Using Machine Learning to Predict Credit Worthiness of Individuals

A comparative analysis of different machine learning algorithms

MATH2191 – Applied Research Program

Credit Risk Scoring 2

Final Project Report

Saurabh Mallik (s3623575) | Dilip Chandra (s3574580)

Shiyuan Lou (s3639669) | Mohammad (3650497)

Table of Contents

I. Project Contribution Report

Table I - Contribution Table

Item	**Contribution Weight (%)**	**Saurabh Mallik**	**Dilip Chandra**	**Mohammad**	**Shiyuan Lou**
Meeting with Industry supervisor	10%	3%	3%	3%	1%
Project Proposal	15%	4%	5%	5%	1%
Data Preparation	25%	13%	7%	2.5%	2.5%
Model Specification & Tuning	30%	10%	10%	4.5%	5.5%
Project Wrap Up	20%	5%	5%	5%	5%
Total Contribution	**100%**	**35%**	**30%**	**20%**	**15%**

Saurabh Mallik (s3623575): 35% Contribution (Data Preparation 70%, Logistic Regression Models)

Dilip Chandra (s3574580): 30% Contribution (Data Preparation 30%, Random Forest Models)

Shiyuan Lou (s3639669): 15% Contribution (Data Visualization)

Mohammad (3650497): 20% Contribution (Gradient Boosting Model)

II. Disclaimer

We declare the following to be our own work, unless otherwise referenced, as defined by the University's policy on plagiarism.

III. Acknowledgement

We would like to thank RMIT Univeristy for providing us with such an amazing opportunity and interesting project. Special thanks go out to Dr. Yan Wang and Denwick Munjeri for providing valuable feedback during project progress.

We would also like to extend thanks to Nick Jonker, for providing us with such an interesting dataset, and providing us with guidance during the entirety of the project.

1. Executive Summary

The purpose of the report is to use machine learning algorithms to create models that can successfully predict whether an individual is good or bad credit risk. The datasets for this project have been sourced from Home Credit Default Risk on Kaggle (https://www.kaggle.com/c/home-credit-default-risk). The project was divided into three main stages and the key results and inferences derived from them are explained below.

Data Understanding and Pre-Processing

- The datasets were joined using means to summarize the multiple IDs.
- 26% of the values (17 million values) were missing in the dataset, which were imputed based on column means.
- 92% of the target values were good and 8% bad in the training dataset.

Feature Selection

In this stage, chi-squared weights of the features were used to determine the absolute top 100 features that predict the credit worthiness. From the feature selection phase, we noticed that variables from other tables like EXT_SOURCE_3 (from bureau table) came with highest weight (.051), even variables like code_reject came quite high.

	attr_importance
EXT_SOURCE_3	0.05121313
EXT_SOURCE_2	0.04992867
NAME_CONTRACT_STATUS	0.04441452
DAYS_FIRST_DRAWING	0.04131077
CODE_REJECT_REASON	0.03939756
DAYS_CREDIT	0.03913177
NAME_EDUCATION_TYPE	0.03467883
CREDIT_ACTIVE	0.03376334
DAYS_CREDIT_UPDATE	0.03357824
CODE_GENDER	0.03349399

Model Fitting and Tuning

Area under ROC curve was used to compare the model performance.

1. Logistic Regression: The best model was found with 12 features and it gave an AUROC score of 0.724.
2. Random Forest: We used all 100 features in the second model with 10 variable splits at every node, this gave us an AUROC score of 0.727.
3. Gradient Boosting: For gradient boosting, we used a numerical version of our dataset to get accurate model scores, the xgboost best iteration for the model was 0.761

All models seemed to give AUROC scores between .72 and .76, which is extremely reassuring, given the depth and vastness of the dataset. For this project, the best model was the gradient boosting model with an AUROC score 0.761.

Imputation of missing values was the biggest challenge, as each variable needed to be treated individually, and understanding the variable and deciding on best method for imputation per variable was a hard task.

For future work, the key recommendation is using all the variables from all 7 datasets and using information gain for feature selection as well, to help compare results with the chi-squared results and fine tune feature selection.

2. Introduction

Over the many years that banks, and other lending institutions have existed, the one tool they have prominently used to generate revenues since the very beginning are loans. The motivations that drive individuals to these financial organizations are of varying kinds, ranging between gaining assistance to accomplish their investments to realizing their personal goals. Here, the lending institutions profit from the interest rate they charge on the lent amount. The crucial role that loans play in a nation's economic development cannot be overlooked. It is this availing of loans that facilitates the collection and increment of capital which is instrumental for investment decisions.

Credit scoring can be described as a set of decision models that aids in the process of lending. It is a key instrument for all types of lenders as it is these very techniques that the lenders utilize to determine who is worthy of how much credit, what is the charge that they should be granted the loan at, and what operational strategies can be employed in order to maximise the profitability (Guegan & Hassani, 2018). This task of credit scoring is made easier when an individual has an existing credit history, as it becomes much easier for lenders to assume how credible is the borrower and if they are at risk of defaulting. Although extensively beneficial, the requirement of a credit history, disables banks from tapping into the potential of the population that is not linked to these banks. The population that has not used any bank related service until the need for a loan is vast, and this market remains untapped due to the lack of a credit history. Banks today want to unleash the potential that this group holds as it would generate an even greater revenue.

However, in order to make decisions that are accurate and at the same time would aid in reducing the credit risk these lending institutions are relying on getting a closer look into big data and deciphering the pattern. The growth in borrowing of loans has caused the process to evolve over the past few years (Kruppa et al, 2013). Changes in the format of application and channels of interaction can be observed. Instead of visiting banks and personally approaching officials, today people can fill out loan applications on online platforms, in turn eliminating tedious steps and making the process even more accessible. Banks and other lending organisation have been known to use techniques that would help flag default risk at initial stages to avoid losses and mitigate uncertainty. These organisations use detailed credit prediction models to decide whether or not to approve a loan request.

> A good prediction model has the potential to help the lender gain leverage to advance the maximum borrowing potential to its clients (Khandani, Kim & Lo, 2010), while ensuring that the risk threshold is not breached. Our main aim in this project is to understand which machine learning algorithms predicts the default risk of an individual, in an accurate and effective manner. Today, machine learning is being used across industries all over the world, to provide simple solutions to complex questions. These models have the ability to self-learn and automatically apply appropriate mathematical and statistical computations to the data, to derive desired results (Sumaiya & Aswani, 2017).

The purpose of this project is to identify accuracy scores through area under the ROC of each of the machine learning algorithms and compare which approach handles the data in a better manner. We would also like to understand the limitations of the approaches and further studies required to improve our results.

3. Data Description:

- application_train.csv
  - This is the main table.
  - Static data for all applications. One row represents one loan in our data sample.
  - 122 variables and 308k observations (including contract type, gender, amount credit etc.). The main table which has been broken down into train and test.
- 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).
  - 121 variables and 48.7k observations (same variables as train set minus the target variable)
- bureau_balance.csv
  - Monthly balances of previous credits in Credit Bureau.
  - 3 variable and 27.3 million observations. This dataset shows monthly balances of previous credits of client.
- POS_CASH_balance.csv
  - Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
  - 8 Variables and 10 million observations. This dataset shows the monthly balance of previous point of sale and cash loans of applicant.

- credit_card_balance.csv
  - Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
  - 23 variables and 3.84 million observations. Showing monthly credit card balances of the applicants.
- previous_application.csv
  - All previous applications for Home Credit loans of clients who have loans in our sample.
  - 37 Variables and 1.67 million observations. An entire record of all previous applications for loans.
- installments_payments.csv
  - Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample.
  - 8 Variables and 13.6 million observations, which talk about repayment history for previous credits of the institution.

4. Objectives:

The main aim for this project is to identify which predictive analysis method predicts an individual's default risk most accurately. For this we would be using Logistic regression, random forest and gradient boosting classifiers to train our models. Comparison will be based on AUROC scores, and hence our aim is also to see which model gives the best AUROC score.

In order to achieve the above, we will also have to undergo data pre-processing and feature selection, and so our secondary aims are as follows:

- Identify the descriptive features which are most important and will act as the building block in predicting the default risk.
- Understand the challenges in pre-processing, to be able to have an advantage in fine tuning similar models in the future.

5. Methodology

In this section of the report we describe the technique used at various stage of the project. The main objective is to understand which analysis techniques are most suitable for big data pertaining to Home Credit Default Risk. The methodology is divided into the following parts:

- Stage I – Data Pre-Processing
- Stage II – Feature Selection
- Stage III – Model Fitting and Tuning

Data Pre-Processing

In this step, we will check the dimensions of the data and structure, understand the variables in the data set, and understand the meaning of each value for the variables. Dataset used for this project consisted of 7 data tables which were joined using unique ID assigned to each applicant. Every row in the bureau data set is identified by the feature SK_ID_ BUREAU. Every row in the loan data set is identified by the feature SK_ID_CURR. Each row in the previous application data set is identified by the feature SK_ID_PREV. All the data sets were later grouped by SK_ID_CURR and SK_ID_ BUREAU and SK_ID_PREV were dropped as it would cause data redundancy. At this stage of the project, we were successful in complying 6 data tables to have a full dataset consisting of all variables. Storing data in a consistent form that matches the semantics of the data set is important for further modelling analysis. Test data set was dropped as we created a split within the final data set (70:30) for model tuning. Full data consisted of 307,511 observation and 212 variables in total.

We also defined one new feature named Days before Due (DBD = DAYS_INSTALMENT - DAYS_ENTRY_PAYMENT). This feature shows different between when the instalment of previous credit was supposed to be paid and when was the instalments of previous credit paid actually. A positive outcome states that instalments were paid after the instalment due date indicating a delay in payments.

Next step is to apply data manipulation techniques. We start by checking for the plausibility of values, identifying and handling outliers, dealing with missing values and cleaning data for obvious errors. There were no obvious inconsistency errors in the dataset. To identify missing values we have used is.na() function which returns a logical vector with TRUE in the element locations that contain missing values represented by NA. Out of total 65 million values, 17 million values were missing. Imputation techniques were used to impute missing. To fix missing values we used u Hmisc() package which has a convenient wrapper function allowing one to specify what function is used to compute imputed values from the non-missing. In most of the instances, we replaced missing values with mean.

To eliminate outliers for numeric variables, we created cap function which involves replacing the outliers with the quantile range. Capping involves replacing the outliers with the nearest neighbours that are not outliers. Outliers that lie outside the outlier fences on a box-plot, will be replaced by those observations outside the lower limit with the value of 5th percentile and those that lie above the upper limit, with the value of 95th percentile.

Below we can see the cap quantile function used on numerical variables.

```
```{r}
cap <- function(x){
    quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
    x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
    x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
    x
}
```
```

Next use the factor function for variables that takes only predefined, finite number of values.

Below we can see two examples of variables being factored.

```
fullset_final$NAME_INCOME_TYPE <- factor(fullset_final$NAME_INCOME_TYPE, levels = c("1", "2","3","4","5","6","7","8"),
labels = c("Businessman","Commercial-associate","Maternity-leave","Pensioner","State-servant","Student","Unemployed","W
orking"))

fullset_final$NAME_EDUCATION_TYPE <- factor(fullset_final$NAME_EDUCATION_TYPE, levels = c("1", "2","3","4","5"), labels
= c("Academic degree","Higher education","Incomplete higher","Lower secondary","Secondary/secondary special"))

fullset_final$NAME_HOUSING_TYPE <- factor(fullset_final$NAME_HOUSING_TYPE, levels = c("1", "2","3","4","5","6"), labels
= c("Co-op apartment","House / apartment","Municipal apartment","Office apartment","Rented apartment","With parents" ))
```

We have also used the binning function to categorize a number of continuous values into a smaller number of buckets (bins) where each bucket defines a numerical interval. For example, AMT_GOODS_PRICE variable is measured by continuous values ranged between zero and 1 million. Binning places each value into one bucket if the value falls into the interval that the bucket covers. Below we can see an example for binning function for this project.

```
fullset_final$AMT_GOODS_PRICE.x <- fullset_final$AMT_GOODS_PRICE.x %>% cap()
fullset_final$AMT_GOODS_PRICE.x <- cut(fullset_final$AMT_GOODS_PRICE.x, breaks = c(0, 100000, 250000, 400000,550000,
700000, 850000, Inf), labels = c("0-100k","101k - 250k","251k - 400k","401k - 550k","551k-700k","700k-850K", "850k+"))
```

All variables have been simplified by the above tasks and we have clean data available for the development and deployment of statistical analysis and modelling.

Feature Selection

Identifying the most important predictor variables, that explains the major variance of the target variable is key to build high performing models. Feature selection is a process to a subset of the original set of variables which are best representatives of the data. The original data set consisted of 212 variables. We used Chi Square method to give weights to the feature's attribute importance. Based on these weights and data exploration inferences, we were able to select the top 100 variables which would be discussed in the results section of the report.

The screenshot below shows the codes used for features selection.

```
```{r}
options(java.parameters = "-Xmx4096m")
library(rJava)
library(FSelector)
fin <- fullset_final %>% select(-SK_ID_CURR)
weights<- chi.squared(TARGET~., fin)
```
```

Model Fitting and Tuning

Model fitting is a performance measure of how well the machine learning model performs to similar data on which it was trained. Each model has different performance characteristics. Model tuning is essential to produce practical and applicable insights to the practical business problem. A well fitted model produces more accurate outcome. If the model is overfitted, the outcome would match too closely to the data and a model that is underfitted doesn't match closely enough. Based on the summary of the dataset, we chosen the following three supervised machine learning models.

- Logistic Regression
- Random Forest
- Extreme Gradient Boosting

Logistic regression:

These models use independent descriptive features which were selected from the weighted chi-squared selected. Two logistic regression models were created which included 8 and 12 predictive features targeting good or bad credit risk. Some of the descriptive features chosen in this model include Age, Gender, Occupation Type, Education Type and External Sources. Performance comparison of the models is measured by using the AUC values. Codes used for to generate the model can be seen below.

```
```{r}
model_log = glm(data = fullset_final,TARGET ~ CODE_GENDER + Age + AMT_CREDIT.x + OCCUPATION_TYPE + NAME_EDUCATION_TYPE +
NAME_INCOME_TYPE + FLAG_OWN_CAR + ORGANIZATION_TYPE,family=binomial(link=logit))
```
```

```
```{r}
model_log2 = glm(data = fullset_final,TARGET ~ CODE_GENDER + Age + AMT_CREDIT.x + OCCUPATION_TYPE + NAME_EDUCATION_TYPE +
AMT_CREDIT_SUM + REGION_RATING_CLIENT + EXT_SOURCE_1 + EXT_SOURCE_2 + EXT_SOURCE_3 + FLAG_OWN_CAR + ORGANIZATION_TYPE +
FLAG_OWN_CAR + OWN_CAR_AGE,family=binomial(link=logit))
```
```

Random forest:

This classifier model creates a set of decision trees from randomly selected sample data set from the original data set. The model aggregates the votes from different decision trees to decide the final class of the test object. Random Forest uses mtry feature to generate outcome (mtry = sqrt(p) where P refers to the number of descriptive features). For this project, p = 102, hence root of p is 10.09. Two random forest models were created which used mtry of 10 where all 100 variables were chosen & mtry 3 where top 10 descriptive features were selected based in logistic regression model. AUC scores have been used to measure the high level of accuracy. Codes used for to generate the model can be seen below.

```
```{r}
model2 <- randomForest(TARGET ~ ., data = training_data, ntree = 500, mtry = 10, importance = TRUE)
model2
```
```

```
```{r}
model3 <- randomForest(TARGET ~
CODE_GENDER+Age+AMT_CREDIT.x+AMT_CREDIT_SUM+REGION_RATING_CLIENT+EXT_SOURCE_1+EXT_SOURCE_2+EXT_SOURCE_3+NAME_EDUCATION_
TYPE+OCCUPATION_TYPE+ORGANIZATION_TYPE+OWN_CAR_AGE, data = training_data, ntree = 500, mtry = 3, importance = TRUE)
```
```

Extreme Gradient Boosting:

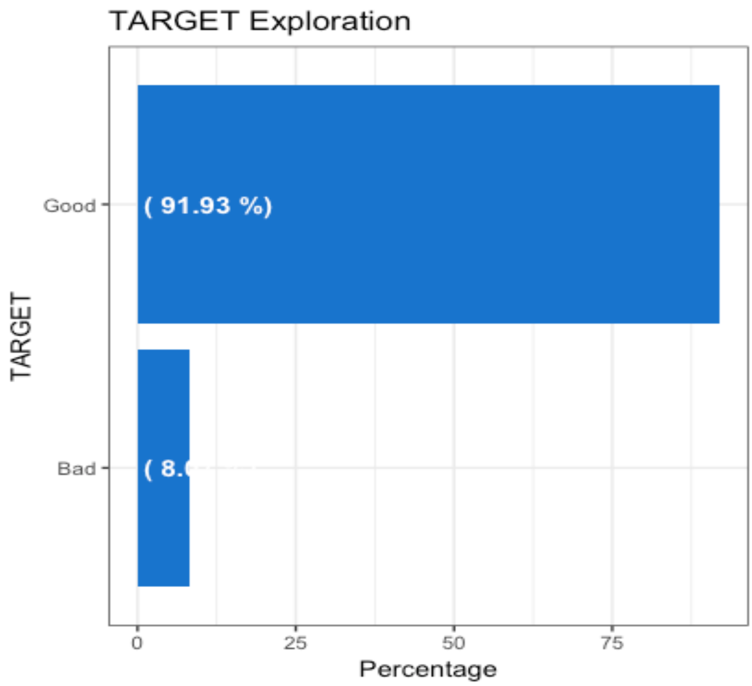
This was the final model used for this project. The model uses multiple base learner types to exploit a computer's hardware to speed up gradient descent components. For this model we needed to convert all the variables in the data set to unique numeric values. All 100 descriptive features were used in this model and the iteration was to 3000 rounds with an early stopping after 200 rounds. AUC scores have been used to measure the high level of accuracy. The results of these models would be discussed in the results stage of the report.

6. Results

Data Exploration

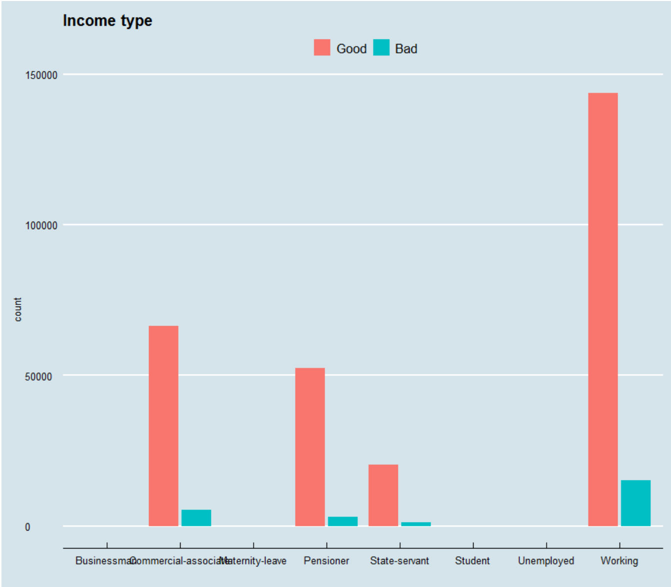
The data provided by Home Credit describes lines of credit (loans) to the unbanked population. Predicting whether or not a client repays a loan or delay is usually of great importance. Home Credit hosted this competition on Kaggle to figure out what sort of models may reduce the risk of overdue.

The target of the model is the data named as **TARGET** with 1 denoting repaid loan and 0 for un-paid loan. The paid loan is 282686 while unpaid one is 24825. In other words, the majority of clients (91.92712%) paid their debt. Non-performing loans only take a small account.



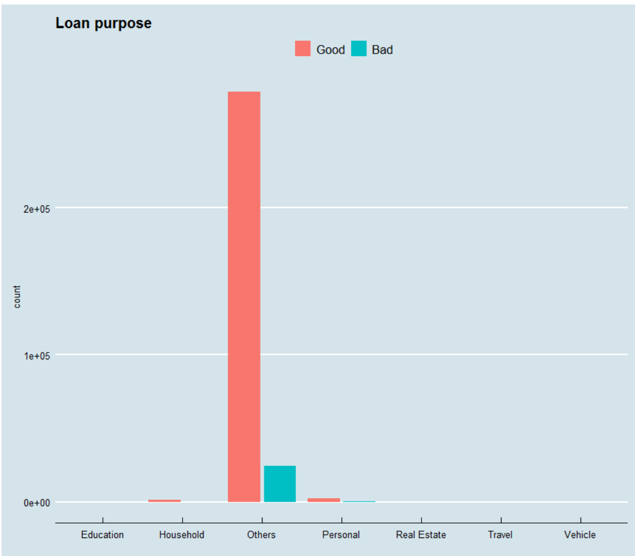
Before credit arrangement, the background of clients is inspected as key feature, such as education type, income type, occupation type etc. A brief summary of key features helps give an overall perspective of data set.

Income Type



Income type reflects the repayment ability of the clients in background inspection. From the plot above, those who have **professional jobs** (Commercial-associate and Working) are primary part of paying credits (0.6997051). What's more, the proportion of on-time repayments is also quite high.

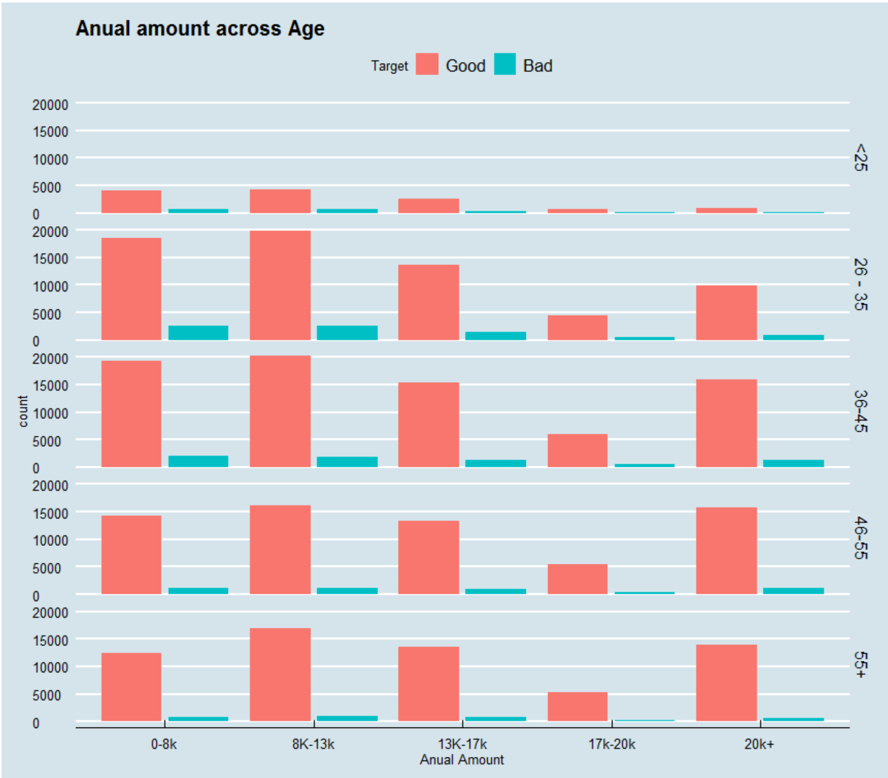
Loan Purpose



The loan purpose above gives a brief summary of field of consumption. Most credit lying in others implies that people are more intending to enjoy their life without the stress of living.

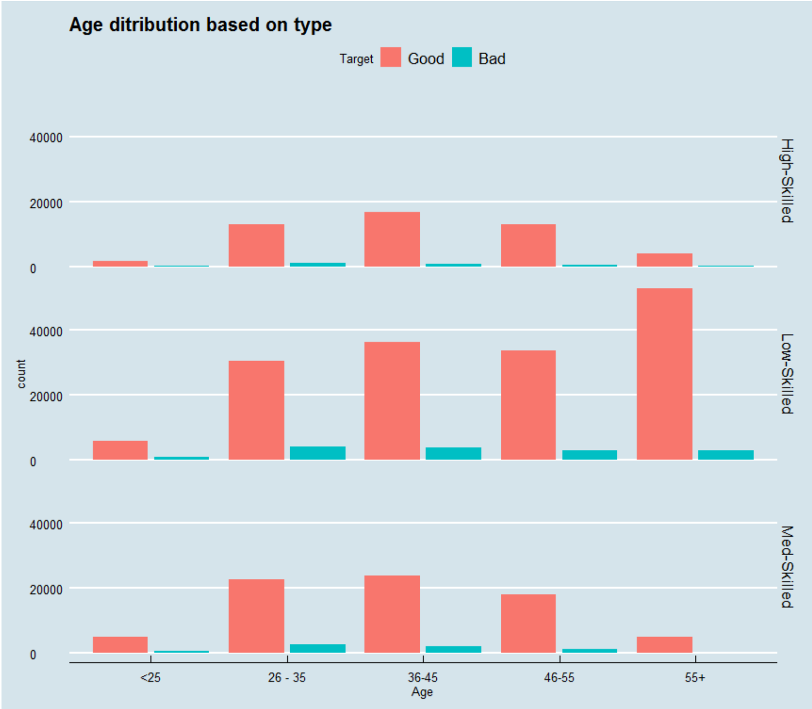
These two features merely give an overview of data set. Since there are more than 200 variables, it's unnecessary to show all the plots of each feature.

Annual amount across age



The annual receivable amount represents the repayment ability of clients. As the common sense goes, the higher income, the more ability to repay. The plot above illustrates the saying vividly. An interesting condition is the elder people are less willing to apply for credit. The possible reason lies in their deposit reduce their dependence on applying for credit.

Age distribution based on occupation type



According to the plot above, it clarifies the clients with low-skilled tend to apply for credit.

Feature selection

Since the model cannot be trained with missing data, it is essential to turn them into meaningful value. The common practice is to replace missing value with mean.

Feature selection plays an important role in machine learning process. Although various criteria work, we use chi square based feature selection to measure attribute importance of each feature and giving them weights. This helped select the top 100 variables that formed the final dataset.

Below is a table showing the top 10 variables based on the chi square test.

	attr_importance
EXT_SOURCE_3	0.05121313
EXT_SOURCE_2	0.04992867
NAME_CONTRACT_STATUS	0.04441452
DAYS_FIRST_DRAWING	0.04131077
CODE_REJECT_REASON	0.03939756
DAYS_CREDIT	0.03913177
NAME_EDUCATION_TYPE	0.03467883
CREDIT_ACTIVE	0.03376334
DAYS_CREDIT_UPDATE	0.03357824
CODE_GENDER	0.03349399

From the above we can see that EXT_SOURCE_3 and 2, which are important variables in the bureau dataset, have ranked highest based on attribute importance weights in the chi squared feature selection. This shows that banks give importance to credit bureau data. The reason of rejection (CODE_REJECT_REASON) has also ranked in the top 10, along with gender (CODE_GENDER) and education type (NAME_EDUCATION_TYPE).

Next section of results will discuss the model summaries and achieved AUROC (Area under curves) scores, which will in turn help us decide which model performs best.

Model fitting and tuning

Logistic regression:

The area under the ROC curve for logistic regression quantifies the overall ability to discriminate the reasons or variables directly impacting the 'TARGET' variable and the total amount of variables selected for the logit model were eight in total with a level of significance with an AUC score of 0.637 is considered to be a reliable start.

The base model included a total of 8 features selected out of a total of 100 variables which were reduced in the pre-processing stage from a total of 212 variables. The best model included a total of 12 features selected out of a total of 100 variables with all of them to be highly significant. The trapezoids under the curve represents an approximation of area and a parametric method which used a maximum likelihood estimator to fit a smooth curve to the data points.

The summary on the right shows that most of the variables were significant, and some were highly significant with 10 fisher scoring iterations and having **Akaike information criterion** value of 167014 which calculates the quality of each model.

```
Call:
glm(formula = TARGET ~ CODE_GENDER + Age + AMT_CREDIT.x + OCCUPATION_TYPE +
    NAME_EDUCATION_TYPE + NAME_INCOME_TYPE + FLAG_OWN_CAR + ORGANIZATION_TYPE,
    family = binomial(link = logit), data = fullset_final)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0780  -0.4490  -0.3735  -0.3044   2.9725

Coefficients:
                                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                                 -12.867396  61.573720  -0.209  0.83447
CODE_GENDERF                                  0.380002   0.015519  24.487  < 2e-16 ***
CODE_GENDERNot Specified                     -9.149268  98.238459  -0.093  0.92580
Age26 - 35                                   -0.090401   0.028975  -3.120  0.00181 **
Age36-45                                     -0.374947   0.029589 -12.672  < 2e-16 ***
Age46-55                                     -0.568979   0.030952 -18.383  < 2e-16 ***
Age55+                                       -0.837501   0.038492 -21.758  < 2e-16 ***
AMT_CREDIT.x101k - 250k                       0.288083   0.059334   4.855 1.20e-06 ***
AMT_CREDIT.x251k - 400k                       0.534853   0.058762   9.102  < 2e-16 ***
AMT_CREDIT.x401k - 550k                       0.695534   0.058706  11.848  < 2e-16 ***
AMT_CREDIT.x551k-700k                         0.615847   0.059927  10.277  < 2e-16 ***
AMT_CREDIT.x700k-850K                         0.482347   0.060899   7.920 2.37e-15 ***
AMT_CREDIT.x850k+                             0.278605   0.059309   4.698 2.63e-06 ***
OCCUPATION_TYPELow-Skilled                    0.223619   0.022229  10.060  < 2e-16 ***
OCCUPATION_TYPEMed-Skilled                    0.210524   0.023011   9.149  < 2e-16 ***
NAME_EDUCATION_TYPEHigher education           1.090363   0.584158   1.867  0.06196 .
NAME_EDUCATION_TYPEIncomplete higher          1.341496   0.585058   2.293  0.02185 *
NAME_EDUCATION_TYPELower secondary            1.833162   0.586326   3.127  0.00177 **
NAME_EDUCATION_TYPESecondary/secondary special 1.599261  0.583994   2.738  0.00617 **
NAME_INCOME_TYPECommercial-associate          8.472589  61.570906   0.138  0.89055
NAME_INCOME_TYPEMaternity-leave              10.840157  61.577699   0.176  0.86026
NAME_INCOME_TYPEPensioner                     8.451646  61.570916   0.137  0.89082
NAME_INCOME_TYPEState-servant                 8.411598  61.570913   0.137  0.89133
NAME_INCOME_TYPEStudent                      -0.823493  76.748498  -0.011  0.99144
NAME_INCOME_TYPEUnemployed                   10.563898  61.572550   0.172  0.86378
NAME_INCOME_TYPEWorking                       8.674580  61.570906   0.141  0.88796
FLAG_OWN_CARN                                -0.346857   0.015573 -22.273  < 2e-16 ***
ORGANIZATION_TYPEBusiness                     0.127304   0.037970   3.353  0.00080 ***
ORGANIZATION_TYPEGovernment                  -0.134428   0.046644  -2.882  0.00395 **
ORGANIZATION_TYPEIndustry                    -0.017630   0.047767  -0.369  0.71207
ORGANIZATION_TYPEOthers                       0.082368   0.042185   1.953  0.05088 .
ORGANIZATION_TYPEService                      0.003625   0.042557   0.085  0.93212
ORGANIZATION_TYPETrade                        0.103734   0.046877   2.213  0.02690 *
ORGANIZATION_TYPETransport                    0.107025   0.051834   2.065  0.03895 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 172542  on 307510  degrees of freedom
Residual deviance: 166946  on 307477  degrees of freedom
AIC: 167014

Number of Fisher Scoring iterations: 10
```

The summary on the right shows that all of the variables were significant, and some were highly significant with 8 fisher scoring iterations and having **Akaike information criterion** value of 156678 which calculates the quality of each model.

```
Call:
glm(formula = TARGET ~ CODE_GENDER + Age + AMT_CREDIT.x + OCCUPATION_TYPE +
    NAME_EDUCATION_TYPE + AMT_CREDIT_SUM + REGION_RATING_CLIENT +
    EXT_SOURCE_1 + EXT_SOURCE_2 + EXT_SOURCE_3 + FLAG_OWN_CAR +
    ORGANIZATION_TYPE, family = binomial(link = logit), data = fullset_final)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4065  -0.4362  -0.3276  -0.2415   3.2683

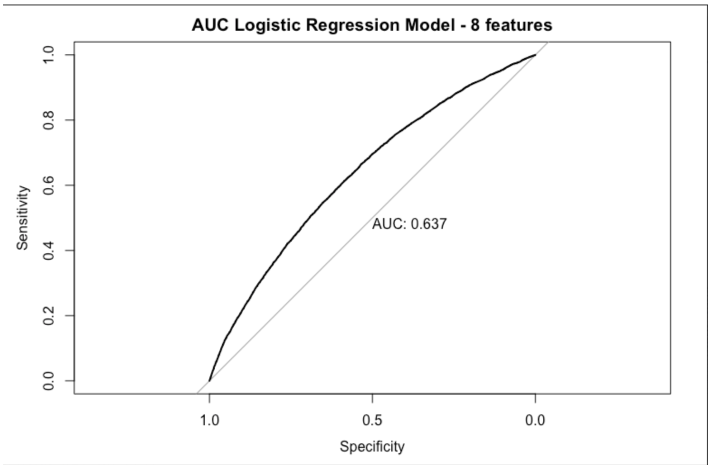
Coefficients:
                                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                                 -3.262401   0.595992  -5.474 4.40e-08 ***
CODE_GENDERF                                 0.341406   0.015936  21.423  < 2e-16 ***
CODE_GENDERNot Specified                    -7.415548  35.189449  -0.211 0.833096
Age26 - 35                                   0.159798   0.029845   5.354 8.60e-08 ***
Age36-45                                     0.064562   0.030822   2.095 0.036200 *
Age46-55                                    -0.074436   0.032647  -2.280 0.022606 *
Age55+                                      -0.393409   0.036839 -10.679  < 2e-16 ***
AMT_CREDIT.x101k - 250k                      0.283739   0.060291   4.706 2.52e-06 ***
AMT_CREDIT.x251k - 400k                      0.512832   0.059735   8.585  < 2e-16 ***
AMT_CREDIT.x401k - 550k                      0.672067   0.059717  11.254  < 2e-16 ***
AMT_CREDIT.x551k-700k                        0.619416   0.060976  10.158  < 2e-16 ***
AMT_CREDIT.x700k-850K                        0.525492   0.061968   8.480  < 2e-16 ***
AMT_CREDIT.x850k+                            0.394246   0.060377   6.530 6.59e-11 ***
OCCUPATION_TYPELow-Skilled                   0.147788   0.022566   6.549 5.78e-11 ***
OCCUPATION_TYPEMed-Skilled                   0.151844   0.023576   6.441 1.19e-10 ***
NAME_EDUCATION_TYPEHigher education          1.226880   0.590388   2.078 0.037701 *
NAME_EDUCATION_TYPEIncomplete higher         1.418975   0.591319   2.400 0.016409 *
NAME_EDUCATION_TYPELower secondary           1.787104   0.592655   3.015 0.002566 **
NAME_EDUCATION_TYPESecondary/secondary special 1.632235 0.590227  2.765 0.005685 **
AMT_CREDIT_SUM100-200k                      -0.070302   0.021098  -3.332 0.000862 ***
AMT_CREDIT_SUM200-300k                       0.001858   0.024467   0.076 0.939479
AMT_CREDIT_SUM300-400k                       0.372685   0.020895  17.836  < 2e-16 ***
AMT_CREDIT_SUM>400k                         -0.024771   0.023069  -1.074 0.282925
REGION_RATING_CLIENTL2                       0.261953   0.028370   9.234  < 2e-16 ***
REGION_RATING_CLIENTL3                       0.446331   0.031627  14.112  < 2e-16 ***
EXT_SOURCE_10.51 - 0.75                     -0.007452   0.016981  -0.439 0.660754
EXT_SOURCE_10.76 - 1                        -0.493371   0.024411 -20.211  < 2e-16 ***
EXT_SOURCE_20.26 - 0.50                     -0.570510   0.018628 -30.626  < 2e-16 ***
EXT_SOURCE_20.51 - 0.75                     -1.031157   0.018223 -56.585  < 2e-16 ***
EXT_SOURCE_20.76 - 1                        -1.602432   0.055261 -28.998  < 2e-16 ***
EXT_SOURCE_30.26 - 0.50                     -0.752131   0.019842 -37.906  < 2e-16 ***
EXT_SOURCE_30.51 - 0.75                     -1.271217   0.019207 -66.187  < 2e-16 ***
EXT_SOURCE_30.76 - 1                        -1.755416   0.039869 -44.029  < 2e-16 ***
FLAG_OWN_CARN                               -0.308962   0.016053 -19.246  < 2e-16 ***
ORGANIZATION_TYPEBusiness                    0.189424   0.038462   4.925 8.44e-07 ***
ORGANIZATION_TYPEGovernment                 -0.135364   0.047343  -2.859 0.004247 **
ORGANIZATION_TYPEIndustry                    0.014194   0.048686   0.292 0.770643
ORGANIZATION_TYPEOthers                      0.068434   0.041407   1.653 0.098390 .
ORGANIZATION_TYPEService                     0.006479   0.043520   0.149 0.881645
ORGANIZATION_TYPETrade                       0.121896   0.047608   2.560 0.010455 *
ORGANIZATION_TYPETransport                   0.168940   0.052966   3.190 0.001425 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

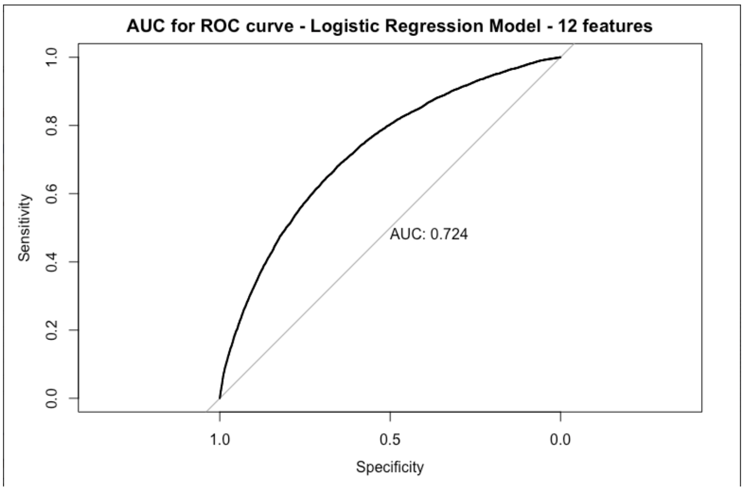
    Null deviance: 172542  on 307510  degrees of freedom
Residual deviance: 156596  on 307470  degrees of freedom
AIC: 156678

Number of Fisher Scoring iterations: 8
```

The AUC Score for the base model in the graph below, shows the value as 0.637 which denotes a fair classifier and illustrates that there is a room for improvement in the model and it is required to be near to 1 for a perfect model but if the case of perfect 1 (accuracy) then it is considered to be overfitted.



The AUC Score for the best model shows the value as 0.724 which denotes a good classifier that there is a less room for improvement in the model and it is required to be near to 1 for a perfect model but if the case of perfect 1 (accuracy) then it is considered to be overfitted. The best model would be the second one which comes under the category of excellent classifier with a good AUC score of 0.724.

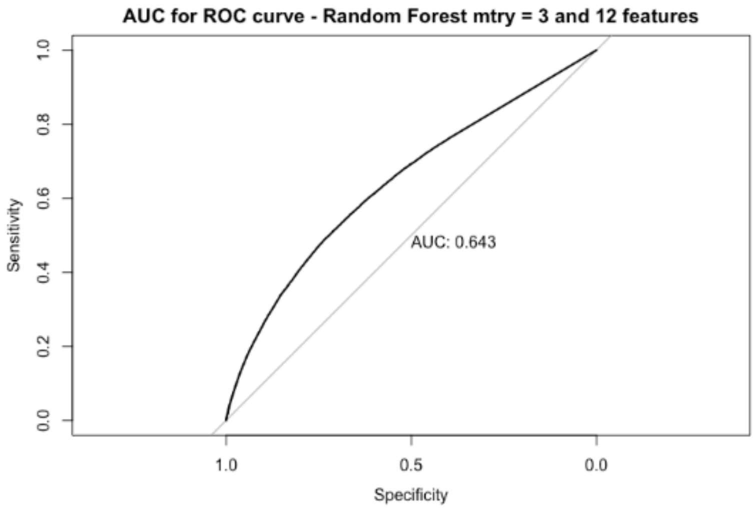


Random Forest:

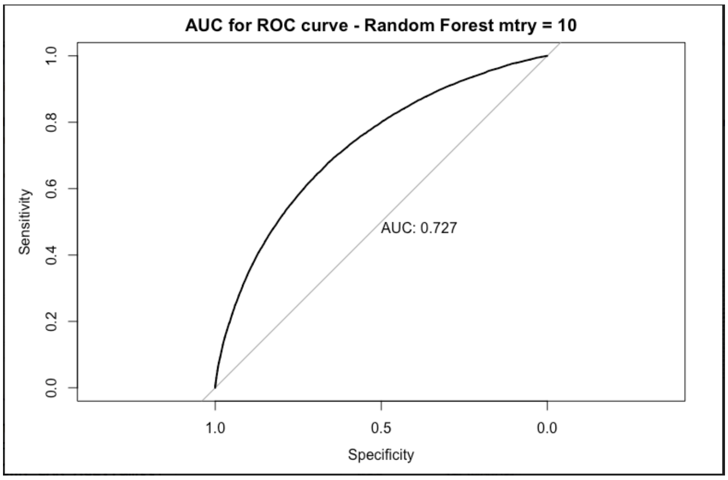
The method of classification was used to construct a multitude of decision trees at the time of training and outputting the classification of individual trees. The training algorithm for random forest involved techniques such as bagging included in running the model and finding the AUC score and defining the curve.

The results were further interpreted with a room for improvement from the beginning to the best models and there were different MTR values considered during the process to bootstrap aggregate the trees.

The area under the ROC curve for Random forest consisted of deep trees and it randomly selected the features which directly impacted the 'TARGET'. We tried various MTR values beginning from 3 to 10. It resulted in different AUC scores ranging for 0.643 to 0.727 using all and some selected features.



The base random forest model had a score of 0.643 with a fair classification and has the mtry=3 and 12 features selected with a potential to improve. The mtry value describes the three variables to be split on each node.



The best random forest model had a score of 0.727 with a good classification and has the mtry=10 and all features selected. The mtry value describes the ten variables to be split on each node.

Gradient Boosting:

The gbm package also adopts the stochastic gradient boosting strategy, a small but important tweak on the basic algorithm. Gradient boosting allowed to improve the variations of previous algorithms which had the potential of improvement in accordance to predictive performance and interpretability. It involved the low variance regression methods and application of robust regression resulting in an improvement of AUC score from the previous model to the best score of 0.761 which is the most accurate model, it included all the variables.

We included ideas from robust regression which resulted in non-parametric regression procedures with many desirable properties. The by-product lead to learn huge datasets and improve the accuracy of the model in a reduced time limitation.

```
[1251]  val-auc:0.760884
[1301]  val-auc:0.761116
[1351]  val-auc:0.761004
[1401]  val-auc:0.761050
[1451]  val-auc:0.760967
[1501]  val-auc:0.760738
Stopping. Best iteration:
[1305]  val-auc:0.761158

>
> xgb.importance(cols, model=m_xgb) %>%
+   xgb.plot.importance(top_n = 30)
>
> #--------------------------
> read_csv("../input/sample_submission.csv") %>%
+   mutate(SK_ID_CURR = as.integer(SK_ID_CURR),
+          TARGET = predict(m_xgb, dtest)) %>%
+   write_csv(paste0("tidy_xgb_", round(m_xgb$best_score, 4), ".csv"))
```

The iteration was at 1501 out of the 3000 iteration which were set in the code with an AUC value of 0.761158. The lowest AUC score that was achieved was from the first iteration which gave an AUC value of 0.708.

7. Discussions

7.1 Conclusions

Our main aim was to identify which model works best at predicting an individual's default risk. From the above sections we can see that all the models are working accurately and giving AUCROC scores between .63 and .76. The table below lists the models, AUROC scores and ranks.

Table Showing Models with Ranking and AUROC scores

Rank	Model Name	AUROC Score
1	Extreme Gradient Boosting	0.761
2	Random Forest (100 features, mtry = 10)	0.727
3	Logistic Regression (12 features)	0.724
4	Random Forest (12 features, mtry = 3)	0.643
5	Logistic Regression (8 features)	0.637

The extreme gradient boosting is giving the best AUROC score of .761 and can be said to be predicting the TARGET most accurately. The random forest and logistic regression models are working well too, however with a lower AUROC score.

The random forest model which used all features with 10 variable splits at every node and 500 trees gave the best score amongst other random forest models but could still not surpass the AUROC for the gradient boosted model. Similarly, the logistic regression best model which had all significant features turned up a lower AUROC score as well. Therefore, from this research, the gradient boosting and random forest models would be ideal choices for big datasets containing mixed data types and multiple factors.

As far as which features had the most importance, we use chi square test to measure the attribute importance and give weights to each variable on this basis. The top 10 most important variables are given below.

	attr_importance
EXT_SOURCE_3	0.05121313
EXT_SOURCE_2	0.04992867
NAME_CONTRACT_STATUS	0.04441452
DAYS_FIRST_DRAWING	0.04131077
CODE_REJECT_REASON	0.03939756
DAYS_CREDIT	0.03913177
NAME_EDUCATION_TYPE	0.03467883
CREDIT_ACTIVE	0.03376334
DAYS_CREDIT_UPDATE	0.03357824
CODE_GENDER	0.03349399

The data suggests that credit bureau variables are important in feature selection as the top 2 variables are from that set. The third most important feature is from the POS cash balance dataset and the fourth and fifth from previous application data. This helps us attain one of our secondary objectives, of identifying descripting features for model training.

The other secondary objective pertaining to challenges is mentioned in the limitations section that follows.

7.2 Limitations

This project had a few limitations which effected each stage and are mentioned below.

- The dataset had 26% missing values, and imputation and dealing with each variable individually was cumbersome. The missing values needed to be treated with a lot more care.
- 92% of the values in the target variable were good and only 8% were bad. Hence, for the model to be able to accurately predict bad credit risk was a huge limitation, and we constantly got a lot of false negative values during confusion matrix-based miscalculation computing.
- The main feature selection on the full dataset was done based on chi squared feature scoring, which gave attribute importance-based weights to each feature. Given time, more approaches such as information gain should be implemented to be able to correctly choose the predictors.
- The Pre-Processing of this data took too long, and factor reduction was based on user input factors (to avoid overfitting). A more holistic approach is required to treat each individual variable separately.
- Given the vastness of the data for the project, working on laptops was causing the data to make R studio crash multiple times. A fix would be purchasing or using a cloud server to upload the data and work of it remotely to save memory space for processing.
- Another major challenge faced was time, as each individual had prior commitments to other subjects as well, everyone didn't pull their weight, and given the limited scope of time, the analysis bore basic results.

7.3 Further Studies and Recommendations

Based on the previous parts of the report, we have seen that the models are working accurately and giving an AUROC score of greater than 0.7. For the purpose of this project, we used 0.65 as a benchmark score of 0.6 by our industry supervisor. The scores we arrived at for each model were only just slightly above the benchmark scores, however, there is scope of huge improvement, given the time and resources.

Basis the challenges faced, and the outcomes received, we have been able to recommend key areas for future work and these have been mentioned below.

1. **Data Pre-Processing:**

   a. **Missing Values:** The missing values in each variable need to be handled in an efficient manner. There are some variables which depict whether the individual owned a house/apartment as collateral and whether the loan was to build property. For the missing values, just putting it off as mean would be incorrect, and hence affect the model. A detailed imputation per variable needs to be carried out.

   b. **Outliers and Transforming:** Variables pertaining to credit sums and amounts need to be transformed to check the skewness and irregularity in data (caused by imputing missing values for a significant percentage of values). This transformation needs to happen per variable and is a time-consuming task depending on the nature of the variables in question.

2. **Feature Selection:** Information gain-based feature selection should be experimented with, on the full imputed dataset. The attribute importance weights from the chi-squared test will be able to give a benchmark for variable selection. It would also help compare what features stand out based on the new feature selection process. This will help to pick the correct predictive features from the data set.

3. **Models:**

    a. **Logistic Regression:** On the basis of the base model (8 features) and best model (12 features) that were obtained in this project, selecting more significant features using information gain would result in better model accuracy and performance. The logistic regression model so far is having the lowest scores based on chi square-based feature selection. This score can improve based on information or probability-based learning methods.

    b. **Random Forest:** The AUROC score of .727 was reached for the random forest having 100 features as predictors and 10 variable splits at each node. We recommend applying this model to the new pre-processed data (as recommended in point 1 of section 5.3 above) and using all 210 variables that were joined in the final dataset as features, and using mtry as 14, 15 and 16 to split variables at each node.

    c. **Gradient Boosting:** The gradient boosting model was created using all features as predictors. It is recommended to feature rank and use limited number of features to get a score. This will reduce overfitting in the data. Also, a better model would be based on the new pre-processed data (as recommended in point 1 of section 5.3 above) and using all 210 variables that were joined in the final dataset as features and using information gain to select features as predictors to give a more realistic model.

Model Recommendation

Based on our present study, it is recommended to build upon the random forest models, as it is engineered to work well with category-based data. As random forest consists of decision trees, information gain or entropy-based measures for feature selection seems ideal based on our findings. The gradient boosting model is generally used with numerical data, and there is significant information loss when converting a large magnitude of continuous and factored data to unique numbers. Even though the chi-squared method of feature selection is apt for categorical data, it would be interesting to see whether information gain-based features are different from the ones we have generated, and what impact it has on the models.

8. References

- Khandani, Kim, & Lo, 2010. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking and Finance,34*(11), 2767-2787.

- Kruppa, Schwarz, Arminger, Ziegler, & Kruppa, J. 2013. Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications, 40*(13), 5125-5131.

- Guegan, D., & Hassani, B., 2018. Regulatory learning: How to supervise machine learning models? An application to credit scoring. *IDEAS Working Paper Series from RePEc, 6*(2), IDEAS Working Paper Series from RePEc, 2018.

- Sumaiya Thaseen, & Aswani Kumar. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *Journal of King Saud University - Computer and Information Sciences, 29*(4), 462-472.

- Hosmer, David W, Lemeshow, Stanley., & Sturdivant, Rodney X. (2013). *Applied logistic regression.* (3rd ed., Wiley series in probability and statistics; 398). Hoboken, New Jersey: Wiley.

- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A., & Gulin, A. (2017). CatBoost: Unbiased boosting with categorical features.

- Tang, Cai, & Ouyang. (2018). Applying a nonparametric random forest algorithm to assess the credit risk of the energy industry in China. *Technological Forecasting & Social Change*.

9. Appendices

7.1 Appendix A – R Codes for Data Pre-Processing

Loading relevant packages.

library(tidyverse)

library(xgboost)

library(magrittr)

library(Hmisc)

library(plotly)

library(GGally)

library(skimr)

library(data.table)

library(caret)

library(DT)

library(viridis)

library(mlr)

library(outliers)

library(lubridate)

library(stringi)

library(pROC)

library(randomForest)

library(xgboost)

Loading Cap function for outlier treatment

```
cap <- function(x){
  quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
  x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
  x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
  x
}
----------------------------
## Loading Datasets
burbal <- read_csv("bureau_balance.csv")
bur <- read_csv("bureau.csv")
ccbal <- read_csv("credit_card_balance.csv")
payments <- read_csv("installments_payments.csv")
pcbal <- read_csv("POS_CASH_balance.csv")
prev <- read_csv("previous_application.csv")
train <- read_csv("application_train.csv")
test <- read_csv("application_test.csv")
----------------------------
## Joining Tables
total_burbal <- burbal %>%
    mutate_if(is.character, funs(factor(.) %>% as.integer)) %>%
    group_by(SK_ID_BUREAU) %>%
    summarise_all(funs(mean, .args = list(na.rm = TRUE)))
rm(burbal); gc()
```

```
total_bur <- bur %>%
    left_join(total_burbal, by = "SK_ID_BUREAU") %>%
    select(-SK_ID_BUREAU) %>%
    mutate_if(is.character, funs(factor(.) %>% as.integer)) %>%
    group_by(SK_ID_CURR) %>%
    summarise_all(funs(mean, .args = list(na.rm = TRUE)))
rm(bur, total_burbal); gc()
total_ccbal <- ccbal %>%
    select(-SK_ID_PREV) %>%
    mutate_if(is.character, funs(factor(.) %>% as.integer)) %>%
    group_by(SK_ID_CURR) %>%
    summarise_all(funs(mean, .args = list(na.rm = TRUE)))
rm(ccbal); gc()
total_payments <- payments %>%
    select(-SK_ID_PREV) %>%
    mutate(PAYMENT_DIFF = AMT_INSTALMENT - AMT_PAYMENT,
        DPD = DAYS_ENTRY_PAYMENT - DAYS_INSTALMENT,
        DBD = DAYS_INSTALMENT - DAYS_ENTRY_PAYMENT,
        DPD = ifelse(DPD > 0, DPD, 0),
        DBD = ifelse(DBD > 0, DBD, 0)) %>%
    group_by(SK_ID_CURR) %>%
    summarise_all(funs(mean, .args = list(na.rm = TRUE)))
rm(payments); gc()
total_pcbal <- pcbal %>%
```

```
    select(-SK_ID_PREV) %>%
    mutate_if(is.character, funs(factor(.) %>% as.integer)) %>%
    group_by(SK_ID_CURR) %>%
    summarise_all(funs(mean, .args = list(na.rm = TRUE)))
rm(pcbal); gc()
total_prev <- prev %>%
    select(-SK_ID_PREV) %>%
    mutate_if(is.character, funs(factor(.) %>% as.integer)) %>%
    mutate(DAYS_FIRST_DRAWING = ifelse(DAYS_FIRST_DRAWING == 365243, NA,
DAYS_FIRST_DRAWING),
        DAYS_FIRST_DUE = ifelse(DAYS_FIRST_DUE == 365243, NA, DAYS_FIRST_DUE),
        DAYS_LAST_DUE_1ST_VERSION = ifelse(DAYS_LAST_DUE_1ST_VERSION == 365243,
NA, DAYS_LAST_DUE_1ST_VERSION),
        DAYS_LAST_DUE = ifelse(DAYS_LAST_DUE == 365243, NA, DAYS_LAST_DUE),
        DAYS_TERMINATION = ifelse(DAYS_TERMINATION == 365243, NA,
DAYS_TERMINATION)) %>%
    group_by(SK_ID_CURR) %>%
    summarise_all(funs(mean, .args = list(na.rm = TRUE)))
rm(prev); gc()
fullset <- train %>%
    left_join(total_bur, by = "SK_ID_CURR") %>%
    left_join(total_ccbal, by = "SK_ID_CURR") %>%
    left_join(total_payments, by = "SK_ID_CURR") %>%
    left_join(total_pcbal, by = "SK_ID_CURR") %>%
```

```
    left_join(total_prev, by = "SK_ID_CURR") %>%
    mutate_if(is.character, funs(factor(.) %>% as.integer)) %>%
    mutate(na = apply(., 1, function(x) sum(is.na(x))))
rm(func, total_bur, total_ccbal, total_payments, total_pcbal, total_prev); gc()
----------------------------
## Dealing with missing values
colSums(is.na(fullset))
----------------------------
## Imputing mean values in the full set + check for special
fullset_impute <- data.frame(
    sapply(
        fullset,
        function(x) ifelse(is.na(x),
                           mean(x, na.rm = TRUE),
                           x)))
colSums(is.na(fullset_impute))
is.special <- function(x){
    if (is.numeric(x)) !is.finite(x) else is.na(x)
}
sum(is.special(fullset_impute))
sum(is.na(fullset))
----------------------------
## Feature selection
options(java.parameters = "-Xmx4096m")
```

```
library(rJava)

library(Fselector)

fin <- fullset_final %>% select(-SK_ID_CURR)

weights<- chi.squared(TARGET~., fin)

----------------------------

## Subsetting Final set based on top 100 by attribute importance weight in chi-squared
feature test.

fullset_final <- fullset_impute %>%    select(SK_ID_CURR,TARGET, NAME_CONTRACT_TYPE.x,
CODE_GENDER, FLAG_OWN_CAR, CNT_CHILDREN, AMT_CREDIT.x, AMT_GOODS_PRICE.x,
NAME_INCOME_TYPE,  NAME_EDUCATION_TYPE,  NAME_HOUSING_TYPE,
REGION_POPULATION_RELATIVE,  DAYS_BIRTH,DAYS_EMPLOYED, DAYS_REGISTRATION,
DAYS_ID_PUBLISH, OWN_CAR_AGE,FLAG_EMP_PHONE,  FLAG_WORK_PHONE, FLAG_PHONE,
OCCUPATION_TYPE,
REGION_RATING_CLIENT,REGION_RATING_CLIENT_W_CITY,HOUR_APPR_PROCESS_START.x,R
EG_CITY_NOT_LIVE_CITY,    REG_CITY_NOT_WORK_CITY, LIVE_CITY_NOT_WORK_CITY,
ORGANIZATION_TYPE, EXT_SOURCE_1,EXT_SOURCE_2,EXT_SOURCE_3, APARTMENTS_AVG,
ELEVATORS_AVG, FLOORSMAX_AVG, FLOORSMIN_AVG,
LIVINGAREA_AVG,APARTMENTS_MODE, ELEVATORS_MODE, FLOORSMAX_MODE,
LIVINGAREA_MODE,APARTMENTS_MEDI,ELEVATORS_MEDI,FLOORSMAX_MEDI,FLOORSMIN_
MEDI,LIVINGAREA_MEDI, TOTALAREA_MODE, DEF_30_CNT_SOCIAL_CIRCLE,
DEF_60_CNT_SOCIAL_CIRCLE, DAYS_LAST_PHONE_CHANGE, FLAG_DOCUMENT_3,
FLAG_DOCUMENT_6, AMT_REQ_CREDIT_BUREAU_YEAR, CREDIT_ACTIVE, DAYS_CREDIT,
DAYS_CREDIT_ENDDATE, DAYS_ENDDATE_FACT, AMT_CREDIT_SUM,  DAYS_CREDIT_UPDATE,
MONTHS_BALANCE.x, STATUS, MONTHS_BALANCE.y,
```

AMT_BALANCE,AMT_DRAWINGS_ATM_CURRENT,AMT_DRAWINGS_CURRENT,

AMT_INST_MIN_REGULARITY,

AMT_RECEIVABLE_PRINCIPAL,AMT_RECIVABLE,AMT_TOTAL_RECEIVABLE,

CNT_DRAWINGS_ATM_CURRENT,CNT_DRAWINGS_CURRENT,CNT_DRAWINGS_POS_CURREN

T,NUM_INSTALMENT_VERSION,DAYS_INSTALMENT, DAYS_ENTRY_PAYMENT,

AMT_INSTALMENT, AMT_PAYMENT, PAYMENT_DIFF,

DBD,MONTHS_BALANCE,CNT_INSTALMENT, CNT_INSTALMENT_FUTURE, AMT_ANNUITY,

AMT_APPLICATION,AMT_DOWN_PAYMENT,HOUR_APPR_PROCESS_START.y,RATE_DOWN_PA

YMENT, NAME_CASH_LOAN_PURPOSE, NAME_CONTRACT_STATUS,

DAYS_DECISION,NAME_PAYMENT_TYPE, CODE_REJECT_REASON, NAME_TYPE_SUITE.y,

NAME_GOODS_CATEGORY, NAME_PRODUCT_TYPE,

CHANNEL_TYPE,CNT_PAYMENT,PRODUCT_COMBINATION,DAYS_FIRST_DRAWING,DAYS_FIRS

T_DUE,

DAYS_LAST_DUE_1ST_VERSION,DAYS_LAST_DUE,DAYS_TERMINATION)

```
## Factoring and binning – Tidying data.

fullset_final$NAME_CONTRACT_TYPE.x <- factor(fullset_final$NAME_CONTRACT_TYPE.x,
levels = c("1", "2"), labels = c("Cash", "Revolving"))

fullset_final$CODE_GENDER <- factor(fullset_final$CODE_GENDER, levels = c("1", "2","3"),
labels = c("M", "F","Not Specified"))

fullset_final$FLAG_OWN_CAR <- factor(fullset_final$FLAG_OWN_CAR, levels = c("1", "2"),
labels = c("Y", "N"))

fullset_final$CNT_CHILDREN <- as.factor(ifelse(fullset_final$CNT_CHILDREN == 0,"0",
                    ifelse(fullset_final$CNT_CHILDREN == 1,"1",">=2")))
```

```
fullset_final$AMT_CREDIT.x <- fullset_final$AMT_CREDIT.x %>% cap()
fullset_final$AMT_CREDIT.x <- cut(fullset_final$AMT_CREDIT.x, breaks = c(0, 100000, 250000,
400000,550000, 700000, 850000, Inf), labels = c("0-100k","101k - 250k","251k - 400k","401k -
550k","551k-700k","700k-850K", "850k+"))
fullset_final$AMT_GOODS_PRICE.x <- fullset_final$AMT_GOODS_PRICE.x %>% cap()
fullset_final$AMT_GOODS_PRICE.x <- cut(fullset_final$AMT_GOODS_PRICE.x, breaks = c(0,
100000, 250000, 400000,550000, 700000, 850000, Inf), labels = c("0-100k","101k -
250k","251k - 400k","401k - 550k","551k-700k","700k-850K", "850k+"))
fullset_final$NAME_INCOME_TYPE <- factor(fullset_final$NAME_INCOME_TYPE, levels =
c("1", "2","3","4","5","6","7","8"), labels = c("Businessman","Commercial-
associate","Maternity-leave","Pensioner","State-
servant","Student","Unemployed","Working"))
fullset_final$NAME_EDUCATION_TYPE <- factor(fullset_final$NAME_EDUCATION_TYPE, levels
= c("1", "2","3","4","5"), labels = c("Academic degree","Higher education","Incomplete
higher","Lower secondary","Secondary/secondary special"))
fullset_final$NAME_HOUSING_TYPE <- factor(fullset_final$NAME_HOUSING_TYPE, levels =
c("1", "2","3","4","5","6"), labels = c("Co-op apartment","House / apartment","Municipal
apartment","Office apartment","Rented apartment","With parents" ))
fullset_final$REGION_POPULATION_RELATIVE <-
fullset_final$REGION_POPULATION_RELATIVE %>% cap()
fullset_final$REGION_POPULATION_RELATIVE <-
cut(fullset_final$REGION_POPULATION_RELATIVE, breaks = c(0, .01, .02, .03,.04,Inf), labels =
c("Lowest","Lower","Moderate","Higher","Highest"))
colnames(fullset_final)[13] <- "Age"
```

```
fullset_final$Age <- (fullset_final$Age/365) * (-1)

fullset_final$Age <- round(fullset_final$Age)

fullset_final$Age <- cut(fullset_final$Age, breaks = c(0, 25, 35, 45,55,Inf), labels = c("<25","26 -
35","36-45","46-55","55+"))

colnames(fullset_final)[14] <- "Years_Employed"

fullset_final$Years_Employed <- round((fullset_final$Years_Employed/365) * (-1))

fullset_final$Years_Employed <- cut(fullset_final$Years_Employed, breaks = c(-Inf, 1, 3, 5,
8,10,Inf), labels = c("<1","1 - 3","3 - 5","5 - 8","8 - 10", "10 +"))

colnames(fullset_final)[15] <- "Years_Registered"

fullset_final$Years_Registered <- round((fullset_final$Years_Registered/365) * (-1))

fullset_final$Years_Registered <- cut(fullset_final$Years_Registered, breaks = c(-Inf, 1, 3, 5,
8,10,Inf), labels = c("<1","1 - 3","3 - 5","5 - 8","8 - 10", "10 +"))

colnames(fullset_final)[16] <- "Years_IDPUB"

fullset_final$Years_IDPUB <- round((fullset_final$Years_IDPUB / 365) * (-1))

fullset_final$Years_IDPUB <- cut(fullset_final$Years_IDPUB, breaks = c(-Inf, 1, 3, 5,
8,10,13,Inf), labels = c("<1","1 - 3","3 - 5","5 - 8","8 - 10", "10 - 13","13 +"))

fullset_final$OWN_CAR_AGE <- fullset_final$OWN_CAR_AGE %>% cap()

fullset_final$OWN_CAR_AGE <- cut(fullset_final$OWN_CAR_AGE, breaks = c(0, 3, 6, 9, 12,Inf),
labels = c("<3","3 - 6","6 - 9","9 - 12","12 +"))

fullset_final$FLAG_EMP_PHONE <- factor(fullset_final$FLAG_EMP_PHONE, levels = c("0", "1"),
labels = c("Y", "N"))

fullset_final$FLAG_WORK_PHONE <- factor(fullset_final$FLAG_WORK_PHONE, levels = c("0",
"1"), labels = c("Y", "N"))
```

```
fullset_final$FLAG_PHONE <- factor(fullset_final$FLAG_PHONE, levels = c("0", "1"), labels =
c("Y", "N"))
fullset_final$OCCUPATION_TYPE <- factor(
    ifelse(fullset_final$OCCUPATION_TYPE %in% c("1","6","7","8","11","12"), "High-Skilled",
        ifelse(fullset_final$OCCUPATION_TYPE %in% c("2","3","4","13","14","15","16","17"),
"Med-Skilled", "Low-Skilled")
))
fullset_final$REGION_RATING_CLIENT <- factor(fullset_final$REGION_RATING_CLIENT , levels
= c("1", "2", "3"), labels = c("L1", "L2", "L3"))
fullset_final$REGION_RATING_CLIENT_W_CITY <-
factor(fullset_final$REGION_RATING_CLIENT_W_CITY , levels = c("1", "2", "3"), labels = c("L1",
"L2", "L3"))
fullset_final$HOUR_APPR_PROCESS_START.x <-
    ifelse(fullset_final$HOUR_APPR_PROCESS_START.x %in%
c("7","8","9","10","11","12"),"Morning",
        ifelse(fullset_final$HOUR_APPR_PROCESS_START.x %in%
c("13","14","15","16"),"Afternoon",
        ifelse(fullset_final$HOUR_APPR_PROCESS_START.x %in%
c("17","18","19","20"),"Evening","Night")))
fullset_final$HOUR_APPR_PROCESS_START.x <-
factor(fullset_final$HOUR_APPR_PROCESS_START.x)
fullset_final$REG_CITY_NOT_LIVE_CITY <- factor(fullset_final$REG_CITY_NOT_LIVE_CITY,
levels = c("0", "1"), labels = c("Y", "N"))
```

```
fullset_final$REG_CITY_NOT_WORK_CITY <- factor(fullset_final$REG_CITY_NOT_WORK_CITY,
levels = c("0", "1"), labels = c("Y", "N"))
fullset_final$LIVE_CITY_NOT_WORK_CITY <- factor(fullset_final$LIVE_CITY_NOT_WORK_CITY,
levels = c("0", "1"), labels = c("Y", "N"))
round(fullset_final$ORGANIZATION_TYPE)
fullset_final$ORGANIZATION_TYPE <-
        ifelse( fullset_final$ORGANIZATION_TYPE %in% c("4", "5", "6", "43"), "Business",
        ifelse( fullset_final$ORGANIZATION_TYPE %in%
c("15","16","17","18","19","20","21","22","23","24","25","26","27"), "Industry",
        ifelse( fullset_final$ORGANIZATION_TYPE %in%
c("46","47","48","49","50","51","52"), "Trade",
        ifelse( fullset_final$ORGANIZATION_TYPE %in% c("53","54","55","56"),
"Transport",
        ifelse( fullset_final$ORGANIZATION_TYPE %in%
c("11","12","35","32","36","42"),"Government",
        ifelse( fullset_final$ORGANIZATION_TYPE %in%
c("39","40","41","44","45","1","33","13","31","38","57"), "Service",
        ifelse( fullset_final$ORGANIZATION_TYPE %in% c("2","10","7","29","9"), "Blue
collar",
        ifelse( fullset_final$ORGANIZATION_TYPEE %in% c("8","37","14"), "Real Estate",
        ifelse( fullset_final$ORGANIZATION_TYPE %in% c("3","28","30"), "Banking",
"Others")))))))))
fullset_final$ORGANIZATION_TYPE <- stri_replace_na(fullset_final$ORGANIZATION_TYPE,
replacement = "Others")
```

```
fullset_final$ORGANIZATION_TYPE <- factor(fullset_final$ORGANIZATION_TYPE)


round(fullset_final$NAME_CASH_LOAN_PURPOSE)
fullset_final$NAME_CASH_LOAN_PURPOSE <-
        ifelse( fullset_final$NAME_CASH_LOAN_PURPOSE %in% c("1", "3", "4", "5"),
"Real Estate",
        ifelse( fullset_final$NAME_CASH_LOAN_PURPOSE %in% c("6", "7", "8"),
"Vehicle",
        ifelse( fullset_final$NAME_CASH_LOAN_PURPOSE %in% c("9"), "Education",
        ifelse( fullset_final$NAME_CASH_LOAN_PURPOSE %in%
c("10","11","12","13","15","21"), "Household",
        ifelse( fullset_final$NAME_CASH_LOAN_PURPOSE %in% c("16"), "Travel",
        ifelse( fullset_final$NAME_CASH_LOAN_PURPOSE %in% c("23","19","22"),
"Personal", "Others"
        ))))))
fullset_final$NAME_CASH_LOAN_PURPOSE <- fullset_final$NAME_CASH_LOAN_PURPOSE
%>% factor()
fullset_final$EXT_SOURCE_1 <- fullset_final$EXT_SOURCE_1 %>% cap()
fullset_final$EXT_SOURCE_1 <- cut(fullset_final$EXT_SOURCE_1, breaks = c(0, 0.25, 0.50,
0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$EXT_SOURCE_2 <- fullset_final$EXT_SOURCE_2 %>% cap()
fullset_final$EXT_SOURCE_2 <- cut(fullset_final$EXT_SOURCE_2, breaks = c(-Inf, 0.25, 0.50,
0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$EXT_SOURCE_3 <- fullset_final$EXT_SOURCE_3 %>% cap()
```

```
fullset_final$EXT_SOURCE_3 <- cut(fullset_final$EXT_SOURCE_3, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$APARTMENTS_AVG <- cut(fullset_final$APARTMENTS_AVG, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$ELEVATORS_AVG <- cut(fullset_final$ELEVATORS_AVG, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("<0.25","0.26 - 0.50","0.51 - 0.75","0.76+"))
fullset_final$FLOORSMAX_AVG <- cut(fullset_final$FLOORSMAX_AVG, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$FLOORSMIN_AVG <- cut(fullset_final$FLOORSMIN_AVG, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$LIVINGAREA_AVG <- cut(fullset_final$LIVINGAREA_AVG, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$APARTMENTS_MODE <- cut(fullset_final$APARTMENTS_MODE, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$ELEVATORS_MODE <- cut(fullset_final$ELEVATORS_MODE, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("<0.25","0.26 - 0.50","0.51 - 0.75","0.76+"))
fullset_final$FLOORSMAX_MODE <- cut(fullset_final$FLOORSMAX_MODE, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$LIVINGAREA_MODE <- cut(fullset_final$LIVINGAREA_MODE, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$APARTMENTS_MEDI <- cut(fullset_final$APARTMENTS_MEDI, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$ELEVATORS_MEDI <- cut(fullset_final$ELEVATORS_MEDI, breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
```

```
fullset_final$FLOORSMAX_MEDI <- cut(fullset_final$FLOORSMAX_MEDI, breaks = c(-Inf, 0.25,
0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$FLOORSMIN_MEDI <- cut(fullset_final$FLOORSMIN_MEDI, breaks = c(-Inf, 0.25,
0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$LIVINGAREA_MEDI <- cut(fullset_final$LIVINGAREA_MEDI, breaks = c(-Inf, 0.25,
0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$TOTALAREA_MODE <- cut(fullset_final$TOTALAREA_MODE, breaks = c(-Inf, 0.25,
0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 - 1"))
fullset_final$DAYS_LAST_PHONE_CHANGE <-
fullset_final$DAYS_LAST_PHONE_CHANGE <-
round((fullset_final$DAYS_LAST_PHONE_CHANGE /365)* (-1))
fullset_final$DAYS_LAST_PHONE_CHANGE <- cut(fullset_final$DAYS_LAST_PHONE_CHANGE,
breaks = c( -Inf, 1, 2, 3,4,Inf), labels = c("<1","1 - 2","2 - 3","3 - 4","4 +"))
fullset_final$DEF_30_CNT_SOCIAL_CIRCLE <- fullset_final$DEF_30_CNT_SOCIAL_CIRCLE %>%
cap()
fullset_final$DEF_30_CNT_SOCIAL_CIRCLE <- cut(fullset_final$DEF_30_CNT_SOCIAL_CIRCLE,
breaks = c(-Inf, 0.25, 0.50, 0.75,1), labels = c("0.00 - 0.25","0.26 - 0.50","0.51 - 0.75","0.76 -
1"))
round(fullset_final$DEF_60_CNT_SOCIAL_CIRCLE)
fullset_final$DEF_60_CNT_SOCIAL_CIRCLE <- cut(fullset_final$DEF_60_CNT_SOCIAL_CIRCLE,
breaks = c(-Inf, 2, 4, 6,Inf), labels = c("<2","2 - 4","4 - 6","6+"))
fullset_final$FLAG_DOCUMENT_3 <- factor(fullset_final$FLAG_DOCUMENT_3, levels = c("0",
"1"), labels = c("Y", "N"))
```

```
fullset_final$FLAG_DOCUMENT_6 <- factor(fullset_final$FLAG_DOCUMENT_6, levels = c("0",
"1"), labels = c("Y", "N"))
fullset_final$AMT_REQ_CREDIT_BUREAU_YEAR <-
fullset_final$AMT_REQ_CREDIT_BUREAU_YEAR %>% cap()
fullset_final$AMT_REQ_CREDIT_BUREAU_YEAR <-
cut(fullset_final$AMT_REQ_CREDIT_BUREAU_YEAR, breaks = c(-Inf,0, 2, 4, 6,Inf), labels =
c("<0","0 - 2","2 - 4","4 - 6", "6+"))
fullset_final$CREDIT_ACTIVE <- factor(round(fullset_final$CREDIT_ACTIVE))
if (levels(fullset_final$CREDIT_ACTIVE) == 2) {
  fullset_final$CREDIT_ACTIVE <- "closed"
} else {
    fullset_final$CREDIT_ACTIVE <- "active"
  }
fullset_final$CREDIT_ACTIVE <- factor(fullset_final$CREDIT_ACTIVE)
#Change days to year
D2Y_scale <- function(x){
  x <- (-x/365)
}
fullset_final$DAYS_CREDIT <- D2Y_scale(fullset_final$DAYS_CREDIT)
fullset_final$DAYS_CREDIT_ENDDATE <- D2Y_scale(fullset_final$DAYS_CREDIT_ENDDATE)
fullset_final$DAYS_CREDIT_UPDATE <- D2Y_scale(fullset_final$DAYS_CREDIT_UPDATE)
fullset_final$DAYS_DECISION <- D2Y_scale(fullset_final$DAYS_DECISION)
fullset_final$DAYS_ENDDATE_FACT <- D2Y_scale(fullset_final$DAYS_ENDDATE_FACT)
fullset_final$DAYS_INSTALMENT <- D2Y_scale(fullset_final$DAYS_INSTALMENT)
```

```
fullset_final$DAYS_ENTRY_PAYMENT <- D2Y_scale(fullset_final$DAYS_ENTRY_PAYMENT)
fullset_final$DAYS_CREDIT <- cut(fullset_final$DAYS_CREDIT, breaks = c(-Inf,2,4,6,Inf), labels =
c("<2","2-4","4-6","6+"))
fullset_final$DAYS_CREDIT_ENDDATE <- cut(fullset_final$DAYS_CREDIT_ENDDATE, breaks =
c(-Inf,1, 2, 3, Inf), labels = c("<1", "1-2", "2-3", "3+"))
fullset_final$DAYS_CREDIT_UPDATE <- cut(fullset_final$DAYS_CREDIT_UPDATE, breaks = c(-
Inf,0.5, 1, 1.5, 2, Inf), labels = c("<0.5", "0.5-1", "1-1.5", "1.5-2", "2+"))
fullset_final$DAYS_DECISION <- cut(fullset_final$DAYS_DECISION, breaks = c(0, 2, 4, 6, 8, Inf),
labels = c("<2", "2-4", "4-6", "6-8", "8+"))
fullset_final$DAYS_ENDDATE_FACT <- cut(fullset_final$DAYS_ENDDATE_FACT, breaks = c(-
Inf,1, 2, 3, 4, 5, Inf), labels = c("<1", "1-2", "2-3", "3-4", "4-5", "5+"))
fullset_final$DAYS_INSTALMENT <- cut(fullset_final$DAYS_INSTALMENT, breaks = c(-Inf, 1, 2,
3, 4, 5, Inf), labels = c("<1", "1-2", "2-3", "3-4", "4-5", "5+"))
fullset_final$DAYS_ENTRY_PAYMENT <- cut(fullset_final$DAYS_ENTRY_PAYMENT, breaks = c(-
Inf, 1, 2, 3, 4, 5, Inf), labels = c("<1", "1-2", "2-3", "3-4", "4-5", "5+"))
fullset_final$AMT_CREDIT_SUM <- cut(fullset_final$AMT_CREDIT_SUM, breaks = c(-Inf,
100000, 200000, 300000, 400000, Inf), labels = c("<100k", "100-200k", "200-300k", "300-
400k", ">400k"))
fullset_final$AMT_BALANCE <- cut(fullset_final$AMT_BALANCE, breaks = c(-Inf, 20000, 40000,
60000, 80000, Inf), labels = c("<20k", "20~ 40k", "40~60k", "60~80k", ">80k"))
fullset_final$AMT_DRAWINGS_ATM_CURRENT <-
cut(fullset_final$AMT_DRAWINGS_ATM_CURRENT, breaks = c(-Inf, 10000, 20000, Inf), labels =
c("<10k", "10~20k", ">20k"))
```

```
fullset_final$AMT_DRAWINGS_CURRENT <- cut(fullset_final$AMT_DRAWINGS_CURRENT,
breaks = c(-Inf, 10000, 20000, Inf), labels = c("<10k", "10~20k", ">20k"))
fullset_final$AMT_INST_MIN_REGULARITY <- cut(fullset_final$AMT_INST_MIN_REGULARITY,
breaks = c(-Inf,  3000, 6000, 9000, Inf), labels = c("<3k", "3~6k", "6~9k", ">9k"))
fullset_final$AMT_RECEIVABLE_PRINCIPAL <- cut(fullset_final$AMT_RECEIVABLE_PRINCIPAL,
breaks = c(-Inf, 40000, 80000, 120000, Inf), labels = c("<40k", "40~80k", "80~120k", ">120k"))
fullset_final$AMT_RECIVABLE <- cut(fullset_final$AMT_RECIVABLE, breaks = c(-Inf, 40000,
80000, Inf), labels = c("<40k", "40~80k", ">120k"))
fullset_final$AMT_TOTAL_RECEIVABLE <- cut(fullset_final$AMT_TOTAL_RECEIVABLE, breaks =
c(-Inf, 40000, 80000, Inf), labels = c("<40k", "40~80k", ">120k"))
fullset_final$AMT_INSTALMENT <- cut(fullset_final$AMT_INSTALMENT, breaks = c(-Inf, 6000,
12000, 18000, 24000, Inf), labels = c("<6k", "6~12k", "12~18k", "18~24k", ">24k")
fullset_final$AMT_PAYMENT <- cut(fullset_final$AMT_PAYMENT, breaks = c(-Inf, 5000, 10000,
15000, 20000, Inf), labels = c("<5k", "5~10k", "10~15k", "15~20k", ">20k")
fullset_final$CNT_DRAWINGS_ATM_CURRENT <-
cut(fullset_final$CNT_DRAWINGS_ATM_CURRENT, breaks = c(-Inf, 0.3, 0.6, 1, Inf), labels =
c("<0.3", "0.3~0.6", "0.6~1", ">1"))
fullset_final$CNT_DRAWINGS_CURRENT <- cut(fullset_final$CNT_DRAWINGS_CURRENT,
breaks = c(-Inf, 1, 2, Inf), labels = c("<1", "1~2", ">2"))
fullset_final$CNT_DRAWINGS_POS_CURRENT <-
cut(fullset_final$CNT_DRAWINGS_POS_CURRENT, breaks = c(-Inf, 1, 2, Inf), labels = c("<1",
"1~2", ">2"))
fullset_final$NUM_INSTALMENT_VERSION <-
factor(round(fullset_final$NUM_INSTALMENT_VERSION))
```

```
fullset_final$MONTHS_BALANCE.x <- cut(fullset_final$MONTHS_BALANCE.x, breaks = c(-Inf, -
24, -12, Inf), labels = c("2+", "1 - 2", "<1"))
fullset_final$MONTHS_BALANCE.y <- cut(fullset_final$MONTHS_BALANCE.y, breaks = c(-Inf, -
24, -12, Inf), labels = c("2+", "1 - 2", "<1"))
fullset_final$NUM_INSTALMENT_VERSION <-
factor(fullset_final$NUM_INSTALMENT_VERSION
fullset_final$PAYMENT_DIFF<-
  cut(fullset_final$PAYMENT_DIFF, breaks = c(-Inf, 50, 150,350,Inf),
      labels = c("<50", "50-150", "150-350", "350+"),
      include.lowest = TRUE)
fullset_final$MONTHS_BALANCE<-
  cut(fullset_final$MONTHS_BALANCE, breaks = c(-Inf, -36, -24,-12,Inf),
      labels = c("> 3 years", "2-3 years", "1-2 years","< 1 year"),
      include.lowest = TRUE)
fullset_final$CNT_INSTALMENT <-
  cut(fullset_final$CNT_INSTALMENT, breaks = c(-Inf,9, 12,18,Inf),
      labels = c("<9", "9-12", "12-18",">18"),
      include.lowest = TRUE)
fullset_final$CNT_INSTALMENT_FUTURE<-
  cut(fullset_final$CNT_INSTALMENT_FUTURE, breaks = c(0,5, 8,11,Inf),
      labels = c("<5", "5-8", "8-11",">11"),
      include.lowest = TRUE)
fullset_final$AMT_ANNUITY<-
  cut(fullset_final$AMT_ANNUITY, breaks = c(0,8000, 13000,17000,20000,Inf),
```

```
     labels = c("0-8k", "8K-13k", "13K-17k","17k-20k","20k+"),
     include.lowest = TRUE
fullset_final$AMT_APPLICATION<-
  cut(fullset_final$AMT_APPLICATION, breaks = c(0, 64082 , 112500 ,145336,183219,Inf),
     labels = c("0-64k", "64K-113k", "113K-146k","146k-184k","184k+"),
     include.lowest = TRUE)
fullset_final$AMT_DOWN_PAYMENT<-
  cut(fullset_final$AMT_DOWN_PAYMENT, breaks = c(-Inf, 4500 ,6000,7500,Inf),
     labels = c("<4.5k", "4.5k - 6k","6k - 7k",">7.5K"),
     include.lowest = TRUE)
fullset_final$HOUR_APPR_PROCESS_START.y <-
      ifelse(fullset_final$HOUR_APPR_PROCESS_START.y %in%
c("7","8","9","10","11","12"),"Morning",
            ifelse(fullset_final$HOUR_APPR_PROCESS_START.y %in%
c("13","14","15","16"),"Afternoon",
            ifelse(fullset_final$HOUR_APPR_PROCESS_START.y %in%
c("17","18","19","20"),"Evening","Night")))
fullset_final$HOUR_APPR_PROCESS_START.y <-
factor(fullset_final$HOUR_APPR_PROCESS_START.y)
fullset_final$STATUS <- fullset_final$STATUS %>% round() %>% factor(levels =
c("1","2","3","4","5","6","7","8"), labels = c("0", "1", "2" ,"3" ,"4", "5", "C", "X"))
fullset_final$RATE_DOWN_PAYMENT<-
  cut(fullset_final$RATE_DOWN_PAYMENT, breaks = c(-Inf,0, 0.0819243 ,0.1077803,Inf),
     labels = c("No Down Payment","0-8%", "8-10%", ">10%") ,
```

```
    include.lowest = TRUE)
fullset_final$DBD<-
  cut(fullset_final$DBD, breaks = c(-Inf,7,14,21,28,Inf),
    labels = c("< 1 Week","1-2 Weeks","2-3 Weeks","3-4 Weeks",">4 Weeks") ,
    include.lowest = TRUE)
fullset_final$NAME_CONTRACT_STATUS <- fullset_final$NAME_CONTRACT_STATUS %>%
round() %>% factor(levels = c("1","2","3","4"), labels = c("Approved"  ,  "Canceled"  ,
"Refused"   ,  "Unused offer"))
fullset_final$NAME_PAYMENT_TYPE <- fullset_final$NAME_PAYMENT_TYPE %>% round()
%>% factor(levels = c("1","2","3","4"), labels = c("Cash through bank","Cashless from
employer","Non-cash from own account","Others"))
fullset_final$CODE_REJECT_REASON <- fullset_final$CODE_REJECT_REASON %>% round() %>%
factor(levels = c("1","2","3","4","5","6","7","8","9"), labels =
c("CLIENT","HC","LIMIT","SCO","SCOFR","SYSTEM","VERIF","XAP", "XNA"   ))
fullset_final$NAME_TYPE_SUITE.y <- round(fullset_final$NAME_TYPE_SUITE.y)
fullset_final$NAME_TYPE_SUITE.y <-
     ifelse(fullset_final$NAME_TYPE_SUITE.y == 1 , "Kids",
     ifelse(fullset_final$NAME_TYPE_SUITE.y == 2, "Family",
     ifelse(fullset_final$NAME_TYPE_SUITE.y == 6, "Partner",
     ifelse(fullset_final$NAME_TYPE_SUITE.y == 7, "Unaccompanied","Others"
         ))))
fullset_final$NAME_TYPE_SUITE.y <- factor(fullset_final$NAME_TYPE_SUITE.y)
fullset_final$NAME_PRODUCT_TYPE <- fullset_final$NAME_PRODUCT_TYPE %>% round()
%>% factor(levels = c("1","2","3"), labels = c("walk-in", "x-sell" , "Others"  ))
```

```
fullset_final$NAME_GOODS_CATEGORY <- round(fullset_final$NAME_GOODS_CATEGORY)
fullset_final$NAME_GOODS_CATEGORY <-
        ifelse(fullset_final$NAME_GOODS_CATEGORY %in% c("2"), "Animals",
         ifelse(fullset_final$NAME_GOODS_CATEGORY %in% c("3","6","8","20","23"),
"Technology",
          ifelse(fullset_final$NAME_GOODS_CATEGORY %in% c("4","26"), "Auto",
           ifelse(fullset_final$NAME_GOODS_CATEGORY %in% c("7","12","13","14","15"), "Real
Estate/ Home",
            ifelse(fullset_final$NAME_GOODS_CATEGORY %in% c("5","17","24","25","10"),
"Personal",
             ifelse(fullset_final$NAME_GOODS_CATEGORY %in% c("11","19","16","18"), "Animals",
"Others"
                    ))))))
fullset_final$NAME_GOODS_CATEGORY <- factor(fullset_final$NAME_GOODS_CATEGORY)
fullset_final$CHANNEL_TYPE <- fullset_final$CHANNEL_TYPE %>% round() %>% factor(levels =
c("1","2","3","4","5","6","7","8"), labels = c("AP+ (Cash loan)","Car dealer","corporate
sales","Contact center","Country-wide","Credit and cash offices","Regional / Local","Stone"  ))
fullset_final$CNT_PAYMENT <-
  cut(fullset_final$CNT_PAYMENT, breaks = c(-Inf,10,15,20,Inf),
      labels = c("< 10","10 ~ 15","15 ~ 20","20+") ,
      include.lowest = TRUE)
fullset_final$PRODUCT_COMBINATION <- round(fullset_final$PRODUCT_COMBINATION)
```

```
fullset_final$PRODUCT_COMBINATION <- factor(fullset_final$PRODUCT_COMBINATION,levels
= c("1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17"), labels =
c("Card Street" ,"Card X-Sell","Cash", "Cash Street: high", "Cash Street: low","Cash Street:
middle","Cash X-Sell: high","Cash X-Sell: low","Cash X-Sell: middle","POS household with
interest","POS household without interest","POS industry with interest","POS industry without
interest","POS mobile with interest","POS mobile without interest","POS other with
interest","POS others without interest"))
fullset_final$DAYS_FIRST_DRAWING <- cut(fullset_final$DAYS_FIRST_DRAWING , breaks = c(-
Inf,-1095,-730,-365,Inf),
    labels = c(">3 years","2-3 years","1-2 years","<1 year") ,
    include.lowest = TRUE)
fullset_final$DAYS_FIRST_DUE <- cut(fullset_final$DAYS_FIRST_DUE , breaks = c(-Inf,-1095,-
730,-365,Inf),
    labels = c(">3 years","2-3 years","1-2 years","<1 year") ,
    include.lowest = TRUE)
fullset_final$DAYS_LAST_DUE_1ST_VERSION <- fullset_final$DAYS_LAST_DUE_1ST_VERSION
%>% cap()
fullset_final$DAYS_LAST_DUE_1ST_VERSION <-
cut(fullset_final$DAYS_LAST_DUE_1ST_VERSION , breaks = c(-Inf,-1095,-730,-365,Inf),
    labels = c(">3 years","2-3 years","1-2 years","<1 year") ,
    include.lowest = TRUE)
fullset_final$DAYS_LAST_DUE <- cut(fullset_final$DAYS_LAST_DUE , breaks = c(-Inf,-1095,-
730,-365,Inf),
    labels = c(">3 years","2-3 years","1-2 years","<1 year") ,
```

```
    include.lowest = TRUE)
fullset_final$DAYS_TERMINATION <- cut(fullset_final$DAYS_TERMINATION , breaks = c(-Inf,-
1095,-730,-365,Inf),
    labels = c(">3 years","2-3 years","1-2 years","<1 year") ,
    include.lowest = TRUE)
--------------------------------
## Checking for inconsistencies and factor distribution
sum(is.na(fullset_final))
sapply( fullset_final[ sapply(fullset_final, is.factor)], table)
--------------------------------
```

7.2 Appendix B – R Codes for Logistic Regression Models

Model 1

model_log = glm(data = fullset_final,TARGET ~ CODE_GENDER + Age + AMT_CREDIT.x +

OCCUPATION_TYPE + NAME_EDUCATION_TYPE + NAME_INCOME_TYPE + FLAG_OWN_CAR +

ORGANIZATION_TYPE,family=binomial(link=logit))

CODE_GENDER + Age + AMT_CREDIT.x + OCCUPATION_TYPE + NAME_EDUCATION_TYPE +

NAME_INCOME_TYPE + FLAG_OWN_CAR + ORGANIZATION_TYPE

summary(model_log)

Model 2

model_log2 = glm(data = fullset_final,TARGET ~ CODE_GENDER + Age + AMT_CREDIT.x +

OCCUPATION_TYPE + NAME_EDUCATION_TYPE + AMT_CREDIT_SUM +

REGION_RATING_CLIENT + EXT_SOURCE_1 + EXT_SOURCE_2 + EXT_SOURCE_3 +

FLAG_OWN_CAR + ORGANIZATION_TYPE + FLAG_OWN_CAR +

OWN_CAR_AGE,family=binomial(link=logit))

summary(model_log2)

model_log2_red <- glm(data = fullset_final, TARGET ~ 1, family=binomial(link=logit))

1-(logLik(model_log2))/(logLik(model_log2_red))

PLOTTING ROC CURVES – Model 1

fit_glm <- glm(TARGET ~ CODE_GENDER + Age + AMT_CREDIT.x + OCCUPATION_TYPE +

NAME_EDUCATION_TYPE + NAME_INCOME_TYPE + FLAG_OWN_CAR + ORGANIZATION_TYPE,

training_data, family=binomial(link="logit"))

glm_link_scores <- predict(fit_glm, test_data, type="link")

glm_prob_scores <- predict(fit_glm, test_data, type="terms")

```
glm_response_scores <- predict(fit_glm, test_data, type="response")

roc_full_resolution <- roc(test_data$TARGET, glm_response_scores)

rounded_scores <- round(glm_response_scores, digits=2)

roc_rounded <- roc(test_data$TARGET, rounded_scores)

plot(roc_full_resolution, print.auc=TRUE, main = "AUC Logistic Regression Model - 8 features")

--------------------------------

## PLOTTING ROC CURVES – Model 2

fit_glm3 <- glm(TARGET ~ CODE_GENDER + Age + AMT_CREDIT.x + OCCUPATION_TYPE +
NAME_EDUCATION_TYPE + AMT_CREDIT_SUM + REGION_RATING_CLIENT + EXT_SOURCE_1 +
EXT_SOURCE_2 + EXT_SOURCE_3  + ORGANIZATION_TYPE + OWN_CAR_AGE , training_data,
family=binomial(link="logit"))

glm_link_scores3 <- predict(fit_glm3, test_data, type="link")

glm_prob_scores3 <- predict(fit_glm3, test_data, type="terms")

glm_response_scores3 <- predict(fit_glm3, test_data, type="response")

roc_full_resolution3 <- roc(test_data$TARGET, glm_response_scores3)

rounded_scores3 <- round(glm_response_scores3, digits=2)

roc_rounded3 <- roc(test_data$TARGET, rounded_scores3)

plot(roc_full_resolution3, print.auc=TRUE, main = "AUC for ROC curve - Logistic Regression
Model - 12 features")
```

7.3 Appendix C – R Codes for Random Forest Models

```
set.seed(1234)

training_index <- sample(nrow(fullset_final)*0.66)

test_index     <- setdiff(seq(2:nrow(fullset_final)), training_index )

training_data  <- fullset_final[training_index, ]

test_data      <- fullset_final[test_index, ]

---------------------------

Model - 1

model2 <- randomForest(TARGET ~ ., data = training_data, ntree = 500, mtry = 10, importance
= TRUE)

model2

predTrain <- predict(model2, training_data, type = "class")

# Checking classification accuracy

table(predTrain, training_data$TARGET)

predValid <- predict(model2, test_data, type = "class")

# Checking classification accuracy

mean(predValid == test_data$TARGET)

table(predValid,test_data$TARGET)

library(randomForest)

importance(model2)

varImpPlot(model2)

varImpPlot(model2)

summary(model2)
```

```
require(pROC)

rf.roc<-roc(training_data$TARGET,model2$votes[,2])

plot(rf.roc,print.acu=TRUE)

auc(rf.roc)

---------------------------
```

Model 2.

```
model3 <- randomForest(TARGET ~
CODE_GENDER+Age+AMT_CREDIT.x+AMT_CREDIT_SUM+REGION_RATING_CLIENT+EXT_SOU
RCE_1+EXT_SOURCE_2+EXT_SOURCE_3+NAME_EDUCATION_TYPE+OCCUPATION_TYPE+ORG
ANIZATION_TYPE+OWN_CAR_AGE, data = training_data, ntree = 500, mtry = 3, importance =
TRUE)

require(pROC)

rf.roc1<-roc(training_data$TARGET,model3$votes[,2])

plot(rf.roc1,print.acu=TRUE)

plot(rf.roc1, print.auc = TRUE, main ="AUC for ROC curve - Random Forest mtry = 3 and 12
features")

auc(rf.roc1)

predTrain <- predict(model3, training_data, type = "class")

# Checking classification accuracy

table(predTrain, training_data$TARGET)

predValid <- predict(model3, test_data, type = "class")

# Checking classification accuracy

mean(predValid == test_data$TARGET)

table(predValid,test_data$TARGET)
```

```
library(randomForest)

importance(model3)

varImpPlot(model3)

summary(model3)

plot(rf.roc, print.auc = TRUE, main ="AUC for ROC curve - Random Forest mtry = 10")
```

7.4 Appendix D – R Codes for Gradient Boosting Models

```
library(caTools)

library(tidyverse)

library(caret)

library(knitr)

library(xgboost)

library(LightGBM)

set.seed(1122)

sample = sample.split(fullset_final$TARGET, SplitRatio = .70)

trn_gbm = subset(fullset_final, sample == TRUE)

test_gbm  = subset(fullset_final, sample == FALSE)

----------------

##Transform to Numeric

trn_gbm2 <- trn_gbm %>%

  select(-TARGET)

feat <- colnames(trn_gbm2)

for (a in feat) {

  if ((class(trn_gbm2[[a]])=="factor") || (class(trn_gbm2[[a]])=="character")) {

    levels <- unique(trn_gbm2[[a]])

    trn_gbm2[[a]] <- as.numeric(factor(trn_gbm2[[a]], levels=levels))

  }

}

trn_gbm2$TARGET = NULL

trn_gbm2$TARGET = as.factor(trn_gbm$TARGET)
```

```
levels(trn_gbm2$TARGET) = make.names(unique(trn_gbm2$TARGET))

test_gbm2 = test_gbm

feat <- colnames(test_gbm2)

for (b in feat) {

  if ((class(test_gbm2[[b]])=="factor") || (class(test_gbm2[[b]])=="character")) {

    levels <- unique(test_gbm2[[b]])

    test_gbm2[[b]] <- as.numeric(factor(test_gbm2[[b]], levels=levels))

  }

}

----------------

##Model

form = TARGET ~ .

fitControl <- trainControl(method="none",number = 5,  classProbs = TRUE, summaryFunction
= twoClassSummary)

xgb.Grid <- expand.grid(nrounds = 100,

                  max_depth = 7,

                  eta = .05,

                  gamma = 0,

                  colsample_bytree = .8,

                  min_child_weight = 1,

                  subsample = 1)


set.seed(132)

gbm_1 = train(form, data = trn_gbm2,
```

```
                  method = "xgbTree",trControl = fitControl,
                  tuneGrid = xgb.Grid,na.action = na.pass,metric="ROC"
                 )
gbm_1
----------------
##Variable Imp
imp = varImp(gbm_1)
var_imp <- data.frame(Variables = row.names(imp[[1]]),
                     Importance = round(imp[[1]]$Overall,2))
----------------
# Create ranks
rank_imp <- var_imp %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance)))) %>%
  head(25)
rank_impfull = rank_imp
ggplot(rank_imp, aes(x = reorder(Variables, Importance),
                    y = Importance)) +
  geom_bar(stat='identity',colour="white", fill = “dodgerblue3”) +
  geom_text(aes(x = Variables, y = 1, label = Rank),
        hjust=0, vjust=.5, size = 4, colour = 'black',
        fontface = 'bold') +
  labs(x = 'Variables', title = 'Relative Variable Importance') +
  coord_flip() +
  theme_bw()
```

```
##prediction

pred = predict(gbm_1,test_gbm2,na.action=na.pass,type="prob")

sol <- data.frame('SK_ID_CURR' = as.integer(test_gbm$SK_ID_CURR), 'TARGET' = pred[,2])

##Preprocessing

full <- bind_rows(trn_gbm,test_gbm)

Target <- trn_gbm$TARGET

Id <- test_gbm$SK_ID_CURR

full[,c('SK_ID_CURR','TARGET')] <- NULL

chr <- full[,sapply(full, is.character)]

num <- full[,sapply(full, is.numeric)]

chr[is.na(chr)] <- "Not Available"

fac <- chr %>%

  lapply(as.factor) %>%

  as_data_frame()

full <- bind_cols(fac, num)

rm(chr, fac, num)

full[is.na(full)] <- 0

num <- train[, sapply(train,is.numeric)]

rm(train, test)

train <- full[1:length(Target),]

test <- full[(length(Target)+1):nrow(full),]
```

```
##Create the Data Partition
```

~~~
set.seed(123)

intrn <- createDataPartition(Target, p=.9, list = F)

tr1 <- train[intrn,]

va1 <- train[-intrn,]

tr1_ta <- Target[intrn]

va1_ta <- Target[-intrn]

----------------

##Create the Model

```{r  message=FALSE, warning=FALSE}

lgb.trn = lgb.Dataset(data.matrix(tr1), label = tr1_ta)

lgb.val= lgb.Dataset(data.matrix(va1), label = va1_ta)

params = list(

  objective = "binary"

  , metric = "auc"

  , min_data_in_leaf = 1

  , min_sum_hessian_in_leaf = 100

  , feature_fraction = 1

  , bagging_fraction = 1

  , bagging_freq = 0

)

model1_gb <- lgb.trn(

    params = params

    , data = lgb.train

    , valids = list(val = lgb.val)
~~~

```
   , learning_rate = 0.05
   , num_leaves = 7
   , num_threads = 2
   , nrounds = 3000
   , early_stopping_rounds = 200
   , eval_freq = 50
 )
----------------
##Importance
gbm1_impr = lgb.importance(model1_gb, percentage = TRUE) %>% head(6)
gbm1_impr %>% kable()
var_imp <- data.frame(Variables = gbm1_impr$Feature,
                      Importance = gbm1_impr$Gain)
# Create a rank variable based on importance
rank_imp <- var_imp %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance)))) %>%
  head(6)
rank_impfull = rank_imp
ggplot(rank_imp, aes(x = reorder(Variables, Importance),
                     y = Importance)) +
  geom_bar(stat='identity',colour="white", fill = fillColor2) +
  geom_text(aes(x = Variables, y = 0.1, label = Rank),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
```

```
 labs(x = 'Variables', title = 'Relative Variable Importance') +
 coord_flip() +
 theme_bw()
----------------
##pred
gb_pred <- predict(model1_gb, data = data.matrix(test), n = model1_gb$best_iter)
result <- data.frame(SK_ID_CURR = Id, TARGET = lgb_pred)
```

7.5 Appendix E – R Codes for Data Exploration through Visualization

```
## Univariate

library(ggthemes)

library(ggplot2)

library(ggpubr)

library(rcartocolor)

library(ggmosaic)

library(cowplot)

library(ggcorrplot)

library(vcd)

library(tidyverse)

cor(fullset_final)

bar_theme <- theme(text  = element_text(size = 12),

                   axis.title.y = element_blank(),

                   axis.text.y = element_blank(),

                   axis.text.x = element_blank(),

                   axis.title.x = element_blank(),

                   legend.position="none")

my_colors <- c('#e4f1e1', '#b4d9cc', '#89c0b6', '#63a6a0', '#448c8a', '#287274', '#0d585f')

p1 <- ggplot(data = fullset_final) +

  geom_bar(aes(x = NAME_CONTRACT_TYPE.x, fill =  TARGET), position = position_dodge2()) +

  bar_theme + labs(fill = "", x ="", title ="Contract type")

p2 <- ggplot(data = fullset_final) +

  geom_bar(aes(x = CODE_GENDER, fill =  TARGET), position = position_dodge2()) +
```

```
  bar_theme + labs(fill = "", x ="", title ="Gender")
p3 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLAG_OWN_CAR, fill =  TARGET), position = position_dodge2()) +
  bar_theme  + labs(fill = "", x ="",  title ="Owned car")
p4 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CNT_CHILDREN, fill =  TARGET), position = position_dodge2()) +
  bar_theme  + labs(fill = "", x ="",  title ="Child number")
p5 <- ggplot(data = fullset_final) +
  geom_bar(aes(x =  AMT_CREDIT.x , fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="",  title ="Amount of credit")
p6 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_GOODS_PRICE.x, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="",  title ="Good price")
p7 <- ggplot(data = fullset_final) +
  geom_bar(aes(x =  NAME_INCOME_TYPE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="",  title ="Income type")
p8 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = NAME_EDUCATION_TYPE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="",  title ="Education type")
p9 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = NAME_HOUSING_TYPE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="",  title ="Housing type")
p10 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = REGION_POPULATION_RELATIVE, fill =  TARGET), position =
```

```
position_dodge2()) +
  bar_theme + labs(fill = "", x ="",  title ="Relative")
p11 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = Age, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", y = "", title ="Age")
p12 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = Years_Employed, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", y = "", title ="Employed years")
p13 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = Years_Registered, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", y = "", title ="Registered years")
p14 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = Years_IDPUB, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", y = "", title ="Years ID")
p15 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = OWN_CAR_AGE, fill =  TARGET), position = position_dodge2()) +
  bar_theme  + labs(fill = "", x ="", y = "", title ="Car age")
p16 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLAG_EMP_PHONE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", y = "", title ="FLAG EMP PHONE")
p17 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLAG_WORK_PHONE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", title = "FLAG WORK PHONE")
p18 <- ggplot(data = fullset_final) +
```

```
  geom_bar(aes(x = FLAG_PHONE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", title = "FLAG PHONE")
p19 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = OCCUPATION_TYPE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", title = "Occupation type")
p20 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = REGION_RATING_CLIENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", title = "RATING CLIENT")
p21 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = REGION_RATING_CLIENT_W_CITY, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x ="", title = "RATING CLIENT with CITY")
p22 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = HOUR_APPR_PROCESS_START.x, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x ="", title = "HOUR APPR PROCES]")
p23 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = REG_CITY_NOT_LIVE_CITY, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x ="", title = "LIVE in CITY or Not")
p24 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = REG_CITY_NOT_WORK_CITY, fill =  TARGET), position = position_dodge2())
+
  bar_theme + labs(fill = "", x ="", title = "WORK in CITY or Not")
p25 <- ggplot(data = fullset_final) +
```

```
  geom_bar(aes(x = LIVE_CITY_NOT_WORK_CITY, fill =  TARGET), position = position_dodge2())
+
  bar_theme + labs(fill = "", x ="", title = "LIVE CITY NOT WORK CITY")
sum1 <- plot_grid(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10,
          p11, p12, p13, p14, p15, p16, p18, p19, p20,
          p21, p22, p23, p24, p25, nrow = 5)
sum1 <- add_sub(sum1, "The first 25 variables across on Target (Good and Bad)")
ggdraw(sum1)
legend("bottomright", legend=c("Line 1", "Line 2"),
       col=c("red", "blue"))
p26 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = ORGANIZATION_TYPE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Organization Type")
p27 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = EXT_SOURCE_1, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Extra source 1")
p28 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = EXT_SOURCE_2, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Extra source 2")
p29 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = EXT_SOURCE_3, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Extra source 3")
p30 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = APARTMENTS_AVG, fill =  TARGET), position = position_dodge2()) +
```

```
  bar_theme + labs(fill = "", x = "", title = "Average Appartment")


p31 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = ELEVATORS_AVG, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Average elevator")
p32 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLOORSMAX_AVG, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Maimun of floor")
p33 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLOORSMIN_AVG, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Minimun of floors" )
p34 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = LIVINGAREA_AVG, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Living area")
p35 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = APARTMENTS_MODE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Appartment mode")
p36 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = ELEVATORS_MODE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Elevator mode")
p37 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLOORSMAX_MODE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Floor max mode")
p38 <- ggplot(data = fullset_final) +
```

```
  geom_bar(aes(x = APARTMENTS_MODE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Floor mode")
p39 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = LIVINGAREA_MODE , fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title= "Living area mode" )
p40 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = APARTMENTS_MEDI, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Appartmnent medium")
p41 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = ELEVATORS_MEDI, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title =  "Elevator medium")
p42 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLOORSMAX_MEDI, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Floor maxium medium")
p43 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLOORSMIN_MEDI, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Floor minimun medium" )
p44 <- ggplot(data = fullset_final) +
  geom_bar(aes(x =  LIVINGAREA_MEDI, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Living area medium")
p45 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = TOTALAREA_MODE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Total area mode")
p46 <- ggplot(data = fullset_final) +
```

```
  geom_bar(aes(x = DEF_30_CNT_SOCIAL_CIRCLE, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "30 days social cirlce")
p47 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DEF_60_CNT_SOCIAL_CIRCLE, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "",  title = "60 days social circle")
p48 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_LAST_PHONE_CHANGE, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Days from last phone number change")
p49 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLAG_DOCUMENT_3, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Flag document 3")
p50 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = FLAG_DOCUMENT_6, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Flag document 6")
sum2 <- plot_grid(p26, p27, p28, p29, p30, p31, p32, p33, p34, p35,
          p36, p37, p38, p39, p40, p41, p42, p43, p44, p45,
          p46, p47, p48, p49, p50, nrow = 5)
sum2 <- add_sub(sum2, "From 26th to 50th variables based on Target (Good and bad)")
ggdraw(sum2)
p51 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_REQ_CREDIT_BUREAU_YEAR, fill =  TARGET), position =
```

```
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Year amount")
p52 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CREDIT_ACTIVE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Active credit")
p53 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_CREDIT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Days of credit")
p54 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_CREDIT_ENDDATE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Days of end credit")
p55 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_ENDDATE_FACT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Actual end date")
p56 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_CREDIT_SUM, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Credit sum")
p57 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_CREDIT_UPDATE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Updated days of credit")
p58 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = MONTHS_BALANCE.x, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Monthly balance x")
p59 <- ggplot(data = fullset_final) +
```

```
  geom_bar(aes(x = STATUS, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Status")
p60 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = MONTHS_BALANCE.y, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Monthly blance y")
p61 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_BALANCE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Amount of balance")
p62 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_DRAWINGS_ATM_CURRENT, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Drawing from current account")
p63 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_DRAWINGS_CURRENT, fill =  TARGET), position = position_dodge2())
+
  bar_theme + labs(fill = "", x = "", title = "Current Drawing")
p64 <- ggplot(data = fullset_final) +
  geom_bar(aes(x =  AMT_INST_MIN_REGULARITY, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Monthly minimum amount ")
p65 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_RECEIVABLE_PRINCIPAL, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Amount of principle receive")
```

```
p66 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_RECIVABLE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Receivable amount")
p67 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_TOTAL_RECEIVABLE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Total receivable amount")
p68 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CNT_DRAWINGS_ATM_CURRENT, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Count drawing from atm")
p69 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CNT_DRAWINGS_CURRENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Count of current drawings")
p70 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CNT_DRAWINGS_POS_CURRENT, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Drawings from POS ")
p71 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = NUM_INSTALMENT_VERSION, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Number of instalment")
p72 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_INSTALMENT, fill =  TARGET), position = position_dodge2()) +
```

```
  bar_theme + labs(fill = "", x = "", title = "Days of instalment")


p73 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_ENTRY_PAYMENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Days of entry payment")
p74 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_INSTALMENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Amount of instalment")
p75 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_PAYMENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Amount of payment")
sum3 <- plot_grid(p51, p52, p53, p54, p55, p56, p57, p58, p59, p60,
          p61, p62, p63, p64, p65, p66, p67, p68, p69, p70,
          p71, p72, p73, p74, p75, nrow = 5)
sum3 <- add_sub(sum3, "The 51st ~ 75th variables based on Target (Good and Bad)")
ggdraw(sum3)
p76 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = PAYMENT_DIFF, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Payment difference")
p77 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DBD, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "DBD")
p78 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = MONTHS_BALANCE, fill =  TARGET), position = position_dodge2()) +
```

```
  bar_theme + labs(fill = "", x = "", title = "Months Balance")
p79 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CNT_INSTALMENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Count of instalment")
p80 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CNT_INSTALMENT_FUTURE, fill =  TARGET), position = position_dodge2())
+
  bar_theme + labs(fill = "", x = "", title = "Count of future instalment")
p81 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_ANNUITY, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Annual amount")
p82 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_APPLICATION, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Application amount")
p83 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = AMT_DOWN_PAYMENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Amount of down payment")
p84 <- ggplot(data = fullset_final) +
  geom_bar(aes(x =  HOUR_APPR_PROCESS_START.y, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Hour of starting application")
p85 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = RATE_DOWN_PAYMENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Rate of down payment")
```

```
p86 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = NAME_CASH_LOAN_PURPOSE, fill =  TARGET), position =
position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Loan purpose")
p87 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = NAME_CONTRACT_STATUS, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Contract status")
p88 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_DECISION, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Decision days")
p89 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = NAME_PAYMENT_TYPE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Payment type")
p90 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CODE_REJECT_REASON, fill =  TARGET), position = position_dodge2())+
  bar_theme + labs(fill = "", x = "", title = "Rejection reason")
p91 <- ggplot(data = fullset_final) +
  geom_bar(aes(x =  NAME_TYPE_SUITE.y , fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Suite name")
p92 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = NAME_GOODS_CATEGORY , fill =  TARGET), position = position_dodge2())
+
  bar_theme + labs(fill = "", x = "", title = "Cood category")
p93 <- ggplot(data = fullset_final) +
```

```
  geom_bar(aes(x = NAME_PRODUCT_TYPE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Product type")
p94 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CHANNEL_TYPE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Channel type")
p95 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = CNT_PAYMENT, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Current payment")
p96 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = PRODUCT_COMBINATION, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Combinaition product")
p97 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_FIRST_DRAWING, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Days of first drawing")
p98 <- ggplot(data = fullset_final) +
  geom_bar(aes(x = DAYS_FIRST_DUE, fill =  TARGET), position = position_dodge2()) +
  bar_theme + labs(fill = "", x = "", title = "Days of first due")
sum4 <- plot_grid(p76, p77, p78, p79, p80, p81, p82, p83, p84, p85,
        p86, p87, p88, p89, p90, p91, p92, p93, p94, p95,
        p96, p97, p98)
sum4 <- add_sub(sum4, "Last 25 variables based on Target (Good and bad)")
ggdraw(sum4)
```

Bivariate

```
b1 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = FLAG_OWN_CAR, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Owned Car", title = "Owned car across the target") +
theme_economist() +
  facet_grid( ~ CODE_GENDER)
b2 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = NAME_TYPE_SUITE.y, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Suite type", title = "Suite type across the target") +
theme_economist() +
  facet_grid( NAME_CONTRACT_TYPE.x~.)
b3 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = NAME_INCOME_TYPE, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Income type", title = "Income type based on target") +
theme_economist() +
  facet_grid( CNT_CHILDREN~.)
b4 <- ggplot(data = fullset_final)+
  geom_bar(aes(x =  Age, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Age", title = "Age ditribution based on type") + theme_economist() +
  facet_grid(OCCUPATION_TYPE ~ .)
b5 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = ORGANIZATION_TYPE, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Orgnization Type", title = "Orgnization type based on Occupation
type") + theme_economist() + facet_grid(OCCUPATION_TYPE ~ .)
```

```
b6 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = APARTMENTS_AVG, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Appartment average ", title = "Average appartment across occupation
type") + theme_economist() + facet_grid(  OCCUPATION_TYPE ~ .)
b7 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = LIVINGAREA_MEDI, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Living area", title = "Living area across number of child") +
theme_economist() +
  facet_grid(CNT_CHILDREN ~ .)
b8 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = DAYS_LAST_PHONE_CHANGE, fill = TARGET), position = position_dodge2())
+
  labs(fill = "Target", x = "Days of changing phone",
       title = "Days of changing phone across credit status") + theme_economist() +
  facet_grid( ~ CREDIT_ACTIVE)
b9 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = DAYS_CREDIT, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Owned Car", title = "Credit days across amount of credit") +
theme_economist() +
  facet_grid(~AMT_CREDIT_SUM )
b10 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = STATUS, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Status", title = "Status across Monthly balance") + theme_economist()
```

```
+
  facet_grid( MONTHS_BALANCE.x~ .)
b11 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = AMT_DRAWINGS_CURRENT, fill = TARGET), position = position_dodge2())
+
  labs(fill = "Target", x = "Drawing amount", title = "Drawing amount across principal reveive")
+ theme_economist() +
  facet_grid( ~ AMT_RECEIVABLE_PRINCIPAL)
b12 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = AMT_TOTAL_RECEIVABLE, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Annual receivable amount",
       title = "Annual receivable amount across occupationtype") + theme_economist() +
  facet_grid(OCCUPATION_TYPE ~ .)
b13 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = AMT_PAYMENT, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Amount of payment", title = "Amount of payment across monthly
balance") + theme_economist() +   facet_grid( MONTHS_BALANCE.x ~.)
b14 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = AMT_ANNUITY, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Anual Amount", title = "Anual amount across Age") +
theme_economist() +   facet_grid(Age ~.)
b15 <- ggplot(data = fullset_final)+
  geom_bar(aes(x = NAME_CONTRACT_STATUS, fill = TARGET), position = position_dodge2()) +
  labs(fill = "Target", x = "Contract Status", title = "Contract status across Occupation type") +
```

```
theme_economist() +   facet_grid(~OCCUPATION_TYPE)


#Multivariate
ggplot(data = fullset_final) +
  geom_mosaic(aes(x = product(CODE_GENDER,  NAME_CONTRACT_TYPE.x), fill = TARGET)) +
  theme_economist() + scale_fill_carto_d(palette = 2) +
  labs(title = "Target distribution across gender and contract type")
ggplot(data = fullset_final) +
  geom_mosaic(aes(x = product(NAME_EDUCATION_TYPE, OCCUPATION_TYPE) , fill =
TARGET))+ theme_economist() + scale_fill_carto_d(palette = 2) +
  labs(title = "Target distribution across education type and occupation type")
ggplot(data = fullset_final) +
  geom_mosaic(aes(x = product(NAME_PAYMENT_TYPE, CREDIT_ACTIVE), fill = TARGET)) +
  theme_economist() + scale_fill_carto_d(palette = 2) +
  labs(title = "Target distribution across payment type and credit status")
ggplot(data = fullset_final) +
  geom_mosaic(aes(x = product(NAME_EDUCATION_TYPE, Age), fill =TARGET))+
  theme_economist() + scale_fill_carto_d(palette = 2) +
  labs(title = "Target distribution across age and education level")
ggplot(data = fullset_final )+
  geom_mosaic(aes(x = product(NAME_CONTRACT_STATUS, DAYS_DECISION), fill = TARGET)) +
  theme_economist() + scale_fill_carto_d(palette = 2) +
  labs(title = "Target distribution across contract status and days for decision")
```

7.6 Appendix F – Data Visualizations

Univariate Visualisation

Variables 1 ~ 25

The first 25 variables across on Target (Good and Bad)

Variables 26 ~ 50

From 26th to 50th variables based on Target (Good and bad)

Variables 51 ~ 75

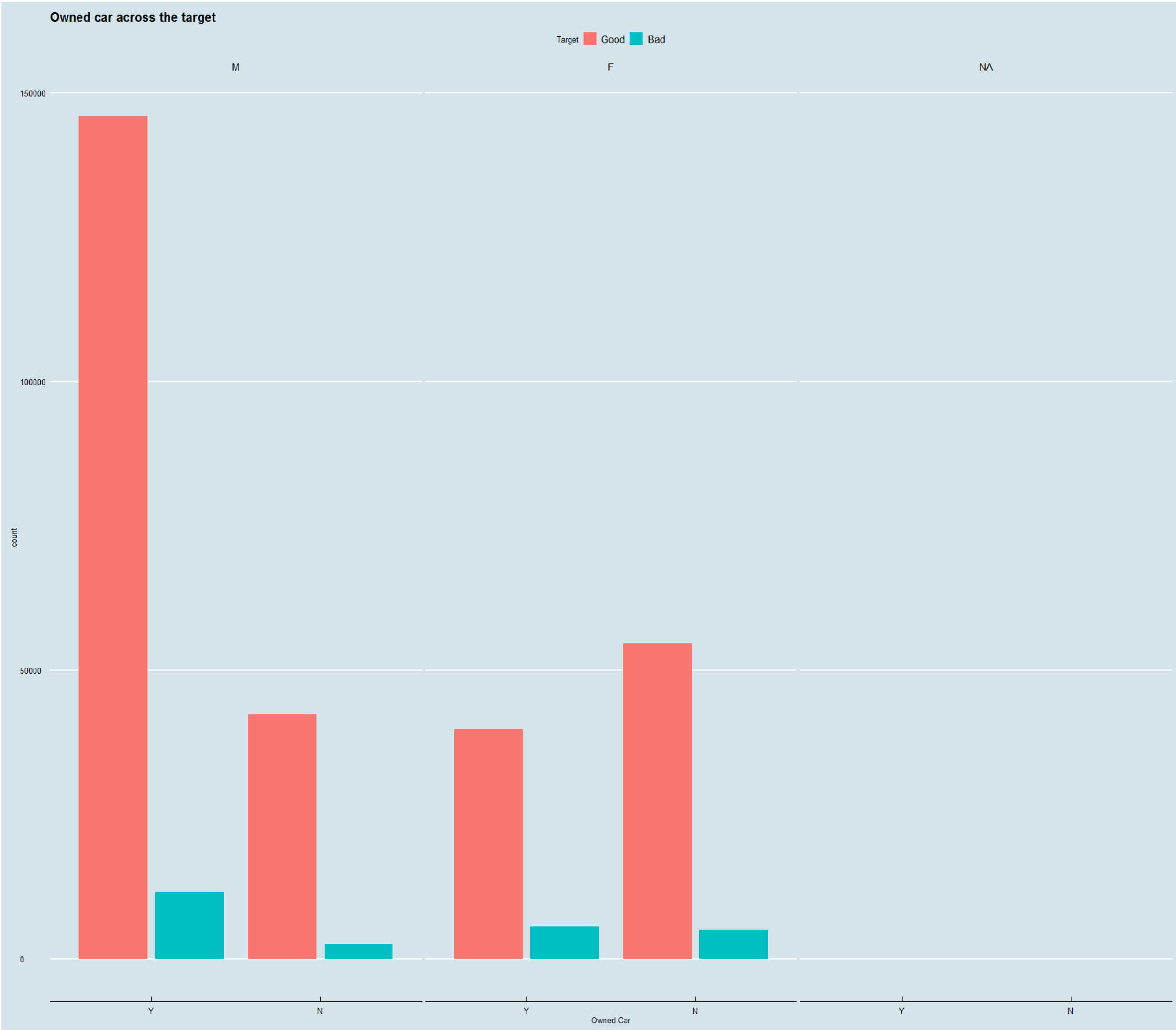
The 51st ~ 75th variables based on Target (Good and Bad)

Variables 76 ~ 98

Last 25 variables based on Target (Good and bad)

Multivariate Visualisation

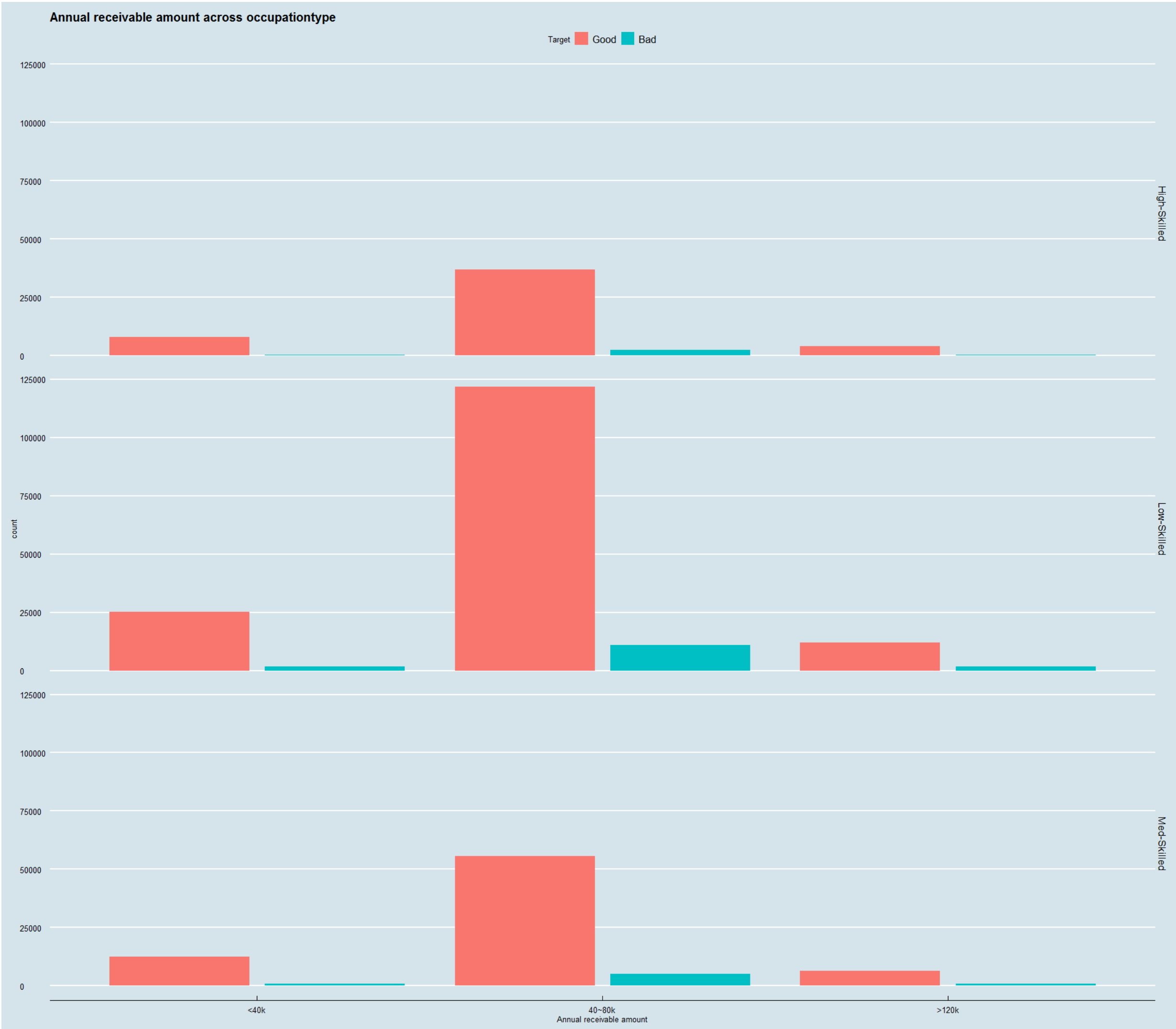
Age of car owned and gender



Age of loan applicant and occupation type

Amount receivable

Amount receivable and occupation type



Amount payment and monthly balance

Contract status and occupation type