone reachable (1=YES, 0=NO) |
| 29 | application\_train.csv | FLAG\_PHONE | Did client provide home phone (1=YES, 0=NO) |
| 30 | application\_train.csv | FLAG\_EMAIL | Did client provide email (1=YES, 0=NO) |
| 31 | application\_train.csv | OCCUPATION\_TYPE | What kind of occupation does the client have |
| 32 | application\_train.csv | CNT\_FAM\_MEMBERS | How many family members does client have |
| 33 | application\_train.csv | REGION\_RATING\_CLIENT | Our rating of the region where client lives (1,2,3) |
| 34 | application\_train.csv | REGION\_RATING\_CLIENT\_W\_CITY | Our rating of the region where client lives with taking city into account (1,2,3) |
| 35 | application\_train.csv | WEEKDAY\_APPR\_PROCESS\_START | On which day of the week did the client apply for the loan |
| 36 | application\_train.csv | HOUR\_APPR\_PROCESS\_START | Approximately at what hour did the client apply for the loan |
| 37 | application\_train.csv | REG\_REGION\_NOT\_LIVE\_REGION | Flag if client's permanent address does not match contact address (1=different, 0=same, at region level) |
| 38 | application\_train.csv | REG\_REGION\_NOT\_WORK\_REGION | Flag if client's permanent address does not match work address (1=different, 0=same, at region level) |
| 39 | application\_train.csv | LIVE\_REGION\_NOT\_WORK\_REGION | Flag if client's contact address does not match work address (1=different, 0=same, at region level) |
| 40 | application\_train.csv | REG\_CITY\_NOT\_LIVE\_CITY | Flag if client's permanent address does not match contact address (1=different, 0=same, at city level) |
| 41 | application\_train.csv | REG\_CITY\_NOT\_WORK\_CITY | Flag if client's permanent address does not match work address (1=different, 0=same, at city level) |
| 42 | application\_train.csv | LIVE\_CITY\_NOT\_WORK\_CITY | Flag if client's contact address does not match work address (1=different, 0=same, at city level) |
| 43 | application\_train.csv | ORGANIZATION\_TYPE | Type of organization where client works |
| 44 | application\_train.csv | EXT\_SOURCE\_1 | Normalized score from external data source |
| 45 | application\_train.csv | EXT\_SOURCE\_2 | Normalized score from external data source |
| 46 | application\_train.csv | EXT\_SOURCE\_3 | Normalized score from external data source |
| 47 | application\_train.csv | APARTMENTS\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 48 | application\_train.csv | BASEMENTAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 49 | application\_train.csv | YEARS\_BEGINEXPLUATATION\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 50 | application\_train.csv | YEARS\_BUILD\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 51 | application\_train.csv | COMMONAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 52 | application\_train.csv | ELEVATORS\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 53 | application\_train.csv | ENTRANCES\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 54 | application\_train.csv | FLOORSMAX\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 55 | application\_train.csv | FLOORSMIN\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 56 | application\_train.csv | LANDAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 57 | application\_train.csv | LIVINGAPARTMENTS\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 58 | application\_train.csv | LIVINGAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 59 | application\_train.csv | NONLIVINGAPARTMENTS\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 60 | application\_train.csv | NONLIVINGAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 61 | application\_train.csv | APARTMENTS\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 62 | application\_train.csv | BASEMENTAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 63 | application\_train.csv | YEARS\_BEGINEXPLUATATION\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 64 | application\_train.csv | YEARS\_BUILD\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 65 | application\_train.csv | COMMONAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 66 | application\_train.csv | ELEVATORS\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 67 | application\_train.csv | ENTRANCES\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 68 | application\_train.csv | FLOORSMAX\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 69 | application\_train.csv | FLOORSMIN\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 70 | application\_train.csv | LANDAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 71 | application\_train.csv | LIVINGAPARTMENTS\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 72 | application\_train.csv | LIVINGAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 73 | application\_train.csv | NONLIVINGAPARTMENTS\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 74 | application\_train.csv | NONLIVINGAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 75 | application\_train.csv | APARTMENTS\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 76 | application\_train.csv | BASEMENTAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 77 | application\_train.csv | YEARS\_BEGINEXPLUATATION\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 78 | application\_train.csv | YEARS\_BUILD\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 79 | application\_train.csv | COMMONAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 80 | application\_train.csv | ELEVATORS\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 81 | application\_train.csv | ENTRANCES\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 82 | application\_train.csv | FLOORSMAX\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 83 | application\_train.csv | FLOORSMIN\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 84 | application\_train.csv | LANDAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 85 | application\_train.csv | LIVINGAPARTMENTS\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 86 | application\_train.csv | LIVINGAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 87 | application\_train.csv | NONLIVINGAPARTMENTS\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 88 | application\_train.csv | NONLIVINGAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 89 | application\_train.csv | FONDKAPREMONT\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 90 | application\_train.csv | HOUSETYPE\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 91 | application\_train.csv | TOTALAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 92 | application\_train.csv | WALLSMATERIAL\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 93 | application\_train.csv | EMERGENCYSTATE\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 94 | application\_train.csv | OBS\_30\_CNT\_SOCIAL\_CIRCLE | How many observation of client's social surroundings with observable 30 DPD (days past due) default |
| 95 | application\_train.csv | DEF\_30\_CNT\_SOCIAL\_CIRCLE | How many observation of client's social surroundings defaulted on 30 DPD (days past due) |
| 96 | application\_train.csv | OBS\_60\_CNT\_SOCIAL\_CIRCLE | How many observation of client's social surroundings with observable 60 DPD (days past due) default |
| 97 | application\_train.csv | DEF\_60\_CNT\_SOCIAL\_CIRCLE | How many observation of client's social surroundings defaulted on 60 (days past due) DPD |
| 98 | application\_train.csv | DAYS\_LAST\_PHONE\_CHANGE | How many days before application did client change phone |
| 99 | application\_train.csv | FLAG\_DOCUMENT\_2 | Did client provide document 2 |
| 100 | application\_train.csv | FLAG\_DOCUMENT\_3 | Did client provide document 3 |
| 101 | application\_train.csv | FLAG\_DOCUMENT\_4 | Did client provide document 4 |
| 102 | application\_train.csv | FLAG\_DOCUMENT\_5 | Did client provide document 5 |
| 103 | application\_train.csv | FLAG\_DOCUMENT\_6 | Did client provide document 6 |
| 104 | application\_train.csv | FLAG\_DOCUMENT\_7 | Did client provide document 7 |
| 105 | application\_train.csv | FLAG\_DOCUMENT\_8 | Did client provide document 8 |
| 106 | application\_train.csv | FLAG\_DOCUMENT\_9 | Did client provide document 9 |
| 107 | application\_train.csv | FLAG\_DOCUMENT\_10 | Did client provide document 10 |
| 108 | application\_train.csv | FLAG\_DOCUMENT\_11 | Did client provide document 11 |
| 109 | application\_train.csv | FLAG\_DOCUMENT\_12 | Did client provide document 12 |
| 110 | application\_train.csv | FLAG\_DOCUMENT\_13 | Did client provide document 13 |
| 111 | application\_train.csv | FLAG\_DOCUMENT\_14 | Did client provide document 14 |
| 112 | application\_train.csv | FLAG\_DOCUMENT\_15 | Did client provide document 15 |
| 113 | application\_train.csv | FLAG\_DOCUMENT\_16 | Did client provide document 16 |
| 114 | application\_train.csv | FLAG\_DOCUMENT\_17 | Did client provide document 17 |
| 115 | application\_train.csv | FLAG\_DOCUMENT\_18 | Did client provide document 18 |
| 116 | application\_train.csv | FLAG\_DOCUMENT\_19 | Did client provide document 19 |
| 117 | application\_train.csv | FLAG\_DOCUMENT\_20 | Did client provide document 20 |
| 118 | application\_train.csv | FLAG\_DOCUMENT\_21 | Did client provide document 21 |
| 119 | application\_train.csv | AMT\_REQ\_CREDIT\_BUREAU\_HOUR | Number of enquiries to Credit Bureau about the client one hour before application |
| 120 | application\_train.csv | AMT\_REQ\_CREDIT\_BUREAU\_DAY | Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application) |
| 121 | application\_train.csv | AMT\_REQ\_CREDIT\_BUREAU\_WEEK | Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application) |
| 122 | application\_train.csv | AMT\_REQ\_CREDIT\_BUREAU\_MON | Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application) |
| 123 | application\_train.csv | AMT\_REQ\_CREDIT\_BUREAU\_QRT | Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application) |
| 124 | application\_train.csv | AMT\_REQ\_CREDIT\_BUREAU\_YEAR | Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application) |
| 125 | bureau.csv | SK\_ID\_CURR | ID of loan in our sample - one loan in our sample can have 0,1,2 or more related previous credits in credit bureau |
| 126 | bureau.csv | SK\_BUREAU\_ID | Recoded ID of previous Credit Bureau credit related to our loan (unique coding for each loan application) |
| 127 | bureau.csv | CREDIT\_ACTIVE | Status of the Credit Bureau (CB) reported credits |
| 128 | bureau.csv | CREDIT\_CURRENCY | Recoded currency of the Credit Bureau credit |
| 129 | bureau.csv | DAYS\_CREDIT | How many days before current application did client apply for Credit Bureau credit |
| 130 | bureau.csv | CREDIT\_DAY\_OVERDUE | Number of days past due on CB credit at the time of application for related loan in our sample |
| 131 | bureau.csv | DAYS\_CREDIT\_ENDDATE | Remaining duration of CB credit (in days) at the time of application in Home Credit |
| 132 | bureau.csv | DAYS\_ENDDATE\_FACT | Days since CB credit ended at the time of application in Home Credit (only for closed credit) |
| 133 | bureau.csv | AMT\_CREDIT\_MAX\_OVERDUE | Maximal amount overdue on the Credit Bureau credit so far (at application date of loan in our sample) |
| 134 | bureau.csv | CNT\_CREDIT\_PROLONG | How many times was the Credit Bureau credit prolonged |
| 135 | bureau.csv | AMT\_CREDIT\_SUM | Current credit amount for the Credit Bureau credit |
| 136 | bureau.csv | AMT\_CREDIT\_SUM\_DEBT | Current debt on Credit Bureau credit |
| 137 | bureau.csv | AMT\_CREDIT\_SUM\_LIMIT | Current credit limit of credit card reported in Credit Bureau |
| 138 | bureau.csv | AMT\_CREDIT\_SUM\_OVERDUE | Current amount overdue on Credit Bureau credit |
| 139 | bureau.csv | CREDIT\_TYPE | Type of Credit Bureau credit (Car, cash,...) |
| 140 | bureau.csv | DAYS\_CREDIT\_UPDATE | How many days before loan application did last information about the Credit Bureau credit come |
| 141 | bureau.csv | AMT\_ANNUITY | Annuity of the Credit Bureau credit |
| 142 | bureau\_balance.csv | SK\_BUREAU\_ID | Recoded ID of Credit Bureau credit (unique coding for each application) - use this to join to CREDIT\_BUREAU table |
| 143 | bureau\_balance.csv | MONTHS\_BALANCE | Month of balance relative to application date (-1 means the freshest balance date) |
| 144 | bureau\_balance.csv | STATUS | Status of Credit Bureau loan during the month (active, closed, DPD0-30,… [C means closed, X means status unknown, 0 means no DPD, 1 means maximal did during month between 1-30, 2 means DPD 31-60,… 5 means DPD 120+ or sold or written off ] ) |
| 145 | POS\_CASH\_balance.csv | SK\_ID\_PREV | ID of previous credit in Home Credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit) |
| 146 | POS\_CASH\_balance.csv | SK\_ID\_CURR | ID of loan in our sample |
| 147 | POS\_CASH\_balance.csv | MONTHS\_BALANCE | Month of balance relative to application date (-1 means the information to the freshest monthly snapshot, 0 means the information at application - often it will be the same as -1 as many banks are not updating the information to Credit Bureau regularly ) |
| 148 | POS\_CASH\_balance.csv | CNT\_INSTALMENT | Term of previous credit (can change over time) |
| 149 | POS\_CASH\_balance.csv | CNT\_INSTALMENT\_FUTURE | Installments left to pay on the previous credit |
| 150 | POS\_CASH\_balance.csv | NAME\_CONTRACT\_STATUS | Contract status during the month |
| 151 | POS\_CASH\_balance.csv | SK\_DPD | DPD (days past due) during the month of previous credit |
| 152 | POS\_CASH\_balance.csv | SK\_DPD\_DEF | DPD during the month with tolerance (debts with low loan amounts are ignored) of the previous credit |
| 153 | credit\_card\_balance.csv | SK\_ID\_PREV | ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit) |
| 154 | credit\_card\_balance.csv | SK\_ID\_CURR | ID of loan in our sample |
| 155 | credit\_card\_balance.csv | MONTHS\_BALANCE | Month of balance relative to application date (-1 means the freshest balance date) |
| 156 | credit\_card\_balance.csv | AMT\_BALANCE | Balance during the month of previous credit |
| 157 | credit\_card\_balance.csv | AMT\_CREDIT\_LIMIT\_ACTUAL | Credit card limit during the month of the previous credit |
| 158 | credit\_card\_balance.csv | AMT\_DRAWINGS\_ATM\_CURRENT | Amount drawing at ATM during the month of the previous credit |
| 159 | credit\_card\_balance.csv | AMT\_DRAWINGS\_CURRENT | Amount drawing during the month of the previous credit |
| 160 | credit\_card\_balance.csv | AMT\_DRAWINGS\_OTHER\_CURRENT | Amount of other drawings during the month of the previous credit |
| 161 | credit\_card\_balance.csv | AMT\_DRAWINGS\_POS\_CURRENT | Amount drawing or buying goods during the month of the previous credit |
| 162 | credit\_card\_balance.csv | AMT\_INST\_MIN\_REGULARITY | Minimal installment for this month of the previous credit |
| 163 | credit\_card\_balance.csv | AMT\_PAYMENT\_CURRENT | How much did the client pay during the month on the previous credit |
| 164 | credit\_card\_balance.csv | AMT\_PAYMENT\_TOTAL\_CURRENT | How much did the client pay during the month in total on the previous credit |
| 165 | credit\_card\_balance.csv | AMT\_RECEIVABLE\_PRINCIPAL | Amount receivable for principal on the previous credit |
| 166 | credit\_card\_balance.csv | AMT\_RECIVABLE | Amount receivable on the previous credit |
| 167 | credit\_card\_balance.csv | AMT\_TOTAL\_RECEIVABLE | Total amount receivable on the previous credit |
| 168 | credit\_card\_balance.csv | CNT\_DRAWINGS\_ATM\_CURRENT | Number of drawings at ATM during this month on the previous credit |
| 169 | credit\_card\_balance.csv | CNT\_DRAWINGS\_CURRENT | Number of drawings during this month on the previous credit |
| 170 | credit\_card\_balance.csv | CNT\_DRAWINGS\_OTHER\_CURRENT | Number of other drawings during this month on the previous credit |
| 171 | credit\_card\_balance.csv | CNT\_DRAWINGS\_POS\_CURRENT | Number of drawings for goods during this month on the previous credit |
| 172 | credit\_card\_balance.csv | CNT\_INSTALMENT\_MATURE\_CUM | Number of paid installments on the previous credit |
| 173 | credit\_card\_balance.csv | NAME\_CONTRACT\_STATUS | Contract status (active signed,...) on the previous credit |
| 174 | credit\_card\_balance.csv | SK\_DPD | DPD (Days past due) during the month on the previous credit |
| 175 | credit\_card\_balance.csv | SK\_DPD\_DEF | DPD (Days past due) during the month with tolerance (debts with low loan amounts are ignored) of the previous credit |
| 176 | previous\_application.csv | SK\_ID\_PREV | ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loan applications in Home Credit, previous application could, but not necessarily have to lead to credit) |
| 177 | previous\_application.csv | SK\_ID\_CURR | ID of loan in our sample |
| 178 | previous\_application.csv | NAME\_CONTRACT\_TYPE | Contract product type (Cash loan, consumer loan [POS] ,...) of the previous application |
| 179 | previous\_application.csv | AMT\_ANNUITY | Annuity of previous application |
| 180 | previous\_application.csv | AMT\_APPLICATION | For how much credit did client ask on the previous application |
| 181 | previous\_application.csv | AMT\_CREDIT | Final credit amount on the previous application. This differs from AMT\_APPLICATION in a way that the AMT\_APPLICATION is the amount for which the client initially applied for, but during our approval process he could have received different amount - AMT\_CREDIT |
| 182 | previous\_application.csv | AMT\_DOWN\_PAYMENT | Down payment on the previous application |
| 183 | previous\_application.csv | AMT\_GOODS\_PRICE | Goods price of good that client asked for (if applicable) on the previous application |
| 184 | previous\_application.csv | WEEKDAY\_APPR\_PROCESS\_START | On which day of the week did the client apply for previous application |
| 185 | previous\_application.csv | HOUR\_APPR\_PROCESS\_START | Approximately at what day hour did the client apply for the previous application |
| 186 | previous\_application.csv | FLAG\_LAST\_APPL\_PER\_CONTRACT | Flag if it was last application for the previous contract. Sometimes by mistake of client or our clerk there could be more applications for one single contract |
| 187 | previous\_application.csv | NFLAG\_LAST\_APPL\_IN\_DAY | Flag if the application was the last application per day of the client. Sometimes clients apply for more applications a day. Rarely it could also be error in our system that one application is in the database twice |
| 188 | previous\_application.csv | NFLAG\_MICRO\_CASH | Flag Micro finance loan |
| 189 | previous\_application.csv | RATE\_DOWN\_PAYMENT | Down payment rate normalized on previous credit |
| 190 | previous\_application.csv | RATE\_INTEREST\_PRIMARY | Interest rate normalized on previous credit |
| 191 | previous\_application.csv | RATE\_INTEREST\_PRIVILEGED | Interest rate normalized on previous credit |
| 192 | previous\_application.csv | NAME\_CASH\_LOAN\_PURPOSE | Purpose of the cash loan |
| 193 | previous\_application.csv | NAME\_CONTRACT\_STATUS | Contract status (approved, cancelled, ...) of previous application |
| 194 | previous\_application.csv | DAYS\_DECISION | Relative to current application when was the decision about previous application made |
| 195 | previous\_application.csv | NAME\_PAYMENT\_TYPE | Payment method that client chose to pay for the previous application |
| 196 | previous\_application.csv | CODE\_REJECT\_REASON | Why was the previous application rejected |
| 197 | previous\_application.csv | NAME\_TYPE\_SUITE | Who accompanied client when applying for the previous application |
| 198 | previous\_application.csv | NAME\_CLIENT\_TYPE | Was the client old or new client when applying for the previous application |
| 199 | previous\_application.csv | NAME\_GOODS\_CATEGORY | What kind of goods did the client apply for in the previous application |
| 200 | previous\_application.csv | NAME\_PORTFOLIO | Was the previous application for CASH, POS, CAR, … |
| 201 | previous\_application.csv | NAME\_PRODUCT\_TYPE | Was the previous application x-sell o walk-in |
| 202 | previous\_application.csv | CHANNEL\_TYPE | Through which channel we acquired the client on the previous application |
| 203 | previous\_application.csv | SELLERPLACE\_AREA | Selling area of seller place of the previous application |
| 204 | previous\_application.csv | NAME\_SELLER\_INDUSTRY | The industry of the seller |
| 205 | previous\_application.csv | CNT\_PAYMENT | Term of previous credit at application of the previous application |
| 206 | previous\_application.csv | NAME\_YIELD\_GROUP | Grouped interest rate into small medium and high of the previous application |
| 207 | previous\_application.csv | PRODUCT\_COMBINATION | Detailed product combination of the previous application |
| 208 | previous\_application.csv | DAYS\_FIRST\_DRAWING | Relative to application date of current application when was the first disbursement of the previous application |
| 209 | previous\_application.csv | DAYS\_FIRST\_DUE | Relative to application date of current application when was the first due supposed to be of the previous application |
| 210 | previous\_application.csv | DAYS\_LAST\_DUE\_1ST\_VERSION | Relative to application date of current application when was the first due of the previous application |
| 211 | previous\_application.csv | DAYS\_LAST\_DUE | Relative to application date of current application when was the last due date of the previous application |
| 212 | previous\_application.csv | DAYS\_TERMINATION | Relative to application date of current application when was the expected termination of the previous application |
| 213 | previous\_application.csv | NFLAG\_INSURED\_ON\_APPROVAL | Did the client requested insurance during the previous application |
| 214 | installments\_payments.csv | SK\_ID\_PREV | ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit) |
| 215 | installments\_payments.csv | SK\_ID\_CURR | ID of loan in our sample |
| 216 | installments\_payments.csv | NUM\_INSTALMENT\_VERSION | Version of installment calendar (0 is for credit card) of previous credit. Change of installment version from month to month signifies that some parameter of payment calendar has changed |
| 217 | installments\_payments.csv | NUM\_INSTALMENT\_NUMBER | On which installment we observe payment |
| 218 | installments\_payments.csv | DAYS\_INSTALMENT | When the installment of previous credit was supposed to be paid (relative to application date of current loan) |
| 219 | installments\_payments.csv | DAYS\_ENTRY\_PAYMENT | When was the installments of previous credit paid actually (relative to application date of current loan) |
| 220 | installments\_payments.csv | AMT\_INSTALMENT | What was the prescribed installment amount of previous credit on this installment |
| 221 | installments\_payments.csv | AMT\_PAYMENT | What the client actually paid on previous credit on this installment |

**Appendix B**

**R-Scripts**

**Data Preprocessing: Merging application\_train and Previous\_application output**

previousapp <- read.csv("C:/Users/dnsingh/Downloads/Harrisburg/06ANYL530/Project/all/previous\_application.csv", stringsAsFactors = FALSE)

str(previousapp)

previousapp\_v2 <- previousapp

previousapp\_v2$SK\_ID\_PREV <- factor(previousapp\_v2$SK\_ID\_PREV)

previousapp\_v2$SK\_ID\_CURR <- factor(previousapp\_v2$SK\_ID\_CURR)

previousapp\_v2$NAME\_CONTRACT\_STATUS <- factor(previousapp\_v2$NAME\_CONTRACT\_STATUS)

previousapp\_v3 <- dplyr::select(previousapp\_v2,SK\_ID\_CURR,NAME\_CONTRACT\_STATUS)

shortpreviousapp\_v3 <- previousapp\_v3

shortpreviousapp\_v3$value <- 1

shortpreviousapp\_v4 <- shortpreviousapp\_v3 %>%

dplyr::group\_by(SK\_ID\_CURR , NAME\_CONTRACT\_STATUS) %>%

dplyr::summarise(total=sum(value))

shortpreviousapp\_v5 <- spread(shortpreviousapp\_v4, NAME\_CONTRACT\_STATUS, total, fill = 0)

shortpreviousapp\_v5$Approved <- as.numeric(shortpreviousapp\_v5$Approved)

shortpreviousapp\_v5$Canceled <- as.numeric(shortpreviousapp\_v5$Canceled)

shortpreviousapp\_v5$Refused <- as.numeric(shortpreviousapp\_v5$Refused)

shortpreviousapp\_v5$`Unused offer` <- as.numeric(shortpreviousapp\_v5$`Unused offer`)

shortpreviousapp\_v6 <- shortpreviousapp\_v5 %>%

mutate(approvedper = Approved/(Approved+Canceled+Refused+`Unused offer`)) %>%

mutate(refusedper = Refused/(Approved+Canceled+Refused+`Unused offer`))

shortpreviousapp\_v7 <- shortpreviousapp\_v6 %>%

dplyr::select(SK\_ID\_CURR,approvedper,refusedper)

shortpreviousapp\_v7$new <- "No"

write.csv(shortpreviousapp\_v7, file = "C:/Users/dnsingh/Downloads/Harrisburg/06ANYL530/Project/all/previous\_application\_output.csv")

shortpreviousapp\_v7$new <- "No"

str(shortpreviousapp\_v7)

currentapp <- read.csv("C:/Users/dnsingh/Downloads/Harrisburg/06ANYL530/Project/all/application\_train.csv", stringsAsFactors = FALSE)

str(currentapp)

currentapp$SK\_ID\_CURR <- factor(currentapp$SK\_ID\_CURR)

Currentwithprevious <- left\_join(currentapp, shortpreviousapp\_v7, by = "SK\_ID\_CURR")

str(Currentwithprevious)

Currentwithprevious$new[is.na(Currentwithprevious$new)] <- "Yes"

Currentwithprevious$approvedper[is.na(Currentwithprevious$approvedper)] <- 1

Currentwithprevious$refusedper[is.na(Currentwithprevious$refusedper)] <- 0

write.csv(Currentwithprevious, file = "C:/Users/dnsingh/Downloads/Harrisburg/06ANYL530/Project/all/Current with previous output.csv")

#merging test data

testapp <- read.csv("C:/Users/dnsingh/Downloads/Harrisburg/06ANYL530/Project/all/application\_test.csv", stringsAsFactors = FALSE)

str(testapp)

testapp$SK\_ID\_CURR <- factor(testapp$SK\_ID\_CURR)

testwithprevious <- left\_join(testapp, shortpreviousapp\_v7, by = "SK\_ID\_CURR")

str(testwithprevious)

testwithprevious$new[is.na(testwithprevious$new)] <- "Yes"

testwithprevious$approvedper[is.na(testwithprevious$approvedper)] <- 1

testwithprevious$refusedper[is.na(testwithprevious$refusedper)] <- 0

write.csv(testwithprevious, file = "C:/Users/dnsingh/Downloads/Harrisburg/06ANYL530/Project/all/test with previous output.csv")

**PART A: Naive Bayes**

library(dplyr)

library(caret)

library(klaR)

library(ranger)

library(randomForest)

library(naivebayes)

library(caTools)

data <- read.csv("C:/Users/GECR/Desktop/Harrisburg/Semesters/ML-I/Machinelearning1/Project/PrevApp\_Bur\_Merge\_train.csv", sep = ',')

data2 <- read.csv("C:/Users/GECR/Desktop/Harrisburg/Semesters/ML-I/Machinelearning1/Project/PrevApp\_Bur\_Merge\_test.csv", sep = ',')

#75/25 split

set.seed(12345)

data\_rand <- data[order(runif(307511)), ]

sample = sample.split(data\_rand,SplitRatio = 0.75)

## train

data\_train =subset(data\_rand,sample ==TRUE)

prop.table(table(data\_train$TARGET))

## test

data\_test=subset(data\_rand, sample==FALSE)

prop.table(table(data\_test$TARGET))

#deal with imbalanced data

data\_train\_pos<-subset(data\_train,data\_train$TARGET=='1')

data\_train\_neg<-subset(data\_train,data\_train$TARGET=='0')

set.seed(12345)

data\_train\_neg\_rand <- data\_train\_neg[order(runif(20000)), ]

#data\_train\_neg\_sample <- data\_rand[1:20000, ]

data\_train\_new <-rbind(data\_train\_neg\_rand,data\_train\_pos)

##

data\_train\_new <- data\_train\_new[,-c(1:3)]

table(data\_train\_new$TARGET)

train <- data\_train\_new[, c("TARGET", "EXT\_SOURCE\_1","EXT\_SOURCE\_2","EXT\_SOURCE\_3", "approvedper", "refusedper", "new", "DAYS\_BIRTH", "DAYS\_EMPLOYED", "NAME\_EDUCATION\_TYPE",

"DAYS\_ID\_PUBLISH", "CODE\_GENDER", "ORGANIZATION\_TYPE","AMT\_ANNUITY", "DAYS\_REGISTRATION", "AMT\_GOODS\_PRICE", "AMT\_CREDIT",

"FLAG\_DOCUMENT\_3","FLAG\_EMP\_PHONE","DAYS\_LAST\_PHONE\_CHANGE","DAYS\_ID\_PUBLISH","AMT\_ANNUITY","REGION\_RATING\_CLIENT\_W\_CITY",

"REG\_CITY\_NOT\_WORK\_CITY","OWN\_CAR\_AGE","FLAG\_OWN\_CAR")]

test <- data\_test[, c("TARGET", "EXT\_SOURCE\_1","EXT\_SOURCE\_2","EXT\_SOURCE\_3", "approvedper", "refusedper", "new", "DAYS\_BIRTH", "DAYS\_EMPLOYED", "NAME\_EDUCATION\_TYPE",

"DAYS\_ID\_PUBLISH", "CODE\_GENDER", "ORGANIZATION\_TYPE","AMT\_ANNUITY", "DAYS\_REGISTRATION", "AMT\_GOODS\_PRICE", "AMT\_CREDIT",

"FLAG\_DOCUMENT\_3","FLAG\_EMP\_PHONE","DAYS\_LAST\_PHONE\_CHANGE","DAYS\_ID\_PUBLISH","AMT\_ANNUITY","REGION\_RATING\_CLIENT\_W\_CITY",

"REG\_CITY\_NOT\_WORK\_CITY","OWN\_CAR\_AGE","FLAG\_OWN\_CAR")]

train[] <- lapply(train, function(x) if(is.integer(x)) as.numeric(x) else x)

train$FLAG\_DOCUMENT\_3 <- as.factor(train$FLAG\_DOCUMENT\_3)

train$FLAG\_EMP\_PHONE <- as.factor(train$FLAG\_EMP\_PHONE)

train$TARGET <- as.factor(train$TARGET)

test[] <- lapply(test, function(x) if(is.integer(x)) as.numeric(x) else x)

test$FLAG\_DOCUMENT\_3 <- as.factor(test$FLAG\_DOCUMENT\_3)

test$FLAG\_EMP\_PHONE <- as.factor(test$FLAG\_EMP\_PHONE)

test$TARGET <- as.factor(test$TARGET)

> ##naive bayes

> set.seed(12345)

> fit1 <- train(train[ ,-1], train$TARGET, data = train, 'naive\_bayes', trControl=trainControl(method='cv',number=10))

> fit1

Naive Bayes

38455 samples

25 predictor

2 classes: '0', '1'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 34609, 34609, 34610, 34610, 34609, 34609, ...

Resampling results across tuning parameters:

usekernel Accuracy Kappa

FALSE 0.6512290 0.3039331

TRUE 0.6624884 0.3226285

Tuning parameter 'laplace' was held constant at a value of 0

Tuning parameter 'adjust' was held constant at a value of 1

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were laplace = 0, usekernel =

TRUE and adjust = 1.

> pred1 <- predict(fit1, test)

> (p <- table(pred1,test$TARGET))

pred1 0 1

0 50033 2356

1 21501 4014

> (precision1 = diag(p)/colSums(p))

0 1

0.6994296 0.6301413

> (recall1 = diag(p)/rowSums(p))

0 1

0.9550287 0.1573192

> (f1 = 2\*precision1\*recall1/(precision1+recall1))

0 1

0.8074853 0.2517798

> (Accuracy1 = sum(diag(p))/sum(p)\*100)

[1] 69.37641

> varImp(fit1)

ROC curve variable importance

only 20 most important variables shown (out of 25)

Importance

OWN\_CAR\_AGE 100.00

EXT\_SOURCE\_1 61.99

EXT\_SOURCE\_2 61.52

DAYS\_BIRTH 33.80

EXT\_SOURCE\_3 29.01

DAYS\_ID\_PUBLISH 22.34

DAYS\_ID\_PUBLISH.1 22.34

refusedper 22.13

DAYS\_LAST\_PHONE\_CHANGE 21.98

approvedper 21.53

CODE\_GENDER 19.30

REGION\_RATING\_CLIENT\_W\_CITY 18.85

DAYS\_REGISTRATION 18.47

NAME\_EDUCATION\_TYPE 17.46

REG\_CITY\_NOT\_WORK\_CITY 16.47

FLAG\_DOCUMENT\_3 15.48

ORGANIZATION\_TYPE 13.85

FLAG\_EMP\_PHONE 13.82

AMT\_GOODS\_PRICE 11.98

DAYS\_EMPLOYED 9.22

### predicting validation naive\_bayes

data2 <- data2[, c("SK\_ID\_CURR", "EXT\_SOURCE\_1","EXT\_SOURCE\_2","EXT\_SOURCE\_3", "approvedper", "refusedper", "new", "DAYS\_BIRTH", "DAYS\_EMPLOYED", "NAME\_EDUCATION\_TYPE",

"DAYS\_ID\_PUBLISH", "CODE\_GENDER", "ORGANIZATION\_TYPE","AMT\_ANNUITY", "DAYS\_REGISTRATION", "AMT\_GOODS\_PRICE", "AMT\_CREDIT",

"FLAG\_DOCUMENT\_3","FLAG\_EMP\_PHONE","DAYS\_LAST\_PHONE\_CHANGE","DAYS\_ID\_PUBLISH","AMT\_ANNUITY","REGION\_RATING\_CLIENT\_W\_CITY",

"REG\_CITY\_NOT\_WORK\_CITY","OWN\_CAR\_AGE","FLAG\_OWN\_CAR")]

data2[] <- lapply(data2, function(x) if(is.integer(x)) as.numeric(x) else x)

data2$FLAG\_DOCUMENT\_3 <- as.factor(data2$FLAG\_DOCUMENT\_3)

data2$FLAG\_EMP\_PHONE <- as.factor(data2$FLAG\_EMP\_PHONE)

data2$SK\_ID\_CURR <- as.integer(data2$SK\_ID\_CURR)

##naive bayes validation

data2$TARGET <- predict(fit1, data2)

submit4 <- data2[,c("SK\_ID\_CURR", "TARGET")]

write.csv(submit4, "submitfile4.csv")

**PART B: Random Forest**

library(dplyr)

library(caret)

library(klaR)

library(ranger)

library(randomForest)

library(naivebayes)

library(caTools)

data <- read.csv("C:/Users/GECR/Desktop/Harrisburg/Semesters/ML-I/Machinelearning1/Project/PrevApp\_Bur\_Merge\_train.csv", sep = ',')

data2 <- read.csv("C:/Users/GECR/Desktop/Harrisburg/Semesters/ML-I/Machinelearning1/Project/PrevApp\_Bur\_Merge\_test.csv", sep = ',')

#75/25 split

set.seed(12345)

data\_rand <- data[order(runif(307511)), ]

sample = sample.split(data\_rand,SplitRatio = 0.75)

## train

data\_train =subset(data\_rand,sample ==TRUE)

prop.table(table(data\_train$TARGET))

## test

data\_test=subset(data\_rand, sample==FALSE)

prop.table(table(data\_test$TARGET))

#deal with imbalanced data

data\_train\_pos<-subset(data\_train,data\_train$TARGET=='1')

data\_train\_neg<-subset(data\_train,data\_train$TARGET=='0')

set.seed(12345)

data\_train\_neg\_rand <- data\_train\_neg[order(runif(20000)), ]

#data\_train\_neg\_sample <- data\_rand[1:20000, ]

data\_train\_new <-rbind(data\_train\_neg\_rand,data\_train\_pos)

##

data\_train\_new <- data\_train\_new[,-c(1:3)]

table(data\_train\_new$TARGET)

train <- data\_train\_new[, c("TARGET", "EXT\_SOURCE\_1","EXT\_SOURCE\_2","EXT\_SOURCE\_3", "approvedper", "refusedper", "new", "DAYS\_BIRTH", "DAYS\_EMPLOYED", "NAME\_EDUCATION\_TYPE",

"DAYS\_ID\_PUBLISH", "CODE\_GENDER", "ORGANIZATION\_TYPE","AMT\_ANNUITY", "DAYS\_REGISTRATION", "AMT\_GOODS\_PRICE", "AMT\_CREDIT",

"FLAG\_DOCUMENT\_3","FLAG\_EMP\_PHONE","DAYS\_LAST\_PHONE\_CHANGE","DAYS\_ID\_PUBLISH","AMT\_ANNUITY","REGION\_RATING\_CLIENT\_W\_CITY",

"REG\_CITY\_NOT\_WORK\_CITY","OWN\_CAR\_AGE","FLAG\_OWN\_CAR")]

test <- data\_test[, c("TARGET", "EXT\_SOURCE\_1","EXT\_SOURCE\_2","EXT\_SOURCE\_3", "approvedper", "refusedper", "new", "DAYS\_BIRTH", "DAYS\_EMPLOYED", "NAME\_EDUCATION\_TYPE",

"DAYS\_ID\_PUBLISH", "CODE\_GENDER", "ORGANIZATION\_TYPE","AMT\_ANNUITY", "DAYS\_REGISTRATION", "AMT\_GOODS\_PRICE", "AMT\_CREDIT",

"FLAG\_DOCUMENT\_3","FLAG\_EMP\_PHONE","DAYS\_LAST\_PHONE\_CHANGE","DAYS\_ID\_PUBLISH","AMT\_ANNUITY","REGION\_RATING\_CLIENT\_W\_CITY",

"REG\_CITY\_NOT\_WORK\_CITY","OWN\_CAR\_AGE","FLAG\_OWN\_CAR")]

train[] <- lapply(train, function(x) if(is.integer(x)) as.numeric(x) else x)

train$FLAG\_DOCUMENT\_3 <- as.factor(train$FLAG\_DOCUMENT\_3)

train$FLAG\_EMP\_PHONE <- as.factor(train$FLAG\_EMP\_PHONE)

train$TARGET <- as.factor(train$TARGET)

test[] <- lapply(test, function(x) if(is.integer(x)) as.numeric(x) else x)

test$FLAG\_DOCUMENT\_3 <- as.factor(test$FLAG\_DOCUMENT\_3)

test$FLAG\_EMP\_PHONE <- as.factor(test$FLAG\_EMP\_PHONE)

test$TARGET <- as.factor(test$TARGET)

##random forest

set.seed(12345)

train <- na.roughfix(train)

test <- na.roughfix(test)

fit2 <- train(TARGET~., data = train, 'ranger', num.tree= 50, trControl=trainControl(method='cv',number=10))

fit2

pred2 <- predict(fit2, test)

(p2 <- table(pred2,test$TARGET))

(precision2 = diag(p2)/colSums(p2))

(recall2 = diag(p2)/rowSums(p2))

(f2 = 2\*precision2\*recall2/(precision2+recall2))

(Accuracy2 = sum(diag(p2))/sum(p2)\*100)

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

Growing trees.. Progress: 74%. Estimated remaining time: 11 seconds.

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

Growing trees.. Progress: 66%. Estimated remaining time: 16 seconds.

Growing trees.. Progress: 82%. Estimated remaining time: 6 seconds.

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

Growing trees.. Progress: 98%. Estimated remaining time: 0 seconds.

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.

> fit2

Random Forest

38455 samples

25 predictor

2 classes: '0', '1'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 34609, 34609, 34610, 34610, 34609, 34609, ...

Resampling results across tuning parameters:

mtry splitrule Accuracy Kappa

2 gini 0.6635289 0.3236261

2 extratrees 0.6470944 0.2911194

43 gini 0.6738522 0.3456182

43 extratrees 0.6738781 0.3458566

85 gini 0.6694834 0.3368706

85 extratrees 0.6713817 0.3408475

Tuning parameter 'min.node.size' was held constant at a value of 1

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were mtry = 43, splitrule

= extratrees and min.node.size = 1.

> pred2 <- predict(fit2, test)

> (p2 <- table(pred2,test$TARGET))

pred2 0 1

0 53251 2634

1 18283 3736

> (precision2 = diag(p2)/colSums(p2))

0 1

0.7444152 0.5864992

> (recall2 = diag(p2)/rowSums(p2))

0 1

0.9528675 0.1696716

> (f2 = 2\*precision2\*recall2/(precision2+recall2))

0 1

0.8358408 0.2632005

> (Accuracy2 = sum(diag(p2))/sum(p2)\*100)

[1] 73.15029

## random forest validation

data3 <- na.roughfix(data2)

data3$TARGET <- predict(fit2, data3)

submit5 <- data3[, c("SK\_ID\_CURR", "TARGET")]

write.csv(submit5, "submitfile5.csv")

**PART C: XgBoost**

library(xgboost)

library(Matrix)

library(dplyr)

library(magrittr)

library(caTools)

data <- read.csv("C:/Users/GECR/Desktop/Harrisburg/Semesters/ML-I/Machinelearning1/Project/PrevApp\_Bur\_Merge\_train.csv", sep = ',')

data2 <- read.csv("C:/Users/GECR/Desktop/Harrisburg/Semesters/ML-I/Machinelearning1/Project/PrevApp\_Bur\_Merge\_test.csv", sep = ',')

data\_N <- data[, c("TARGET", "EXT\_SOURCE\_1","EXT\_SOURCE\_2","EXT\_SOURCE\_3", "approvedper", "refusedper", "new",

"DAYS\_BIRTH", "DAYS\_EMPLOYED", "NAME\_EDUCATION\_TYPE",

"DAYS\_ID\_PUBLISH", "CODE\_GENDER", "ORGANIZATION\_TYPE","AMT\_ANNUITY", "AMT\_INCOME\_TOTAL",

"DAYS\_REGISTRATION", "AMT\_GOODS\_PRICE", "AMT\_CREDIT",

"FLAG\_DOCUMENT\_3","FLAG\_EMP\_PHONE","DAYS\_LAST\_PHONE\_CHANGE","DAYS\_ID\_PUBLISH",

"AMT\_ANNUITY","REGION\_RATING\_CLIENT\_W\_CITY","REG\_CITY\_NOT\_WORK\_CITY","OWN\_CAR\_AGE","FLAG\_OWN\_CAR",

"NAME\_CONTRACT\_TYPE", "NAME\_FAMILY\_STATUS", "REGION\_POPULATION\_RELATIVE")]

data\_N1 <- data2 [, c("SK\_ID\_CURR","EXT\_SOURCE\_1","EXT\_SOURCE\_2","EXT\_SOURCE\_3", "approvedper", "refusedper", "new",

"DAYS\_BIRTH", "DAYS\_EMPLOYED", "NAME\_EDUCATION\_TYPE",

"DAYS\_ID\_PUBLISH", "CODE\_GENDER", "ORGANIZATION\_TYPE","AMT\_ANNUITY", "AMT\_INCOME\_TOTAL",

"DAYS\_REGISTRATION", "AMT\_GOODS\_PRICE", "AMT\_CREDIT",

"FLAG\_DOCUMENT\_3","FLAG\_EMP\_PHONE","DAYS\_LAST\_PHONE\_CHANGE","DAYS\_ID\_PUBLISH",

"AMT\_ANNUITY","REGION\_RATING\_CLIENT\_W\_CITY","REG\_CITY\_NOT\_WORK\_CITY","OWN\_CAR\_AGE","FLAG\_OWN\_CAR",

"NAME\_CONTRACT\_TYPE", "NAME\_FAMILY\_STATUS", "REGION\_POPULATION\_RELATIVE")]

###converting the categorical variables into factors and integers into numbers except TARGET

data\_N[, c('FLAG\_DOCUMENT\_3', 'FLAG\_EMP\_PHONE', 'REGION\_RATING\_CLIENT\_W\_CITY', 'REG\_CITY\_NOT\_WORK\_CITY')] <- lapply(data\_N[, c('FLAG\_DOCUMENT\_3', 'FLAG\_EMP\_PHONE', 'TARGET', 'REGION\_RATING\_CLIENT\_W\_CITY', 'REG\_CITY\_NOT\_WORK\_CITY')], as.factor)

data\_N[] <- lapply(data\_N, function(x) if(is.integer(x)) as.numeric(x) else x)

data\_N1[, c('FLAG\_DOCUMENT\_3', 'FLAG\_EMP\_PHONE', 'REGION\_RATING\_CLIENT\_W\_CITY', 'REG\_CITY\_NOT\_WORK\_CITY')] <- lapply(data\_N1[, c('FLAG\_DOCUMENT\_3', 'FLAG\_EMP\_PHONE','REGION\_RATING\_CLIENT\_W\_CITY', 'REG\_CITY\_NOT\_WORK\_CITY')], as.factor)

data\_N1[] <- lapply(data\_N1, function(x) if(is.integer(x)) as.numeric(x) else x)

## test train down sampling

set.seed(12345)

data\_rand <- data\_N[order(runif(307511)), ]

sample = sample.split(data\_rand,SplitRatio = 0.75)

## train

data\_train =subset(data\_rand,sample ==TRUE)

prop.table(table(data\_train$TARGET))

## test

data\_test=subset(data\_rand, sample==FALSE)

prop.table(table(data\_test$TARGET))

#deal with imbalanced data

data\_train\_pos<-subset(data\_train,data\_train$TARGET=='1')

data\_train\_neg<-subset(data\_train,data\_train$TARGET=='0')

set.seed(12345)

data\_train\_neg\_rand <- data\_train\_neg[order(runif(20000)), ]

#data\_train\_neg\_sample <- data\_rand[1:20000, ]

data\_train\_new <-rbind(data\_train\_neg\_rand,data\_train\_pos)

data\_train\_new <- data\_train\_new[order(runif(38211)), ]

data\_train\_new <- na.omit(data\_train\_new)

### one-hot encoding

##train matrix

options(na.action='na.pass')

trainm <- sparse.model.matrix(TARGET ~. -1, data = data\_train\_new)

dim(trainm)

train\_label <- data\_train\_new[, "TARGET"]

#View(train\_label)

train\_matrix <- xgb.DMatrix(data = as.matrix(trainm), label = train\_label)

##test matrix

data\_test <- na.omit(data\_test)

testm <- sparse.model.matrix(TARGET ~. -1, data = data\_test)

test\_label <- data\_test[, "TARGET"]

test\_matrix <- xgb.DMatrix(data = as.matrix(testm), label= test\_label)

table(train\_label)

table(test\_label)

##parameters

nc <- length(unique(train\_label))

xgb\_params <- list("objective" = "multi:softprob",

"eval\_metric" = "mlogloss",

"num\_class" = nc)

watchlist <- list(train = train\_matrix, test = test\_matrix)

###extreme gradient boosting model

xgmodel <- xgb.train(params = xgb\_params,

data = train\_matrix,

nrounds = 100,

watchlist = watchlist,

eta = 0.02,

max.depth = 10,

gamma = 5,

subsample = 0.5,

colsample\_bytree = 1,

missing = NA,

seed = 1234)

> ###extreme gradient boosting model

> xgmodel <- xgb.train(params = xgb\_params,

+ data = train\_matrix,

+ nrounds = 100,

+ watchlist = watchlist,

+ eta = 0.02,

+ max.depth = 10,

+ gamma = 5,

+ subsample = 0.5,

+ colsample\_bytree = 1,

+ missing = NA,

+ seed = 1234)

[1] train-mlogloss:0.673381 test-mlogloss:0.673378

[2] train-mlogloss:0.654389 test-mlogloss:0.654382

[3] train-mlogloss:0.636125 test-mlogloss:0.636115

[4] train-mlogloss:0.618549 test-mlogloss:0.618537

[5] train-mlogloss:0.601625 test-mlogloss:0.601611

[6] train-mlogloss:0.585316 test-mlogloss:0.585299

[7] train-mlogloss:0.569591 test-mlogloss:0.569572

[8] train-mlogloss:0.554419 test-mlogloss:0.554398

[9] train-mlogloss:0.539774 test-mlogloss:0.539750

[10] train-mlogloss:0.525629 test-mlogloss:0.525604

[11] train-mlogloss:0.511961 test-mlogloss:0.511933

[12] train-mlogloss:0.498746 test-mlogloss:0.498717

[13] train-mlogloss:0.485964 test-mlogloss:0.485934

[14] train-mlogloss:0.473596 test-mlogloss:0.473565

[15] train-mlogloss:0.461623 test-mlogloss:0.461592

[16] train-mlogloss:0.450029 test-mlogloss:0.449996

[17] train-mlogloss:0.438796 test-mlogloss:0.438761

[18] train-mlogloss:0.427909 test-mlogloss:0.427873

[19] train-mlogloss:0.417354 test-mlogloss:0.417317

[20] train-mlogloss:0.407118 test-mlogloss:0.407081

[21] train-mlogloss:0.397187 test-mlogloss:0.397149

[22] train-mlogloss:0.387551 test-mlogloss:0.387512

[23] train-mlogloss:0.378196 test-mlogloss:0.378156

[24] train-mlogloss:0.369112 test-mlogloss:0.369072

[25] train-mlogloss:0.360289 test-mlogloss:0.360248

[26] train-mlogloss:0.351718 test-mlogloss:0.351676

[27] train-mlogloss:0.343389 test-mlogloss:0.343347

[28] train-mlogloss:0.335294 test-mlogloss:0.335250

[29] train-mlogloss:0.327422 test-mlogloss:0.327379

[30] train-mlogloss:0.319768 test-mlogloss:0.319724

[31] train-mlogloss:0.312323 test-mlogloss:0.312278

[32] train-mlogloss:0.305080 test-mlogloss:0.305034

[33] train-mlogloss:0.298032 test-mlogloss:0.297986

[34] train-mlogloss:0.291172 test-mlogloss:0.291126

[35] train-mlogloss:0.284494 test-mlogloss:0.284448

[36] train-mlogloss:0.277993 test-mlogloss:0.277946

[37] train-mlogloss:0.271661 test-mlogloss:0.271614

[38] train-mlogloss:0.265495 test-mlogloss:0.265448

[39] train-mlogloss:0.259488 test-mlogloss:0.259441

[40] train-mlogloss:0.253636 test-mlogloss:0.253588

[41] train-mlogloss:0.247934 test-mlogloss:0.247885

[42] train-mlogloss:0.242376 test-mlogloss:0.242328

[43] train-mlogloss:0.236959 test-mlogloss:0.236910

[44] train-mlogloss:0.231678 test-mlogloss:0.231629

[45] train-mlogloss:0.226529 test-mlogloss:0.226480

[46] train-mlogloss:0.221508 test-mlogloss:0.221459

[47] train-mlogloss:0.216612 test-mlogloss:0.216562

[48] train-mlogloss:0.211835 test-mlogloss:0.211786

[49] train-mlogloss:0.207176 test-mlogloss:0.207126

[50] train-mlogloss:0.202631 test-mlogloss:0.202581

[51] train-mlogloss:0.198196 test-mlogloss:0.198146

[52] train-mlogloss:0.193867 test-mlogloss:0.193817

[53] train-mlogloss:0.189644 test-mlogloss:0.189594

[54] train-mlogloss:0.185521 test-mlogloss:0.185471

[55] train-mlogloss:0.181497 test-mlogloss:0.181446

[56] train-mlogloss:0.177568 test-mlogloss:0.177518

[57] train-mlogloss:0.173732 test-mlogloss:0.173682

[58] train-mlogloss:0.169987 test-mlogloss:0.169937

[59] train-mlogloss:0.166330 test-mlogloss:0.166280

[60] train-mlogloss:0.162758 test-mlogloss:0.162708

[61] train-mlogloss:0.159270 test-mlogloss:0.159220

[62] train-mlogloss:0.155863 test-mlogloss:0.155812

[63] train-mlogloss:0.152535 test-mlogloss:0.152484

[64] train-mlogloss:0.149283 test-mlogloss:0.149232

[65] train-mlogloss:0.146106 test-mlogloss:0.146056

[66] train-mlogloss:0.143003 test-mlogloss:0.142952

[67] train-mlogloss:0.139970 test-mlogloss:0.139919

[68] train-mlogloss:0.137006 test-mlogloss:0.136955

[69] train-mlogloss:0.134109 test-mlogloss:0.134058

[70] train-mlogloss:0.131278 test-mlogloss:0.131228

[71] train-mlogloss:0.128511 test-mlogloss:0.128461

[72] train-mlogloss:0.125807 test-mlogloss:0.125756

[73] train-mlogloss:0.123163 test-mlogloss:0.123112

[74] train-mlogloss:0.120579 test-mlogloss:0.120528

[75] train-mlogloss:0.118052 test-mlogloss:0.118001

[76] train-mlogloss:0.115581 test-mlogloss:0.115531

[77] train-mlogloss:0.113166 test-mlogloss:0.113115

[78] train-mlogloss:0.110804 test-mlogloss:0.110753

[79] train-mlogloss:0.108494 test-mlogloss:0.108443

[80] train-mlogloss:0.106235 test-mlogloss:0.106184

[81] train-mlogloss:0.104026 test-mlogloss:0.103975

[82] train-mlogloss:0.101865 test-mlogloss:0.101815

[83] train-mlogloss:0.099752 test-mlogloss:0.099702

[84] train-mlogloss:0.097685 test-mlogloss:0.097635

[85] train-mlogloss:0.095664 test-mlogloss:0.095613

[86] train-mlogloss:0.093686 test-mlogloss:0.093635

[87] train-mlogloss:0.091751 test-mlogloss:0.091701

[88] train-mlogloss:0.089859 test-mlogloss:0.089808

[89] train-mlogloss:0.088007 test-mlogloss:0.087956

[90] train-mlogloss:0.086195 test-mlogloss:0.086145

[91] train-mlogloss:0.084423 test-mlogloss:0.084372

[92] train-mlogloss:0.082688 test-mlogloss:0.082638

[93] train-mlogloss:0.080991 test-mlogloss:0.080941

[94] train-mlogloss:0.079330 test-mlogloss:0.079280

[95] train-mlogloss:0.077705 test-mlogloss:0.077655

[96] train-mlogloss:0.076115 test-mlogloss:0.076065

[97] train-mlogloss:0.074559 test-mlogloss:0.074509

[98] train-mlogloss:0.073036 test-mlogloss:0.072986

[99] train-mlogloss:0.071545 test-mlogloss:0.071495

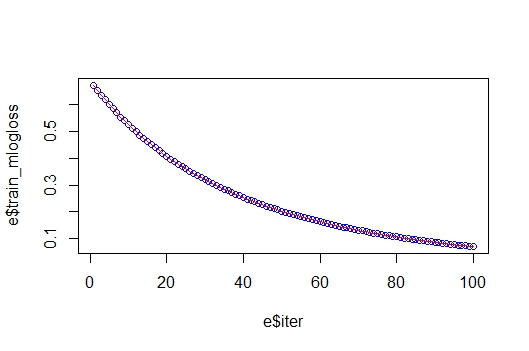
[100] train-mlogloss:0.070086 test-mlogloss:0.070036

> ## errors

> e <- data.frame(xgmodel$evaluation\_log)

> plot(e$iter, e$train\_mlogloss, col = 'blue')

> lines(e$iter, e$test\_mlogloss, col = 'red')



> ## prediction and cf matrix

> p3 <- predict(xgmodel, newdata = test\_matrix)

> pred3 <- matrix(p, nrow = nc, ncol = length(p)/nc)%>%

+ t()%>%

+ data.frame()%>%

+ mutate(lable = test\_label, max\_prob = max.col(.,"last")-1)

> pred3

X1 X2 lable max\_prob

1 0.93236852 0.0676315 0 0

2 0.93236852 0.0676315 0 0

3 0.93236852 0.0676315 0 0

4 0.93236852 0.0676315 0 0

5 0.93236852 0.0676315 0 0

6 0.93236852 0.0676315 0 0

7 0.93236852 0.0676315 0 0

8 0.06776171 0.9322383 1 1

9 0.93236852 0.0676315 0 0

10 0.93236852 0.0676315 0 0

11 0.93236852 0.0676315 0 0

12 0.93236852 0.0676315 0 0

13 0.06776171 0.9322383 1 1

14 0.93236852 0.0676315 0 0

15 0.93236852 0.0676315 0 0

16 0.93236852 0.0676315 0 0

17 0.93236852 0.0676315 0 0

18 0.93236852 0.0676315 0 0

19 0.93236852 0.0676315 0 0

20 0.93236852 0.0676315 0 0

21 0.93236852 0.0676315 0 0

22 0.93236852 0.0676315 0 0

23 0.93236852 0.0676315 0 0

24 0.93236852 0.0676315 0 0

25 0.93236852 0.0676315 0 0

26 0.93236852 0.0676315 0 0

27 0.93236852 0.0676315 0 0

28 0.93236852 0.0676315 0 0

29 0.93236852 0.0676315 0 0

30 0.93236852 0.0676315 0 0

31 0.93236852 0.0676315 0 0

32 0.93236852 0.0676315 0 0

33 0.93236852 0.0676315 0 0

34 0.93236852 0.0676315 0 0

35 0.93236852 0.0676315 0 0

36 0.93236852 0.0676315 0 0

37 0.93236852 0.0676315 0 0

38 0.93236852 0.0676315 0 0

39 0.93236852 0.0676315 0 0

40 0.93236852 0.0676315 0 0

41 0.93236852 0.0676315 0 0

42 0.93236852 0.0676315 0 0

43 0.93236852 0.0676315 0 0

44 0.93236852 0.0676315 0 0

45 0.06776171 0.9322383 1 1

46 0.93236852 0.0676315 0 0

47 0.93236852 0.0676315 0 0

48 0.93236852 0.0676315 0 0

49 0.93236852 0.0676315 0 0

50 0.93236852 0.0676315 0 0

51 0.93236852 0.0676315 0 0

52 0.06776171 0.9322383 1 1

53 0.93236852 0.0676315 0 0

54 0.93236852 0.0676315 0 0

55 0.93236852 0.0676315 0 0

56 0.93236852 0.0676315 0 0

57 0.93236852 0.0676315 0 0

58 0.93236852 0.0676315 0 0

59 0.93236852 0.0676315 0 0

60 0.93236852 0.0676315 0 0

61 0.93236852 0.0676315 0 0

62 0.93236852 0.0676315 0 0

63 0.93236852 0.0676315 0 0

64 0.93236852 0.0676315 0 0

65 0.93236852 0.0676315 0 0

66 0.06776171 0.9322383 1 1

67 0.06776171 0.9322383 1 1

68 0.93236852 0.0676315 0 0

69 0.93236852 0.0676315 0 0

70 0.93236852 0.0676315 0 0

71 0.93236852 0.0676315 0 0

72 0.93236852 0.0676315 0 0

73 0.93236852 0.0676315 0 0

74 0.93236852 0.0676315 0 0

75 0.93236852 0.0676315 0 0

76 0.93236852 0.0676315 0 0

77 0.93236852 0.0676315 0 0

78 0.93236852 0.0676315 0 0

79 0.93236852 0.0676315 0 0

80 0.93236852 0.0676315 0 0

81 0.93236852 0.0676315 0 0

82 0.06776171 0.9322383 1 1

83 0.93236852 0.0676315 0 0

84 0.93236852 0.0676315 0 0

85 0.93236852 0.0676315 0 0

86 0.93236852 0.0676315 0 0

87 0.93236852 0.0676315 0 0

88 0.93236852 0.0676315 0 0

89 0.93236852 0.0676315 0 0

90 0.93236852 0.0676315 0 0

91 0.93236852 0.0676315 0 0

92 0.06776171 0.9322383 1 1

93 0.93236852 0.0676315 0 0

94 0.93236852 0.0676315 0 0

95 0.93236852 0.0676315 0 0

96 0.93236852 0.0676315 0 0

97 0.93236852 0.0676315 0 0

98 0.93236852 0.0676315 0 0

99 0.93236852 0.0676315 0 0

100 0.93236852 0.0676315 0 0

101 0.06776171 0.9322383 1 1

102 0.93236852 0.0676315 0 0

103 0.93236852 0.0676315 0 0

104 0.93236852 0.0676315 0 0

105 0.06776171 0.9322383 1 1

106 0.93236852 0.0676315 0 0

107 0.93236852 0.0676315 0 0

108 0.93236852 0.0676315 0 0

109 0.93236852 0.0676315 0 0

110 0.93236852 0.0676315 0 0

111 0.93236852 0.0676315 0 0

112 0.93236852 0.0676315 0 0

113 0.93236852 0.0676315 0 0

114 0.93236852 0.0676315 0 0

115 0.93236852 0.0676315 0 0

116 0.93236852 0.0676315 0 0

117 0.93236852 0.0676315 0 0

118 0.93236852 0.0676315 0 0

119 0.93236852 0.0676315 0 0

120 0.93236852 0.0676315 0 0

121 0.93236852 0.0676315 0 0

122 0.93236852 0.0676315 0 0

123 0.93236852 0.0676315 0 0

124 0.93236852 0.0676315 0 0

125 0.93236852 0.0676315 0 0

126 0.93236852 0.0676315 0 0

127 0.93236852 0.0676315 0 0

128 0.93236852 0.0676315 0 0

129 0.93236852 0.0676315 0 0

130 0.93236852 0.0676315 0 0

131 0.93236852 0.0676315 0 0

132 0.93236852 0.0676315 0 0

133 0.93236852 0.0676315 0 0

134 0.93236852 0.0676315 0 0

135 0.06776171 0.9322383 1 1

136 0.93236852 0.0676315 0 0

137 0.93236852 0.0676315 0 0

138 0.93236852 0.0676315 0 0

139 0.93236852 0.0676315 0 0

140 0.06776171 0.9322383 1 1

141 0.93236852 0.0676315 0 0

142 0.93236852 0.0676315 0 0

143 0.93236852 0.0676315 0 0

144 0.93236852 0.0676315 0 0

145 0.93236852 0.0676315 0 0

146 0.93236852 0.0676315 0 0

147 0.93236852 0.0676315 0 0

148 0.93236852 0.0676315 0 0

149 0.93236852 0.0676315 0 0

150 0.93236852 0.0676315 0 0

151 0.93236852 0.0676315 0 0

152 0.93236852 0.0676315 0 0

153 0.93236852 0.0676315 0 0

154 0.93236852 0.0676315 0 0

155 0.93236852 0.0676315 0 0

156 0.93236852 0.0676315 0 0

157 0.93236852 0.0676315 0 0

158 0.93236852 0.0676315 0 0

159 0.93236852 0.0676315 0 0

160 0.93236852 0.0676315 0 0

161 0.93236852 0.0676315 0 0

162 0.93236852 0.0676315 0 0

163 0.93236852 0.0676315 0 0

164 0.93236852 0.0676315 0 0

165 0.93236852 0.0676315 0 0

166 0.93236852 0.0676315 0 0

167 0.93236852 0.0676315 0 0

168 0.06776171 0.9322383 1 1

169 0.93236852 0.0676315 0 0

170 0.93236852 0.0676315 0 0

171 0.93236852 0.0676315 0 0

172 0.93236852 0.0676315 0 0

173 0.93236852 0.0676315 0 0

174 0.93236852 0.0676315 0 0

175 0.93236852 0.0676315 0 0

176 0.93236852 0.0676315 0 0

177 0.93236852 0.0676315 0 0

178 0.93236852 0.0676315 0 0

179 0.93236852 0.0676315 0 0

180 0.93236852 0.0676315 0 0

181 0.93236852 0.0676315 0 0

182 0.93236852 0.0676315 0 0

183 0.06776171 0.9322383 1 1

184 0.93236852 0.0676315 0 0

185 0.93236852 0.0676315 0 0

186 0.93236852 0.0676315 0 0

187 0.93236852 0.0676315 0 0

188 0.93236852 0.0676315 0 0

189 0.93236852 0.0676315 0 0

190 0.93236852 0.0676315 0 0

191 0.93236852 0.0676315 0 0

192 0.06776171 0.9322383 1 1

193 0.93236852 0.0676315 0 0

194 0.93236852 0.0676315 0 0

195 0.93236852 0.0676315 0 0

196 0.93236852 0.0676315 0 0

197 0.93236852 0.0676315 0 0

198 0.06776171 0.9322383 1 1

199 0.93236852 0.0676315 0 0

200 0.93236852 0.0676315 0 0

201 0.93236852 0.0676315 0 0

202 0.93236852 0.0676315 0 0

203 0.93236852 0.0676315 0 0

204 0.93236852 0.0676315 0 0

205 0.93236852 0.0676315 0 0

206 0.93236852 0.0676315 0 0

207 0.93236852 0.0676315 0 0

208 0.93236852 0.0676315 0 0

209 0.93236852 0.0676315 0 0

210 0.93236852 0.0676315 0 0

211 0.93236852 0.0676315 0 0

212 0.93236852 0.0676315 0 0

213 0.93236852 0.0676315 0 0

214 0.93236852 0.0676315 0 0

215 0.93236852 0.0676315 0 0

216 0.93236852 0.0676315 0 0

217 0.93236852 0.0676315 0 0

218 0.93236852 0.0676315 0 0

219 0.93236852 0.0676315 0 0

220 0.93236852 0.0676315 0 0

221 0.93236852 0.0676315 0 0

222 0.93236852 0.0676315 0 0

223 0.93236852 0.0676315 0 0

224 0.93236852 0.0676315 0 0

225 0.93236852 0.0676315 0 0

226 0.93236852 0.0676315 0 0

227 0.93236852 0.0676315 0 0

228 0.93236852 0.0676315 0 0

229 0.93236852 0.0676315 0 0

230 0.93236852 0.0676315 0 0

231 0.93236852 0.0676315 0 0

232 0.06776171 0.9322383 1 1

233 0.93236852 0.0676315 0 0

234 0.06776171 0.9322383 1 1

235 0.93236852 0.0676315 0 0

236 0.93236852 0.0676315 0 0

237 0.93236852 0.0676315 0 0

238 0.93236852 0.0676315 0 0

239 0.93236852 0.0676315 0 0

240 0.93236852 0.0676315 0 0

241 0.93236852 0.0676315 0 0

242 0.93236852 0.0676315 0 0

243 0.93236852 0.0676315 0 0

244 0.93236852 0.0676315 0 0

245 0.93236852 0.0676315 0 0

246 0.93236852 0.0676315 0 0

247 0.93236852 0.0676315 0 0

248 0.93236852 0.0676315 0 0

249 0.93236852 0.0676315 0 0

250 0.93236852 0.0676315 0 0

[ reached getOption("max.print") -- omitted 10721 rows ]

> p4 <- table(Predictied = pred$max\_prob, Actual= pred$lable)

> p4

Actual

Predictied 0 1

0 10258 0

1 0 713

> (precision3 = diag(p4)/colSums(p4))

0 1

1 1

> (recall3 = diag(p4)/rowSums(p4))

0 1

1 1

> (f3 = 2\*precision3\*recall3/(precision3+recall3))

0 1

1 1

> (Accuracy3 = sum(diag(p4))/sum(p4)\*100)

[1] 100

> data\_N1 <- na.omit(data\_N1)

> #data\_N1$TARGET <- NA

> valm <- sparse.model.matrix(SK\_ID\_CURR ~.-1, data = data\_N1)

> val\_label <- data\_N1[,"TARGET"]

Error in `[.data.frame`(data\_N1, , "TARGET") : undefined columns selected

> val\_matrix <- xgb.DMatrix(data = as.matrix(valm))

>

> taret <- predict(xgmodel,val\_matrix)

Error in predict.xgb.Booster(xgmodel, val\_matrix) :

Feature names stored in `object` and `newdata` are different!

>

**PART D : RPART model**

data <- read.csv("C:/Users/dnsingh/Downloads/Harrisburg/06ANYL530/Project/Machine learning project/PrevApp\_Bur\_Merge\_train.csv", stringsAsFactors = TRUE)

summary(data)

data$TARGET <- factor(data$TARGET)

prop.table(table(data$TARGET))

#80/20 split

set.seed(12345)

data\_rand <- data[order(runif(307511)), ]

#data\_train <- data\_rand[1:184506, ]

#data\_test <- data\_rand[184507:307511, ]

data\_train <- data\_rand[1:246008, ]

data\_test <- data\_rand[246009:307511, ]

prop.table(table(data\_train$TARGET))

prop.table(table(data\_test$TARGET))

#deal with imbalanced data

data\_train\_pos<-subset(data\_train,data\_train$TARGET=='1')

data\_train\_neg<-subset(data\_train,data\_train$TARGET=='0')

set.seed(12345)

data\_train\_rand <- data\_train\_neg[order(runif(169656)), ]

data\_train\_neg\_sample <- data\_rand[1:20000, ]

data\_train\_new <-rbind(data\_train\_neg\_sample,data\_train\_pos)

prop.table(table(data\_train\_new$TARGET))

#rpart

data\_train\_new$TARGET <-as.factor(data\_train\_new$TARGET)

rpart.mod <- rpart(TARGET~NAME\_CONTRACT\_TYPE + FLAG\_OWN\_CAR + FLAG\_OWN\_REALTY + NAME\_INCOME\_TYPE + HOUSETYPE\_MODE + EMERGENCYSTATE\_MODE + CREDIT\_ACTIVEBad.debt + EXT\_SOURCE\_1+EXT\_SOURCE\_2+EXT\_SOURCE\_3+approvedper+refusedper+new+DAYS\_BIRTH+DAYS\_EMPLOYED+NAME\_EDUCATION\_TYPE+DAYS\_ID\_PUBLISH+CODE\_GENDER+AMT\_ANNUITY+DAYS\_REGISTRATION+AMT\_GOODS\_PRICE+AMT\_CREDIT,data=data\_train\_new)

summary(rpart.mod)

pred1 <- predict(rpart.mod, data\_test,type="class")

pred1

(conf1 <- table(pred1, data\_test$TARGET))

(Accuracy <- sum(diag(conf1))/sum(conf1)\*100)

precision <- posPredValue(pred1, data\_test$TARGET, positive="1")

precision

recall <- sensitivity(pred1, data\_test$TARGET, positive="1")

recall

F1 <- (2 \* precision \* recall) / (precision + recall)

F1

roc.curve(data\_test$TARGET,pred1)

data\_final2 <- data\_final

data\_final2$TARGET <- predict(rpart.mod, data\_final2,type="class")

str(data\_final2)

submitfile2 <- data\_final2[,c("SK\_ID\_CURR", "TARGET")]

str(submitfile2)

write.csv(submitfile2, "C:/Users/dnsingh/Downloads/Harrisburg/06ANYL530/Project/Machine learning project/output/submitfile2.csv")