# Default Payments of Credit Clients in Taiwan from 2005

By

*Shasha Li,* sl6805b@student.american.edu

*Alema  Fissuh,* fissuh@american.edu

*Hafid Pradipta,* hp5685a@student.american.edu

*Zhikun Chen Brandon, zc0733a@american.student.edu*

Instructor: J. Alberto Espinosa

**Course:**   2018S Predictive Analytics (ITEC-621-001) (275872)

## Table of Contents

## 1. Background

This report contains the following four parts: Introduction, Exploratory Analysis (Data Pre-processing), Predictive models and summary. Each part will briefly explain the procedures and methodologies followed.

## 2. Introduction

Default credit cards happen when clients fail to adhere to the credit card agreement, by not paying the monthly bill.  Thus, the primary objective of this analysis will focus on finding the best model that predicts well the likelihood of customers' default on credit card. In other words, it will help us to predict, with a reasonable accuracy, whether the customer would fail or succeed in making the next payment.   Various models will be run, compared and the best one with better prediction accuracy will be chosen. The developed models will take into account all possible factors in the data set.

Since our response variable is categorical and binary, we will use the following several predictive models such as Logistic Regression, KNN, Discriminant analysis, Classification trees, Random forest, and bagging. And, finally, we will choose the best method that gives the best accuracy after cross-validation. The final chosen model will benefit the bank before they make any decisions against their customers. The overall target is to minimize the risk of having loan loss.

## 3. Objective

The overall objective will be the development of a system capable of detecting clients that will not be able to pay the debt that they borrow. It focuses on the prediction of defaulters for Credit Card Bank Customers in Taiwan. R programming will be used for exploratory data analysis, visualization purposes and modeling computations. Knowing the probability of credit default can prevent the credit card company to give more credit line if the user cannot pay it in the future.

## 4. Dataset description

We will be using the dataset from UCL repository machine learning linked: (https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients).

Our dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

There is a total of 25 variables in our data set. They are listed below as follows:

| | |
|---|---|
| ID | The ID of each client |
| LIMIT_BAL | Amount of given credit in NT dollars (includes individual and family/supplementary credit |
| SEX | Gender (1=male, 2=female) |
| EDUCATION | (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown) |
| MARRIAGE | Marital status (1=married, 2=single, 3=others) |
| AGE | Age in years |
| PAY_0 | Repayment status in September 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above) |

| | |
|---|---|
| PAY_2 | Repayment status in August 2005 (scale same as above) |
| PAY_3 | Repayment status in July 2005 (scale same as above) |
| PAY_4 | Repayment status in June 2005 (scale same as above) |
| PAY_5 | Repayment status in May 2005 (scale same as above) |
| PAY_6 | Repayment status in April 2005 (scale same as above) |
| BILL_AMT1 | Amount of bill statement in September 2005 (NT dollar) |
| BILL_AMT2 | Amount of bill statement in August 2005 (NT dollar) |
| BILL_AMT3 | Amount of bill statement in July 2005 (NT dollar) |
| BILL_AMT4 | Amount of bill statement in June 2005 (NT dollar) |
| BILL_AMT5 | Amount of bill statement in May 2005 (NT dollar) |
| BILL_AMT6 | Amount of bill statement in April 2005 (NT dollar) |
| PAY_AMT1 | Amount of previous payment in September 2005 (NT dollar) |
| PAY_AMT2 | Amount of previous payment in August 2005 (NT dollar) |
| PAY_AMT3 | Amount of previous payment in July 2005 (NT dollar) |
| PAY_AMT4 | Amount of previous payment in June 2005 (NT dollar) |
| PAY_AMT5 | Amount of previous payment in May 2005 (NT dollar) |

| | |
|---|---|
| PAY_AMT6 | Amount of previous payment in April 2005 (NT dollar) |
| Default payment next month | Default payment (1=yes, 0=no) |

**Further explanation about Dataset:**

- There is 30000 Customer's credit card behavior for dates: April – Sep. 2005
- There are 23 independent variables (predictors)
- Dependent Variable is Default payment next month, with Default (1=yes, 0=no)
- ID: ID of each client – which will be removed from any statistical computations

Among these data points, we can easily notice the following key variables.

1. Nominal variables include sex, education, marriage, repayment statuses (PAY_X), etc.
2. Numeric variables contain age, amount of given credit (LIMIT_BAL), amount of bill statements (BILL_AMT), and amount of previous payments (PAY_AMT).
3. The outcome categorical binomial variable (y) indicates whether that customer had default payment the next month or not. If yes, it is labeled 1, otherwise, set to 0.

We converted Numeric variables **BILL_AMT1** through **BILL_AMT6** to **billamt1** through **billamt6** and **PAY_AMT1** through **PAY_AMT6** and **payamt1** through **payamt6** to US dollars by multiplying them by **0.034** to make the interpretation of data easier. In addition to this, in our dataset, **PAY_0** through **PAY_6** has the value of **-2** which is not explained. We assumed that -2 (the lowest value) is the value that shows that customers pay duly. We will shift this dataset from -2 to 0 by adding 2.

By using various dimensionality reduction mechanisms, our variables of interest will be determined from a total of 23 independent variables and will be used with dependent variable listed above.

## 5. Problem definition

The following three questions will be addressed in our analysis.

1. How does the probability of default payment vary by categories of different demographic variables?
2. What is the proportion of payment defaults for specific customers?
3. What is the best method to predict credit card default?

## 6. Exploratory Data Analysis

### 6.1 Visualization based on the characteristic of default customers.

The following are a visualization of some of the different demographic variable categories that play roles in our default credit prediction.

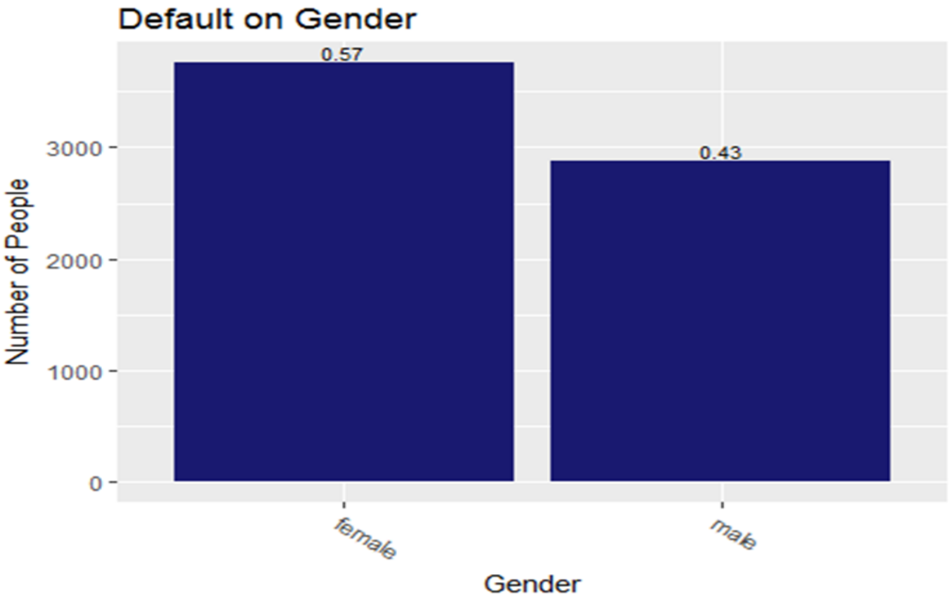


Figure 1. Distribution of default based on gender

**Conclusion about the impact to default:** Female persons have more chances to default in general, according to our analysis.

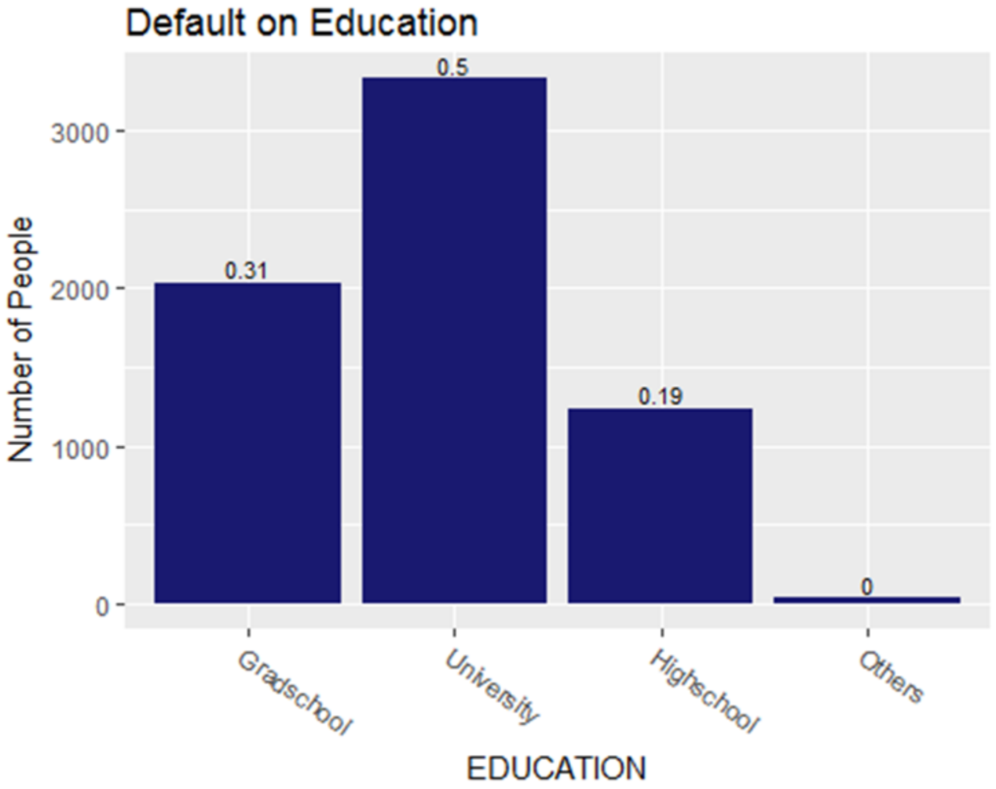


Figure 2. Distribution of default group based on education

**Conclusion about the impact to default:** The university students have higher chances to default in general according to our analysis.

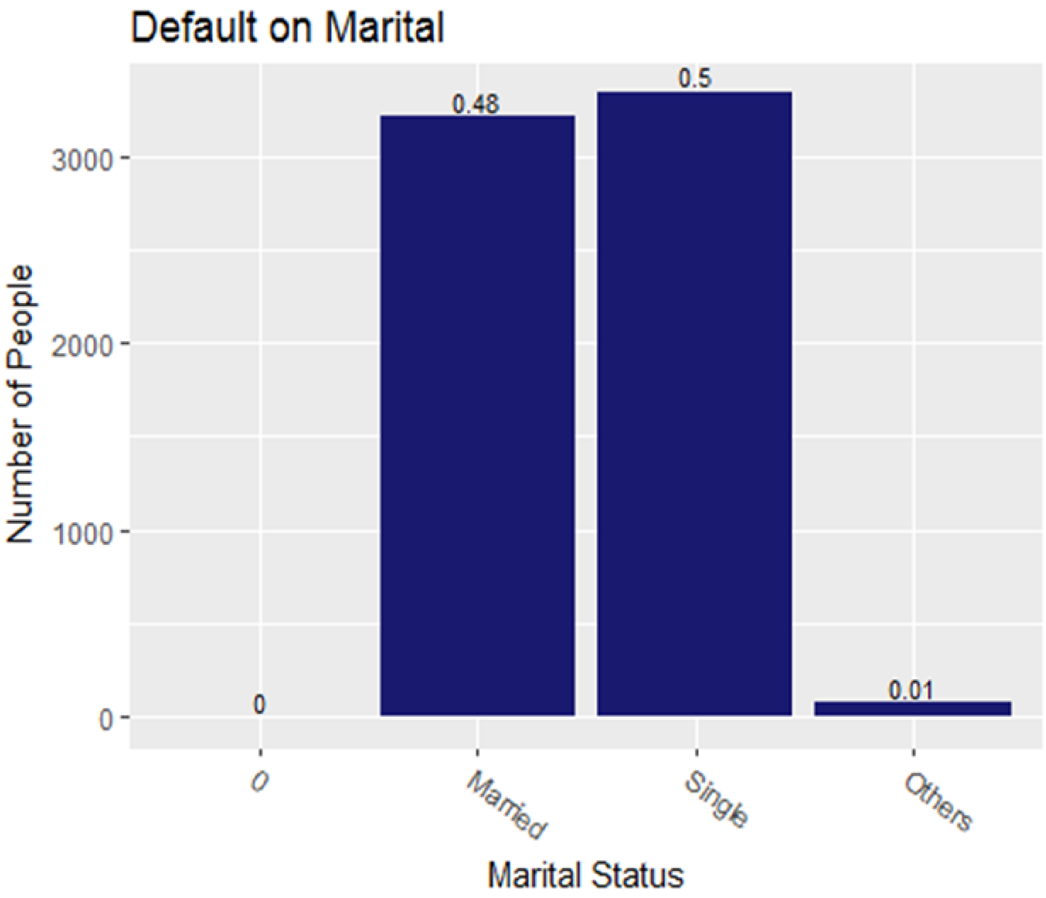


Figure 3.  Distribution of marital status in the default group

**Conclusion about the impact to default:** Single persons have more chances to default.

Based on our analysis and visualization, the group that tends to default more are females, university students, and singles. It seems that more females seem to default payment and in case of education more clients with university education default more.   Marital Status of the client also shows that singles seem to default more.

### 6.2 Visualization of Limit balance in USD

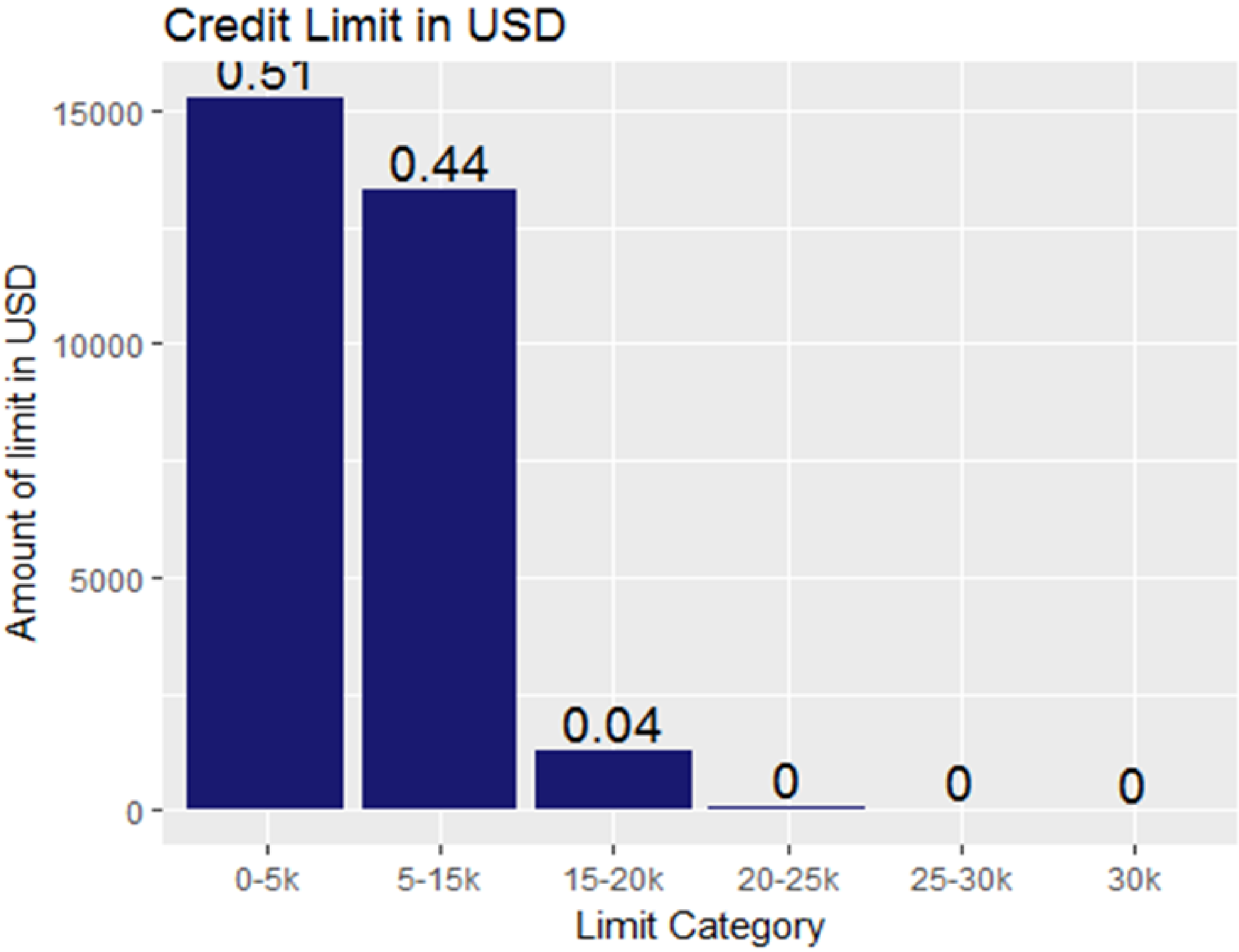


Figure 4. Credit limit distribution in the US

We split the limit balance per 5k and based on the graph, most of the customers have to limit balance between $0 - $15000 which account for 95% of all customers. This variable shows a right-skewed type of probability distribution. It was normalized during our model prediction processes.

### 6.3 Plot of Two-Dimensional PCA

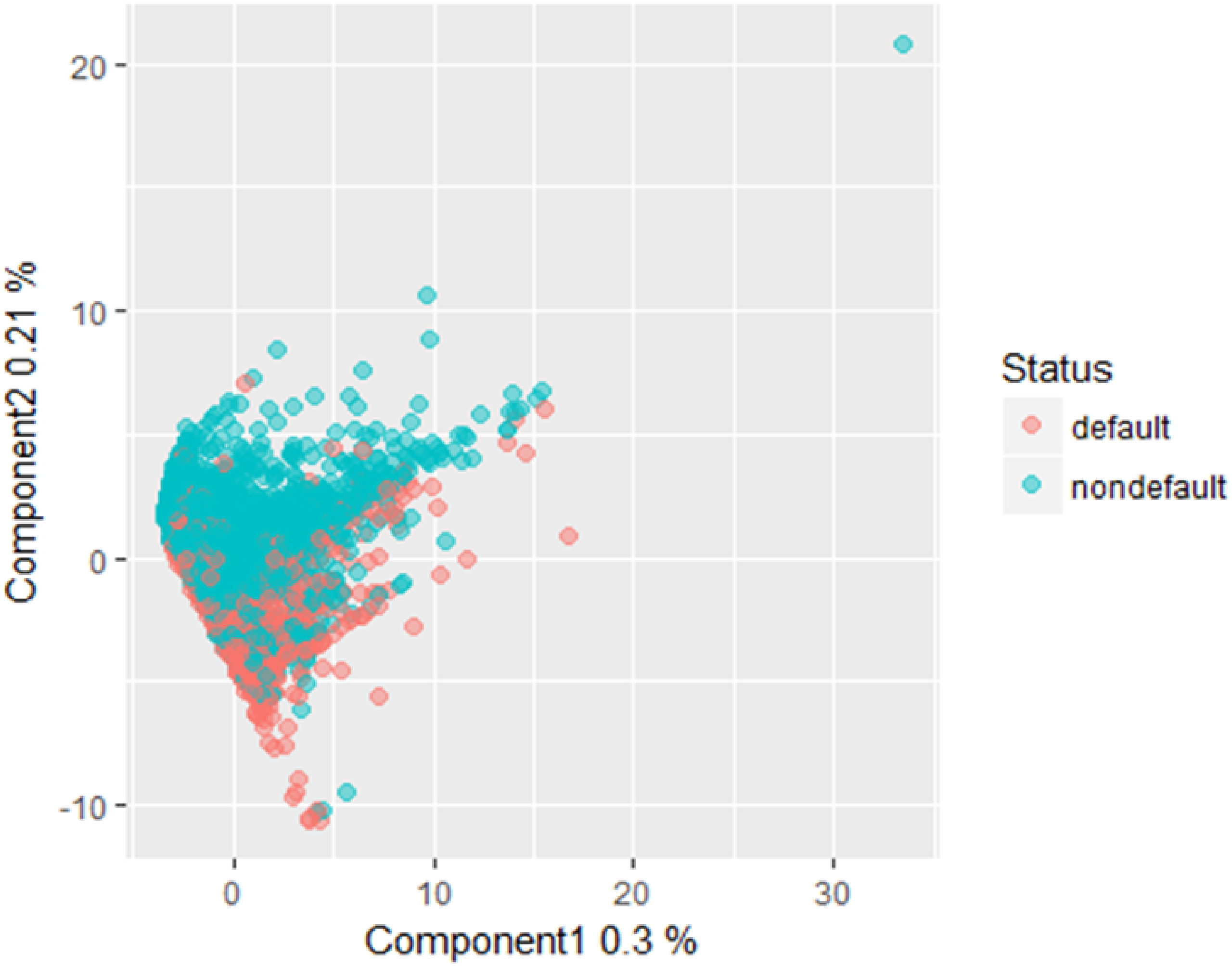
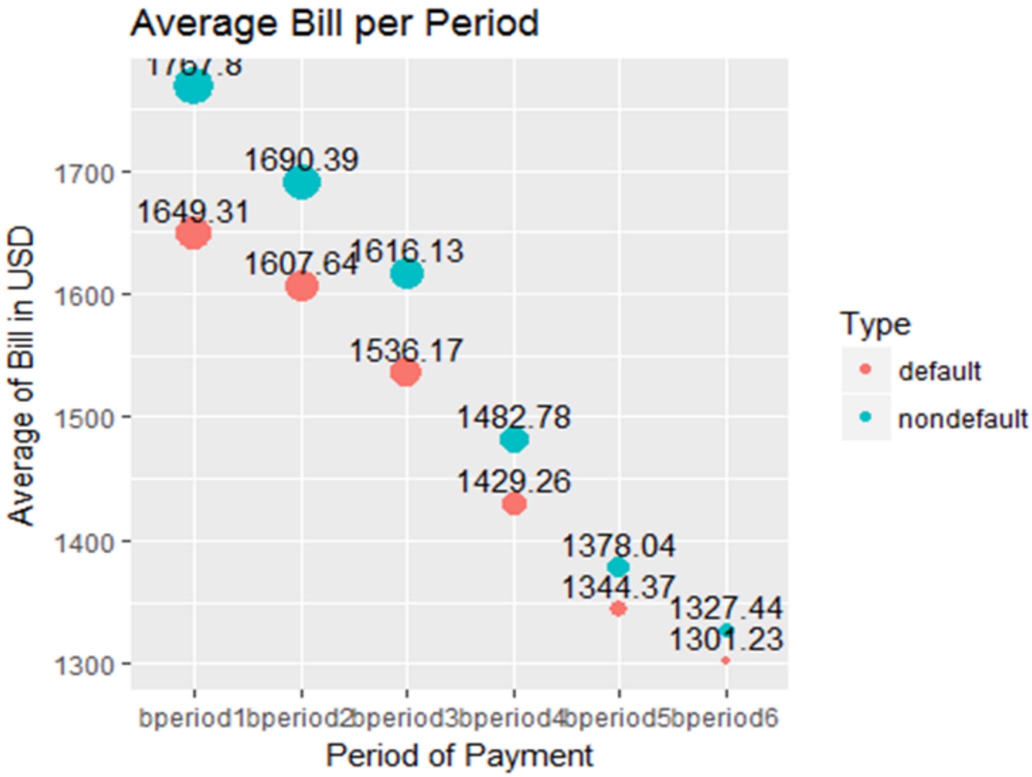


Figure 5. Two-Dimensional principal component analysis (PCA)

Based on the principal component analysis, as shown above, we create two dimensions that are orthogonal to each other and tried to plot the component score. 55.1 % of total variation in the dataset is explained by this graph. It can easily see that there is no clear-cut differentiation between default and non-default. There are some overlaps as can be seen and there is no hope of getting a good result from its predictive analysis.

### 6.4 Visualization of average payment, bill statement and payment punctuality across time.



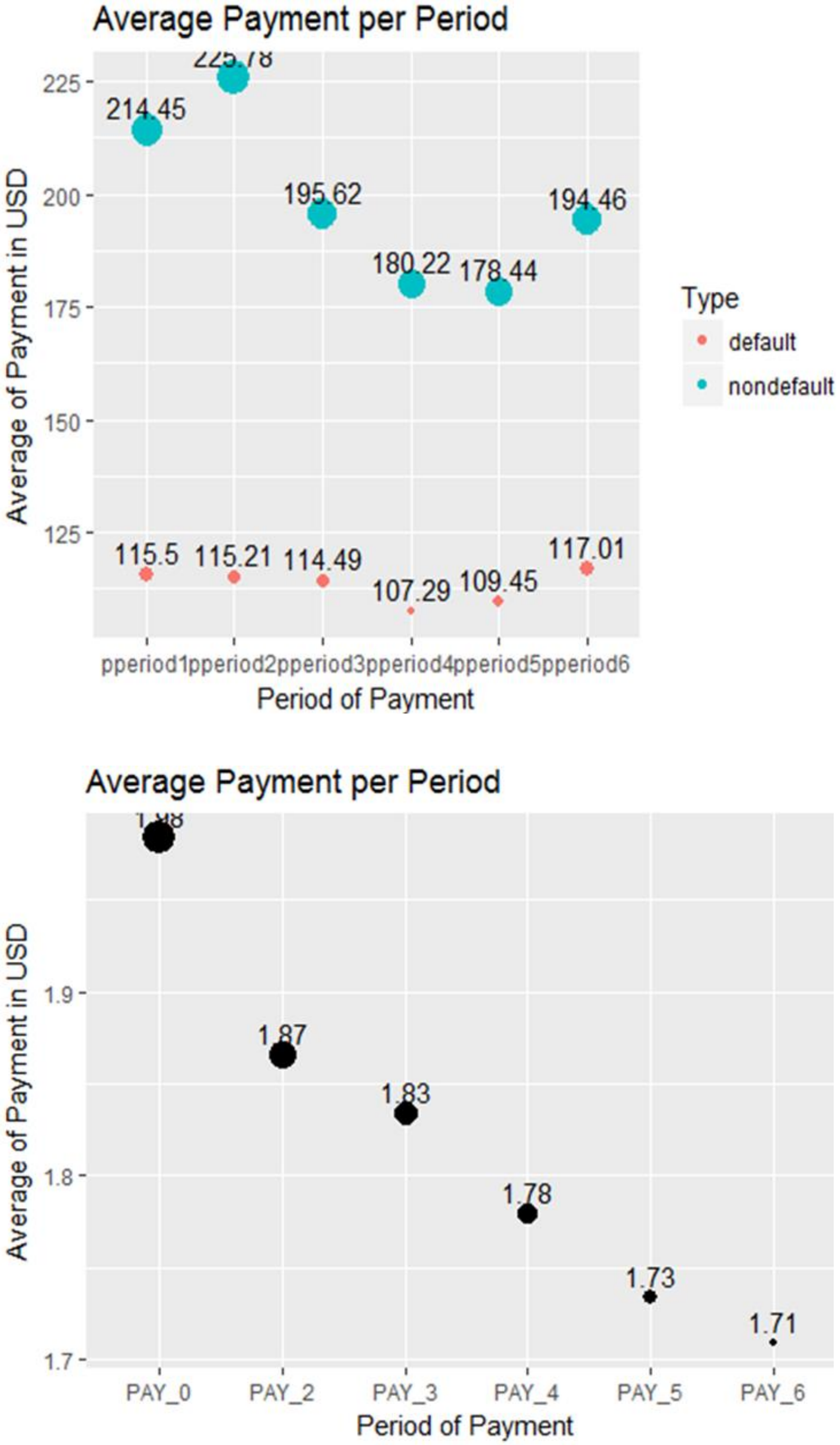


Figure 6. visualization of average bill period for group default and non-default

Here above is the visualization of average bill period for group default and non-default. Even though during the first period, there are differences in the billing (around 90$ on average), it is gradually getting less and less when we move towards period six from period 1 (it narrows down to $20 difference on average only).  However, both groups have significant payments. The non-default group pays almost double when compared with the default group, and their payment tends to vary between $178 (lowest) and $225 (highest). On the other side, default group payments are on average $110.

From this exploratory data analysis, we can see that both of group have a similar bill, but default group only pay half of their bill. The question that can be raised from this visualization is why there is a huge gap between average bill amount and payment? A further study and manipulation of data are required to answer this question. Visualization regarding the lateness of the payment also does not give us much information as the average payment is late by 3.78 months. The question here: does the late payment occur because of several groups who pay late?

### 6.5 Correlation Matrix

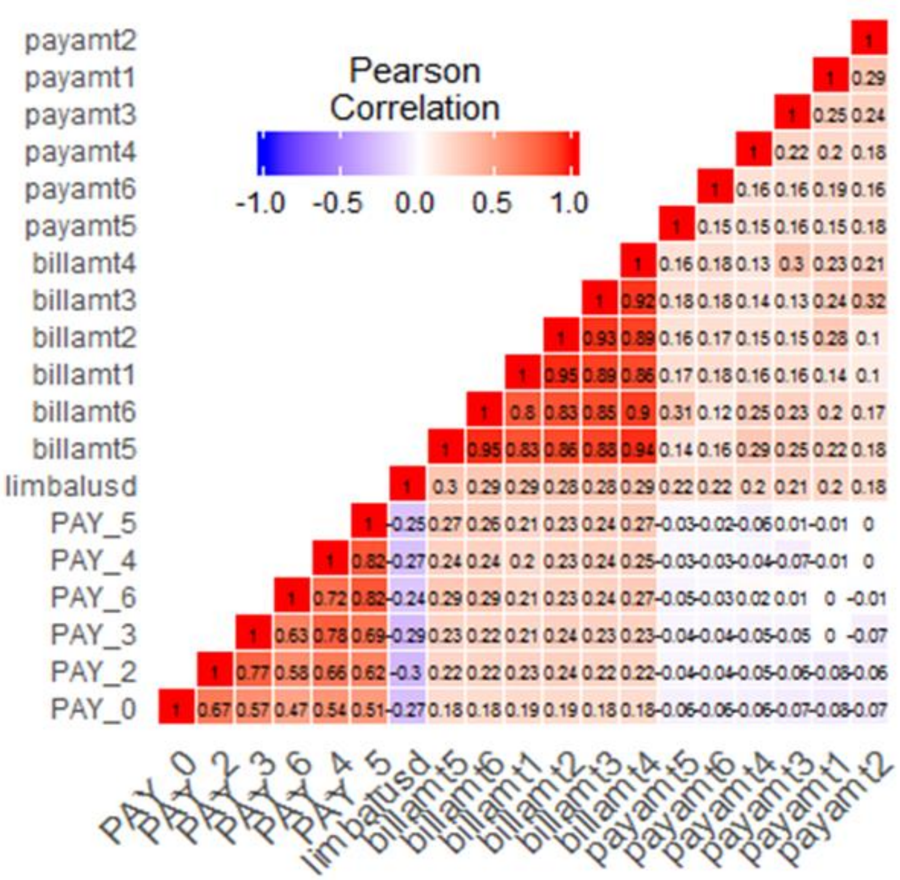


Figure 7. Correlation Matrix

When we reflect the correlations between limit balances, bill amounts, and payments amounts; it presents us that there’s a low correlation between the limit balances and payments and bill amounts. However, it can be seen that bill amounts have a high correlation between each other as expected since the bills a reflecting the cumulative amounts.

## 7. **Data preprocessing**

Before doing modeling, the data set was assessed and went through various proper data preprocessing methods. Our dataset was fairly in clean condition when we first obtained it through Machine Learning Repository site. But still, there were some minor issues to be corrected and was properly corrected.

Some of the issues that were found were as follows. The EDUCATION attribute in which values one through 4 were defined but still some values are greater than four, which goes until six. In this case, using RStudio, undefined values were coded into the more appropriate “Others” category. The same was done for the MARITAL_STATUS attribute, as it contained the same error as our EDUCATION categorical variable.

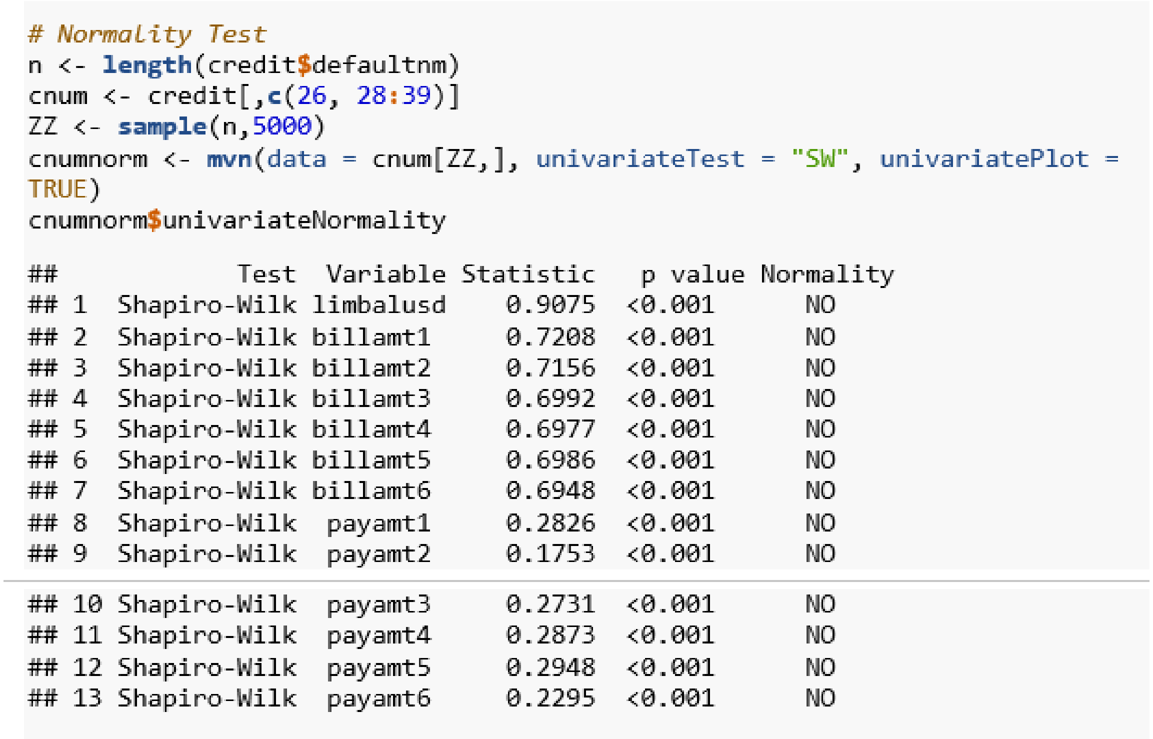
### 7.1 Transformation of Data

The next step in data preprocessing was data transform. The dataset from the UCI Machine Learning Repository came in well-optimized form, which is good for data mining and apply various analytical capabilities. Most of the data values were in numerical form, and those categorical variables such as SEX were changed into the numerical state by encoding as “1” and “2” to represent male and female respectively.

By this way, we transformed SEX, EDUCATION, and ARITAL_STATUS (which were string representations) into numeric using R -code.

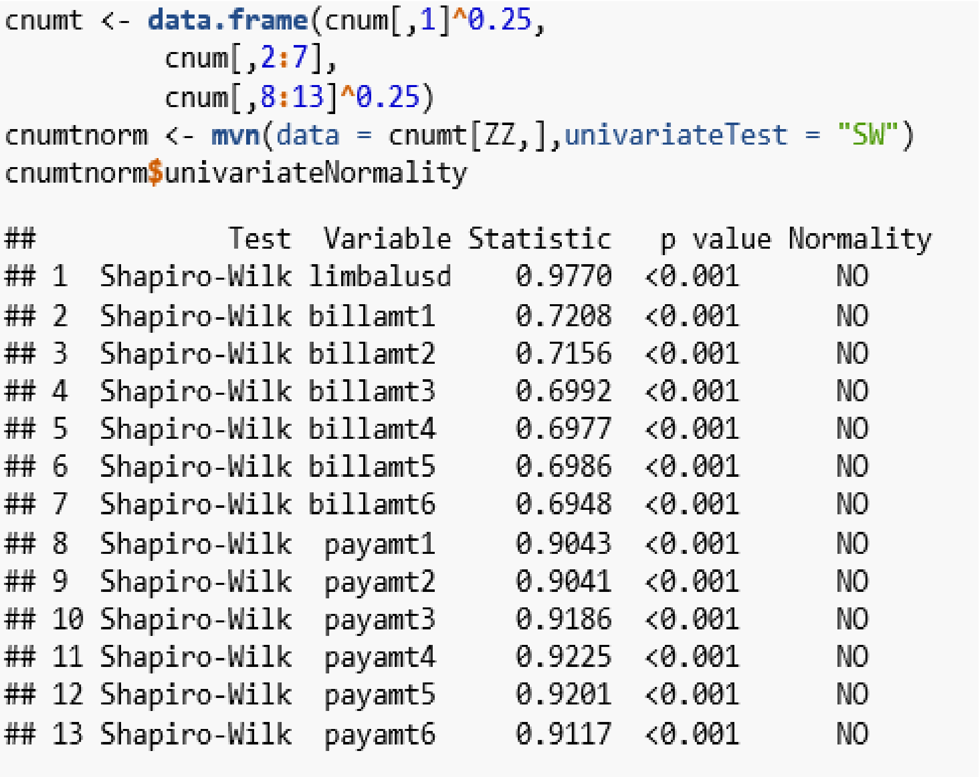
### 7.2 Normalization of Data

Almost all of the independent variables were right skewed and as a result, was normalized as the part of data preprocessing step by using appropriate methodologies depending on the nature of Data set. Besides, Standardization of variables was conducted as needed to get best fit and accuracy of our predictive models. Standardization of independent variables was done because of different units that we have in our data set. Most of the units are in dollars, but some are in months (time unit), etc. thus, to correct this, data was standardized.



```
# Normality Test
n <- length(credit$defaultnm)
cnum <- credit[,c(26, 28:39)]
ZZ <- sample(n,5000)
cnumnorm <- mvn(data = cnum[ZZ,], univariateTest = "SW", univariatePlot =
TRUE)
cnumnorm$univariateNormality

##              Test  Variable Statistic   p value Normality
## 1  Shapiro-Wilk limbalusd    0.9075  <0.001      NO
## 2  Shapiro-Wilk billamt1     0.7208  <0.001      NO
## 3  Shapiro-Wilk billamt2     0.7156  <0.001      NO
## 4  Shapiro-Wilk billamt3     0.6992  <0.001      NO
## 5  Shapiro-Wilk billamt4     0.6977  <0.001      NO
## 6  Shapiro-Wilk billamt5     0.6986  <0.001      NO
## 7  Shapiro-Wilk billamt6     0.6948  <0.001      NO
## 8  Shapiro-Wilk  payamt1     0.2826  <0.001      NO
## 9  Shapiro-Wilk  payamt2     0.1753  <0.001      NO
## 10 Shapiro-Wilk  payamt3     0.2731  <0.001      NO
## 11 Shapiro-Wilk  payamt4     0.2873  <0.001      NO
## 12 Shapiro-Wilk  payamt5     0.2948  <0.001      NO
## 13 Shapiro-Wilk  payamt6     0.2295  <0.001      NO
```

Based on the Shapiro Wilk test, the distributions of the variables (Bill amounts and Payment amounts) are not normal. The value of the variables from payment1 to payment6 are not normal.  In this case, Tukey transformation is used to find the optimum value of the power that will help to make it more normal. By this way, variable limit balance and payment in all period were transformed to the power of 0.25. Variable bill amount can’t be transformed further due to the nature of the distribution.



```
cnumt <- data.frame(cnum[,1]^0.25,
          cnum[,2:7],
          cnum[,8:13]^0.25)
cnumtnorm <- mvn(data = cnumt[ZZ,],univariateTest = "SW")
cnumtnorm$univariateNormality
```

```
##            Test  Variable Statistic   p value Normality
## 1  Shapiro-Wilk limbalusd    0.9770  <0.001      NO
## 2  Shapiro-Wilk billamt1     0.7208  <0.001      NO
## 3  Shapiro-Wilk billamt2     0.7156  <0.001      NO
## 4  Shapiro-Wilk billamt3     0.6992  <0.001      NO
## 5  Shapiro-Wilk billamt4     0.6977  <0.001      NO
## 6  Shapiro-Wilk billamt5     0.6986  <0.001      NO
## 7  Shapiro-Wilk billamt6     0.6948  <0.001      NO
## 8  Shapiro-Wilk  payamt1     0.9043  <0.001      NO
## 9  Shapiro-Wilk  payamt2     0.9041  <0.001      NO
## 10 Shapiro-Wilk  payamt3     0.9186  <0.001      NO
## 11 Shapiro-Wilk  payamt4     0.9225  <0.001      NO
## 12 Shapiro-Wilk  payamt5     0.9201  <0.001      NO
## 13 Shapiro-Wilk  payamt6     0.9117  <0.001      NO
```

After transformation, the variables from **payamt1** through to **payamt6** are not still well normally distributed, even though test statistic indicates that it is more normally distributed than the previous status.

## 8. Model Specifications

First of all, Data was divided into 90% training and 10% testing sets. And 10-fold cross validation (10KFCV) was done for each model. And, a total of five models was run to get the best prediction accuracy. Those models were a logistic regression, QDA, LDA, KNN and random forest.

Every model have undergone four steps.

1. all models were run by using original variables without transformation (original data). They were trained and tested. Cross-validation was done on four of them.

2. All five models were run using selected significant variables from original data – no transformation. The four models were trained, tested and cross-validated
3. All models were run using a transformed variable and the remaining steps on #2
4. Lastly, all models were run using selected significant variables of transformed data. All models were cross-validated by 10FCV.

For variable selection, stepwise variable selection (i.e. both forward and backward) method was used. Both backward and forward selection gave us the same result. The variables finally selected and used were

LIMIT_BAL + SEX + EDUCATION + MARRIAGE +AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + billamt1 + billamt2 + billamt5 + payamt1 + payamt2 + payamt3 + payamt4 + payamt5 + payamt6.

## 9. Models building

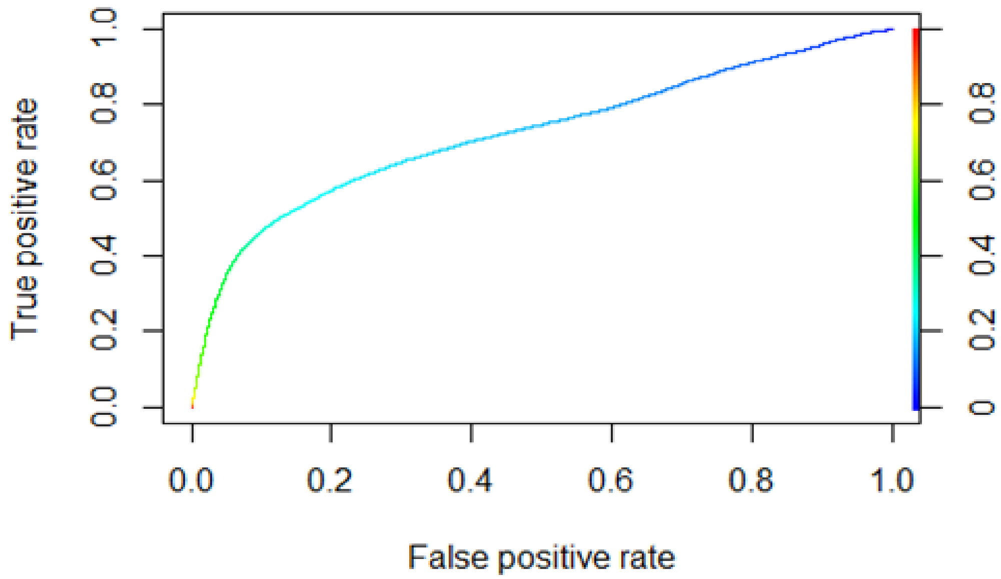
This section focuses on building and choosing best model for predicting the default payment outcome. Before building the model, the dataset was divided in training and test data set.

- Train Data Set = 90%
- Test Data Set = 10%

### 9.1 Logistic Regression

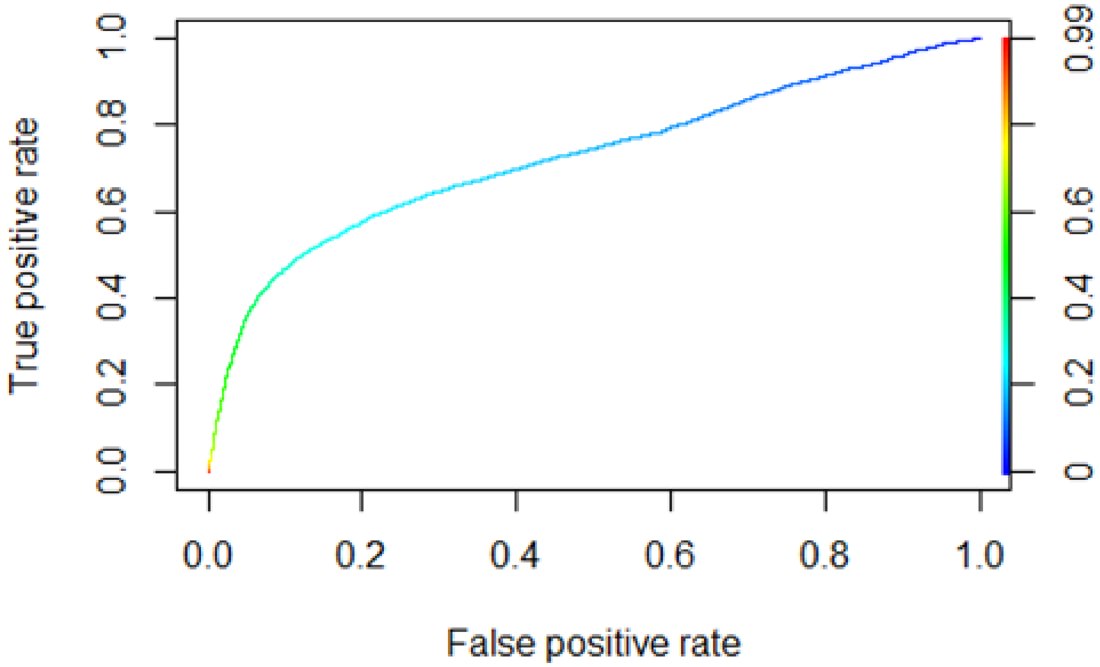
#### 9.1.1 Logistic Regression before transforming variables

Initial model run without normalizing the input variables (predictors and it shows that education, pay_5, pay_6, limit balance, billamt2-billamt5, payamt3 and payamt6 are not significant variables. As a result, those insignificant variables were removed, and the obtained prediction accuracy was 78.5%.



```
## [1] "Area under the ROC curve" "0.724279874520893"
```

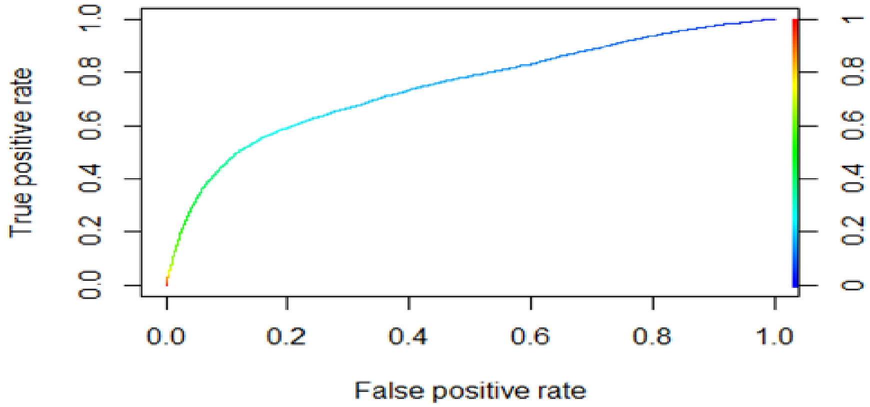
Here also Logistic regression using stepwise variable selection was run, and the results showed that education and payamt6 are not significant. And the obtained model prediction accuracy was 78.5%



```
## [1] "Area under the ROC curve" "0.725117347764003"
```

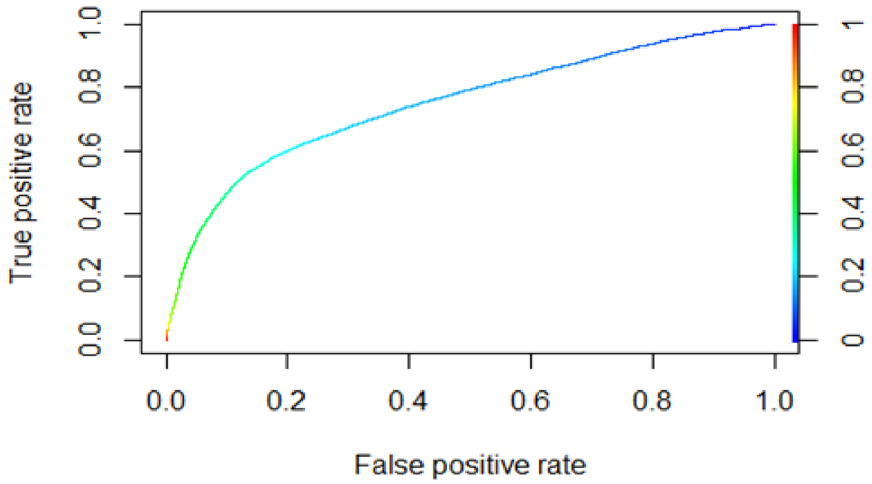
### 9.1.2 Logistic Regression with Transformed Data

Transformed data makes prediction accuracy higher only by 4%. This model uses limbalusd, SEX, MARRIAGE, AGE, PAY_0, PAY_2, PAY_3, PAY_5, PAY_6, billamt6, payamt1, payamt2, payamt3, payamt4 , payamt5 as independent predictor variables and the prediction accuracy obtained was 79%.



```
## [1] "Area under the ROC curve" "0.745748707317294"
```

Logistic regression on transformed data accompanied by variable selection does not give us much better prediction compared to the previous models. Prediction accuracy for this model was 79.1%.



### 9.2 Discriminant Analysis

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) was the second models that were run. Since the assumption of linear discriminant analysis (LDA) is multivariate normality, we conducted the analysis only for comparison purpose to another model (QDA).

LDA accompanied with variable selection method on original (untransformed) variables gave us 81.1% prediction accuracy. After transformation, the prediction accuracy decreases slightly to 80.8% for both original and variable selection model.

QDA for original variables can’t be executed because the data has multicollinearity and the variable selection model gives prediction accuracy of only 45%. Even after transformation, QDA for all variables can't be done because of the multicollinearity issues of the variables. Furthermore, the variable selection of transformed data shows 75% of prediction accuracy.

### 9.3 K- Nearest Neighborhood

Since KNN is nonparametric statistical methods, there is no assumption while conducting this method. The maximum prediction accuracy from KNN was obtained when k =19. It means that 19 neighborhoods were used to determine the vote of the classification (default or not default).  By this way, it was able to predict 80% of default and 20% of non-default group correctly. The prediction accuracy before the transformation was 76%. However, after transformation it went up to 80%.

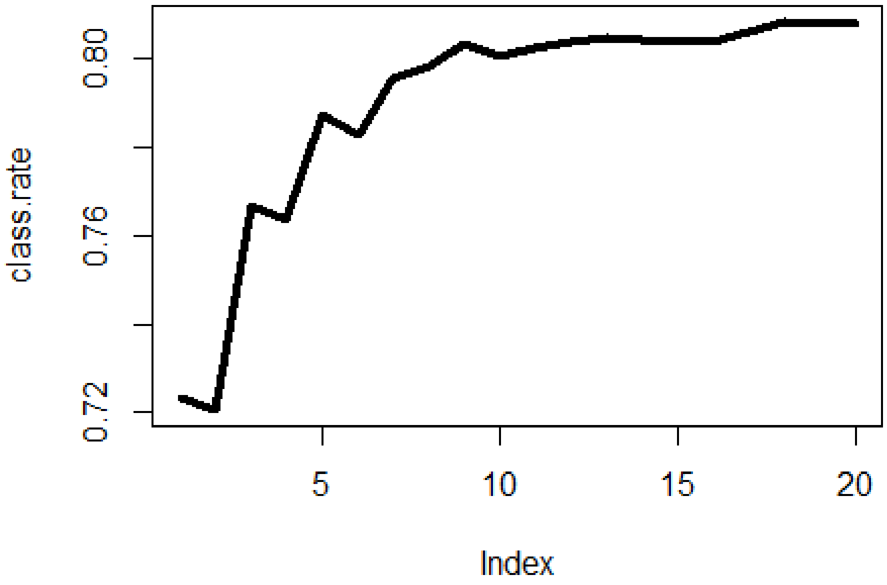


Figure 8. k =19 neighborhoods used to determine the vote of default or not default

### 9.4 Random Forest

We conduct an iteration of random forest to find the optimum number independent variables available at each node and number of optimum trees. Based on this model, we get the prediction accuracy of 70.1%

## 10. Summary of results

The dataset was not univariate, rather multivariate which were not normally distributed. Most of the predictors were right-skewed.

A huge gap was noticed between the average **bill amount** to the customers and the average **payment made each month**. The bill for nondefault and default group is $1500 and $1400 respectively meanwhile the payment is $200 and $100 respectively. The average payment is late by 1.7 months.

Female, university and single customers tend to default more in this dataset. Most of the customers borrow credit card with the limit between $0 to $15,000.

The plot of two highest principal component shows that there is no clear-cut for the customer's default and nondefault.

Various models were run, compared and the best one with better prediction accuracy is chosen. The developed models took into account all possible factors and data. This final chosen model would benefit the bank before they make any decisions against that customers.

Analysis using five different methods with and without transformation shows that Linear Discriminant Analysis (LDA) get the best prediction accuracy for 80.9%. However, since the assumption of LDA is multivariate normality, this model will not be chosen. Rather, we chose

KNN to run on transformed data set as the best model for prediction because this model is a nonparametric method and does not hold any assumption. However, for further interpretation, we would look deeper into the logistic regression of variable selection and transformed data to understand the inference of the nature of this dataset.

| **Models** | **Model Prediction Accuracy** | |
|---|---|---|
| | **Original Data** | **Transformed Data** |
| Logistic Regression | 78.5% | 78.5% |
| Logistic Regression Variable Selection | 79.3% | 79.1% |
| LDA | 81.2% | 81.1% |
| LDA Variable Selection | 80.9% | 80.8% |

| | | |
|---|---|---|
| QDA | - | 44.4% |
| QDA Variable Selection | - | 75% |
| KNN | 76.9% | 80.4% |
| Random Forest | 73.7% | 70.7% |

## 11. Interpretation of Results:

➢ Customers who are females, with a university education and singles, have the highest rate of a payment default.

➢ A customer with the least chance of defaulting are married persons, who have a graduate and high school education.

➢ KNN run using transformed data chosen as best model that can predict customer payment default well.

## Appendix

```
credit2 <- credit
credit <- credit %>%
  mutate(SEX=ifelse(credit$SEX==1,"male","female"))
credit$MARRIAGE <- as.factor(credit$MARRIAGE)
credit$MARRIAGE <- credit$MARRIAGE %>%
  recode(`1` = "Married",
         `2` = "Single",
         `3` = "Others")

credit$EDUCATION <- as.factor(credit$EDUCATION)

credit$EDUCATION <-  credit$EDUCATION %>%
  recode(`1`= "Gradschool",
         `2` = "University",
         `3` = "Highschool",
         `4` = "Others",
         `5` = "Others",
         `6` = "Others")
colnames(credit)[25] <- ("defaultnm")
credit <- credit %>%
  mutate(defaultnm = ifelse(credit$defaultnm==1, "default","nondefault"))
```

```
credit <- credit %>%
  mutate(limbalusd =LIMIT_BAL*0.034)

credit <- credit %>%
  mutate(flimbalusd = cut(limbalusd, c(0, 5000, 15000, 20000 ,25000,30000, Inf), right =
FALSE,
                        labels = c("0-5k","5-15k","15-20k","20-25k","25-30k","30k")))

credit <- credit %>%
  mutate(billamt1 = BILL_AMT1*0.034,
         billamt2 = BILL_AMT2*0.034,
         billamt3 = BILL_AMT3*0.034,
         billamt4 = BILL_AMT4*0.034,
         billamt5 = BILL_AMT5*0.034,
         billamt6 = BILL_AMT6*0.034,
         payamt1 = PAY_AMT1*0.034,
         payamt2 = PAY_AMT2*0.034,
         payamt3 = PAY_AMT3*0.034,
         payamt4 = PAY_AMT4*0.034,
         payamt5 = PAY_AMT5*0.034,
         payamt6 = PAY_AMT6*0.034)

# Normality Test
n <- length(credit$defaultnm)
```

```
cnum <- credit[,c(26, 28:39)]

ZZ <- sample(n,5000)

cnumnorm <- mvn(data = cnum[ZZ,], univariateTest = "SW", univariatePlot = TRUE)

cnumnorm$univariateNormality

##            Test  Variable Statistic   p value Normality
## 1  Shapiro-Wilk limbalusd    0.9069  <0.001      NO
## 2  Shapiro-Wilk billamt1     0.6959  <0.001      NO
## 3  Shapiro-Wilk billamt2     0.6947  <0.001      NO
## 4  Shapiro-Wilk billamt3     0.6477  <0.001      NO
## 5  Shapiro-Wilk billamt4     0.6892  <0.001      NO
## 6  Shapiro-Wilk billamt5     0.6849  <0.001      NO
## 7  Shapiro-Wilk billamt6     0.6803  <0.001      NO
## 8  Shapiro-Wilk  payamt1     0.2865  <0.001      NO
## 9  Shapiro-Wilk  payamt2     0.1190  <0.001      NO
## 10 Shapiro-Wilk  payamt3     0.2496  <0.001      NO
## 11 Shapiro-Wilk  payamt4     0.2454  <0.001      NO
## 12 Shapiro-Wilk  payamt5     0.2445  <0.001      NO
## 13 Shapiro-Wilk  payamt6     0.2636  <0.001      NO

scnum <- scale(cnum, scale = TRUE)

scnumnorm <- mvn(data = scnum[ZZ,], univariateTest = "SW", univariatePlot = TRUE)

ccnum <- scale(cnum, scale = FALSE, center = TRUE)

ccnumnorm <- mvn(data = scnum[ZZ,], univariateTest = "SW")

nresult <- tibble(names = colnames(cnum),Skew=cnumnorm$Descriptives$Skew,

Kurt=cnumnorm$Descriptives$Kurtosis,
                ScaleSkew=scnumnorm$Descriptives$Skew,
ScaleKurt=scnumnorm$Descriptives$Kurtosis,
                CenterSkew=ccnumnorm$Descriptives$Skew,
CenterKurt=ccnumnorm$Descriptives$Kurtosis)



#Tukey Transformation
a <- transformTukey(cnum$limbalusd[ZZ],plotit=FALSE)#0.25

##
##     lambda      W Shapiro.p.value
## 413    0.3 0.9773       1.565e-27
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ab1 <- transformTukey(cnum$billamt1[ZZ],plotit=FALSE)#1

## Warning in log(x): NaNs produced

##
##     lambda      W Shapiro.p.value
## 441      1 0.6959        9.14e-70
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ab2 <- transformTukey(cnum$billamt2[ZZ],plotit=FALSE)#1

## Warning in log(x): NaNs produced

##
##     lambda      W Shapiro.p.value
## 441      1 0.6947       7.566e-70
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ab3<- transformTukey(cnum$billamt3[ZZ],plotit=FALSE)#1

## Warning in log(x): NaNs produced

##
##     lambda      W Shapiro.p.value
## 441      1 0.6477       9.249e-73
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ab4 <- transformTukey(cnum$billamt4[ZZ],plotit=FALSE)#1

## Warning in log(x): NaNs produced
```

```
##
##     lambda      W Shapiro.p.value
## 441      1 0.6892       3.326e-70
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ab5<- transformTukey(cnum$billamt5[ZZ],plotit=FALSE)#1

## Warning in log(x): NaNs produced

##
##     lambda      W Shapiro.p.value
## 441      1 0.6849       1.743e-70
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ab6 <- transformTukey(cnum$billamt6[ZZ],plotit=FALSE)#1

## Warning in log(x): NaNs produced

##
##     lambda      W Shapiro .p.value
##    441      1 0.6803       8.844e-71
##
```

```
## if (lambda >  0){TRANS = x ^ lambda}

## if (lambda == 0){TRANS = log(x)}

## if (lambda <  0){TRANS = -1 * x ^ lambda}

ap1 <- transformTukey(cnum$payamt1[ZZ],plotit=FALSE)#0.3

##

##     lambda      W Shapiro. p.value

## 413    0.3 0.9142       1.167e-46

##

## if (lambda >  0){TRANS = x ^ lambda}

## if (lambda == 0){TRANS = log(x)}

## if (lambda <  0){TRANS = -1 * x ^ lambda}

ap2 <- transformTukey(cnum$payamt2[ZZ],plotit=FALSE)#0.25

##

##     lambda      W Shapiro.p.value

## 412  0.275 0.9003       3.625e-49

##

## if (lambda >  0){TRANS = x ^ lambda}

## if (lambda == 0){TRANS = log(x)}

## if (lambda <  0){TRANS = -1 * x ^ lambda}

ap3 <- transformTukey(cnum$payamt3[ZZ],plotit=FALSE)#0.25

##

##     lambda      W Shapiro.p.value
## 412  0.275 0.9151       1.718e-46
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ap4 <- transformTukey(cnum$payamt4[ZZ],plotit=FALSE)#0.25

##
##     lambda      W Shapiro.p.value
## 412  0.275 0.9227       5.599e-45
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ap5 <- transformTukey(cnum$payamt5[ZZ],plotit=FALSE)#0.25

##
##     lambda     W Shapiro.p.value
## 412  0.275 0.919       9.888e-46
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}

ap6 <- transformTukey(cnum$payamt6[ZZ],plotit=FALSE)#0.25
```

```
##
##     lambda      W Shapiro.p.value
## 411   0.25 0.9123       5.074e-47
##
## if (lambda >  0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda <  0){TRANS = -1 * x ^ lambda}
```

```
cnumt <- data.frame(cnum[,1]^0.25,
           cnum[,2:7],
           cnum[,8:13]^0.25)
cnumtnorm <- mvn(data = cnumt[ZZ,],univariateTest = "SW")
cnumtnorm$univariateNormality
```

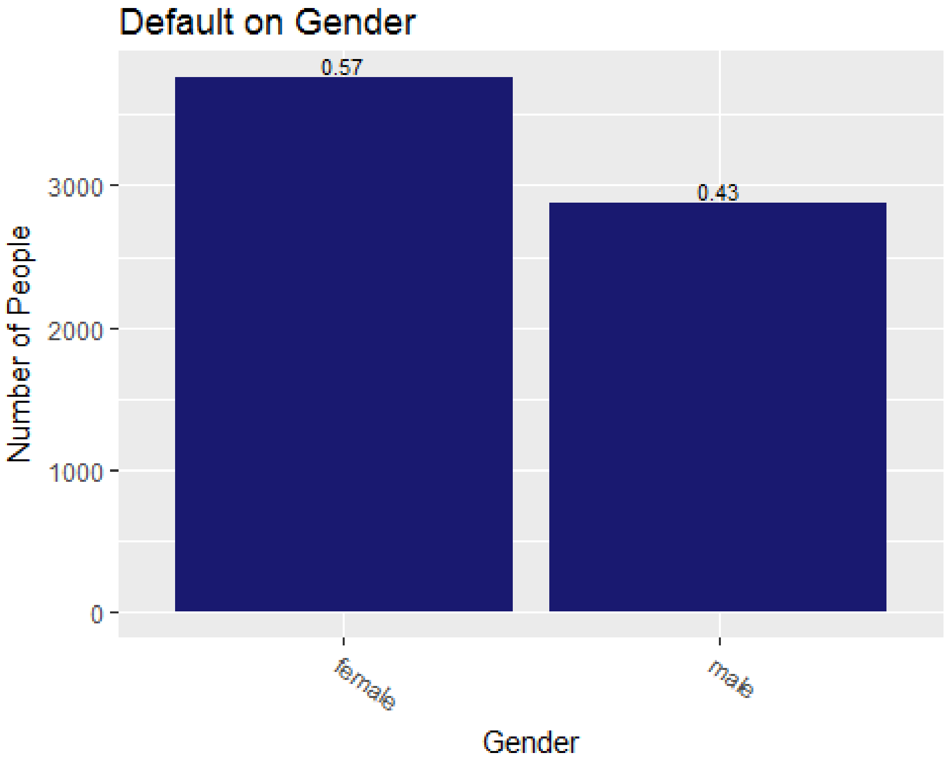
```
##            Test  Variable Statistic   p value Normality
## 1  Shapiro-Wilk limbalusd    0.9765  <0.001      NO
## 2  Shapiro-Wilk billamt1     0.6959  <0.001      NO
## 3  Shapiro-Wilk billamt2     0.6947  <0.001      NO
## 4  Shapiro-Wilk billamt3     0.6477  <0.001      NO
## 5  Shapiro-Wilk billamt4     0.6892  <0.001      NO
## 6  Shapiro-Wilk billamt5     0.6849  <0.001      NO
## 7  Shapiro-Wilk billamt6     0.6803  <0.001      NO
## 8  Shapiro-Wilk  payamt1     0.9057  <0.001      NO
## 9  Shapiro-Wilk  payamt2     0.8979  <0.001      NO
## 10 Shapiro-Wilk  payamt3     0.9134  <0.001      NO
## 11 Shapiro-Wilk  payamt4     0.9206  <0.001      NO
## 12 Shapiro-Wilk  payamt5     0.9164  <0.001      NO

## 13 Shapiro-Wilk  payamt6     0.9123  <0.001      NO
```

```
creditt <- credit

creditt[,26:39] <- cnumt

creditt$flimbalusd <- credit$flimbalusd

names(creditt)
```

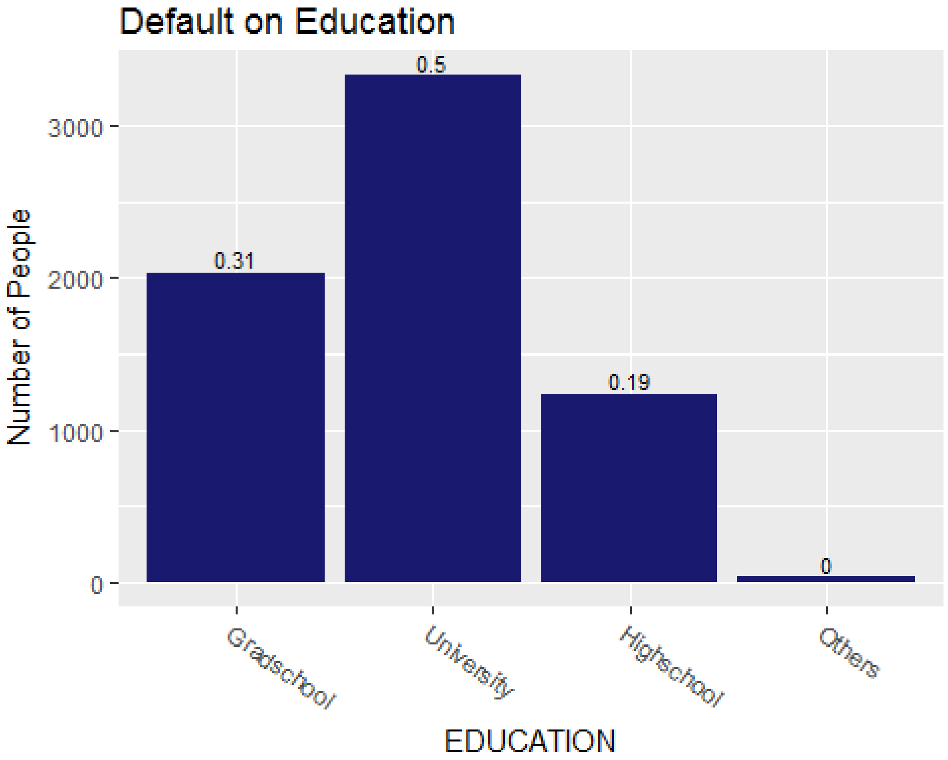
```
##  [1] "ID"          "LIMIT_BAL"  "SEX"         "EDUCATION"  "MARRIAGE"

##  [6] "AGE"         "PAY_0"       "PAY_2"       "PAY_3"       "PAY_4"

## [11] "PAY_5"       "PAY_6"       "BILL_AMT1"  "BILL_AMT2"  "BILL_AMT3"

## [16] "BILL_AMT4"  "BILL_AMT5"  "BILL_AMT6"  "PAY_AMT1"   "PAY_AMT2"

## [21] "PAY_AMT3"   "PAY_AMT4"   "PAY_AMT5"   "PAY_AMT6"   "defaultnm"

## [26] "limbalusd"  "flimbalusd" "billamt1"   "billamt2"   "billamt3"

## [31] "billamt4"   "billamt5"   "billamt6"   "payamt1"    "payamt2"

## [36] "payamt3"    "payamt4"    "payamt5"    "payamt6"
```

```
abc <- credit %>% gather(paste0("billamt",1:6), key = "billperiod", value = "billamount")

abc <- abc %>% gather(paste0("payamt",1:6), key = "payperiod", value = "payamount")

longdata <- abc[,c(25,28:31)]


credit[,7:12] <- credit[,7:12]+2

def <- credit %>% gather(PAY_0,paste0("PAY_",2:6), key = "payperiod", value =

"paypunc")

longdata2 <- def[,34:35]
```

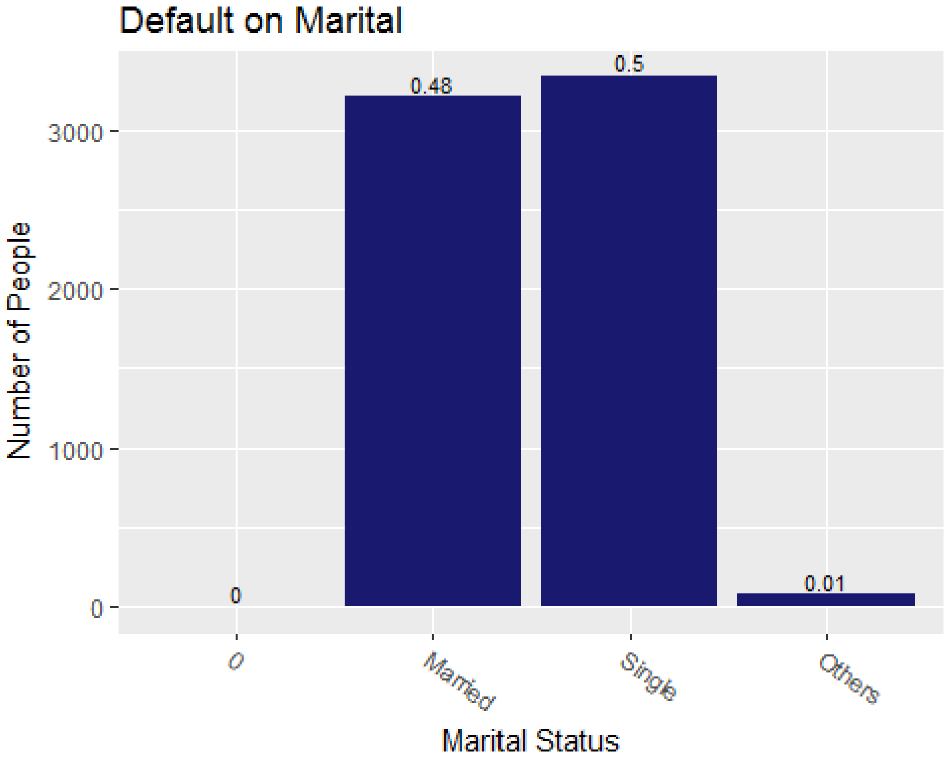
```
default <- credit %>% filter(defaultnm == "default")


d.sex<- default %>% group_by(SEX) %>%
  summarise(count = n()) %>%
  ggplot(aes(x =SEX, y = count), label = count)+
  geom_bar(stat="identity", fill = "midnightblue")+
  geom_text(aes(label = round(count/sum(count),2)), size=3, vjust = -0.25)+
  xlab("Gender")+
  ylab("Number of People")+
  theme(axis.text.x = element_text(angle = 325, hjust = 0))+
  ggtitle("Default on Gender")


d.edu <- default %>% group_by(EDUCATION) %>%
  summarise(count = n()) %>%
  ggplot(aes(x =EDUCATION, y = count), label = count)+
  geom_bar(stat="identity",fill = "midnightblue")+
  geom_text(aes(label = round(count/sum(count),2)), size = 3, vjust = -0.25)+
  ylab("Number of People")+
  theme(axis.text.x = element_text(angle = 325, hjust = 0))+
  ggtitle("Default on Education")


d.mar <-default %>% group_by(MARRIAGE) %>%
  summarise(count = n()) %>%
  ggplot(aes(x =MARRIAGE, y = count))+
  geom_bar(stat="identity", fill = "midnightblue", position = position_stack())+

  geom_text(aes(label = round(count/sum(count),2)), size = 3, vjust = -0.25)+

  ylab("Number of People")+

  xlab("Marital Status")+

   theme(axis.text.x = element_text(angle = 325, hjust = 0))+

  ggtitle("Default on Marital")


d.sex
```



```
d.edu
```

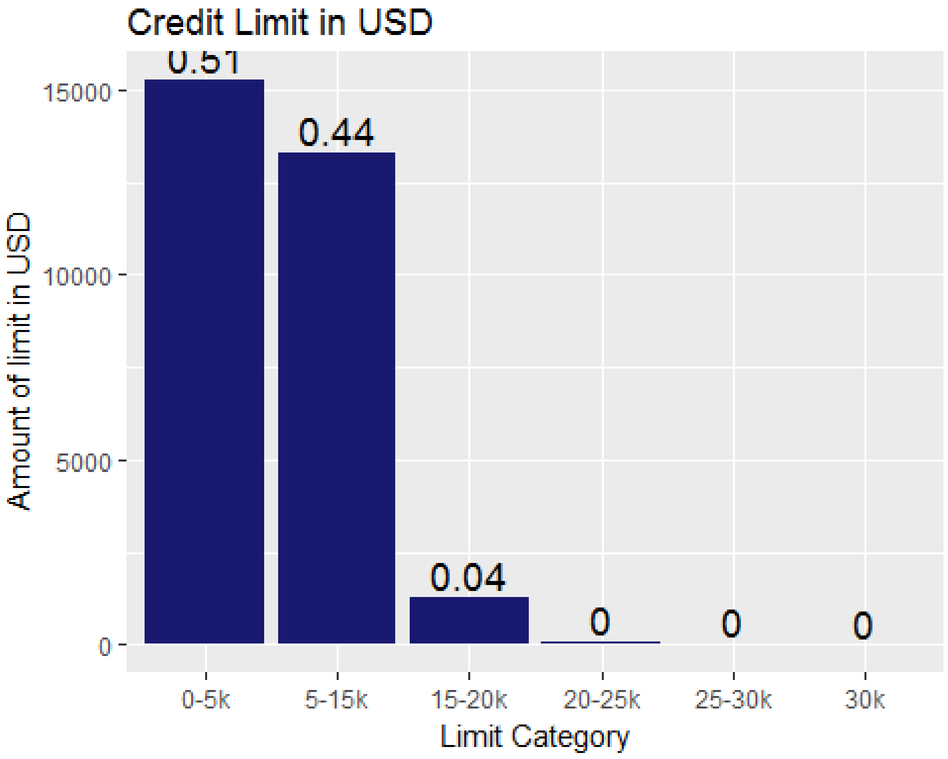

Default on Education
Number of People
3000
2000
1000
0
0.31
0.5
0.19
0
Gradschool
University
Highschool
Others
EDUCATION

```
d.mar
```


Default on Marital
Number of People
3000
2000
1000
0
0
0.48
0.5
0.01
0
Married
Single
Others
Marital Status

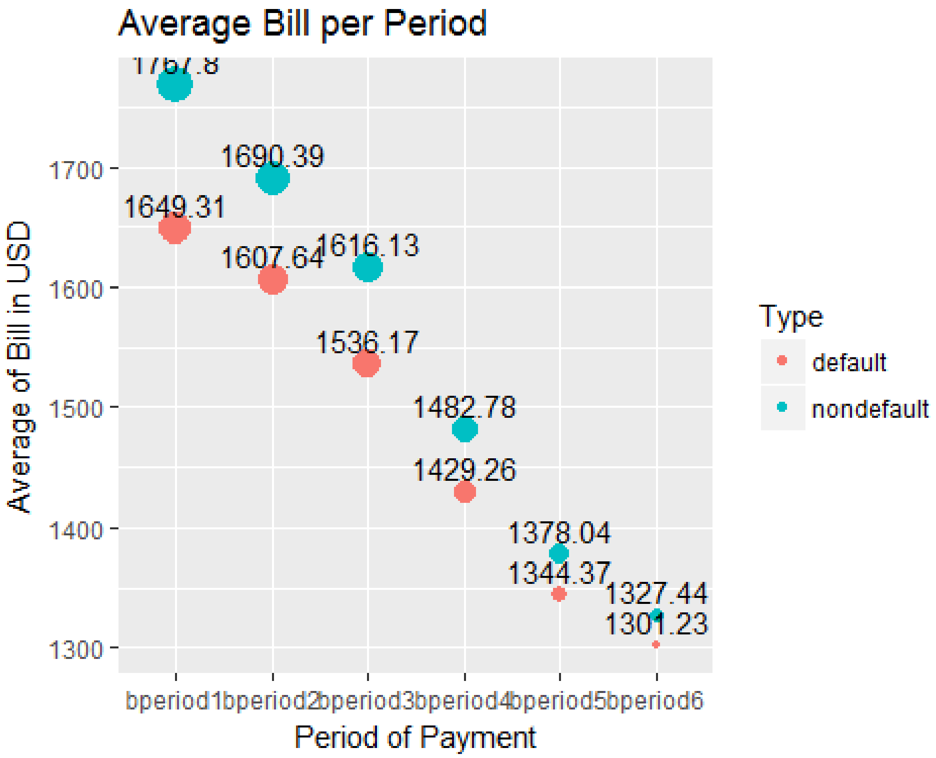
Based on the visualization of the group of default customer, the group that tend to default more are female, university student and single.

```
#cxredit limit visualization
credit %>%
  group_by(flimbalusd) %>%
  summarise(count = n()) %>%
  ggplot(aes(x =flimbalusd, y = count), label = count)+
  geom_bar(stat="identity", fill = "midnightblue")+
  geom_text(aes(label = round(count/sum(count),2)), size = 5, vjust = -0.25)+
  theme(legend.position="none")+
  xlab("Limit Category")+
  ylab("Amount of limit in USD")+
  ggtitle("Credit Limit in USD")
```

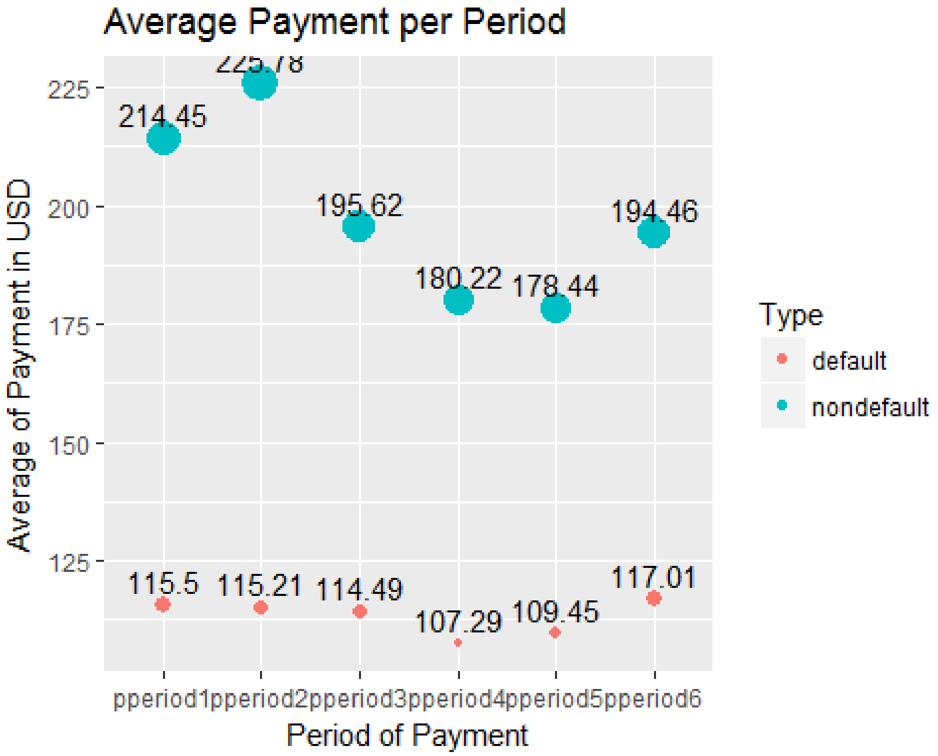


We split the limit balance per 5k and based on the graph, most of the customers has limit balance between $5000 - %15000 which account for 95% of all customers.

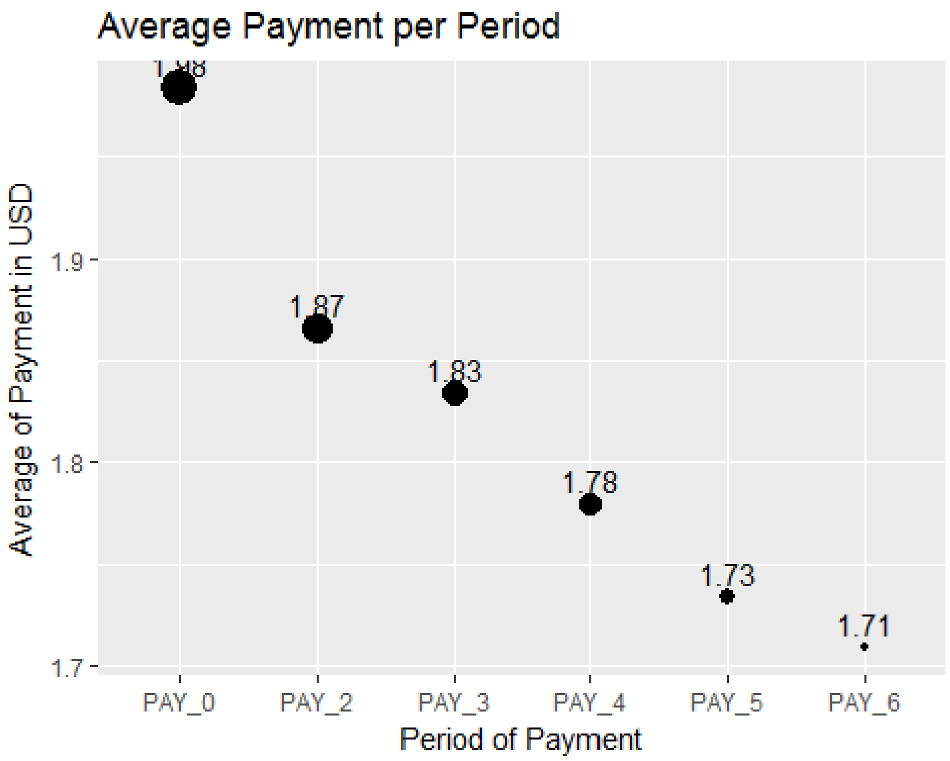
```
badef <- longdata %>%
  filter(defaultnm=="default") %>%
  group_by(billperiod) %>%
  summarise(average= mean(billamount))
bandef <- longdata %>%
  filter(defaultnm=="nondefault") %>%
  group_by(billperiod) %>%
  summarise(average= mean(billamount))

diffba <- badef %>% left_join(bandef, by = "billperiod")
colnames(diffba)[2:3] <- c("default","nondefault")
diffba$billperiod <- diffba$billperiod %>%
  recode("billamt1"="bperiod1",
         "billamt2"="bperiod2",
         "billamt3"="bperiod3",
         "billamt4"="bperiod4",
         "billamt5"="bperiod5",
         "billamt6"="bperiod6")



diffba %>% gather(default, nondefault, key= "Type", value = "amount") %>%
  ggplot(aes(x= billperiod, y=amount))+
  geom_point(aes(color = Type, size = amount))+

  geom_text(aes(label = round(amount,2)), vjust = -0.5)+

  guides(size = FALSE)+

  xlab("Period of Payment")+

  ylab("Average of Bill in USD")+

  ggtitle("Average Bill per Period")
```



```
ppdef <- longdata %>%

 filter(defaultnm=="default") %>%

 group_by(payperiod) %>%

 summarise(average = mean(payamount))



ppnondef <- longdata %>%

 filter(defaultnm=="nondefault") %>%
  group_by(payperiod) %>%
  summarise(average = mean(payamount))



diffpp <- ppdef %>% left_join(ppnondef, by = "payperiod")
colnames(diffpp)[2:3] <- c("default","nondefault")
#################################################################################
diffpp$payperiod <- as.factor(diffpp$payperiod)
diffpp$payperiod <- diffpp$payperiod %>%
  recode("payamt1"="pperiod1",
         "payamt2"="pperiod2",
         "payamt3"="pperiod3",
         "payamt4"="pperiod4",
         "payamt5"="pperiod5",
         "payamt6"="pperiod6")

diffpp %>% gather(default, nondefault, key= "Type", value = "amount") %>%
  ggplot(aes(x= payperiod, y=amount))+
  geom_point(aes(color = Type, size = amount))+
  guides(size = FALSE)+
  geom_text(aes(label = round(amount,2)), vjust = -0.5)+
  xlab("Period of Payment")+
  ylab("Average of Payment in USD")+
  ggtitle("Average Payment per Period")
```



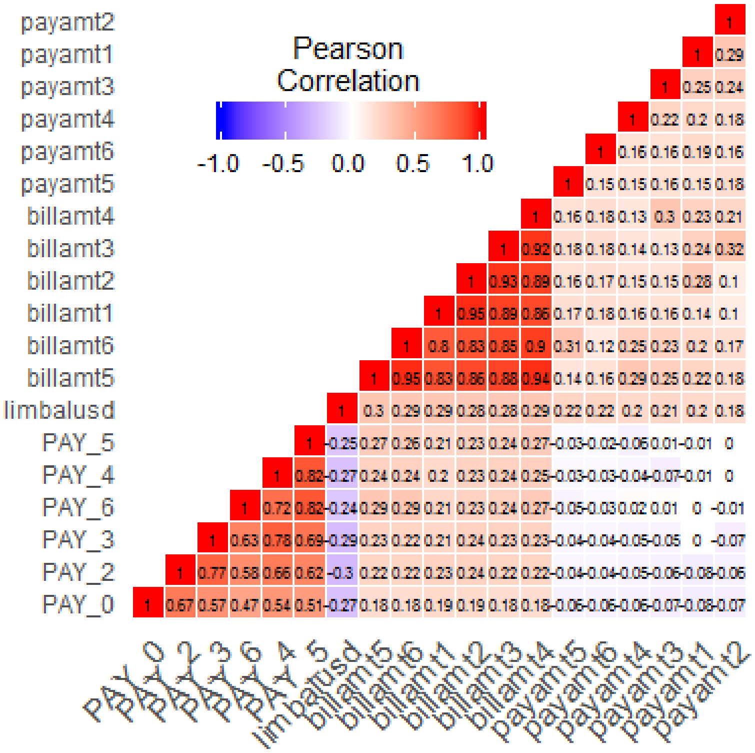
```
longdata2 %>%
 group_by(payperiod) %>%
 summarise(average = mean(paypunc)) %>%
  ggplot(aes(x= payperiod, y=average))+
 geom_point(aes( size = average))+
 guides(size = FALSE)+
 geom_text(aes(label = round(average,2)), vjust = -0.5)+
 xlab("Period of Payment")+
 ylab("Average of Payment in USD")+
 ggtitle("Average Payment per Period")
```



```
library(reshape2)
creditnum <- credit[,c(7:12,26,28:39)]
cormat <- round(cor(creditnum),2)
melted_cormat <- melt(cormat)

# Get lower triangle of the correlation matrix
get_lower_tri<-function(cormat){
  cormat[upper.tri(cormat)] <- NA
  return(cormat)
}
# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
```

```r
  return(cormat)
}
upper_tri <- get_upper_tri(cormat)
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Heatmap
library(ggplot2)

ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                     midpoint = 0, limit = c(-1,1), space = "Lab",
                     name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                size = 12, hjust = 1))+
  coord_fixed()

####reorder
reorder_cormat <- function(cormat){
  # Use correlation between variables as distance
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
```

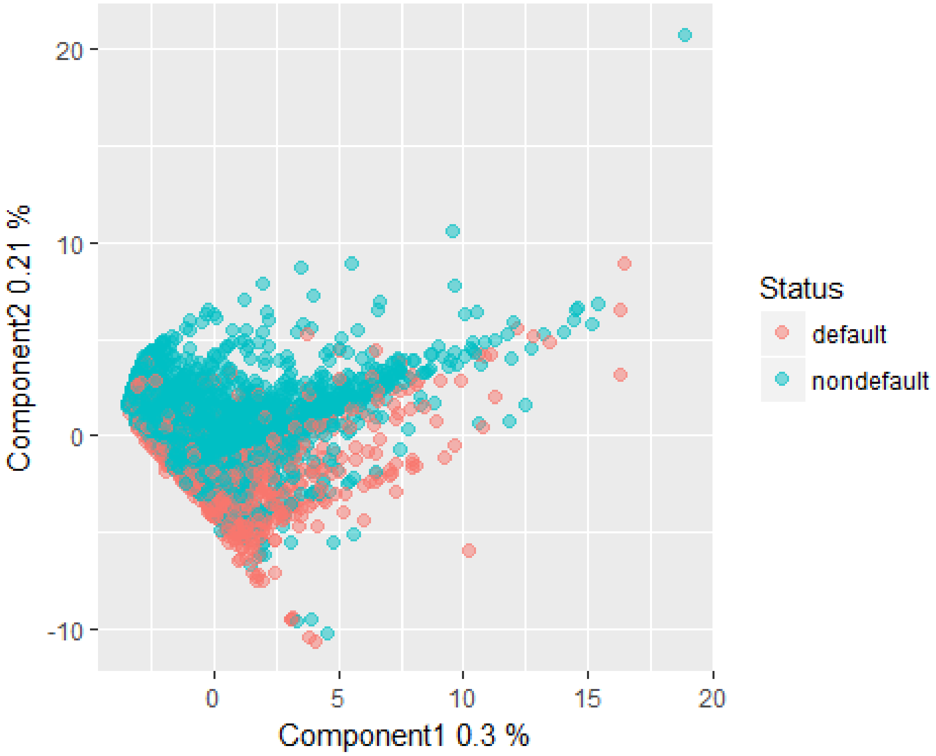
```
}


cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
              midpoint = 0, limit = c(-1,1), space = "Lab",
              name="Pearson\nCorrelation") +
 theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
                     size = 12, hjust = 1))+
 coord_fixed()


ggheatmap +
 geom_text(aes(Var2, Var1, label = value), color = "black", size = 2) +
 theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),

  axis.ticks = element_blank(),

  legend.justification = c(1, 0),

  legend.position = c(0.6, 0.7),

  legend.direction = "horizontal")+

guides(fill = guide_colorbar(barwidth = 7, barheight = 1,

                             title.position = "top", title.hjust = 0.5))
```



The correlation matrix does not give us many information because it is obvious that a particular payment depends on the preious payment.

```
#method1

creditusd <- credit[,c(1:12,25, 26, 28:39)]

creditusd <- creditusd %>% mutate(defaultnm = ifelse (defaultnm == "default", 1, 0))
```

```
creditusd$defaultnm <- as.numeric(creditusd$defaultnm)

prc <- prcomp(creditusd[,-c(1,3:5)], scale = TRUE)


seig1 <- prc$sdev[1]^2/sum(prc$sdev^2)

seig2 <- prc$sdev[2]^2/sum(prc$sdev^2)


grid <- tibble(y = as.factor(creditusd$defaultnm), pc1 =prc$x[,1], pc2 =  prc$x[,2])

gs <- sample(nrow(grid), 5000)

grid <- grid %>%

  mutate(y = ifelse(y == 1, "default", "nondefault"))


grid[gs,] %>% ggplot(aes(x = pc1, y = pc2, color = y))+

  geom_point(alpha = 0.5, size = 2)+

 xlab (paste("Component1", round(seig1,2), "%"))+

ylab (paste("Component2", round(seig2,2), "%"))+

 labs(color = "Status")
```



```
#logistic Regression Initial model

mod1 <- glm(defaultnm~., data = creditusd, family = binomial(link="logit"))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(mod1)

##

## Call:

## glm(formula = defaultnm ~ ., family = binomial(link = "logit"),

##     data = creditusd)

##

## Deviance Residuals:

##     Min       1Q   Median       3Q      Max
## -3.1369  -0.7021  -0.5442  -0.2836   3.8647

##

## Coefficients: (1 not defined because of singularities)

##                        Estimate Std. Error z value Pr(>|z|)

## (Intercept)         -1.479e+01  8.243e+01  -0.179  0.85757

## ID                  -1.147e-06  1.751e-06  -0.655  0.51230

## LIMIT_BAL           -7.030e-07  1.577e-07  -4.457 8.30e-06 ***

## SEXmale              1.122e-01  3.073e-02   3.652  0.00026 ***

## EDUCATIONGradschool  1.080e+01  8.243e+01   0.131  0.89574

## EDUCATIONUniversity  1.072e+01  8.243e+01   0.130  0.89655

## EDUCATIONHighschool  1.070e+01  8.243e+01   0.130  0.89676

## EDUCATIONOthers      9.669e+00  8.243e+01   0.117  0.90663

## MARRIAGEMarried      1.318e+00  5.159e-01   2.555  0.01063 *

## MARRIAGESingle       1.129e+00  5.160e-01   2.189  0.02862 *

## MARRIAGEOthers       1.237e+00  5.328e-01   2.322  0.02023 *

## AGE                  5.485e-03  1.861e-03   2.948  0.00320 **

## PAY_0                5.771e-01  1.771e-02  32.584  < 2e-16 ***

## PAY_2                8.180e-02  2.020e-02   4.049 5.14e-05 ***

## PAY_3                7.100e-02  2.262e-02   3.139  0.00170 **

## PAY_4                2.351e-02  2.504e-02   0.939  0.34773

## PAY_5                3.362e-02  2.690e-02   1.250  0.21137

## PAY_6                6.882e-03  2.215e-02   0.311  0.75603

## limbalusd                   NA         NA      NA       NA

## billamt1            -1.621e-04  3.346e-05  -4.845 1.27e-06 ***
## billamt2          7.030e-05  4.428e-05   1.588  0.11239
## billamt3          4.035e-05  3.896e-05   1.036  0.30026
## billamt4         -2.622e-06  3.988e-05  -0.066  0.94757
## billamt5          2.088e-05  4.489e-05   0.465  0.64185
## billamt6          5.583e-06  3.525e-05   0.158  0.87417
## payamt1          -4.015e-04  6.787e-05  -5.916 3.29e-09 ***
## payamt2          -2.804e-04  6.143e-05  -4.565 5.00e-06 ***
## payamt3          -7.703e-05  5.051e-05  -1.525  0.12727
## payamt4          -1.185e-04  5.255e-05  -2.256  0.02408 *
## payamt5          -9.365e-05  5.233e-05  -1.789  0.07354 .
## payamt6          -6.105e-05  3.813e-05  -1.601  0.10942
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31705  on 29999  degrees of freedom
## Residual deviance: 27831  on 29970  degrees of freedom
## AIC: 27891
##
## Number of Fisher Scoring iterations: 11

#defaultnm~ LIMIT_BAL + SEX + Marriage + AGE + PAY_0 + PAY_2 + PAY_3 + billamt1 +
payamt1 + payamt2 + payamt4
```

```
n <- length(creditusd$defaultnm)

Z <- sample(n,n/10)

c.test <- creditusd[Z,]

mod1.1 <- glm(defaultnm~ LIMIT_BAL + SEX + MARRIAGE + AGE + PAY_0 + PAY_2 +

PAY_3 + billamt1 + payamt1 + payamt2 + payamt4 , data = creditusd[-Z,], family =

binomial(link = "logit"))

summary(mod1.1)


##

## Call:

## glm(formula = defaultnm ~ LIMIT_BAL + SEX + MARRIAGE + AGE +

##     PAY_0 + PAY_2 + PAY_3 + billamt1 + payamt1 + payamt2 + payamt4,

##     family = binomial(link = "logit"), data = creditusd[-Z, ])

##

## Deviance Residuals:

##     Min       1Q   Median       3Q      Max

## -3.1436  -0.6965  -0.5492  -0.3039   3.5948

##

## Coefficients:

##                   Estimate Std. Error z value Pr(>|z|)

## (Intercept)     -3.933e+00  5.274e-01  -7.458 8.81e-14 ***

## LIMIT_BAL       -7.360e-07  1.555e-07  -4.734 2.20e-06 ***

## SEXmale          1.124e-01  3.225e-02   3.487 0.000489 ***

## MARRIAGEMarried  1.126e+00  5.206e-01   2.163 0.030520 *

## MARRIAGESingle   9.591e-01  5.208e-01   1.842 0.065499 .
## MARRIAGEOthers   1.040e+00  5.391e-01   1.929 0.053751 .  
## AGE              5.914e-03  1.899e-03   3.114 0.001846 ** 
## PAY_0            5.851e-01  1.848e-02  31.662  < 2e-16 ***
## PAY_2            8.032e-02  2.093e-02   3.839 0.000124 ***
## PAY_3            1.237e-01  1.922e-02   6.437 1.22e-10 ***
## billamt1        -5.679e-05  8.170e-06  -6.952 3.61e-12 ***
## payamt1         -3.314e-04  6.293e-05  -5.266 1.40e-07 ***
## payamt2         -2.137e-04  5.314e-05  -4.022 5.77e-05 ***
## payamt4         -1.134e-04  4.682e-05  -2.422 0.015453 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 28521  on 26999  degrees of freedom
## Residual deviance: 25180  on 26986  degrees of freedom
## AIC: 25208
## 
## Number of Fisher Scoring iterations: 5
```

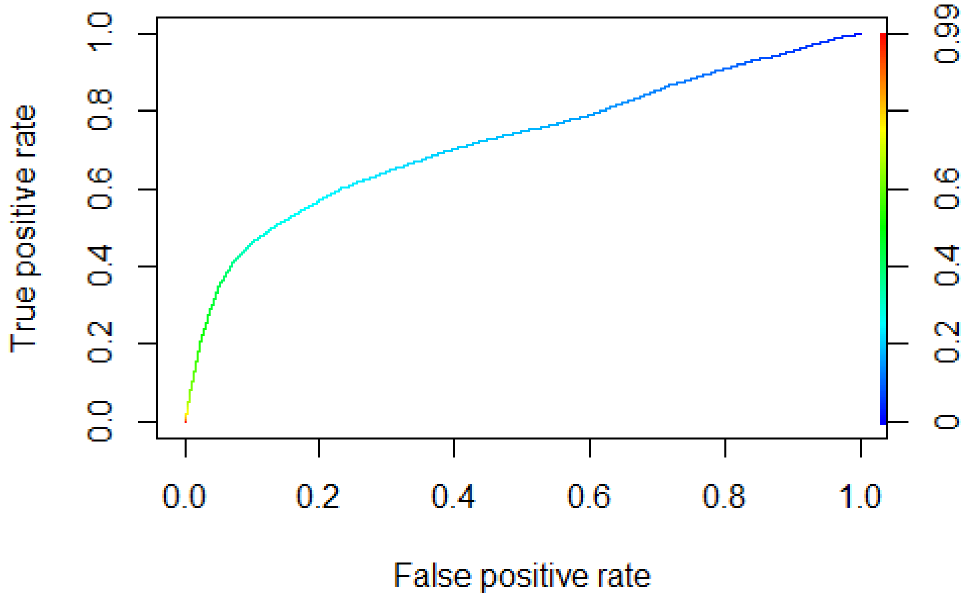
```
prob = predict(mod1.1, data.frame(c.test))
yespred <- 1*(prob>0.5)


table(creditusd$defaultnm[Z], yespred)
```

```
##    yespred
##       0    1
##   0 2318   13
##   1  615   54
```

```
log.in <- mean( creditusd$defaultnm[Z] == yespred)
log.in
```

```
## [1] 0.7906667
```

```
attach(creditusd)
creditusd.prob.test <- predict(mod1.1, creditusd, type ="response")[-Z]
pred <- prediction(creditusd.prob.test,creditusd$defaultnm[-Z] )
perf <- performance(pred, "tpr","fpr")
plot(perf, colorize = TRUE)
```



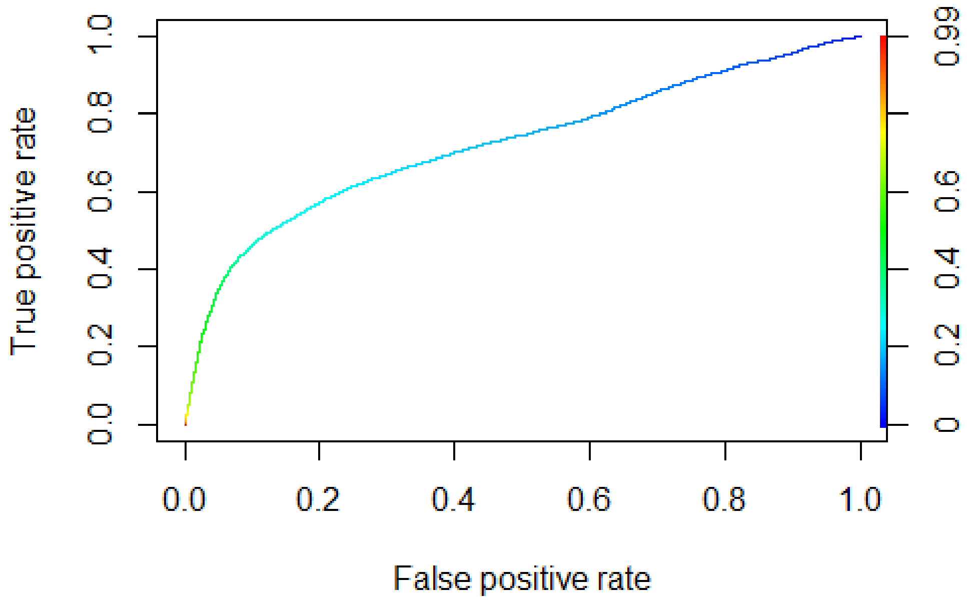
```
auc=performance(pred, "auc")
c(auc@y.name[[1]], auc@y.values[[1]])

## [1] "Area under the ROC curve" "0.721778180283167"

#any kind of threshold will gave the same result
```

Based on the logistic regression of initial model, the prediction accuracy is 78%. However, based on ROC curve, there is no optimum probabilty that gives minimum False Positive Rate while maximizing True Positive Rate. Both of them proportionate to each other. In other case, this model cannot really predict the default of credit card. This model is no more different than flipping a coin. It gives us False positive as wel as True positive proportionately.

```
null = glm( defaultnm ~ 1, data=creditusd, family = binomial(link="logit") )
full = glm( defaultnm ~ ., data=creditusd, family = binomial(link="logit") )

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

#step( null, scope=list(lower=null, upper=full), direction="forward" )


#glm(formula = defaultnm ~ PAY_0 + LIMIT_BAL + PAY_3 + payamt1 +
 #   billamt1 + MARRIAGE + EDUCATION + payamt2 + billamt3 + PAY_2 +
  #  SEX + PAY_5 + AGE + payamt4 + payamt5 + payamt6 + payamt3 +
   # billamt2, family = binomial(link = "logit"), data = creditusd)


#step( full, scope=list(lower=null, upper=full), direction="backward" )
#glm(formula = defaultnm ~ LIMIT_BAL + SEX + EDUCATION + MARRIAGE +
 #AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + billamt1 + billamt2 +
# billamt5 + payamt1 + payamt2 + payamt3 + payamt4 + payamt5 +
# payamt6, family = binomial(link = "logit"), data = creditusd)



modvs <- glm(formula = defaultnm ~ LIMIT_BAL + SEX + EDUCATION + MARRIAGE +
AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + billamt1 + billamt2 +    billamt5 + payamt1 +
payamt2 + payamt3 + payamt4 + payamt5 +    payamt6, family = binomial(link = "logit"),
data = creditusd)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(modvs)

##
## Call:
## glm(formula = defaultnm ~ LIMIT_BAL + SEX + EDUCATION + MARRIAGE +
##     AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + billamt1 + billamt2 +
##     billamt5 + payamt1 + payamt2 + payamt3 + payamt4 + payamt5 +
##     payamt6, family = binomial(link = "logit"), data = creditusd)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1370  -0.7019  -0.5440  -0.2833   3.8915
##
## Coefficients:
```

```
##                         Estimate Std. Error z value Pr(>|z|)

## (Intercept)          -1.481e+01  8.240e+01  -0.180 0.857332

## LIMIT_BAL             -7.141e-07  1.573e-07  -4.540 5.62e-06 ***

## SEXmale               1.124e-01  3.072e-02   3.658 0.000255 ***

## EDUCATIONGradschool  1.080e+01  8.240e+01   0.131 0.895707

## EDUCATIONUniversity  1.072e+01  8.240e+01   0.130 0.896517

## EDUCATIONHighschool  1.069e+01  8.240e+01   0.130 0.896731

## EDUCATIONOthers      9.664e+00  8.240e+01   0.117 0.906634

## MARRIAGEMarried      1.322e+00  5.160e-01   2.562 0.010403 *

## MARRIAGESingle       1.134e+00  5.162e-01   2.197 0.028033 *

## MARRIAGEOthers       1.245e+00  5.330e-01   2.336 0.019471 *

## AGE                  5.504e-03  1.861e-03   2.958 0.003100 **

## PAY_0                5.783e-01  1.767e-02  32.727  < 2e-16 ***

## PAY_2                8.128e-02  2.017e-02   4.029 5.59e-05 ***

## PAY_3                8.140e-02  2.034e-02   4.002 6.29e-05 ***

## PAY_5                5.149e-02  1.790e-02   2.876 0.004026 **

## billamt1            -1.616e-04  3.325e-05  -4.861 1.17e-06 ***

## billamt2             9.507e-05  3.776e-05   2.517 0.011822 *

## billamt5             3.915e-05  1.953e-05   2.005 0.044982 *

## payamt1             -4.055e-04  6.778e-05  -5.982 2.21e-09 ***

## payamt2             -2.457e-04  5.450e-05  -4.509 6.53e-06 ***

## payamt3             -9.814e-05  4.484e-05  -2.188 0.028637 *

## payamt4             -1.265e-04  4.761e-05  -2.658 0.007871 **

## payamt5             -8.890e-05  4.427e-05  -2.008 0.044644 *
## payamt6           -6.119e-05  3.759e-05  -1.628 0.103581
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31705  on 29999  degrees of freedom
## Residual deviance: 27834  on 29976  degrees of freedom
## AIC: 27882
##
## Number of Fisher Scoring iterations: 11

#Education and payamount6 is not significant.
#PAY_0 + payamt2 + payamt1 + PAY_3 + payamt6 + payamt5 + billamt3 + PAY_2 +
billamt1


modvs2 <- glm(formula = defaultnm ~ LIMIT_BAL + SEX  + MARRIAGE +     AGE +
PAY_0 + PAY_2 + PAY_3 + PAY_5 + billamt1 + billamt2 +    billamt5 + payamt1 + payamt2
+ payamt3 + payamt4 + payamt5 , family = binomial(link = "logit"), data = creditusd[-Z,])


prob = predict(modvs2, data.frame(c.test))
yespred <- 1*(prob>0.5)
table(creditusd$defaultnm[Z], yespred)

##    yespred
##       0    1
##   0 2318   13

##   1  612   57

log.vs <- mean( creditusd$defaultnm[Z] == yespred)

log.vs

## [1] 0.7916667

#logistic regression predict 78% correctly




#TPR = rep(0,100)

#FPR = rep(0,100)

#for(k in 1:100){

#  fit = glm(formula = defaultnm ~ limbalusd + SEX  + MARRIAGE +     AGE + PAY_0 +

PAY_2 + PAY_3 + PAY_5 + billamt1 + billamt2 +    billamt5 + payamt1 + payamt2 +

payamt3 + payamt4 + payamt5 , family = binomial(link = "logit"), data = creditusd[-Z,])

#  prob  = predict(fit, data.frame(creditusd[Z,]), type = "response")

 # Yhat = 1*(prob > k/100)

  #TPR[k] = sum(Yhat ==1 & defaultnm==1)/ sum(defaultnm==1)

  #FPR[k] = sum(Yhat ==1 & defaultnm==0)/ sum(defaultnm==0)

#}

#plot(FPR, TPR, xlab="False positive rate", ylab="True positive rate", main="ROC curve")
```

```
creditusd.prob.test <- predict(modvs2, creditusd, type ="response")[-Z]

pred <- prediction(creditusd.prob.test,creditusd$defaultnm[-Z] )

perf <- performance(pred, "tpr","fpr")

plot(perf, colorize = TRUE)
```



```
auc=performance(pred, "auc")

c(auc@y.name[[1]], auc@y.values[[1]])

## [1] "Area under the ROC curve" "0.722885791184621"
```

Another logistic regression model based on the variable selection of stepwise gives us the same accuracy. However, the ROC curve shows that there are equal False Positive Rate as well as True Positive Rate.

### Linear Discriminant Analysis

```
lda.fit <- lda(defaultnm~., data= creditusd, CV = TRUE)

## Warning in lda.default(x, grouping, ...): variables are collinear

table(creditusd$defaultnm, lda.fit$class)

##
##         0     1
##   0 22631   733
##   1  4915  1721

lda.in <- mean(creditusd$defaultnm == lda.fit$class)
lda.in

## [1] 0.8117333

lda.fitvs <- lda( defaultnm ~ LIMIT_BAL + SEX  + MARRIAGE + AGE + PAY_0 + PAY_2 +
PAY_3 + PAY_5 + billamt1 + billamt2 +    billamt5 + payamt1 + payamt2 + payamt3 +
payamt4 + payamt5 , data = creditusd, CV = TRUE)
table(creditusd$defaultnm, lda.fitvs$class)

##
##         0     1
##   0 22644   720
##   1  4940  1696

lda.vs <- mean(creditusd$defaultnm == lda.fitvs$class)
lda.vs
```

```
## [1] 0.8113333
```

Both Linead Discriminant Analysis based on all variables and variable selection give us 81.1% of prediction accuracy.

```
###qda for all variable can't be done due to rank deficiency in one group. It means that the
data has too many multicollinearity.


qda.fitvs <- qda( defaultnm ~ LIMIT_BAL + SEX  + MARRIAGE + AGE + PAY_0 + PAY_2
+ PAY_3 + PAY_5 + billamt1 + billamt2 +    billamt5 + payamt1 + payamt2 + payamt3 +
payamt4 + payamt5 , data = creditusd, CV = TRUE)
table(defaultnm, qda.fitvs$class)

##
## defaultnm     0     1
##         0  7452 15907
##         1   871  5765

qda.vs<- mean(defaultnm == qda.fitvs$class,na.rm=TRUE)
qda.vs

## [1] 0.4406401
```

We can not run QDA using all variables because our predictors variable has strong correlation with each others. It creates rank defficiency in one of the group. It means that there are not too many information(variance) in the predictors to predict the response variable.

However, based on the variable selection, QDA predict 44% correctly for this datset. It is a clear sign that the difference between default group and non default group is not in a curvature area.

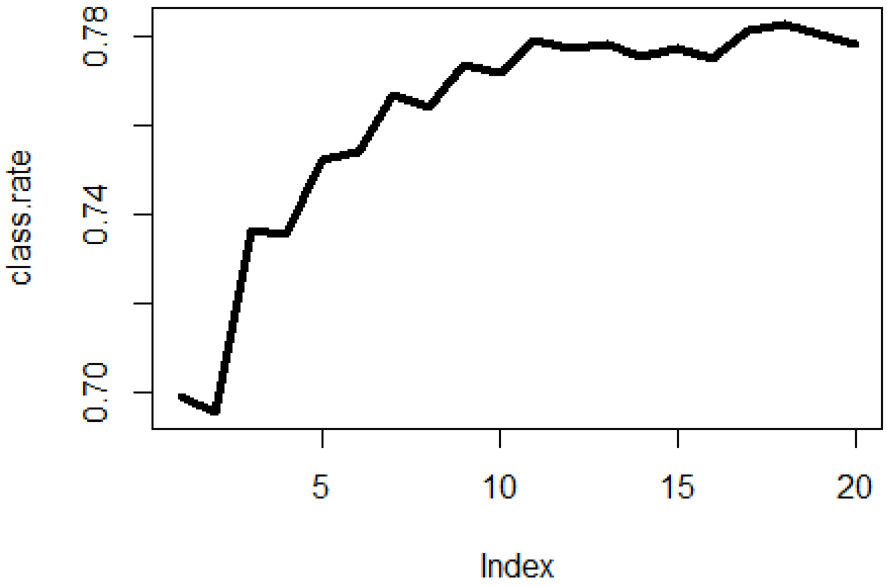
```
creditusd2 <- creditusd
creditusd2[,3:5] <- credit2[,3:5]
creditusd2 <- creditusd2[,-1]
creditusd2$defaultnm <- as.factor(creditusd2$defaultnm)
n = length(creditusd2$defaultnm)
Z = sample( n, n/10)
X = model.matrix(defaultnm~., creditusd2)
Y = creditusd2$defaultnm
c.train = creditusd2[-Z,]
c.test = creditusd2[Z,]
x.train <- X[-Z,]
x.test <- X[Z,]
y.train <- Y[-Z]
y.test <- Y[Z]

library(class)
knn.result <- knn(x.train, x.test,y.train,3)
table(y.test, knn.result)

##       knn.result
## y.test    0    1
##      0 2065  265
##      1  527  143
```

```
class.rate = rep(0,20)
for(k in 1:20){
  knn.result <- knn(x.train, x.test,y.train,k)
  class.rate[k] <- mean(y.test == knn.result)
}
which.max(class.rate)
```

```
## [1] 18
```

```
plot(class.rate, type = "line", lwd = 4)
```

```
## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to
## first character
```



```
knn.in <- class.rate[which.max(class.rate)]

knn.in

## [1] 0.783
```

Non parametric approach of KNN shows that the maximum prediction accuracy is obtained when k =19. It means that using 19 neighborhood to determine the vote of the classification, we are able to predict 79% of default and non default group correctly.

```
rf = randomForest(defaultnm ~ ., data=creditusd2, subset=-Z)

Yhat = predict(rf,creditusd2, type="class")

mean(Yhat[-Z] != creditusd2$defaultnm[-Z])

## [1] 0.006222222

cv.err = rep(0,7)

n.trees= rep(0,7)

 for (m in 1:7){

rf.m = randomForest( defaultnm ~ ., data=creditusd2[-Z,], mtry=m )

opt.trees = which.min(rf.m$err.rate)

rf.m = randomForest( defaultnm ~ ., data=creditusd2[-Z,], mtry=m, ntree=opt.trees )

Yhat = predict(rf.m,newdata=creditusd2[Z,], type="class")

pred.err = mean( (Yhat == creditusd2$defaultnm[Z])^2 )

cv.err[m] = pred.err

n.trees[m] = opt.trees

 }

which.min(cv.err)
```

```
## [1] 1

n.trees[which.min(cv.err)]

## [1] 705

rf.optimal = randomForest(defaultnm~., data = creditusd2, mtry = which.min(cv.err), ntree
= n.trees[which.min(cv.err)])
rf.optimal

##
## Call:
##  randomForest(formula = defaultnm ~ ., data = creditusd2, mtry = which.min(cv.err),
ntree = n.trees[which.min(cv.err)])
##                Type of random forest: classification
##                      Number of trees: 705
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 18.92%
## Confusion matrix:
##       0    1 class.error
## 0 22579  785   0.0335987
## 1  4891 1745   0.7370404

names(rf.optimal)

##  [1] "call"            "type"            "predicted"
##  [4] "err.rate"        "confusion"       "votes"
##  [7] "oob.times"       "classes"         "importance"
## [10] "importanceSD"    "localImportance" "proximity"
## [13] "ntree"           "mtry"            "forest"
## [16] "y"               "test"            "inbag"
## [19] "terms"

rf.optimal$confusion[2,3]

## [1] 0.7370404
```

### conclusion

```
result <- tibble(InitialLogistic = log.in,
VSLogistic = log.vs,
InitialLDA = lda.in,
VSLDA = lda.vs,
VSQDA = qda.vs,
InitialKNN = knn.in
#,randomForest = rf.optimal$confusion[2,3]
)
result

## # A tibble: 1 x 6
##   InitialLogistic VSLogistic InitialLDA VSLDA VSQDA InitialKNN
##             <dbl>      <dbl>      <dbl> <dbl> <dbl>      <dbl>
## 1           0.791      0.792      0.812 0.811 0.441      0.783
```

### repeat the method using transformation data

### method1

```
creditusd <- creditt[,c(1:12,25, 26, 28:39)]

cnumt2<- creditusd[,c(6:12,14:26)]

scnumt2 <- scale(cnumt2, scale = TRUE)

creditusd[,c(6:12,14:26)] <- scnumt2

creditusd <- creditusd %>% mutate(defaultnm = ifelse (defaultnm == "default", 1, 0))

creditusd$defaultnm <- as.numeric(creditusd$defaultnm)

#logistic Regression Initial model

creditusd <- creditusd[,-c(1:2)]

creditusd$defaultnm <- as.factor(creditusd$defaultnm)

mod1 <- glm(defaultnm~., data = creditusd, family = binomial(link="logit"))

summary(mod1)

##

## Call:

## glm(formula = defaultnm ~ ., family = binomial(link = "logit"),

##     data = creditusd)

##

## Deviance Residuals:

##     Min       1Q   Median       3Q      Max

## -3.3210  -0.6661  -0.5322  -0.2456   3.1260

##

## Coefficients: (1 not defined because of singularities)
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -13.36502   80.16027  -0.167 0.867584
## SEXmale               0.08697    0.03128   2.781 0.005424 **
## EDUCATIONGradschool  10.58518   80.15855   0.132 0.894942
## EDUCATIONUniversity  10.52498   80.15855   0.131 0.895536
## EDUCATIONHighschool  10.49696   80.15856   0.131 0.895813
## EDUCATIONOthers       9.47996   80.15877   0.118 0.905858
## MARRIAGEMarried       1.41270    0.52480   2.692 0.007105 **
## MARRIAGESingle        1.23010    0.52500   2.343 0.019127 *
## MARRIAGEOthers        1.34393    0.54200   2.480 0.013153 *
## AGE                   0.04323    0.01737   2.488 0.012830 *
## PAY_0                 0.51850    0.02004  25.867  < 2e-16 ***
## PAY_2                 0.04681    0.02652   1.765 0.077629 .
## PAY_3                 0.08890    0.03209   2.771 0.005593 **
## PAY_4                 0.05197    0.03402   1.528 0.126515
## PAY_5                 0.08884    0.03484   2.550 0.010769 *
## PAY_6                 0.09970    0.02816   3.541 0.000399 ***
## limbalusd            -0.12434    0.01963  -6.333 2.41e-10 ***
## billamt1             -0.02301    0.05655  -0.407 0.684078
## billamt2              0.05165    0.07176   0.720 0.471608
## billamt3              0.08323    0.07302   1.140 0.254363
## billamt4              0.01907    0.07684   0.248 0.804015
## billamt5              0.06393    0.05959   1.073 0.283375
## billamt6             -0.21661    0.02274  -9.524  < 2e-16 ***
## payamt1              -0.20120    0.02395  -8.402  < 2e-16 ***
## payamt2              -0.16960    0.02412  -7.032 2.04e-12 ***
## payamt3              -0.12304    0.02402  -5.123 3.00e-07 ***
## payamt4              -0.07053    0.02185  -3.227 0.001249 **
## payamt5              -0.06061    0.02007  -3.020 0.002524 **
## payamt6                    NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31705  on 29999  degrees of freedom
## Residual deviance: 27157  on 29972  degrees of freedom
## AIC: 27213
##
## Number of Fisher Scoring iterations: 11
```

```
#defaultnm~ limbalusd + SEX + MARRIAGE + AGE + PAY_0 + PAY_2 + PAY_3 +PAY_5
+PAY_6 + billamt6 + payamt1 + payamt2 + payamt3 + payamt4 + payamt5
creditusd
```

```
## # A tibble: 30,000 x 24
##    SEX    EDUCATION  MARRIAGE    AGE   PAY_0  PAY_2  PAY_3  PAY_4  PAY_5
##    <chr>  <fct>      <fct>      <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 female University Married  -1.25    1.79    1.78  -0.697 -0.667 -1.53
##  2 female University Single   -1.03   -0.875   1.78   0.139  0.189  0.235
##  3 female University Single   -0.161   0.0149  0.112  0.139  0.189  0.235
##  4 female University Married   0.164   0.0149  0.112  0.139  0.189  0.235
##  5 male   University Married   2.33   -0.875   0.112 -0.697  0.189  0.235
##  6 male   Gradschool Single    0.164   0.0149  0.112  0.139  0.189  0.235
##  7 male   Gradschool Single   -0.704   0.0149  0.112  0.139  0.189  0.235
##  8 female University Single   -1.35    0.0149 -0.724 -0.697  0.189  0.235
##  9 female Highschool Married  -0.812   0.0149  0.112  1.81   0.189  0.235
## 10 male   Highschool Single   -0.0527 -1.76   -1.56  -1.53  -1.52  -0.648
## # ... with 29,990 more rows, and 15 more variables: PAY_6 <dbl>,
## #   defaultnm <fct>, limbalusd <dbl>, billamt1 <dbl>, billamt2 <dbl>,
## #   billamt3 <dbl>, billamt4 <dbl>, billamt5 <dbl>, billamt6 <dbl>,
## #   payamt1 <dbl>, payamt2 <dbl>, payamt3 <dbl>, payamt4 <dbl>,
## #   payamt5 <dbl>, payamt6 <dbl>
```

```
n <- length(creditusd$defaultnm)
Z <- sample(n,n/10)
c.test <- creditusd[Z,]
mod1.1 <- glm(defaultnm~ limbalusd + SEX + MARRIAGE + AGE + PAY_0 + PAY_2 +
PAY_3 +PAY_5 +PAY_6 + billamt6 + payamt1 + payamt2 + payamt3 + payamt4 +
payamt5 , data = creditusd[-Z,], family = binomial(link = "logit"))
summary(mod1.1)
```

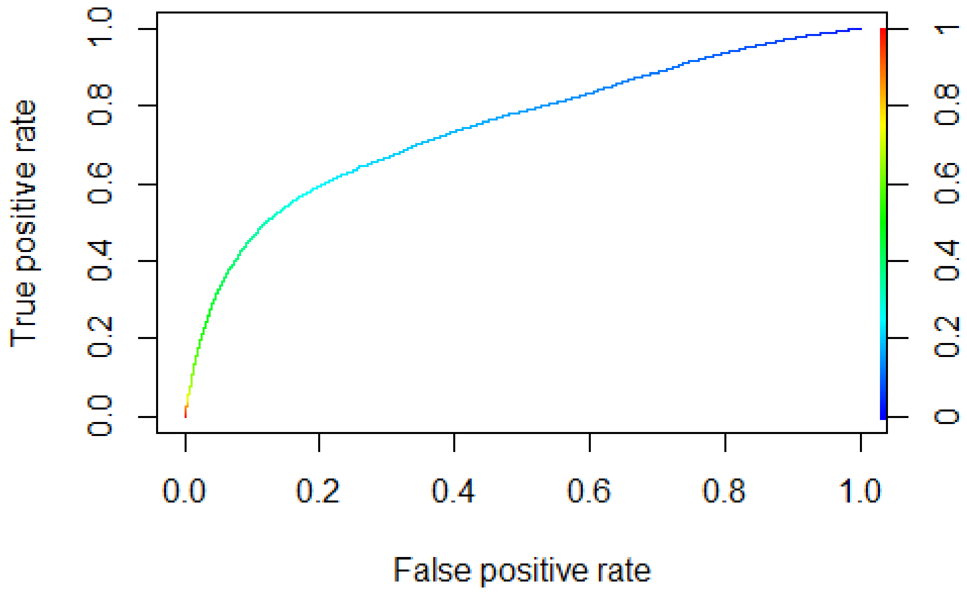
```
##
## Call:
## glm(formula = defaultnm ~ limbalusd + SEX + MARRIAGE + AGE +
##     PAY_0 + PAY_2 + PAY_3 + PAY_5 + PAY_6 + billamt6 + payamt1 +
##     payamt2 + payamt3 + payamt4 + payamt5, family = binomial(link = "logit"),
##     data = creditusd[-Z, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3277  -0.6730  -0.5245  -0.2636   3.0413
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.75546    0.53059  -5.193 2.07e-07 ***
## limbalusd        -0.06535    0.01843  -3.546 0.000392 ***
## SEXmale           0.11327    0.03278   3.455 0.000550 ***
## MARRIAGEMarried  1.34133    0.53096   2.526 0.011529 *
## MARRIAGESingle   1.15873    0.53112   2.182 0.029133 *
## MARRIAGEOthers   1.30560    0.54929   2.377 0.017460 *
## AGE               0.03101    0.01776   1.747 0.080723 .
## PAY_0             0.54151    0.02072  26.139  < 2e-16 ***
## PAY_2             0.05456    0.02716   2.009 0.044500 *
## PAY_3             0.13471    0.02947   4.570 4.87e-06 ***
## PAY_5             0.10901    0.03155   3.455 0.000551 ***
## PAY_6             0.11852    0.02917   4.063 4.85e-05 ***
## billamt6         -0.20461    0.02261  -9.048  < 2e-16 ***
## payamt1          -0.16980    0.02179  -7.794 6.51e-15 ***
## payamt2          -0.15564    0.02155  -7.221 5.16e-13 ***
## payamt3        -0.10253    0.02364  -4.337 1.44e-05 ***
## payamt4        -0.04059    0.02102  -1.932 0.053419 .
## payamt5        -0.05030    0.02087  -2.410 0.015948 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28617  on 26999  degrees of freedom
## Residual deviance: 24596  on 26982  degrees of freedom
## AIC: 24632
##
## Number of Fisher Scoring iterations: 5
```

```
prob = predict(mod1.1, data.frame(c.test))
yespred <- 1*(prob>0.5)


table(creditusd$defaultnm[Z], yespred)
```

```
##    yespred
##        0    1
##   0 2331   38
##   1  560   71
```

```
log.in <- mean( creditusd$defaultnm[Z] == yespred)
log.in
```

```
## [1] 0.8006667

attach(creditusd)


#TPR = rep(0,100)

#FPR = rep(0,100)

#for(k in 1:100){

#  fit = glm(defaultnm~ limbalusd + SEX + MARRIAGE + AGE + PAY_0 + PAY_2 + PAY_3
+PAY_5 +PAY_6 + billamt6 + payamt1 + payamt2 + payamt3 + payamt4 + payamt5 , data
= creditusd[-Z,], family = binomial(link = "logit"))

#  prob  = predict(fit, data.frame(creditusd[Z,]), type = "response")

#  Yhat = 1*(prob > k/100)

#  TPR[k] = sum(Yhat ==1 & defaultnm==1)/ sum(defaultnm==1)

#  FPR[k] = sum(Yhat ==1 & defaultnm==0)/ sum(defaultnm==0)

#}

#plot(FPR, TPR, xlab="False positive rate", ylab="True positive rate", main="ROC curve")

#any kind of threshold will gave the same result




creditusd.prob.test <- predict(mod1.1, creditusd, type ="response")[-Z]

pred <- prediction(creditusd.prob.test,creditusd$defaultnm[-Z] )

perf <- performance(pred, "tpr","fpr")

plot(perf, colorize = TRUE)
```



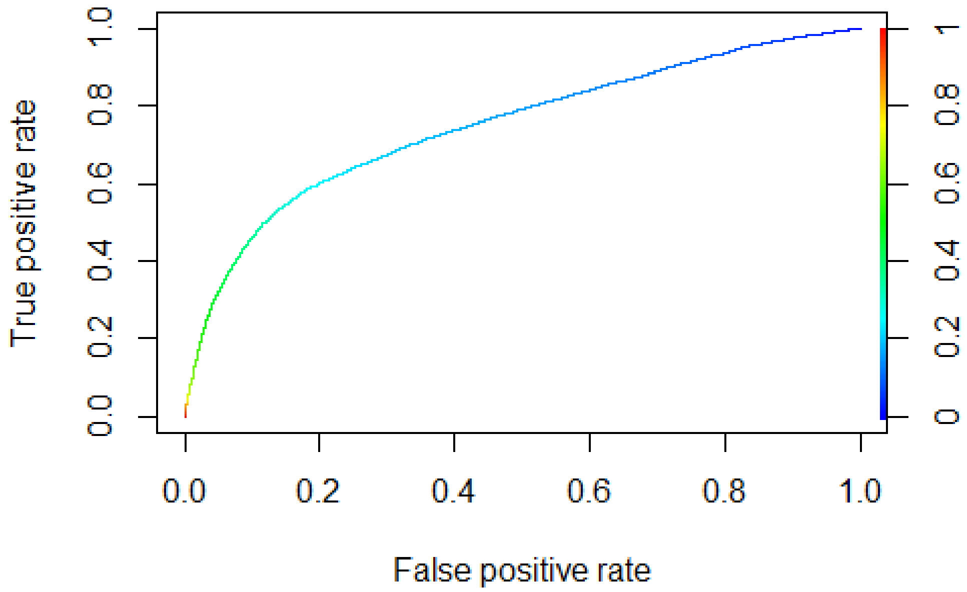
```
auc=performance(pred, "auc")

c(auc@y.name[[1]], auc@y.values[[1]])

## [1] "Area under the ROC curve" "0.745697986456059"
```

Based on the logistic regression of initial model, the prediction accuracy is 78%. However, based on ROC curve, there is no optimum probabilty that gives minimum False Positive Rate while maximizing True Positive Rate. Both of them proportionate to each other. In other case, this model cannot really predict the default of credit card. This model is no more different than flipping a coin. It gives us False positive as wel as True positive proportionately.

```
null = glm( defaultnm ~ 1, data=creditusd, family = binomial(link="logit") )

full = glm( defaultnm ~ ., data=creditusd, family = binomial(link="logit") )
```

```
#step( null, scope=list(lower=null, upper=full), direction="forward" )


#glm(formula = defaultnm ~ PAY_0 + payamt1 + PAY_4 + billamt6 +

 #   payamt4 + MARRIAGE + EDUCATION + limbalusd + billamt2 + payamt3 +

  #  PAY_6 + payamt2 + PAY_3 + SEX + billamt5 + payamt5 + PAY_5 +

   # AGE + PAY_2, family = binomial(link = "logit"), data = creditusd)


#step( full, scope=list(lower=null, upper=full), direction="backward" )


#glm(formula = defaultnm ~ SEX + EDUCATION + MARRIAGE + AGE +

 #   PAY_0 + PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6 + limbalusd +

  #  billamt3 + billamt5 + billamt6 + payamt1 + payamt2 + payamt3 +

   # payamt4 + payamt5, family = binomial(link = "logit"), data = creditusd)



modvs <- glm(formula = defaultnm ~ SEX + EDUCATION + MARRIAGE + AGE +

PAY_0 + PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6 + limbalusd +

  billamt3 + billamt5 + billamt6 + payamt1 + payamt2 + payamt3 +

 payamt4 + payamt5, family = binomial(link = "logit"), data = creditusd)

summary(modvs)

##

## Call:

## glm(formula = defaultnm ~ SEX + EDUCATION + MARRIAGE + AGE +

##     PAY_0 + PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6 + limbalusd +
##     billamt3 + billamt5 + billamt6 + payamt1 + payamt2 + payamt3 +
##     payamt4 + payamt5, family = binomial(link = "logit"), data = creditusd)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3213  -0.6661  -0.5320  -0.2462   3.1173
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -13.36383   80.12283  -0.167 0.867534
## SEXmale               0.08688    0.03126   2.779 0.005452 **
## EDUCATIONGradschool  10.58276   80.12112   0.132 0.894917
## EDUCATIONUniversity  10.52249   80.12112   0.131 0.895512
## EDUCATIONHighschool  10.49481   80.12112   0.131 0.895786
## EDUCATIONOthers       9.47862   80.12133   0.118 0.905827
## MARRIAGEMarried       1.41403    0.52479   2.694 0.007050 **
## MARRIAGESingle        1.23134    0.52499   2.345 0.019004 *
## MARRIAGEOthers        1.34534    0.54197   2.482 0.013054 *
## AGE                   0.04327    0.01737   2.490 0.012757 *
## PAY_0                 0.51886    0.02002  25.916  < 2e-16 ***
## PAY_2                 0.04627    0.02616   1.769 0.076910 .
## PAY_3                 0.09087    0.03195   2.844 0.004450 **
## PAY_4                 0.05022    0.03389   1.482 0.138328
## PAY_5                 0.08968    0.03466   2.588 0.009661 **
## PAY_6              0.09925    0.02807   3.535 0.000407 ***
## limbalusd         -0.12397    0.01955  -6.340 2.29e-10 ***
## billamt3           0.11806    0.04437   2.661 0.007798 **
## billamt5           0.07476    0.04396   1.701 0.088960 .
## billamt6          -0.21695    0.02183  -9.939  < 2e-16 ***
## payamt1           -0.19663    0.02301  -8.543  < 2e-16 ***
## payamt2           -0.17270    0.02359  -7.320 2.48e-13 ***
## payamt3           -0.12168    0.02314  -5.259 1.45e-07 ***
## payamt4           -0.07177    0.02127  -3.374 0.000740 ***
## payamt5           -0.06032    0.02006  -3.008 0.002633 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31705  on 29999  degrees of freedom
## Residual deviance: 27157  on 29975  degrees of freedom
## AIC: 27207
##
## Number of Fisher Scoring iterations: 11

#Education, pay_4, andpay_2 is not significant.
#PAY_0 + payamt2 + payamt1 + PAY_3 + payamt6 + payamt5 + billamt3 + PAY_2 +
billamt1
```

```
modvs2 <- glm(formula = defaultnm ~ SEX  + MARRIAGE + AGE +

PAY_0  + PAY_3 + PAY_5 + PAY_6 + limbalusd +

  billamt3 + billamt6 + payamt1 + payamt2 + payamt3 +

 payamt4 + payamt5, family = binomial(link = "logit"), data = creditusd[-Z,])


prob = predict(modvs2, data.frame(c.test))

yespred <- 1*(prob>0.5)

table(creditusd$defaultnm[Z], yespred)

##    yespred

##        0    1

##   0 2331   38

##   1  563   68

log.vs <- mean( creditusd$defaultnm[Z] == yespred)

log.vs

## [1] 0.7996667

#logistic regression predict 78% correctly





#TPR = rep(0,100)

#FPR = rep(0,100)

#for(k in 1:100){

#  fit = glm(formula = defaultnm ~ SEX  + MARRIAGE + AGE +
#PAY_0  + PAY_3 + PAY_5 + PAY_6 + limbalusd +

#  billamt3 + billamt6 + payamt1 + payamt2 + payamt3 +

# payamt4 + payamt5, family = binomial(link = "logit"), data = creditusd[-Z,])

 # prob  = predict(fit, data.frame(creditusd[Z,]), type = "response")

  #Yhat = 1*(prob > k/100)

  #TPR[k] = sum(Yhat ==1 & defaultnm==1)/ sum(defaultnm==1)

  #FPR[k] = sum(Yhat ==1 & defaultnm==0)/ sum(defaultnm==0)

#}

#plot(FPR, TPR, xlab="False positive rate", ylab="True positive rate", main="ROC curve")

#we can use any kind of threshold





creditusd.prob.test <- predict(modvs2, creditusd, type ="response")[-Z]

pred <- prediction(creditusd.prob.test,creditusd$defaultnm[-Z] )

perf <- performance(pred, "tpr","fpr")

plot(perf, colorize = TRUE)
```



```
auc=performance(pred, "auc")
c(auc@y.name[[1]], auc@y.values[[1]])

## [1] "Area under the ROC curve" "0.749011130083495"
```

**Linear Discriminant Analysis**

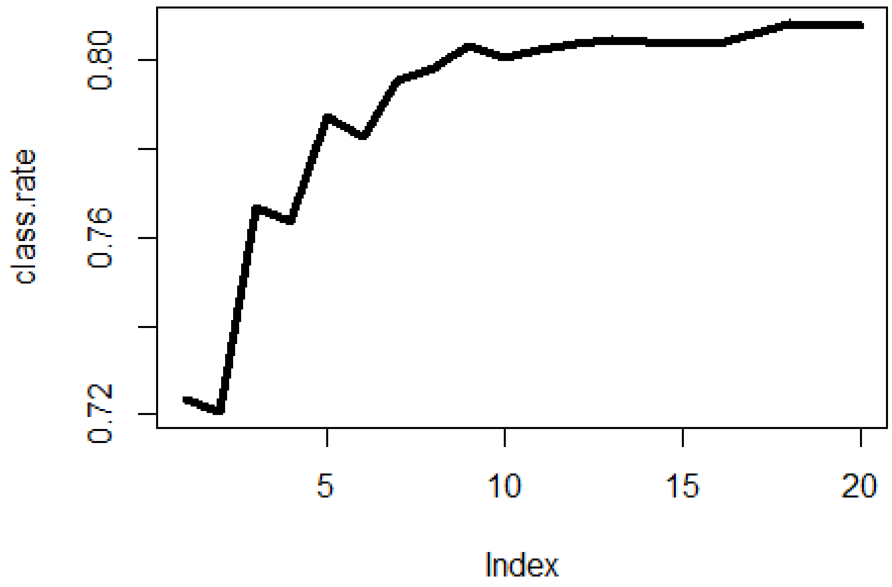
```
lda.fit <- lda(defaultnm~., data= creditusd, CV = TRUE)

## Warning in lda.default(x, grouping, ...): variables are collinear

table(creditusd$defaultnm, lda.fit$class)

## 
##         0     1
##   0 22536   828
##   1  4917  1719
```

```
lda.in <- mean(creditusd$defaultnm == lda.fit$class)

lda.in

## [1] 0.8085

lda.fitvs <- lda( defaultnm ~ SEX + EDUCATION + MARRIAGE + AGE +

 PAY_0 + PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6 + limbalusd +

   billamt3 + billamt5 + billamt6 + payamt1 + payamt2 + payamt3 +

  payamt4 + payamt5 , data = creditusd, CV = TRUE)

table(creditusd$defaultnm, lda.fitvs$class)

## 

##        0     1

##   0 22550   814

##   1  4930  1706

lda.vs <- mean(creditusd$defaultnm == lda.fitvs$class)

lda.vs

## [1] 0.8085333
```

Both Linead Discriminant Analysis based on all variables and variable selection give us 81.1% of prediction accuracy.

```
###qda for all variable can't be done due to rank deficiency in one group. It means that the data has too many multicollinearity.


qda.fitvs <- qda( defaultnm ~ limbalusd + SEX  + MARRIAGE + AGE + PAY_0 + PAY_2 +

PAY_3 + PAY_5 + billamt1 + billamt2 +    billamt5 + payamt1 + payamt2 + payamt3 +
payamt4 + payamt5 , data = creditusd, CV = TRUE)

table(defaultnm, qda.fitvs$class)

## 
## defaultnm     0     1
##         0 18557  4806
##         1  2687  3949

qda.vs<- mean(defaultnm == qda.fitvs$class,na.rm=TRUE)

qda.vs

## [1] 0.750225

creditusd2 <- creditusd

creditusd2[,3:5] <- credit2[,3:5]

creditusd2 <- creditusd2[,-1]

creditusd2$defaultnm <- as.factor(creditusd2$defaultnm)

n = length(creditusd2$defaultnm)

Z = sample( n, n/10)

X = model.matrix(defaultnm~., creditusd2)

Y = creditusd2$defaultnm

c.train = creditusd2[-Z,]

c.test = creditusd2[Z,]

x.train <- X[-Z,]

x.test <- X[Z,]

y.train <- Y[-Z]

y.test <- Y[Z]
```

```
library(class)

knn.result <- knn(x.train, x.test,y.train,3)

table(y.test, knn.result)

##       knn.result
## y.test    0    1
##      0 2063  289
##      1  410  238

class.rate = rep(0,20)

for(k in 1:20){

  knn.result <- knn(x.train, x.test,y.train,k)

  class.rate[k] <- mean(y.test == knn.result)

}

which.max(class.rate)

## [1] 18

plot(class.rate, type = "line", lwd = 4)

## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to

## first character
```



```
knn.in <- class.rate[which.max(class.rate)]

knn.in


## [1] 0.8083333
```

Non parametric approach of KNN shows that the maximum prediction accuracy is obtained when k =19. It means that using 19 neighborhood to determine the vote of the classification, we are able to predict 79% of default and non default group correctly.

```
rf = randomForest(defaultnm ~ ., data=creditusd2, subset=-Z)

Yhat = predict(rf,creditusd2, type="class")

mean(Yhat[-Z] != creditusd2$defaultnm[-Z])


## [1] 0.01022222
```

```
cv.err = rep(0,7)
n.trees= rep(0,7)
 for (m in 1:7){
rf.m = randomForest( defaultnm ~ ., data=creditusd2[-Z,], mtry=m )
opt.trees = which.min(rf.m$err.rate)
rf.m = randomForest( defaultnm ~ ., data=creditusd2[-Z,], mtry=m, ntree=opt.trees )
Yhat = predict(rf.m,newdata=creditusd2[Z,], type="class")
pred.err = mean( (Yhat == creditusd2$defaultnm[Z])^2 )
cv.err[m] = pred.err
n.trees[m] = opt.trees
 }
which.min(cv.err)

## [1] 5

n.trees[which.min(cv.err)]

## [1] 986

rf.optimal = randomForest(defaultnm~., data = creditusd2, mtry = which.min(cv.err), ntree
= n.trees[which.min(cv.err)])
rf.optimal

##
## Call:
##  randomForest(formula = defaultnm ~ ., data = creditusd2, mtry = which.min(cv.err),
ntree = n.trees[which.min(cv.err)])
##                Type of random forest: classification
##                      Number of trees: 986
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 19.82%
## Confusion matrix:
##       0    1 class.error
## 0 22065 1299  0.05559836
## 1  4647 1989  0.70027125

names(rf.optimal)

##  [1] "call"            "type"            "predicted"
##  [4] "err.rate"        "confusion"       "votes"
##  [7] "oob.times"       "classes"         "importance"
## [10] "importanceSD"    "localImportance" "proximity"
## [13] "ntree"           "mtry"            "forest"
## [16] "y"               "test"            "inbag"
## [19] "terms"

rf.optimal$confusion[2,3]

## [1] 0.7002712
```

Optimum with the random forest with m= 1 and trees of 848. error rate = 27% which means the prediction accuracy is 73.8%.

### conclusion

```
result2 <- tibble(InitialLogistic = log.in,

VSLogistic2 = log.vs,

InitialLDA2 = lda.in,

VSLDA2 = lda.vs,

VSQDA2 = qda.vs,

InitialKNN2 = knn.in

,randomForest = rf.optimal$confusion[2,3]

)

result

## # A tibble: 1 x 6

##   InitialLogistic VSLogistic InitialLDA VSLDA VSQDA InitialKNN

##             <dbl>      <dbl>      <dbl> <dbl> <dbl>      <dbl>

## 1           0.791      0.792      0.812 0.811 0.441      0.783

result2

## # A tibble: 1 x 7

##   InitialLogistic VSLogistic2 InitialLDA2 VSLDA2 VSQDA2 InitialKNN2

##             <dbl>       <dbl>       <dbl>  <dbl>  <dbl>       <dbl>

## 1           0.801       0.800       0.808  0.809  0.750       0.808

## # ... with 1 more variable: randomForest <dbl>
```