An analysis of the possibility of default with data mining techniques

Student ID: 31488404
Word count: 2200
MANG6054– Credit Scoring and Data Mining
University of Southampton

Table of Contents

1. Introduction……. 3

2. Literature review……..……..... 3

2.1 Main argument and problem….……..……..... 3

2.2 Techniques..…….. ….…….…… 3

2.3 Result and discussion...…….....…..…… 3

3. Methodology…….....…..……...…….. 4

3.1 LDA……...……..…..…….... 4

3.2 QDA…….……..…..……..…. 4

3.3 Lasso…….……..…..…….. 4

3.4 Random forest…….…….……... 4

4. Exploratory data analysis ….……..……..... 4

4.1 Data processing.……..……..... 9

5. Modelling and result……..……..……. 9

5.1 Logistic regression result……..……... 9

6. Conclusion and discussion……..…….. 11

Reference……...……..... 12

Appendices.……...……..... 13

1. Introduction

In the past decades, there has been a significant growth in using credit scoring to evaluate the possibility of a loan application being defaulted (Mester, 1997; Chye et al., 2004). The development in technologies, machine learning, data mining, and availability of customer credit data is critically associated with the growth in credit scoring. However, Hooman et al. (2013) points out that there is no single data mining method that always outperforms the others. Each method has its weaknesses and advantages. Ince and Aktan (2009) explore the performance of credit scoring with various algorithms. Additionally, with experimental studies, Ince and Aktan (2009) conclude that classification, regression tree, and neural network have a better performance comparing to traditional models in terms of its predictive accuracy.

The objective of the report is to construct a credit scoring model to predict the possibility of default. The report applies data mining methods and reports a relatively better model. The data mining methods include logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), Lasso regression, random forest, decision tree, neural network, and extreme gradient boosting (XGBoost). The rest of this report is organized as follows. In Section 2, we provide a literature review in response to question 2 of the requirement. In Section 3, we introduce the methodologies for modelling and data cleaning. We perform an exploratory data analysis and justify the data cleaning process in Section 4. Model comparison and result from our final chosen model is discussed in Section 5. We conclude the report in Section 6.

2. Literature review

This section is based on Nguyen (2019). We emphasise the paper in following subsections.

2.1 Main argument and problem

Nguyen (2019) investigates the performance of four different classification algorithms with the confusion matrix and Monte Carlo simulation benchmarks in order to predict the possibility of loan defaulting. Nguyen (2019) performs the algorithms on a dataset with 5,960 observations that contain loan-related and customer-related information. Among these algorithms, XGBoost has the highest accuracy and consistency. Nevertheless, other algorithms also have a high accuracy that is above 85%.

2.2 Techniques

The techniques considered by Nguyen (2019) include the following.

Logistic regression is a classic traditional method in modelling the probability that the response variable belongs to a particular category. It is commonly used for binary classification. Logistic regression is suitable for modelling loan defaulting because the defaulting is usually illustrated in two categories.

Decision trees can be applied to both regression and classification problem. For a classification problem, the method predicts the probability that an observation belongs to a certain class. The class or so called the node or the leave is usually determined with the classification error rate.

Neural network is a modern methodology that an output variable is evaluated through multiple computing layers. This method mimics the learning behaviours of a human brain where the brain sends signals through huge number of neurons and layers.

XGBoost is a similar to random forest but a more advanced method. The method produces an ensemble of weak prediction models and generalises the models by optimising the minimum of a loss function (Nishida, 2017).

2.3 Result and discussion

The main result from Nguyen (2019) is that the XGBoost outperforms other techniques with the highest accuracy, prediction, and recall value. Additionally, Nguyen (2019) concludes that the simulation models for these four algorithms result in a reliable and low rate of variation in distribution.

The main limitation could be the quality of the dataset, because the dataset is retrieved from a website where many important values and variables are missing, specifically some variables have up to 20% of missing values. The missing values create challenges in producing a reliable modelling result and in its predictability of the classification models. Missing values should be critically analysed.

In short, even though the result shows that XGBoost has the highest accuracy, the potential weakness in its predictability should be considered. Nevertheless, the paper provides an exploration in the accuracy of different models. Future improvements include utilising a more reliable data source, evaluating methods for imputing missing values and outliers, and applying more advanced data mining algorithms.

3. Methodology

In this section, we introduce the algorithms of data mining methods. We do include logistic regression, decision tree, neural network and XGBoost because they are discussed in Section 2.2. Additionally, the general approaches in developing a final model include: 1. Exploratory data analysis, 2. Data processing, 3. Modelling comparison and selection and 4. Final model.

3.1 LDA

LDA is a method to find a linear combination of elements that separates two or more classes of objects. The main assumption of the LDA is that the observations are from a multivariate Gaussian distribution (James et al, 2017). The observations have a specific mean vector and a covariance matrix that is common to all k classes.

3.2 QDA

The main difference between LDA and QDA is that QDA assumes each class has its own covariance matrix. Because of this assumption, the x appears to be a quadratic function in the function for Bayes classifier (James et al, 2017). The advantage of QDA is its flexibility and beneficial for a large training set where the variance of the classifier is not a major concern.

3.3 Lasso

Lasso is a regression method that is able to perform variable selection and regularization. The key idea of lasso regression estimates the regression coefficients by minimising

$$RSS + \lambda \sum_{j}^{p} |\beta_j|$$

where $\lambda \sum_{j}^{p} |\beta_j|$ is the penalty term and λ is the tuning parameter (James et al, 2017; Santosa et al, 1986).

3.4 Random forest

Random forest is considered as an extension from the bagging approach where the dataset is randomly split based on selected number of features. Random forest overcomes the problem of substantial reduction in variance from bagging by assigning only a subset of the features to each split (James et al., 2017; Alam and Vuong, 2013).

4. Exploratory data analysis

Appendix 1 contains the variable description and abbreviation. Late payment in this report refers to variable past_due_30, past_due_60, and late_90.

Summary of the basic statistics of the original dataset

Basic Statistics	default	unsecured_lines	age	past_due_30	debt_ratio	monthly_income
Number of Observations	150000.000	150000.000	150000.000	150000.000	150000.000	150000.000
Number of missing values	0.000	0.000	0.000	0.000	0.000	29731.000
Minimum	0.000	0.000	0.000	0.000	0.000	0.000
Maximum	1.000	50708.000	109.000	98	329664.000	3008750.000
Mean	0.067	6.048	52.295	0.421	353.005	6670.221
Standard Deviation	0.250	249.755	14.772	4.193	2037.819	14384.670
Skewness	3.469	97.630	0.189	22.597	95.156	114.037
Kurtosis	10.033	14544.035	-0.495	522.352	13733.648	19503.570

Figure 1. A table of basic statistics

Summary of the basic statistics of the original dataset (continue)

Basic Statistics	credit_lines	late_90	real_estate	past_due_60	num_dependents
Number of Observations	150000.000	150000.000	150000.000	150000.000	150000.000
Number of missing values	0.000	0.000	0.000	0.000	3924.000
Minimum	0.000	0.000	0.000	0.000	0.000
Maximum	58.000	98.000	54.000	98.000	20.000
Mean	8.453	0.266	1.018	0.240	0.757
Standard Deviation	5.146	4.169	1.130	4.155	1.115
Skewness	1.215	23.087	3.482	23.331	1.588
Kurtosis	3.091	537.714	60.474	545.657	3.001

Figure 2. A table of basic statistics (continue)

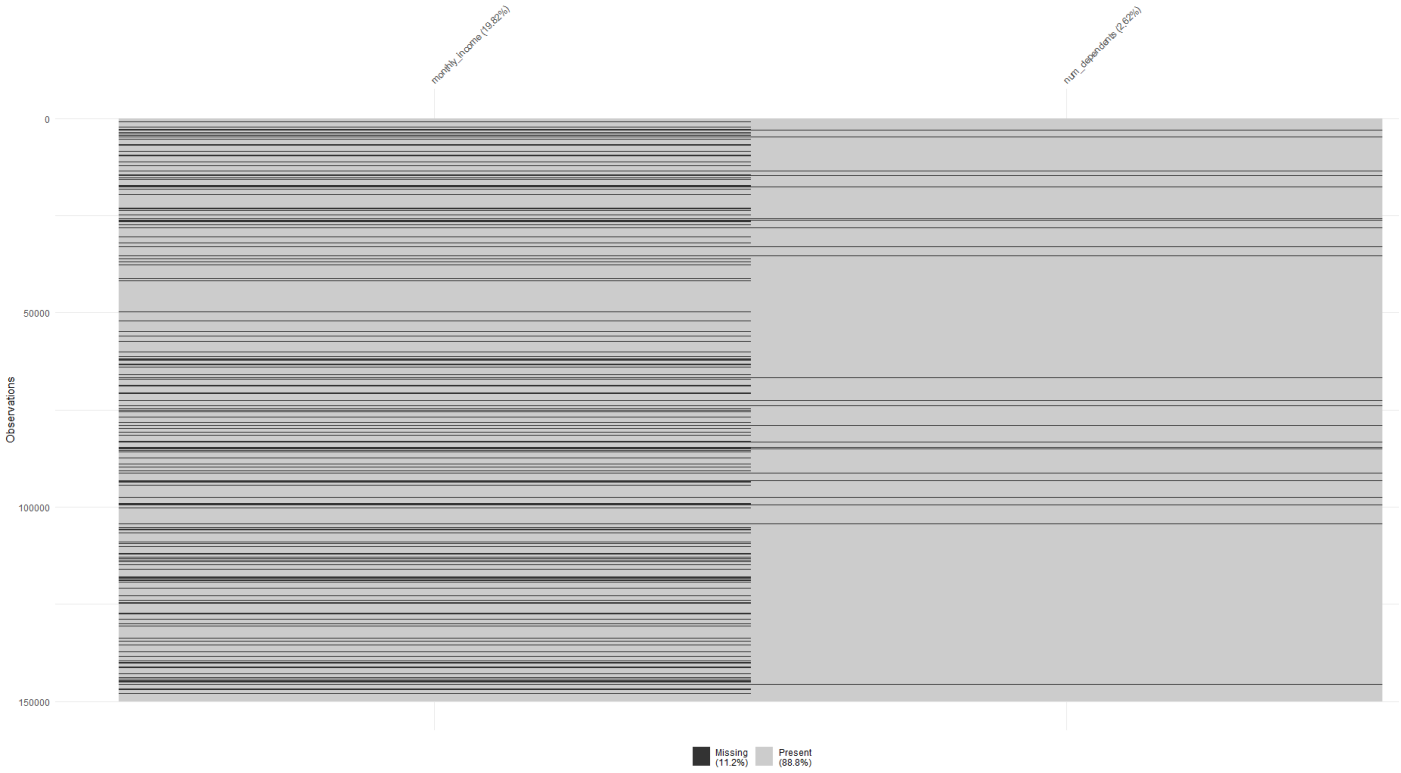


Figure 3. Missing values pattern

According to figure 1 to 3, the dataset contains 150,000 observations for 11 different variables and 29731 (19.92%) and 3924 (2.62%) missing values for monthly_income and num_dependents respectively. According to average, minimum, and maximum, all variables except age and default potentially contain odd values based on extremely high maximum. For all the variables except unsecured_lines, debt_ratio, and montly_income, the data is relatively close to the mean according to relatively low standard deviations. The positive skewness (greater than 0.5) indicates that the observations are right skewed but not approximately symmetric. This observation is supported by figure 4 to 6. Based on kurtosis, all variables potentially have outliers except default, age, credit_lines, and num_dependents.

The boxplots in figure 7 and 8 are created based on different datasets, from left to right is dataset with, all data, data excluding missing num_dependents, and data excluding missing monthly_income. The observations with missing num_dependents also have missing value for montly_income.

The significant amount of observations outside the box does not necessary indicate outliers. The distribution of the variables are likely a Pareto distribution. Therefore, a detail evaluation of the observations is required when performing outlier treatment. Additionally, the boxplots are similar between different datasets which suggest that there is no significant association between missing values and other variables. Hence, we impute monthly_income and num_dependents as they are missing at random.

According to figure 9, there is no significant correlation between variables except relationship between late payment. These three variables represents the late payment which does not raise a concern in modelling.

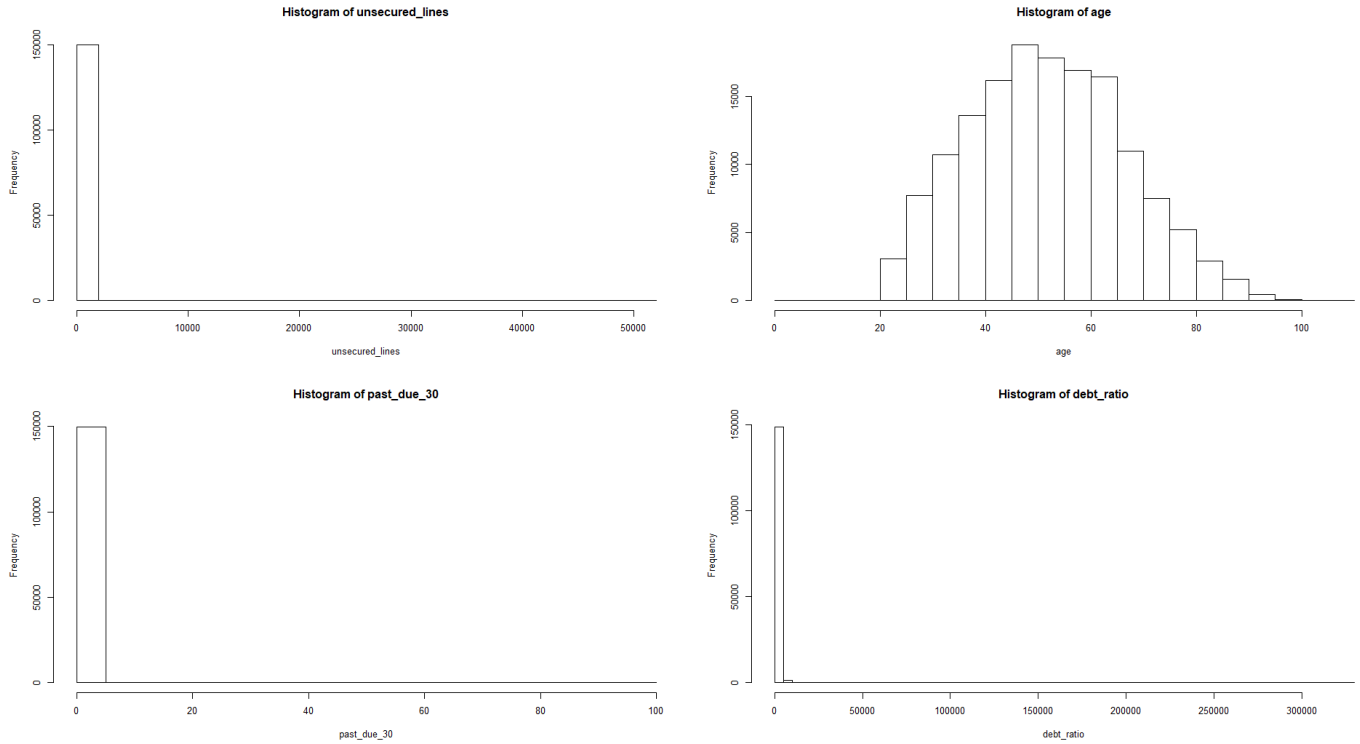


Figure 4. Histogram of variables #1

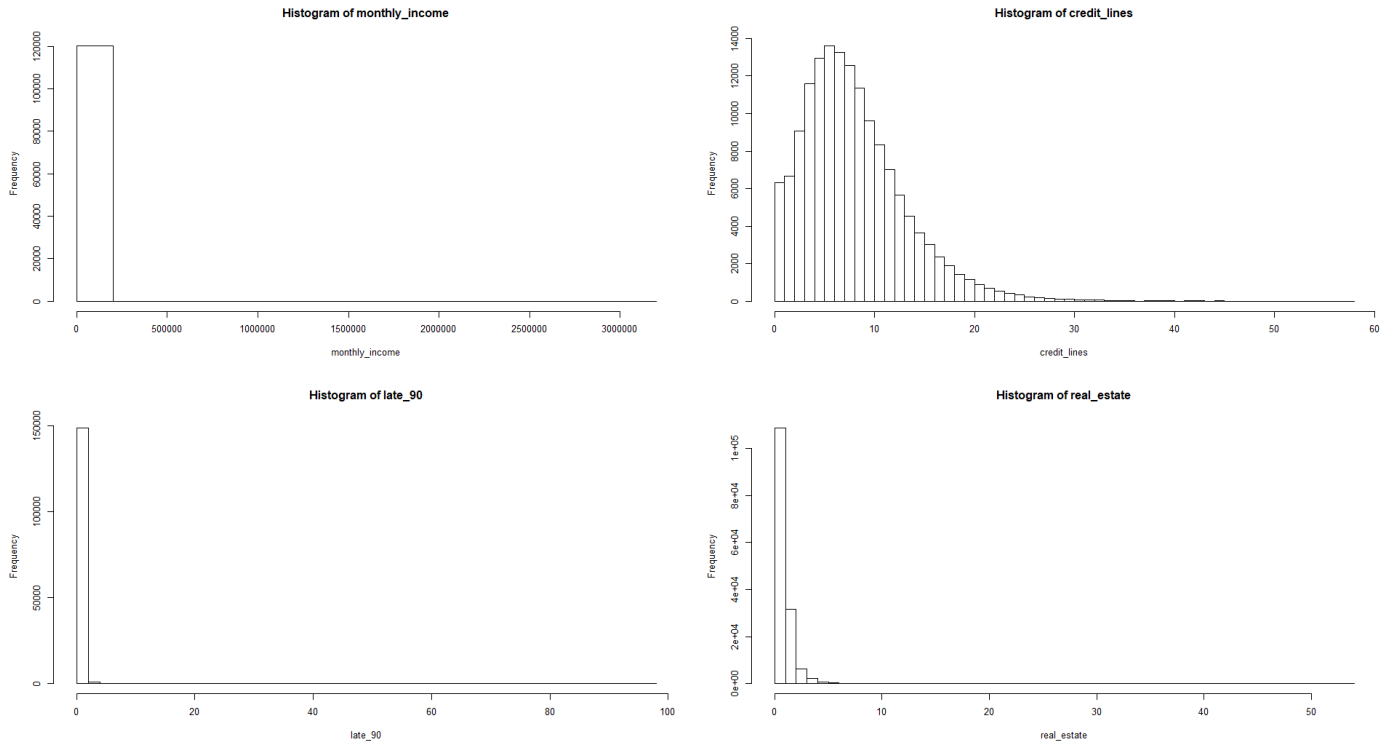


Figure 5. Histogram of variables #2

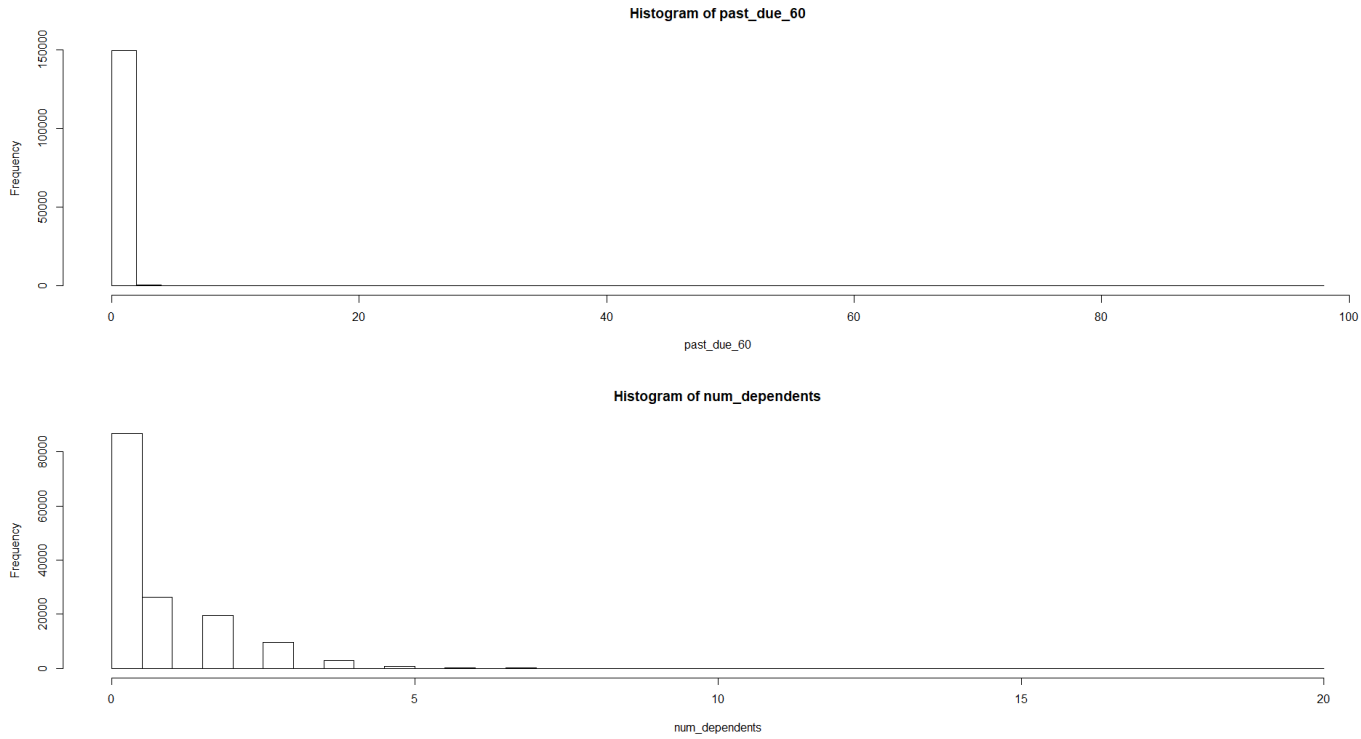


Figure 6. Histogram of variables #3

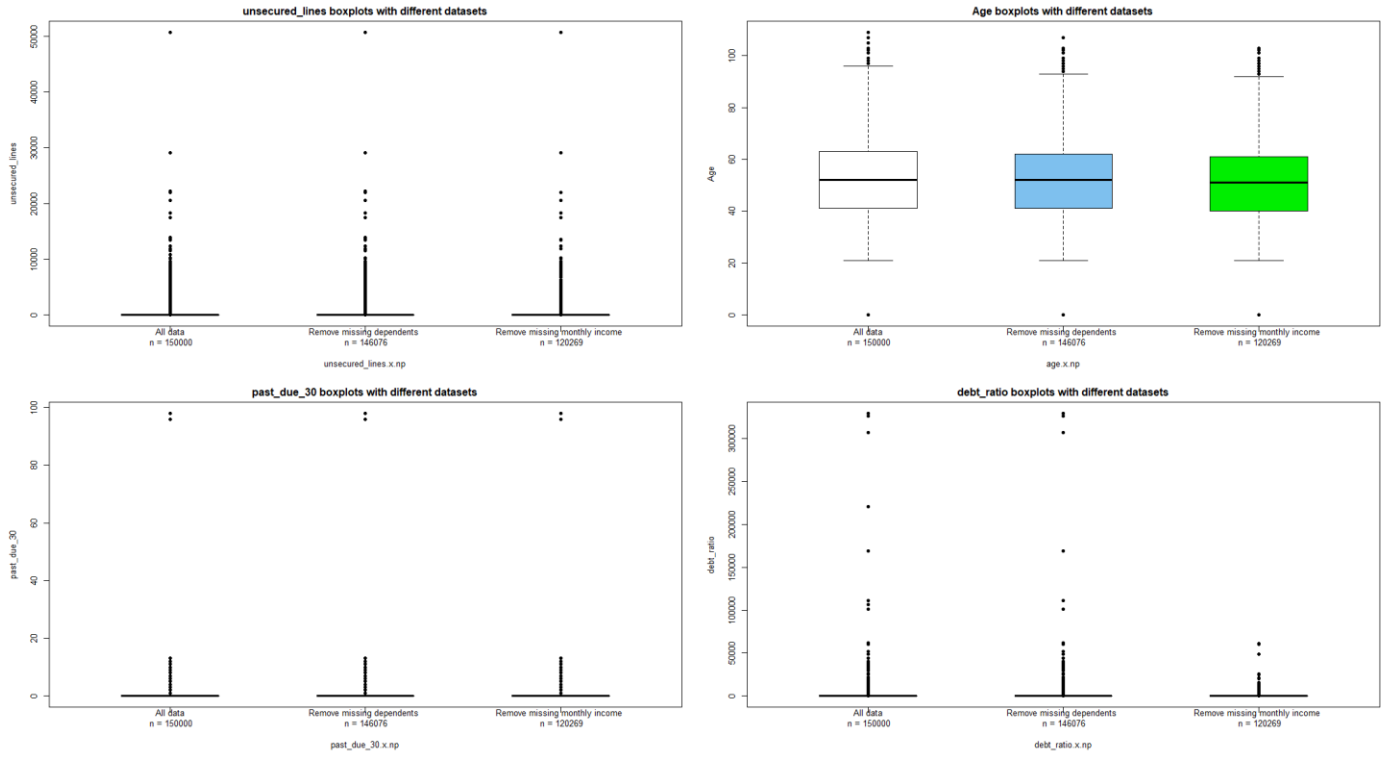


Figure 7. Boxplots of variables #1

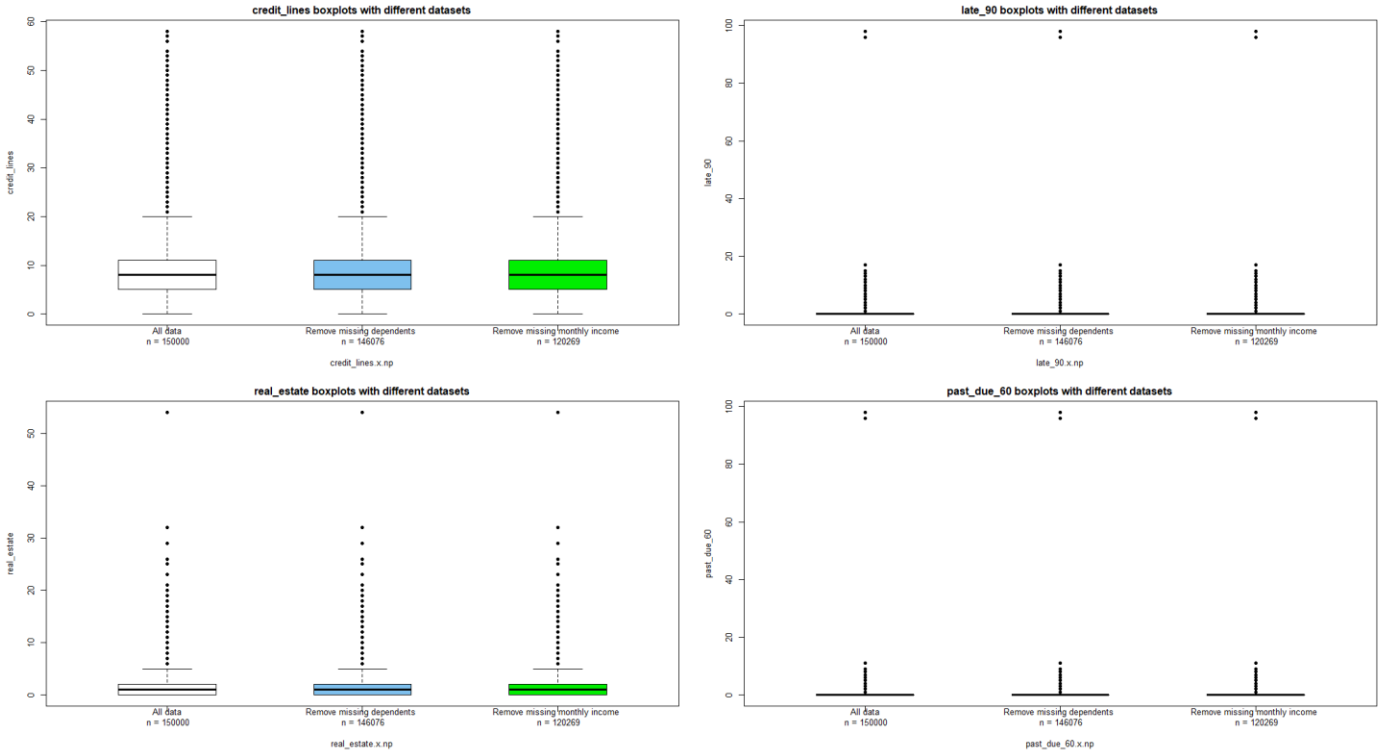


Figure 8. Boxplots of variables #2

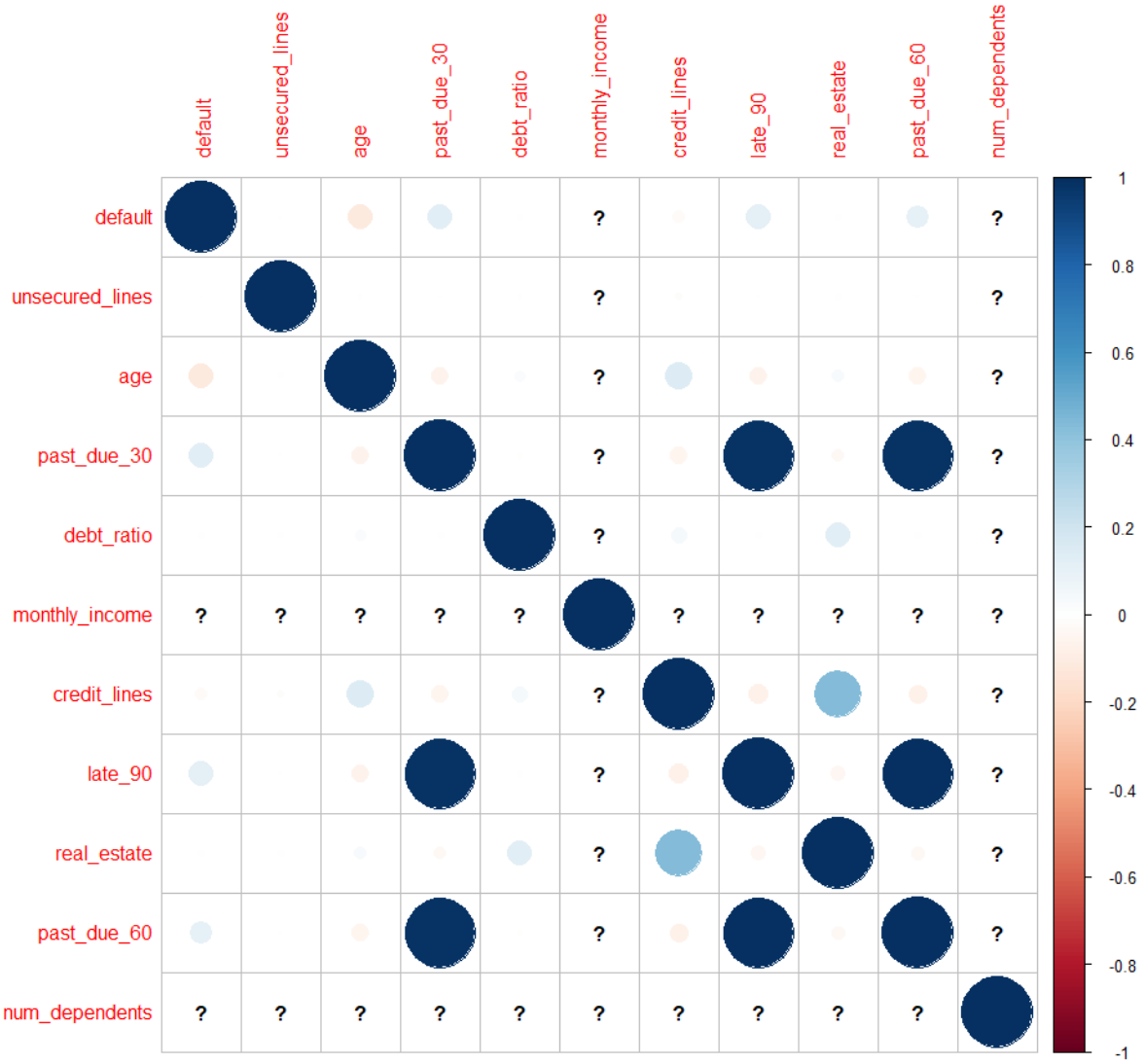


Figure 9. Correlation plot between variables

4.1. Data processing

In this section, we justify outlier treatments for each variable. Truncation based on IQR refers to truncate to $M \pm 3s, M = meidan\ and\ s\ = \frac{IQR}{2\times0.6745}$ (Bijak, 2020).

For late payment, there are 267 observations that share the same values, specifically 96 and 98. These values are considered as some code which are missing not at random. Therefore, we remove observations with late payment equals to 96 and 98. One important evaluation for removing outliers is the amount/percentage of outliers exist. If there are many outliers in a variable, we do not remove all the outliers because there may be some trends in these observations; therefore, removing them results in a potential risk of loss information.

Credit_lines, real_estate, and age do not contain unreasonable values; however, some values are extremely low or high. Therefore, we perform truncation based on IQR. The measurement of the unsecured_line is percentage where the value is normally not an integer. We assume this value can exceed 1. We notice that the unsecured_line above 15 are all integer values. As a result, we believe that this is due to human or mechanical error. We consider these as invalid outliers and therefore replace with missing values.

Applying truncation based on IQR to debt_ratio would remove 20.871% of the data which is significantly large. Therefore, we decide to remove debt_ratio above 97.5th percentile. This would remove 2.5% of the data but does not influence much on proportion of default (see appendix 5). Similar reasoning is applied to monthly_income and num_dependents.

The clean dataset contains 138,999 observations where 7.334% of observations are removed. We impute missing value in the clean dataset with predictive mean matching and five imputations. Predictive mean matching is suitable for quantitative variables that are not normally distributed and the method is robust and produces low biased estimates (Allison, 2015; Buuren, 2018). The final dataset is split into training (75%) and test data (25%).

The kurtosis and correlation significantly decrease after data wrangling (see appendix 3,4,6).

5. Modelling and result

Model performance and confusion matrix

Model	Logistic		LDA		QDA		Lasso	
	Not default	Default	Not default	Default	Not default	Default	Not default	Default
Not default	32206	1876	31748	1541	30857	1279	32324	2027
Default	268	400	726	735	1617	997	150	249
Accuracy	0.93830		0.93476		0.91666		0.93735	
RMSE	0.24839		1.00912		1.05014		0.25030	
AUC	0.85321		0.85901		0.84261		0.55239	
Model	Random forest		Decision Tree		Neural network		Extreme gradient boosting	
	Not default	Default	Not default	Default	Not default	Default	Not default	Default
Not default	32253	1952	32474	2276	32144	1817	32185	1856
Default	221	324	0	0	330	459	289	420
Accuracy	0.93747		0.93450		0.93822		0.93827	
RMSE	0.25006		0.25592		1.07115		1.07327	
AUC	0.56777		0.50000		0.59575		0.58782	

Figure 10. A table of model performance and confusion matrix

Figure 10. shows that logistic regression has the highest accuracy and lowest RMSE and LDA has the highest AUC. However, the difference of AUC between LDA and logistic regression is very small. Based on these indications and interpretability, we choose logistic regression as our final model.

5.1 Logistic regression result

Based on figure 11, the negative estimate of the age represents that an increase in age is associated with a decrease in the probability of having a default. The odds ratio is important which represents the ratio of the odds that an event will occur given the present of the predictor, compared to the odds of the event occurring in the absence of that predictor (Kassambara, 2018). For instance, one unit increase in the age will decrease the odds of having default by exp(-0.117) = 0.983 times.

The deviance provides information on how including the variable affect the model. For instance, adding unsecured_lines alone reduce the deviance significantly by 50554 – 43706 = 6848. In addition, we perform a dominance analysis based on the McFadden index, R^2M to evaluate the relative importance of predictors in our logistic regression(Azen and Traxel, 2009; Soares, 2020). Despite information value shows some variables are weak predictors, we include them in our model for interpretability. Additionally, logistic regression results in significant p-values for these weak predictors. Based on deviance, figure 12. and 13., these measurements result in the same top 5 important variables where the most important variable is unsecured_lines.

Result of logistic regression

	estimate	standard error	deviance	residual deviance	VIF	information value
Intercept	-3.114*	0.067	NA	50554		
unsecured_lines	1.748*	0.039	6848	43706	1.247	1.145
age	-0.017*	0.001	261	43444	1.155	0.256
past_due_30	0.464*	0.014	2482	40962	1.147	0.656
debt_ratio	-0.000331*	0	17	40945	1.283	0.079
monthly_income	-0.0000758*	0	143	40801	1.580	0.068
credit_lines	0.035*	0.004	56	40744	1.542	0.071
late_90	0.674*	0.02	1801	38943	1.132	0.850
real_estate	0.107*	0.018	36	38907	1.448	0.041
past_due_60	0.594*	0.028	475	38431	1.114	0.556
num_dependents	0.074*	0.013	31	38399	1.094	0.038

*p-value < 0.05

Figure 11. A table of logistic regression result

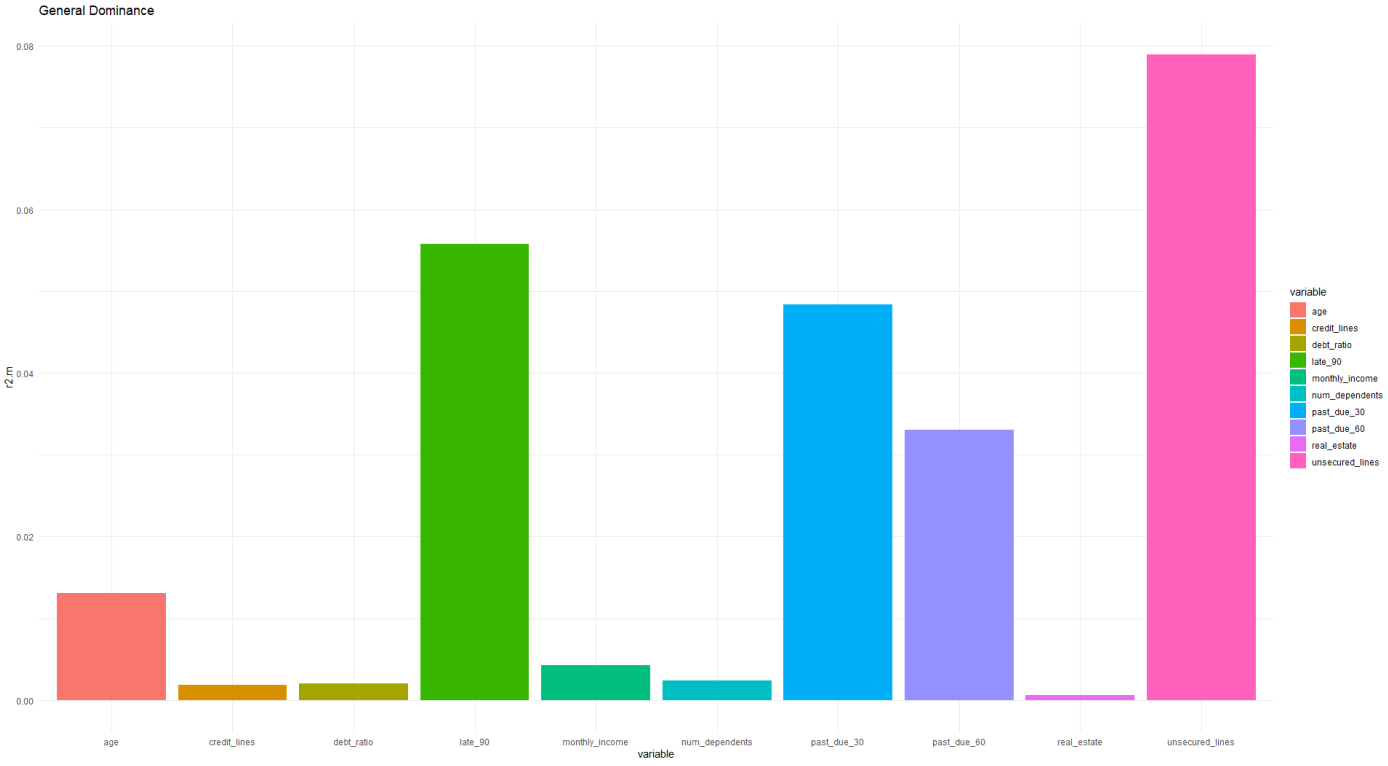


Figure 12. Dominance of variables

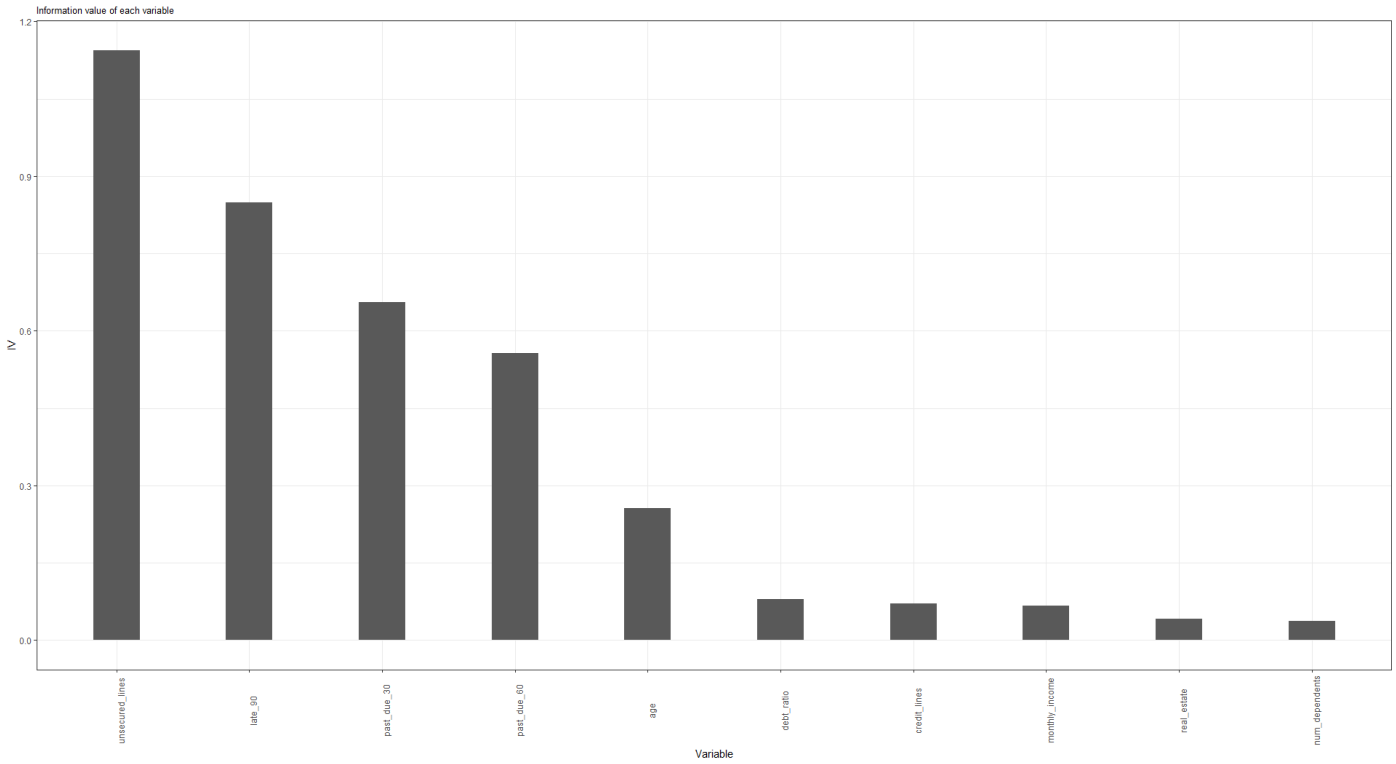


Figure 13. Information value of variables

6. Conclusion and discussion

The goal of credit scoring models is to predict the probability of a person will default on a given financial obligation or not. The logistic model developed in this report allows a bank or a financial institution to minimize the risk of loss by setting criteria regarding which customers should receive loan approvals. Based on the given variables and the logistic model, we are able to uncover potential relationship between a customer's characteristics and his/her possibility of defaulting. A bank can definitely benefit from this information to avoid any defaulting that results in a loss to the bank. However, this relies on the accuracy of a model. An inaccurate model could mislead a decision maker in deciding which customers receive loan approvals; therefore, this results in an enormous loss to the bank.

Broadly speaking, when a customer has history of past due and higher percentage of unsecure_lines, there is a higher chance that he/she would default. Conversely, the possibility of default decreases as the debt_ratio, monthly_income, and age increase. However, we should be aware that these variables have a smaller impact.

Reference

Alam, M. and Vuong, S., 2013. Random Forest Classification for Detecting Android Malware. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*,.

Allison, P., 2015. *Imputation By Predictive Mean Matching: Promise & Peril | Statistical Horizons*. [online] Statisticalhorizons.com. Available at: <https://statisticalhorizons.com/predictive-mean-matching#:~:text=Predictive%20mean%20matching%20(PMM)%20is,that%20are%20not%20normally%20distributed.&text=That's%20because%20the%20imputed%20values,from%20individuals%20with%20real%20data.> [Accessed 4 June 2020].

Azen, R. and Traxel, N., 2009. Using Dominance Analysis to Determine Predictor Importance in Logistic Regression. *Journal of Educational and Behavioral Statistics*, 34(3), pp.319-347.

Bijak, K 2020, MANG6054: Credit Scoring and Data Mining, lecture notes, University of Southampton, delivered March 2020.

Buuren, S., 2018. *Flexible Imputation Of Missing Data*. Vancouver: Chapman & Hall/CRC.

Hooman, A., Marthandan, G. and Karamizadeh, S., 2013. Statistical and Data Mining Methods in Credit Scoring. *SSRN Electronic Journal*,.

Ince, H. and Aktan, B., 2009. A COMPARISON OF DATA MINING TECHNIQUES FOR CREDIT SCORING IN BANKING: A MANAGERIAL PERSPECTIVE. *Journal of Business Economics and Management*, 10(3), pp.233-240.

James, G., Witten, D., Hastie, T. and Tibshirani, R., 2017. *An Introduction To Statistical Learning*. New York: Springer.

Koh Hian Chye, Tan Wei Chin and Goh Chwee Peng., 2004. 'Credit Scoring Using Data Mining Techniques', *Singapore Management Review*, 26(2), pp. 25–47. Available at: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=13577615&site=eds-live (Accessed: 3 June 2020).

Mester, L., 1997. What's the point of credit scoring?. *Business Review*, pp.3-16.

Nguyen, Cuong, "The credit risk evaluation models: an application of data mining techniques" (2019). SAIS 2019 Proceedings. 36 [online]. Available at: https://aisel.aisnet.org/sais2019/36 (Accessed: 1 June 2020)

Nishida, K., 2017. *Introduction To Extreme Gradient Boosting In Exploratory*. [online] Medium. Available at: <https://blog.exploratory.io/introduction-to-extreme-gradient-boosting-in-exploratory-7bbec554ac7> [Accessed 1 June 2020].

Santosa, F. & Symes, W.W., 1986. Linear Inversion of Band-Limited Reflection Seismograms. SIAM Journal on Scientific and Statistical Computing, 7(4), pp.1307–1330.

Soares, F., 2020. *Exploring Predictors' Importance In Binomial Logistic Regressions*. [online] Cran.r-project.org. Available at: <https://cran.r-project.org/web/packages/dominanceanalysis/vignettes/da-logistic-regression.html> [Accessed 1 June 2020].

Kassambara, A., 2018. *Logistic Regression Essentials In R - Articles - STHDA*. [online] Sthda.com. Available at: <http://www.sthda.com/english/articles/36-classification-methods-essentials/151-logistic-regression-essentials-in-r/> [Accessed 2 June 2020].

Appendices

Appendix 1. Variable description

Variable Name	Abbreviation	Description	Type
SeriousDlqin2yrs	default	Person experienced 90 days past due delinquency or worse	Y/N
RevolvingUtilizationOfUnsecuredLines	unsecured_lines	Total balance on credit cards and personal lines of credit expt r. estate and no installment debt ((car loans / (sum of credit limits)	percentage
age	age	Age of borrower in years	integer
NumberOfTime30-59DaysPastDueNotWorse	past_due_30	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.	integer
DebtRatio	debt_ratio	Monthly debt payments, alimony,living costs divided by monthy gross income	percentage
MonthlyIncome	monthly_income	Monthly income	real
NumberOfOpenCreditLinesAndLoans	credit_lines	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
NumberOfTimes90DaysLate	late_90	Number of times borrower has been 90 days or more past due.	integer
NumberRealEstateLoansOrLines	real_estate	Number of mortgage and real estate loans including home equity lines of credit	integer
NumberOfTime60-89DaysPastDueNotWorse	past_due_60	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	integer
NumberOfDependents	num_dependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

Appendix 2. Variables and truncation rule

Variable	Truncation rule
unsecured_lines	Replace value above 15 as missing value
credit_lines, real_estate, age	Truncation based on IQR in step 3a
debt_ratio, monthly_income, num_dependents	Remove observations above 97.5th quantile
late payment	Remove value 96 and 98

Appendix 3. Summary of the basic statistics of the final dataset

Basic Statistics	default	unsecured_lines	age	past_due_30	debt_ratio	monthly_income
Number of Observations	138999	138999	138999	138999	138999	138999
Number of missing values	0	0	0	0	0	29731
Minimum	0	0	0	0	0	0
Maximum	1	14.541	99	13	3491	18000
Mean	0.066	0.323	52.19	0.24	217.437	5142.176
SE Mean	0.248	0.378	14.991	0.686	628.96	3712.586
Skewness	3.507	3.152	0.2	4.239	3.179	0.761
Kurtosis	10.296	67.101	-0.543	25.451	9.541	0.378

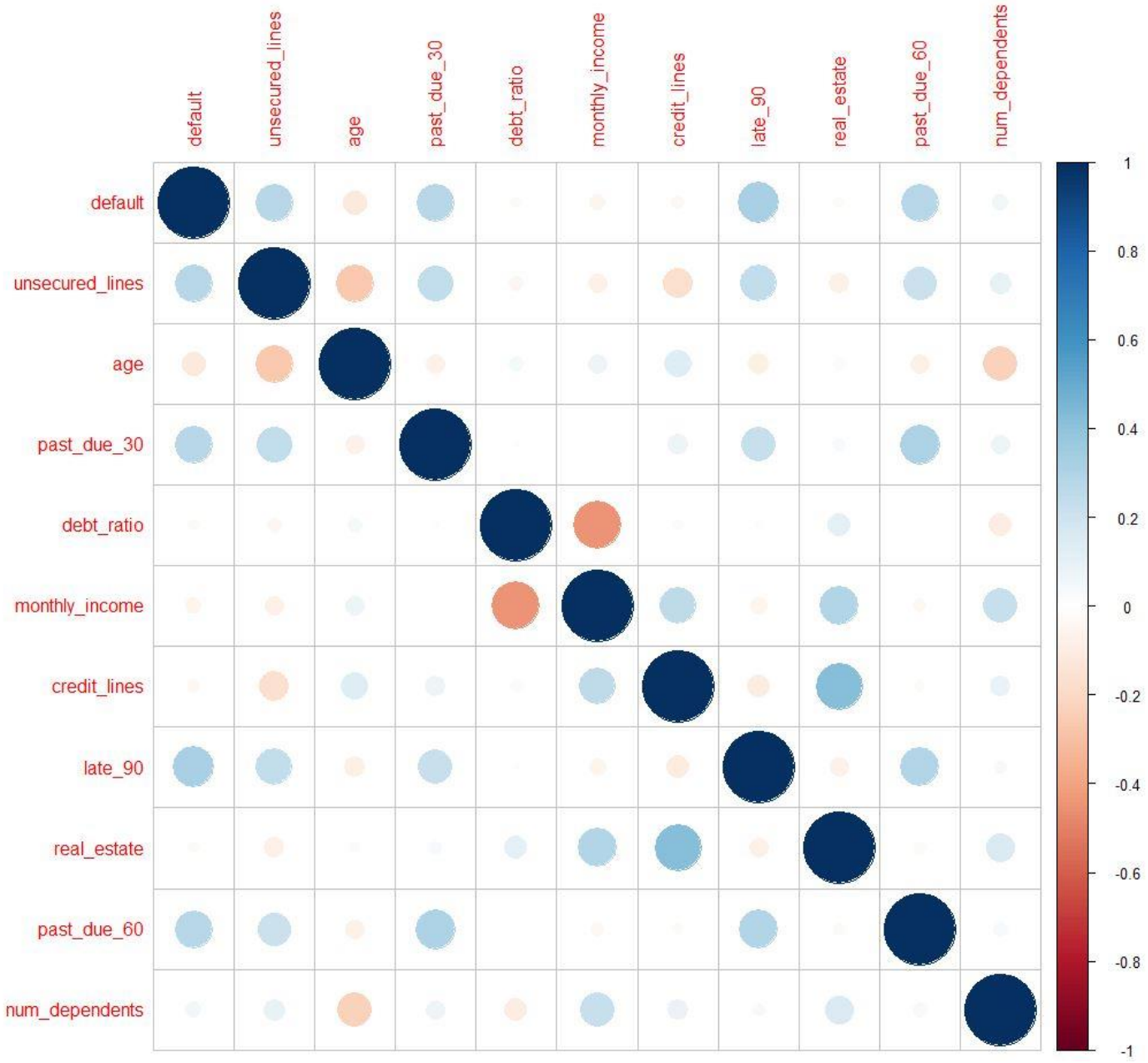
Appendix 4. Summary of the basic statistics of the final dataset (continue from appendix 3)

Basic Statistics	credit_lines	late_90	real_estate	past_due_60	num_dependents
Number of Observations	138999	138999	138999	138999	138999
Number of missing values	0	0	0	0	0
Minimum	0	0	0	0	0
Maximum	21	17	5	11	4
Mean	7.953	0.093	0.923	0.065	0.71
SE Mean	4.428	0.49	0.939	0.329	1.038
Standard Deviation	0.6	9.082	0.965	7.353	1.347
Skewness	-0.061	128.97	0.979	80.981	0.861

Appendix 5. Information loss and proportion of defaulting for debt_ratio

dataset	proportion of defaulting	proportion of data removed
all data	0.06684	0.0000
removing top 97.5th percentile	0.06691	0.0250
remove top 95th percentile	0.06746	0.0499

Appendix 6. Correlation of complete dataset



Code for MANG6054 - Credict Scoring and Data Mining

```
# Import library and dataset

library(dplyr)
library(VIM)
library(tidyverse)
library(ggplot2)
library(mice)
library(fBasics)
library(Metrics)
library(MASS)
library(ISLR)
library(leaps)
library(glmnet)
library(klaR)

setwd("your own working directory")
ori_train_data <- read.table("cs-training.csv", header = TRUE,
                             sep = ",",row.names = 1)

# Rename columns

colnames(ori_train_data)[which(names(ori_train_data) == "SeriousDlqin2yrs")] <- "defau
lt"
colnames(ori_train_data)[which(names(ori_train_data) == "MonthlyIncome")] <- "monthly_
income"
colnames(ori_train_data)[which(names(ori_train_data) == "NumberOfDependents")] <- "num
_dependents"
colnames(ori_train_data)[which(names(ori_train_data) == "RevolvingUtilizationOfUnsecur
edLines")] <- "unsecured_lines"
colnames(ori_train_data)[which(names(ori_train_data) == "NumberOfTime30.59DaysPastDueN
otWorse")] <- "past_due_30"
colnames(ori_train_data)[which(names(ori_train_data) == "NumberOfTime60.89DaysPastDueN
otWorse")] <- "past_due_60"
colnames(ori_train_data)[which(names(ori_train_data) == "DebtRatio")] <- "debt_ratio"
colnames(ori_train_data)[which(names(ori_train_data) == "NumberOfOpenCreditLinesAndLoa
ns")] <- "credit_lines"
colnames(ori_train_data)[which(names(ori_train_data) == "NumberOfTimes90DaysLate")] <-
 "late_90"
colnames(ori_train_data)[which(names(ori_train_data) == "NumberRealEstateLoansOrLines
")] <- "real_estate"
```

Exploratory data analysis

reference: *https://cran.r-project.org/web/packages/dlookr/vignettes/EDA.html*

```
# basic statistic summary
round(basicStats(ori_train_data),3)

# visualising missing patterns
library(naniar)

cols_missing <- ori_train_data[,c("monthly_income", "num_dependents")]
vis_miss(cols_missing)

# visualising correlation
```

```r
cor_ori_train <- cor(ori_train_data[,1:11])

library(corrplot)
corrplot(cor_ori_train, method="circle")

# visualising normality
library(dlookr)
plot_normality(ori_train_data)
```

Histogram

```r
par(mfrow = c(2,2))
hist(ori_train_data$unsecured_lines, n = 30, xlab = "unsecured_lines",
     main = "Histogram of unsecured_lines")
hist(ori_train_data$age, n = 30, xlab = "age",
     main = "Histogram of age")
hist(ori_train_data$past_due_30, n = 30, xlab = "past_due_30",
     main = "Histogram of past_due_30")
hist(ori_train_data$debt_ratio, n = 50, xlab = "debt_ratio",
     main = "Histogram of debt_ratio")

hist(ori_train_data$monthly_income, xlab = "monthly_income",
     main = "Histogram of monthly_income")
hist(ori_train_data$credit_lines, n = 50, xlab = "credit_lines",
     main = "Histogram of credit_lines")
hist(ori_train_data$late_90, n = 50, xlab = "late_90",
     main = "Histogram of late_90")
hist(ori_train_data$real_estate, n = 50, xlab = "real_estate",
     main = "Histogram of real_estate")


par(mfrow = c(2,1))
hist(ori_train_data$past_due_60, n = 50, xlab = "past_due_60",
     main = "Histogram of past_due_60")
hist(ori_train_data$num_dependents, n = 50, xlab = "num_dependents",
     main = "Histogram of num_dependents")
```

Boxplots

```r
no_missing_income <- ori_train_data[!is.na(ori_train_data$monthly_income),]
no_missing_dependent <- ori_train_data[!is.na(ori_train_data$num_dependents),]

par(mfrow=c(2,2))

#unsecured_lines

unsecured_lines.y.bp <- c(ori_train_data$unsecured_lines,
                          no_missing_dependent$unsecured_lines,
                          no_missing_income$unsecured_lines)

unsecured_lines.x.np <- c(rep(1,length(ori_train_data$unsecured_lines)),
                          rep(2,length(no_missing_dependent$unsecured_lines)),
                          rep(3,length(no_missing_income$unsecured_lines)))


boxplot(unsecured_lines.y.bp ~ unsecured_lines.x.np,
        col=c("white","skyblue2","green2"),
        names=c(" \n All data \n n = 150000","Remove missing dependents \n n = 146076
","Remove missing monthly income \n n = 120269"),
```

```
        pch = 19, boxwex = 0.5,
        ylab = "unsecured_lines",
        main = "unsecured_lines boxplots with different datasets")



#AGE

age.y.bp <- c(ori_train_data$age,
              no_missing_dependent$age,
              no_missing_income$age)

age.x.np <- c(rep(1,length(ori_train_data$age)),
              rep(2,length(no_missing_dependent$age)),
              rep(3,length(no_missing_income$age)))

boxplot(age.y.bp ~ age.x.np, col=c("white","skyblue2","green2"),
        names=c(" \n All data \n n = 150000","Remove missing dependents \n n = 146076
","Remove missing monthly income \n n = 120269"),
        pch = 19, boxwex = 0.5,
        ylab = "Age",
        main = "Age boxplots with different datasets")



#past_due_30

past_due_30.y.bp <- c(ori_train_data$past_due_30,
                      no_missing_dependent$past_due_30,
                      no_missing_income$past_due_30)

past_due_30.x.np <- c(rep(1,length(ori_train_data$past_due_30)),
                      rep(2,length(no_missing_dependent$past_due_30)),
                      rep(3,length(no_missing_income$past_due_30)))

boxplot(past_due_30.y.bp ~ past_due_30.x.np,
        col=c("white","skyblue2","green2"),
        names=c(" \n All data \n n = 150000","Remove missing dependents \n n = 146076
","Remove missing monthly income \n n = 120269"),
        pch = 19, boxwex = 0.5,
        ylab = "past_due_30",
        main = "past_due_30 boxplots with different datasets")



#debt_ratio

debt_ratio.y.bp <- c(ori_train_data$debt_ratio,
                     no_missing_dependent$debt_ratio,
                     no_missing_income$debt_ratio)

debt_ratio.x.np <- c(rep(1,length(ori_train_data$debt_ratio)),
                     rep(2,length(no_missing_dependent$debt_ratio)),
                     rep(3,length(no_missing_income$debt_ratio)))

boxplot(debt_ratio.y.bp ~ debt_ratio.x.np,
        col=c("white","skyblue2","green2"),
```

```
        names=c(" \n All data \n n = 150000","Remove missing dependents \n n = 146076
","Remove missing monthly income \n n = 120269"),
        pch = 19, boxwex = 0.5,
        ylab = "debt_ratio",
        main = "debt_ratio boxplots with different datasets")



#credit_lines

credit_lines.y.bp <- c(ori_train_data$credit_lines,
                       no_missing_dependent$credit_lines,
                       no_missing_income$credit_lines)

credit_lines.x.np <- c(rep(1,length(ori_train_data$credit_lines)),
                       rep(2,length(no_missing_dependent$credit_lines)),
                       rep(3,length(no_missing_income$credit_lines)))

boxplot(credit_lines.y.bp ~ credit_lines.x.np,
        col=c("white","skyblue2","green2"),
        names=c(" \n All data \n n = 150000","Remove missing dependents \n n = 146076
","Remove missing monthly income \n n = 120269"),
        pch = 19, boxwex = 0.5,
        ylab = "credit_lines",
        main = "credit_lines boxplots with different datasets")



#late_90

late_90.y.bp <- c(ori_train_data$late_90,
                  no_missing_dependent$late_90,
                  no_missing_income$late_90)

late_90.x.np <- c(rep(1,length(ori_train_data$late_90)),
                  rep(2,length(no_missing_dependent$late_90)),
                  rep(3,length(no_missing_income$late_90)))

boxplot(late_90.y.bp ~ late_90.x.np,
        col=c("white","skyblue2","green2"),
        names=c(" \n All data \n n = 150000","Remove missing dependents \n n = 146076
","Remove missing monthly income \n n = 120269"),
        pch = 19, boxwex = 0.5,
        ylab = "late_90",
        main = "late_90 boxplots with different datasets")



#real_estate

real_estate.y.bp <- c(ori_train_data$real_estate,
                      no_missing_dependent$real_estate,
                      no_missing_income$real_estate)

real_estate.x.np <- c(rep(1,length(ori_train_data$real_estate)),
                      rep(2,length(no_missing_dependent$real_estate)),
                      rep(3,length(no_missing_income$real_estate)))
```

```r
boxplot(real_estate.y.bp ~ real_estate.x.np,
        col=c("white","skyblue2","green2"),
        names=c(" \n All data \n n = 150000","Remove missing dependents \n n = 146076
","Remove missing monthly income \n n = 120269"),
        pch = 19, boxwex = 0.5,
        ylab = "real_estate",
        main = "real_estate boxplots with different datasets")



#past_due_60

past_due_60.y.bp <- c(ori_train_data$past_due_60,
                      no_missing_dependent$past_due_60,
                      no_missing_income$past_due_60)

past_due_60.x.np <- c(rep(1,length(ori_train_data$past_due_60)),
                      rep(2,length(no_missing_dependent$past_due_60)),
                      rep(3,length(no_missing_income$past_due_60)))

boxplot(past_due_60.y.bp ~ past_due_60.x.np,
        col=c("white","skyblue2","green2"),
        names=c(" \n All data \n n = 150000","Remove missing dependents \n n = 146076
","Remove missing monthly income \n n = 120269"),
        pch = 19, boxwex = 0.5,
        ylab = "past_due_60",
        main = "past_due_60 boxplots with different datasets")
```

Exploring outliers and missing values in each variable

unsecured_lines

```r
exploration_data <- ori_train_data

#cut unsecured lines into groups
labs <- c(paste(seq(0, 28, by = 2), seq(0 + 2 - 1, 30 - 1, by = 2),
                sep = "-"), paste(30, "+", sep = ""))

exploration_data$unsecured_lines_group <- cut(exploration_data$unsecured_line, breaks
= c(seq(0, 30, by = 2), Inf), labels = labs, right = FALSE)

unsecured_table<- exploration_data %>%
  group_by(unsecured_lines_group) %>%
  summarise(n=n())

# evaluate the table and the trend in unsecured_lines
unsecured_table

# evaluate proportion of default
unsecured_default <- exploration_data %>%
  dplyr::filter(unsecured_lines >=15) %>%
  summarise(proportion = mean(default), n = n())

unsecured_default2 <- exploration_data %>%
  dplyr::filter(unsecured_lines <15) %>%
  summarise(proportion = mean(default), n = n())
```

real_estate

```
# explore real_estate values
estate_table <- exploration_data %>%
  group_by(real_estate) %>%
  summarise(n = n())

estate_table
```

credit_lines

```
#cut credit_lines into groups
labs <- c(paste(seq(0, 45, by = 5), seq(0 + 45 - 1, 50 - 1, by = 5),
                sep = "-"), paste(50, "+", sep = ""))

exploration_data$credit_lines_group <- cut(exploration_data$credit_lines, breaks = c(s
eq(0, 50, by = 5), Inf), labels = labs, right = FALSE)

#evaluate credit_lines
credit_table <- exploration_data %>%
  group_by(credit_lines_group) %>%
  summarise(n=n())

credit_table
```

debt_ratio

```
#evaluate proportion of default with dataset containing different percentile
debt_table1 <- exploration_data %>%
  summarise(proportion = mean(default), n = n())

debt_boundary_97 <-  quantile(ori_train_data$debt_ratio, probs = 0.975)
debt_boundary_95 <-  quantile(ori_train_data$debt_ratio, probs = 0.95)

debt_table2 <- exploration_data %>%
  dplyr::filter(debt_ratio<= debt_boundary_97) %>%
  summarise(proportion = mean(default), n = n())

debt_table3 <- exploration_data %>%
  dplyr::filter(debt_ratio<= debt_boundary_95) %>%
  summarise(proportion = mean(default), n = n())

#combind result in a table
debt_table <- rbind(debt_table1, debt_table2, debt_table3)
debt_table
```

late payment

```
#explore late payment variables
late_payment_table <- exploration_data %>%
  dplyr::filter(past_due_30 >= 90 & past_due_60 >= 90 & late_90 >=90)
```

Truncation

```
#truncate late payment
clean_train_data <- ori_train_data %>%
  dplyr::filter(past_due_30 < 96 | past_due_60 < 96 | late_90 < 96)

# credit_lines range
Q1.credit <- quantile(clean_train_data$credit_lines, probs = 0.25)
Q2.credit <- quantile(clean_train_data$credit_lines, probs = 0.5)
```

```r
Q3.credit <- quantile(clean_train_data$credit_lines, probs = 0.75)

upper.cut.credit <- Q2.credit + 3*((Q3.credit - Q1.credit)/(2*0.6745))
lower.cut.credit <- Q2.credit - 3*((Q3.credit - Q1.credit)/(2*0.6745))


# real_estate range
Q1.estate <- quantile(clean_train_data$real_estate, probs = 0.25)
Q2.estate <- quantile(clean_train_data$real_estate, probs = 0.5)
Q3.estate <- quantile(clean_train_data$real_estate, probs = 0.75)

upper.cut.estate <- Q2.estate + 3*((Q3.estate - Q1.estate)/(2*0.6745))
lower.cut.estate <- Q2.estate - 3*((Q3.estate - Q1.estate)/(2*0.6745))

# age range
Q1.age <- quantile(clean_train_data$age, probs = 0.25)
Q2.age <- quantile(clean_train_data$age, probs = 0.5)
Q3.age <- quantile(clean_train_data$age, probs = 0.75)

upper.cut.age <- Q2.age + 3*((Q3.age - Q1.age)/(2*0.6745))
lower.cut.age <- Q2.age - 3*((Q3.age - Q1.age)/(2*0.6745))

# debt_ratio range
upper.cut.debt <-  quantile(clean_train_data$debt_ratio, probs = 0.975)

# monthly_income range
upper.cut.income <-quantile(clean_train_data$monthly_income, probs = 0.975, na.rm = T)

# num_dependents range
upper.cut.dependents <-quantile(clean_train_data$num_dependents, probs = 0.975, na.rm
= T)

# truncate variables

clean_train_data2 <- clean_train_data %>%
  dplyr::filter(credit_lines >= lower.cut.credit & credit_lines <= upper.cut.credit) %
>%
  dplyr::filter(real_estate >= lower.cut.estate & real_estate <= upper.cut.estate) %>%
  dplyr::filter(debt_ratio <= upper.cut.debt) %>%
  dplyr::filter(monthly_income <= upper.cut.income | is.na(monthly_income)) %>%
  dplyr::filter(num_dependents <= upper.cut.dependents | is.na(num_dependents)) %>%
  dplyr::filter(age >= lower.cut.age & age <= upper.cut.age)

# replace unsecured_lines outliers with NA
clean_train_data2$unsecured_lines[clean_train_data2$unsecured_lines >= 15] <- NA
```

Imputation

This part is imputation for the missing values in monthly_income, num_dependents, and unsecured_lines

```r
# make 1 datasets with 5 iterations using predictive mean matching

temp_train_data <- mice(clean_train_data2,m=1,maxit=5,meth='pmm',seed=500)

# complete the dataset

train_complete_data = complete(temp_train_data, action=1)
```

Visualising final clean data

```
# visualising correlation
cor_complete_data <- cor(train_complete_data[,1:11])

library(corrplot)
corrplot(cor_complete_data, method="circle")


#histogram
par(mfrow = c(2,2))
hist(train_complete_data$unsecured_lines, n = 30, xlab = "unsecured_lines",
     main = "Histogram of unsecured_lines")
hist(train_complete_data$age, n = 30, xlab = "age",
     main = "Histogram of age")
hist(train_complete_data$past_due_30, n = 30, xlab = "past_due_30",
     main = "Histogram of past_due_30")
hist(train_complete_data$debt_ratio, n = 50, xlab = "debt_ratio",
     main = "Histogram of debt_ratio")

hist(train_complete_data$monthly_income, xlab = "monthly_income",
     main = "Histogram of monthly_income")
hist(train_complete_data$credit_lines, n = 50, xlab = "credit_lines",
     main = "Histogram of credit_lines")
hist(train_complete_data$late_90, n = 50, xlab = "late_90",
     main = "Histogram of late_90")
hist(train_complete_data$real_estate, n = 50, xlab = "real_estate",
     main = "Histogram of real_estate")

par(mfrow = c(2,1))
hist(train_complete_data$past_due_60, n = 50, xlab = "past_due_60",
     main = "Histogram of past_due_60")
hist(train_complete_data$num_dependents, n = 50, xlab = "num_dependents",
     main = "Histogram of num_dependents")

# basic statistics
round(basicStats(train_complete_data),3)
```

Information value

Reference: *https://www.r-bloggers.com/woe-and-iv-variable-screening-with-information-in-r/*

```
infoTables <- create_infotables(data = train_complete_data,

                                y = "default",
                               bins = 10,
                               parallel = T)

# - Plot IV

plotFrame <- infoTables$Summary[order(-infoTables$Summary$IV), ]
plotFrame$Variable <- factor(plotFrame$Variable,

                             levels = plotFrame$Variable[order(-plotFrame$IV)])

ggplot(plotFrame, aes(x = Variable, y = IV)) +
geom_bar(width = .35, stat = "identity") +
ggtitle("Information value of each variable") +
```

```
theme_bw() +
theme(plot.title = element_text(size = 10)) +
theme(axis.text.x = element_text(angle = 90))

plotFrame$Variable
plotFrame$IV
```

Splitting into training and test data

```
# creates a value for dividing the data into train and test.
# The training value is defined as 75% of the dataset

sample.size = floor(0.75*nrow(train_complete_data))

set.seed(1003)   # set seed to ensure having same random numbers generated


# Randomly identifies the rows equal to sample size from  all the rows of AutoPart dat
aset and stores the row number in train_ind

train_ind = sample(seq_len(nrow(train_complete_data)),size = sample.size)

final_train_data = train_complete_data[train_ind,] #creates the training dataset with
row numbers in train_ind
final_test_data = train_complete_data[-train_ind,]  # creates the test dataset excludi
ng the row numbers in train_ind
```

Model fit

Logistic Regression

Reference: *https://cran.r-project.org/web/packages/dominanceanalysis/vignettes/da-logistic-regression.html*

```
library(dominanceanalysis)

glm_fit <- glm(default~., data=final_train_data, family=binomial(link='logit'))

# predict
glm_prob <- predict(glm_fit, final_test_data, type="response")
glm_pred <- ifelse(glm_prob > 0.5, 1, 0)

glm_CM <- table(glm_pred, final_test_data$default)
glm_accuracy <- 1- mean(glm_pred != final_test_data$default)

glm_rmse <- rmse(as.numeric(final_test_data$default), as.numeric(glm_pred))
```

LDA

```
LDA_fit <- lda(default~., data = final_train_data)

# prediction and confusion table

LDA_pred <- predict(LDA_fit, final_test_data)

LDA_CM <- table(LDA_pred$class, final_test_data$default)
LDA_accuracy <- 1- mean(LDA_pred$class != final_test_data$default)  # error rate
```

```
LDA_rmse <- rmse(final_test_data$default, as.numeric(LDA_pred$class))
```

QDA

```
QDA_fit <- qda(default~., data = final_train_data)


# prediction and confusion table

QDA_pred <- predict(QDA_fit, final_test_data)

QDA_CM <- table(QDA_pred$class, final_test_data$default)
QDA_accuracy <- 1 - mean(QDA_pred$class != final_test_data$default)  # error rate

QDA_rmse <- rmse(final_test_data$default, as.numeric(QDA_pred$class))
```

LASSO

```
library(glmnet)
library(caret)

x.train <- model.matrix(default~., final_train_data)[,-1]
y.train <- final_train_data$default

x.test <- model.matrix(default~., final_test_data)[,-1]
y.test <- final_test_data$default

cv.lasso <- cv.glmnet(x.train, y.train, alpha = 1, type.measure = "mae")
plot(cv.lasso)

lasso.best.lam =cv.lasso$lambda.min
lasso.best.lam

lasso.model <- glmnet(x.train, y.train, alpha = 1, family = "binomial")
lasso.coef <- coef(lasso.model,s=lasso.best.lam)[,1]


lasso.prob <- predict(lasso.model, newx = x.test, s = lasso.best.lam)
lasso.pred <- ifelse(lasso.prob > 0.5, 1, 0)

(lasso_info2 <- postResample(lasso.pred, y.test))



lasso_CM <- table(lasso.pred, final_test_data$default)
lasso_accuracy <- mean(lasso.pred == final_test_data$default)

lasso_rmse <- rmse(final_test_data$default, lasso.pred)
```

Random forest

Reference:

https://www.blopig.com/blog/2017/04/a-very-basic-introduction-to-random-forests-using-r/
https://www.listendata.com/2014/11/random-forest-with-r.html#Preparing-Data-for-Random-Forest

```
#import the package
library(randomForest)
```

```
final_train_data$default <- as.factor(final_train_data$default)
final_test_data$default <- as.factor(final_test_data$default)

# Perform training:
rf_fit = randomForest(default ~ ., data = final_train_data,
                              ntree=100, mtry=2, importance=TRUE)

# Validation set assessment #1: looking at confusion matrix

rf_pred <- predict(rf_fit, final_test_data, type = "class")


# Checking classification accuracy
rf_accuracy <- mean(rf_pred == final_test_data$default)
rf_CM <- table(rf_pred,final_test_data$default)


rf_rmse <- rmse(as.numeric(final_test_data$default), as.numeric(rf_pred))
```

##Decision tree

Reference:

https://stats.stackexchange.com/questions/105760/how-we-can-draw-an-roc-curve-for-decision-trees

https://medium.com/analytics-vidhya/a-guide-to-machine-learning-in-r-for-beginners-decision-trees-c24dfd490abb#:~:text=A%20Decision%20Tree%20is%20a,(Classification%20%26%20Regression%20Trees).

https://www.datacamp.com/community/tutorials/decision-trees-R

```
library(tree)

tree_fit <- tree(default~., data = final_train_data)
tree_pred <- predict(tree_fit, final_test_data, type = "class")


tree_CM <- table(tree_pred, final_test_data$default)
tree_accuracy <- mean(tree_pred == final_test_data$default)

tree_rmse <-rmse(as.numeric(final_test_data$default), as.numeric(tree_pred))
```

Neural Network

Reference: *https://machinelearningmastery.com/non-linear-classification-in-r/*

```
library(nnet)

final_train_data$default <- as.factor(final_train_data$default)
final_test_data$default <- as.factor(final_test_data$default)


# fit model
NN_fit <- nnet(default~., data=final_train_data, size=10, decay=0.0001, maxit=500)

# summarize the fit
summary(NN_fit)

# make predictions
NN_pred <- predict(NN_fit, final_test_data, type="class")
```

```
# summarize accuracy
NN_CM <- table(NN_pred, final_test_data$default)

NN_accuracy <- mean(NN_pred == final_test_data$default)

NN_rmse <-rmse(as.numeric(final_test_data$default), as.numeric(NN_pred))
```

XGBoost

Reference:

https://cran.r-project.org/web/packages/xgboost/vignettes/xgboostPresentation.html#measure-learning-progress-with-xgb.train

```
library(xgboost)
library(ROCR)

gbm_train <- final_train_data

xgb_train_data <- xgb.DMatrix(data = as.matrix(final_train_data[,-1])
                      ,label = as.numeric(final_train_data$default)-1)

xgb_test_data <- xgb.DMatrix(data = as.matrix(final_test_data[,-1])
                     , label = as.numeric(final_test_data$default)-1)

watchlist <- list(train=xgb_train_data, test=xgb_test_data)

xgb_fit <- xgb.train(data=xgb_train_data, max.depth=3
                 , eta=0.01, nthread = 2, nround=2000
                 , watchlist=watchlist, eval.metric = "error"
                 , eval.metric = "logloss"
                 , objective = "binary:logistic")

print(xgb.importance(model = xgb_fit))

#plot importance
xgb.plot.importance(importance_matrix = xgb.importance(model = xgb_fit),
                    xlab = "Relative importance (percentage)",
                    main = "Importance")

#model prediction
xgb_prob <- predict(xgb_fit,xgb_test_data)
xgb_pred <- ifelse(xgb_prob > 0.5, 1, 0)

# summarize accuracy
xgb_CM <- table(xgb_pred, final_test_data$default)

xgb_accuracy <- mean(xgb_pred == final_test_data$default)


#rmse

XGB_rmse <-rmse(as.numeric(final_test_data$default), as.numeric(xgb_pred))
```

Creating table of accuracy

```
accuracy_table <- data.frame(
  glm_accuracy = glm_accuracy,
  LDA_accuracy = LDA_accuracy,
  QDA_accuracy = QDA_accuracy,
  lasso_accuracy = lasso_accuracy,
  rf_accuracy = rf_accuracy,
  tree_accuracy = tree_accuracy,
  NN_accuracy = NN_accuracy,
  xgb_accuracy = xgb_accuracy
)
```

Creating table of RMSE

```
RMSE_table <- data.frame(
  glm_rmse = glm_rmse,
  LDA_rmse = LDA_rmse,
  QDA_rmse = QDA_rmse,
  lasso_rmse = lasso_rmse,
  rf_rmse = rf_rmse,
  tree_rmse = tree_rmse,
  NN_rmse = NN_rmse,
  XGB_rmse = XGB_rmse
)

RMSE_table
```

All the confusion matrix

```
glm_CM
LDA_CM
QDA_CM
lasso_CM

rf_CM
tree_CM
NN_CM
xgb_CM
```

Calculating AUC

```
# Logistic regression AUC
glm_auc <- prediction(glm_prob, final_test_data$default) %>%
  performance(measure = "auc") %>%
  .@y.values

# LDA AUC
LDA_auc <- prediction(LDA_pred$posterior[,2], final_test_data$default) %>%
  performance(measure = "auc") %>%
  .@y.values

# QDA AUC
QDA_auc <- prediction(QDA_pred$posterior[,2], final_test_data$default) %>%
  performance(measure = "auc") %>%
  .@y.values

# Lasso auc
```

```r
lasso_auc <- prediction(lasso.pred, final_test_data$default) %>%
  performance(measure = "auc") %>%
  .@y.values

# Random forest AUC
rf_auc <- prediction(as.numeric(rf_pred), final_test_data$default) %>%
  performance(measure = "auc") %>%
  .@y.values

# Decision tree AUC
tree_auc <- prediction(as.numeric(tree_pred), final_test_data$default) %>%
  performance(measure = "auc") %>%
  .@y.values

# Neural network AUC
NN_auc <- prediction(as.numeric(NN_pred), final_test_data$default) %>%
  performance(measure = "auc") %>%
  .@y.values

# xbg auc
xgb_auc <- prediction(xgb_pred, final_test_data$default) %>%
  performance(measure = "auc") %>%
  .@y.values
```

Creating table of AUC

```r
AUC_table <- data.frame(
  glm_auc = glm_auc[[1]],
  LDA_auc = LDA_auc[[1]],
  QDA_auc = QDA_auc[[1]],
  lasso_auc = lasso_auc[[1]],
  rf_auc = rf_auc[[1]],
  tree_auc = tree_auc[[1]],
  NN_auc = NN_auc[[1]],
  xgb_auc = xgb_auc[[1]]
)

AUC_table
```

Importance plot for glm and VIF

```r
dapres<-dominanceAnalysis(glm_fit)

# plot for importance
plot(dapres, which.graph ="general",fit.function = "r2.m")
anova(glm_fit, test="Chisq")

#robustness
plot(glm_fit)

car::vif(glm_fit)
```

Final model result

```r
round(summary(glm_fit)$coefficients,3)
round(anova(glm_fit, test="Chisq"),3)
```