Loan Defaulter's Classification Project

1. Loading Libraries

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.2
```

```
## corrplot 0.84 loaded
```

```
library(grid)
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.2
```

```
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 3.4.2
```

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.4.2
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.4.2
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.2
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:reshape':
##
##     expand

## Loading required package: foreach

## Loaded glmnet 2.0-13

library(MASS)
library(e1071)
library(rpart)
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.4.2

library(RColorBrewer)
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.4.2

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

2. Declaring Functions

```
# Common Functions
set.seed(24287)

bindModel <- function(yLabel, xFeatures){
  # Automates the creation of feature model to be passed into an Classifier o
r Predictive Model
  return (as.formula(paste(yLabel, "~", paste(xFeatures, collapse = '+ '))))
}


# Takes the complete dataframe as an input including the label column
factorToDummy_DF_Builder <- function (dataFrameIN, numericCols, factorCols, l
abelCol){
  # Creates a design matrix by expanding the factors to a set of dummy variab
les and interaction etc.
  xNumeric <- dataFrameIN[, numericCols]
```

```
  xFactor <- dataFrameIN[, c(factorCols,labelCol)]

  factorModel <- bindModel(yLabel=labelCol, xFeatures=factorCols)
  xFactor <- model.matrix(factorModel, data=xFactor)[, -1]

# -1 is provided to exclude the intercept term from the matrix
  yLabel <- dataFrameIN[labelCol]
  return (data.frame(xNumeric, xFactor, yLabel))
}


stratifiedSampling <- function(dataIN, sample_on_col, trainPrcnt){
  trainIndices <- createDataPartition(y=dataIN[[sample_on_col]], p=trainPrcnt
, list=FALSE)
  trainData <- dataIN[trainIndices,]
  testData <- dataIN[-trainIndices,]

  stopifnot(nrow(trainData) + nrow(testData) == nrow(dataIN))
  return (list(trainData, testData))
}


# Plot and calculate the acuracy, precision and recall for different range of
cut-offs

performanceMetric <- function (cutoffRange, y, y_hat){
  y_bin <- y_hat
  actualYesIndex <- which(y==1)
  perfMetric <- matrix(0,length(cutoffRange),3)
  for (i in 1:length(cutoffRange)){
    predYesIndex <- which(y_hat>=cutoffRange[i])
    bothYesIndex <- intersect(actualYesIndex,predYesIndex)

    # Get the Binomial prediction based on cut-off value
    y_bin[predYesIndex] <- 1
    y_bin[-predYesIndex] <- 0

    # Calculate the accuracy, precision and recall
    accuracy <- length(which(y_bin == y))/length(y)
    precision <- length(bothYesIndex)/length(predYesIndex)
    recall <- length(bothYesIndex)/length(actualYesIndex)
    cbind(accuracy, precision, recall)

    perfMetric[i,] <- cbind(accuracy, precision, recall)
  }

  return (perfMetric)

}
```

```r
# Changing the datatypes
changeDataType <- function(dataIN, featureNames, type){
  if (type=='factor'){
    dataIN[featureNames] <- lapply(dataIN[featureNames], factor)
  }
  else if (type=='numeric'){
    dataIN[featureNames] <- lapply(dataIN[featureNames], as.numeric)
  }
  else{
    print ('No Type Specified! Specify a Type Factor or Numeric')
  }
  return (dataIN)
}


aicCompute <- function(fullModel, dataIN){
  glmIN <- glm(fullModel, data = dataIN)
  aic <- AIC(glmIN)
  return (aic)
}



backwardSelection <- function(features, label, dataIN){
  featuresIN <- features
  while (TRUE){
    fullModel <- bindModel(label, featuresIN)
    aic_main <- aicCompute(fullModel, dataIN)

    intermediateAIC <- c()
    for (j in (1:length(featuresIN))){
      newFeatureSet <- featuresIN[-j]
      newModel <- bindModel(label, newFeatureSet)
      aicNew <- aicCompute(newModel, dataIN)
      intermediateAIC <- c(intermediateAIC, aicNew)
    }


    badFeatureIndex <- which(intermediateAIC == min(intermediateAIC))
    featuresIN <- featuresIN[-badFeatureIndex]

    if (aic_main < min(intermediateAIC)){
      return (fullModel)
    }
  }
}
```

```
plotPerfMetric <- function(performanceDF, cutoffRange){
  p <- ggplot() +
    geom_line(data = performanceDF, aes(x = cutoffRange, y = accuracy, color
= "accuracy")) +
    geom_line(data = performanceDF, aes(x = cutoffRange, y = precision, color
= "precision")) +
    geom_line(data = performanceDF, aes(x = cutoffRange, y = recall, color =
"recall")) +
    xlab('Cutoff') +
    ylab('percent.change')
  return (p)
}
```

3. Loading Data

```
dir<- 'C:/Users/Mohit/Documents/R/Dataset/credit_card_dataset.csv'
data <- read.csv (dir,header = TRUE)
head(data)
```

```
##   ID LIMIT_BAL SEX EDUCATION MARRIAGE AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5
## 1  1     20000   2         2        1  24     2     2    -1    -1    -2
## 2  2    120000   2         2        2  26    -1     2     0     0     0
## 3  3     90000   2         2        2  34     0     0     0     0     0
## 4  4     50000   2         2        1  37     0     0     0     0     0
## 5  5     50000   1         2        1  57    -1     0    -1     0     0
## 6  6     50000   1         1        2  37     0     0     0     0     0
##   PAY_6 BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6
## 1    -2      3913      3102       689         0         0         0
## 2     2      2682      1725      2682      3272      3455      3261
## 3     0     29239     14027     13559     14331     14948     15549
## 4     0     46990     48233     49291     28314     28959     29547
## 5     0      8617      5670     35835     20940     19146     19131
## 6     0     64400     57069     57608     19394     19619     20024
##   PAY_AMT1 PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6
## 1        0      689        0        0        0        0
## 2        0     1000     1000     1000        0     2000
## 3     1518     1500     1000     1000     1000     5000
## 4     2000     2019     1200     1100     1069     1000
## 5     2000    36681    10000     9000      689      679
## 6     2500     1815      657     1000     1000      800
##   default.payment.next.month
## 1                          1
## 2                          1
## 3                          0
## 4                          0
## 5                          0
## 6                          0
```

4. Data Preprocessing

```
# Remove the ID column:
credit.data <- subset(data, select=-c(ID))
head(credit.data)
```

```
##   LIMIT_BAL SEX EDUCATION MARRIAGE AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6
## 1     20000   2         2        1  24     2     2    -1    -1    -2    -2
## 2    120000   2         2        2  26    -1     2     0     0     0     2
## 3     90000   2         2        2  34     0     0     0     0     0     0
## 4     50000   2         2        1  37     0     0     0     0     0     0
## 5     50000   1         2        1  57    -1     0    -1     0     0     0
## 6     50000   1         1        2  37     0     0     0     0     0     0
##   BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT1
## 1      3913      3102       689         0         0         0        0
## 2      2682      1725      2682      3272      3455      3261        0
## 3     29239     14027     13559     14331     14948     15549     1518
## 4     46990     48233     49291     28314     28959     29547     2000
## 5      8617      5670     35835     20940     19146     19131     2000
## 6     64400     57069     57608     19394     19619     20024     2500
##   PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 default.payment.next.month
## 1      689        0        0        0        0                          1
## 2     1000     1000     1000        0     2000                          1
## 3     1500     1000     1000     1000     5000                          0
## 4     2019     1200     1100     1069     1000                          0
## 5    36681    10000     9000      689      679                          0
## 6     1815      657     1000     1000      800                          0
```

```
dim(credit.data)
```

```
## [1] 30000    24
```

```
# Change the label column name
colnames(credit.data)[24] <- "default"


# Identifying Categorical, numerical & Logical element

uniqueCount <- function (feature){
  return (length(unlist(unique(credit.data[feature]))))
}
sapply(colnames(credit.data), FUN=uniqueCount)
```

```
## LIMIT_BAL       SEX EDUCATION  MARRIAGE       AGE     PAY_0     PAY_2
##        81         2         7         4        56        11        11
##     PAY_3     PAY_4     PAY_5     PAY_6 BILL_AMT1 BILL_AMT2 BILL_AMT3
##        11        11        10        10     22723     22346     22026
## BILL_AMT4 BILL_AMT5 BILL_AMT6  PAY_AMT1  PAY_AMT2  PAY_AMT3  PAY_AMT4
##     21548     21010     20604      7943      7899      7518      6937
##  PAY_AMT5  PAY_AMT6   default
##      6897      6939         2
```

It seems that [sex, education, marriage, age], [PAY_0,PAY_2,PAY_3,PAY_4,PAY_5,PAY_6] are nominal There's is only one logical variable "default" as it can take only two value either "Yes (True)" or "No (False)"

```
numericCols <- names(which(sapply(credit.data, is.numeric)))
nominalCols <- names(which(sapply(credit.data, is.factor)))
print (nrow(credit.data))
```

```
## [1] 30000
```

```
print (ncol(credit.data))
```

```
## [1] 24
```

```
# Convert into Proper datatypes
credit.nominalCols <- c('SEX','EDUCATION','MARRIAGE')
credit.numericCols <- setdiff(colnames(credit.data), credit.nominalCols)
credit.data <- changeDataType(credit.data, credit.nominalCols, type='factor')
credit.data <- changeDataType(credit.data, credit.numericCols, type='numeric'
)
```

```
# CAPTURE Numeric and nominal and the label Columns
credit.labelCol <- 'default'
credit.numericCols <- setdiff(names(which(sapply(credit.data, is.numeric))),
credit.labelCol)
credit.nominalCols <- names(which(sapply(credit.data, is.factor)))
```

```
str(credit.data)
```

```
## 'data.frame':    30000 obs. of  24 variables:
##  $ LIMIT_BAL: num  20000 120000 90000 50000 50000 50000 500000 100000 1400
00 20000 ...
##  $ SEX      : Factor w/ 2 levels "1","2": 2 2 2 2 1 1 1 2 2 1 ...
##  $ EDUCATION: Factor w/ 7 levels "0","1","2","3",..: 3 3 3 3 3 2 2 3 4 4 .
..
##  $ MARRIAGE : Factor w/ 4 levels "0","1","2","3": 2 3 3 2 2 3 3 3 2 3 ...
##  $ AGE      : num  24 26 34 37 57 37 29 23 28 35 ...
##  $ PAY_0    : num  2 -1 0 0 -1 0 0 0 0 -2 ...
##  $ PAY_2    : num  2 2 0 0 0 0 0 -1 0 -2 ...
##  $ PAY_3    : num  -1 0 0 0 -1 0 0 -1 2 -2 ...
##  $ PAY_4    : num  -1 0 0 0 0 0 0 0 0 -2 ...
##  $ PAY_5    : num  -2 0 0 0 0 0 0 0 0 -1 ...
##  $ PAY_6    : num  -2 2 0 0 0 0 0 -1 0 -1 ...
##  $ BILL_AMT1: num  3913 2682 29239 46990 8617 ...
##  $ BILL_AMT2: num  3102 1725 14027 48233 5670 ...
##  $ BILL_AMT3: num  689 2682 13559 49291 35835 ...
##  $ BILL_AMT4: num  0 3272 14331 28314 20940 ...
##  $ BILL_AMT5: num  0 3455 14948 28959 19146 ...
##  $ BILL_AMT6: num  0 3261 15549 29547 19131 ...
##  $ PAY_AMT1 : num  0 0 1518 2000 2000 ...
```

```
##  $ PAY_AMT2 : num  689 1000 1500 2019 36681 ...
##  $ PAY_AMT3 : num  0 1000 1000 1200 10000 657 38000 0 432 0 ...
##  $ PAY_AMT4 : num  0 1000 1000 1100 9000 ...
##  $ PAY_AMT5 : num  0 0 1000 1069 689 ...
##  $ PAY_AMT6 : num  0 2000 5000 1000 679 ...
##  $ default  : num  1 1 0 0 0 0 0 0 0 0 ...
```

```
credit.labelCol
```

```
## [1] "default"
```

```
credit.numericCols
```

```
##  [1] "LIMIT_BAL" "AGE"       "PAY_0"     "PAY_2"     "PAY_3"
##  [6] "PAY_4"     "PAY_5"     "PAY_6"     "BILL_AMT1" "BILL_AMT2"
## [11] "BILL_AMT3" "BILL_AMT4" "BILL_AMT5" "BILL_AMT6" "PAY_AMT1"
## [16] "PAY_AMT2"  "PAY_AMT3"  "PAY_AMT4"  "PAY_AMT5"  "PAY_AMT6"
```

```
credit.nominalCols
```

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

```
#Check if data is missing

which(is.na(credit.data))
```

```
## integer(0)
```

5. Exploratory Data Analysis

```
# iv. Perform all required EDA on this data set.

# ScatterPlot
set.seed(24287)
samplePrcntg <- 0.10
credit.sampleIndices <- createDataPartition(y = credit.data$default, p=sample
Prcntg, list=FALSE)
credit.sample <- credit.data[credit.sampleIndices , ]

head(credit.sample)
```

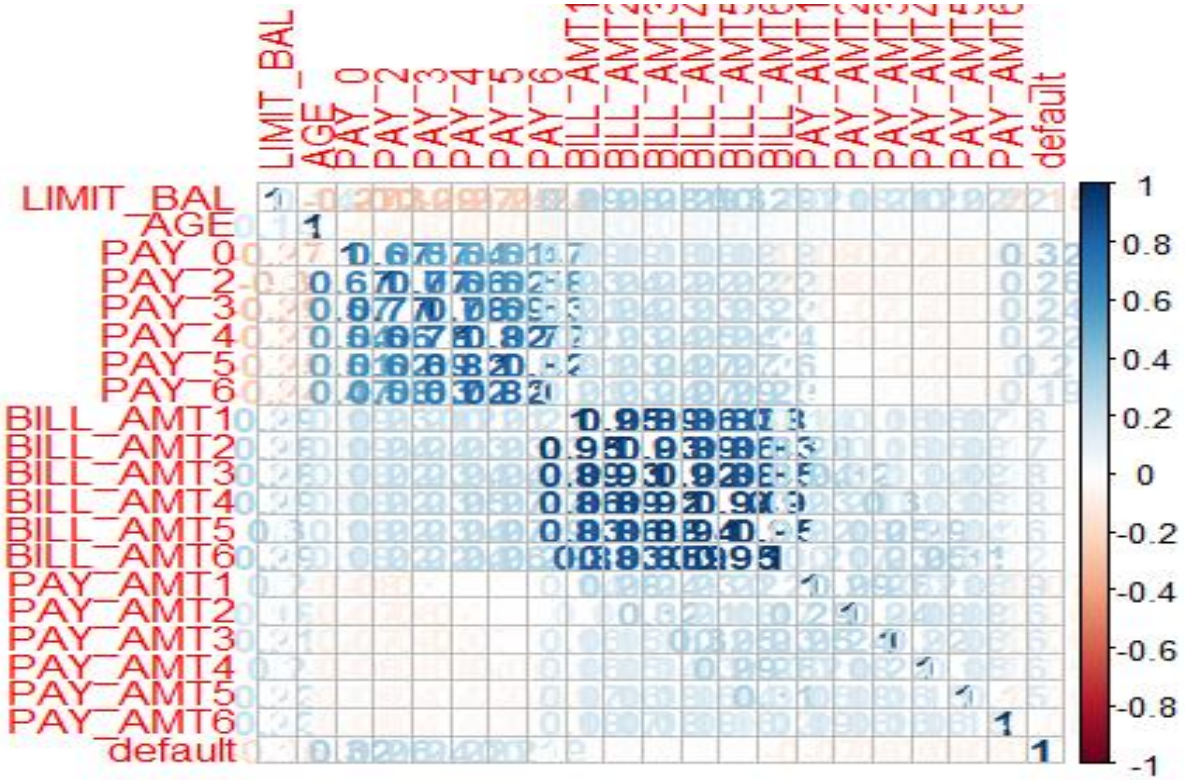
```
##    LIMIT_BAL SEX EDUCATION MARRIAGE AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5
## 9     140000   2         3        1  28     0     0     2     0     0
## 16     50000   2         3        3  23     1     2     0     0     0
## 24    450000   2         1        1  40    -2    -2    -2    -2    -2
## 26     50000   1         3        2  23     0     0     0     0     0
## 27     60000   1         1        2  27     1    -2    -1    -1    -1
## 42     70000   2         1        2  25     0     0     0     0     0
##    PAY_6 BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6
## 9      0     11285     14096     12108     12211     11793      3719
## 16     0     50614     29173     28116     28771     29531     30211
## 24    -2      5512     19420      1473       560         0         0
```

```
## 26     0     47620     41810     36023     28967     29829     30046
## 27    -1      -109      -425       259       -57       127      -189
## 42     0     67521     66999     63949     63699     64718     65970
##    PAY_AMT1 PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 default
## 9      3329        0       432      1000      1000      1000       0
## 16        0     1500      1100      1200      1300      1100       0
## 24    19428     1473       560         0         0      1128       1
## 26     1973     1426      1001      1432      1062       997       0
## 27        0     1000         0       500         0      1000       1
## 42     3000     4500      4042      2500      2800      2500       0
```

```
nrow(credit.sample)
```

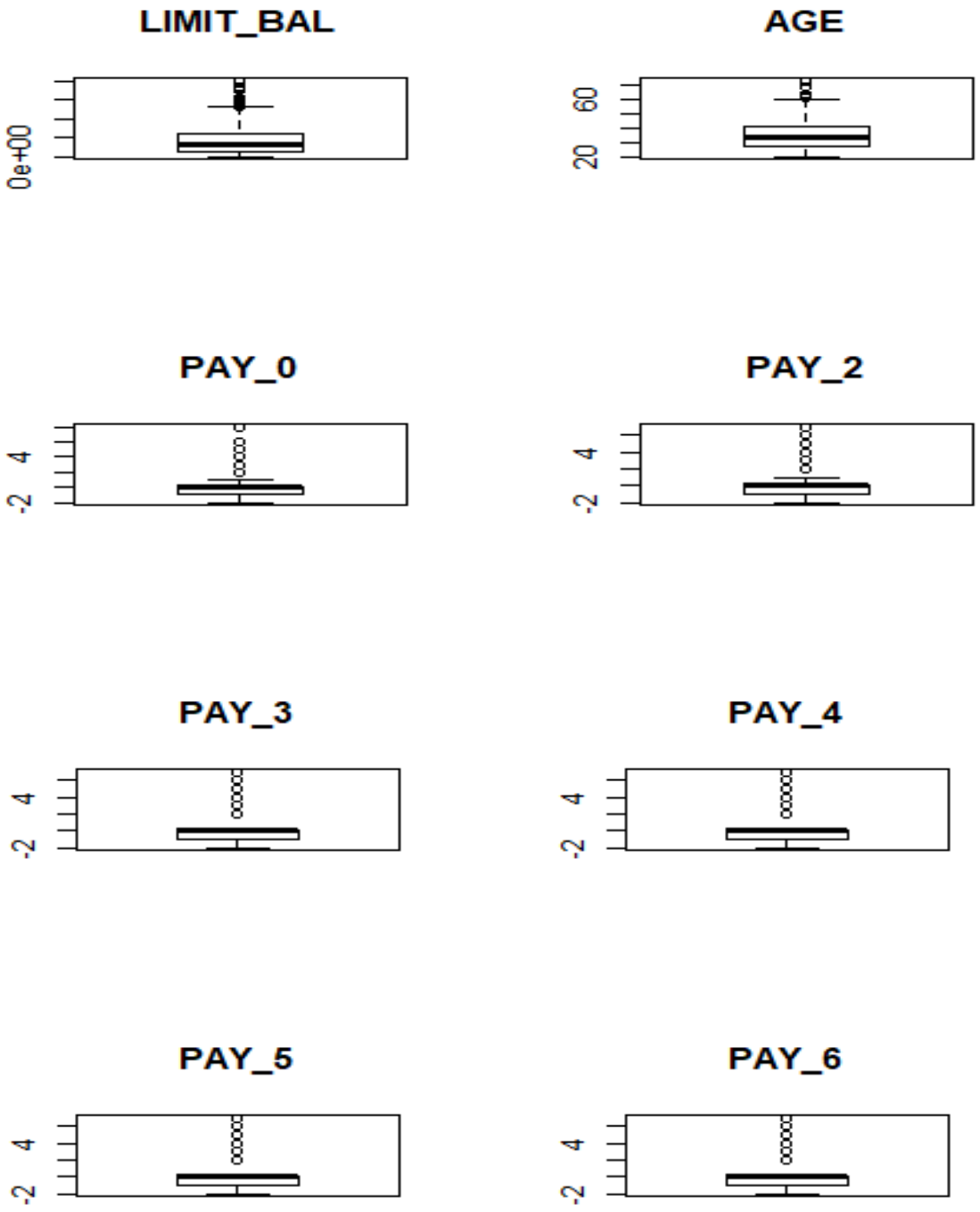
```
## [1] 3000
```

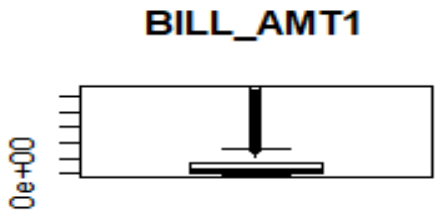
```
# Correlation Matrix
options(repr.plot.width=15, repr.plot.height=10)
credit.cor_matrix <- cor(credit.data[, c(credit.numericCols, credit.labelCol)
])
corrplot(credit.cor_matrix, method="number")
```

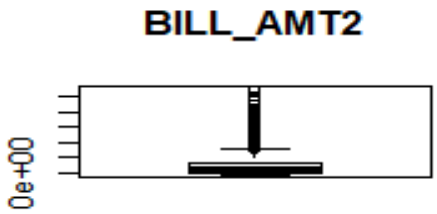


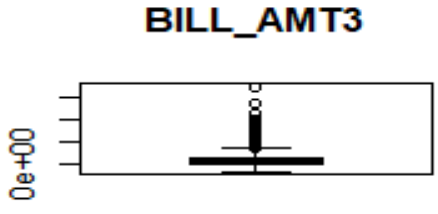

```
#Box Plot
options(repr.plot.width=10, repr.plot.height=15)
par(mfrow=c(2,2))
crearteBoxPlots <- function (column_name, dataIN){

  boxplot(dataIN[column_name], horizontal = FALSE,  main= column_name)
```

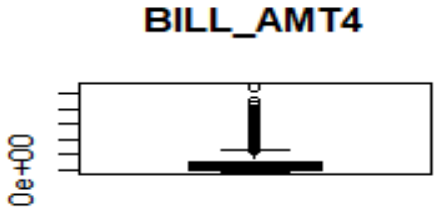
```
}
a <- sapply(credit.numericCols, FUN=crearteBoxPlots, dataIN=credit.sample)
```

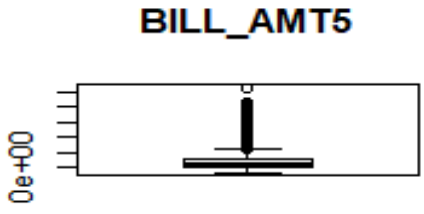


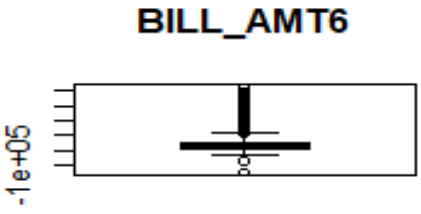

BILL_AMT1
0e+00

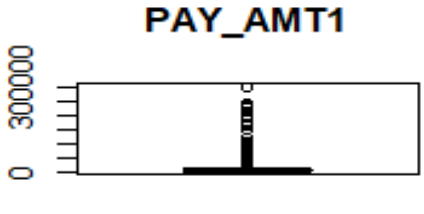

BILL_AMT2
0e+00

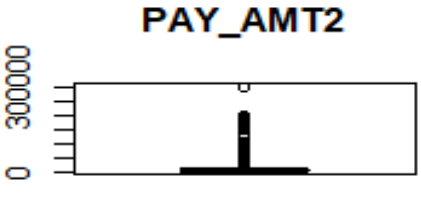

BILL_AMT3
0e+00

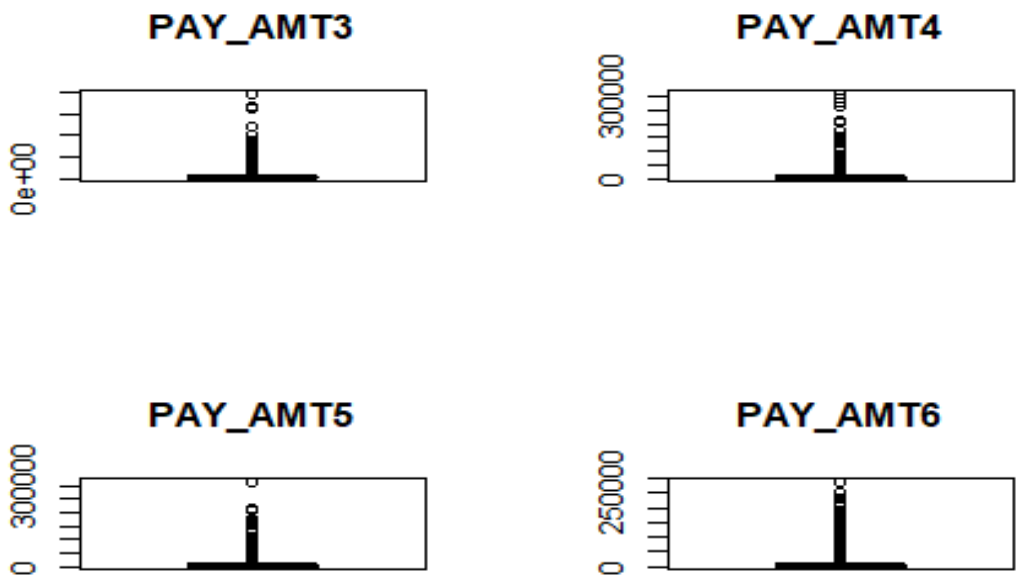

BILL_AMT4
0e+00


BILL_AMT5
0e+00

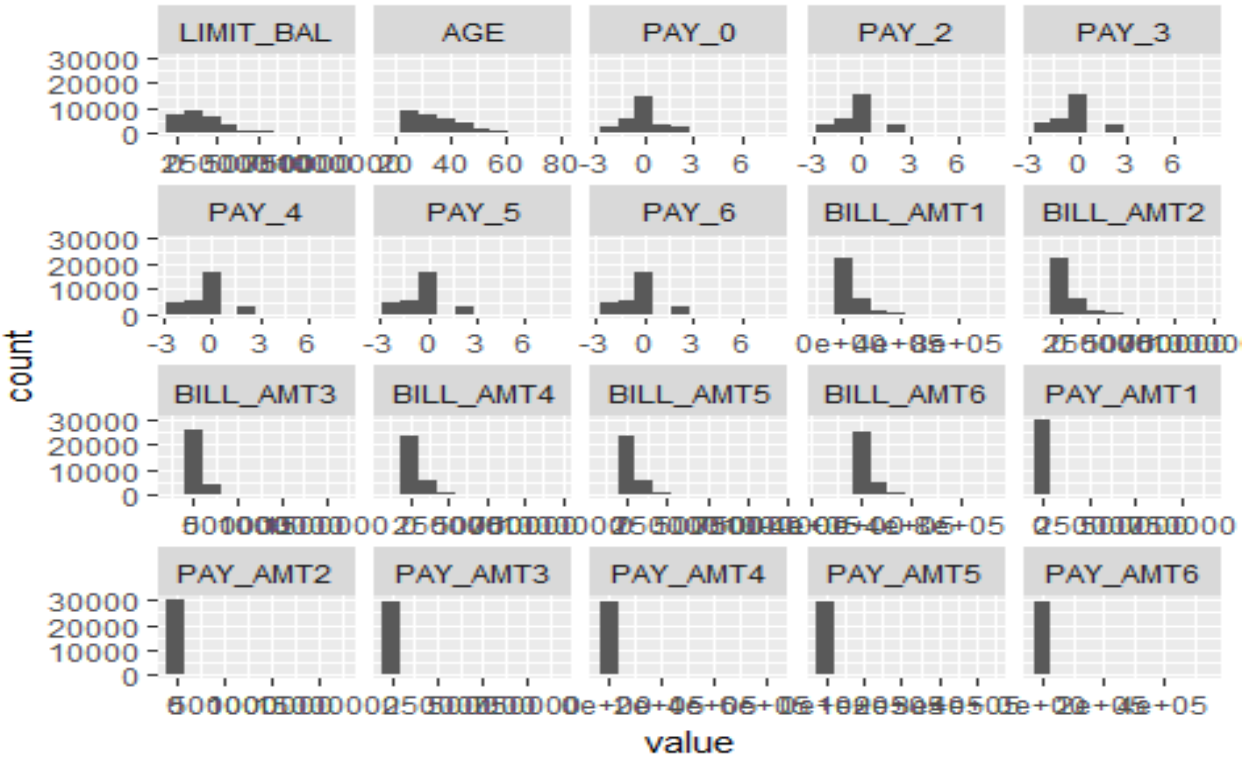

BILL_AMT6
-1e+05


PAY_AMT1
0
300000

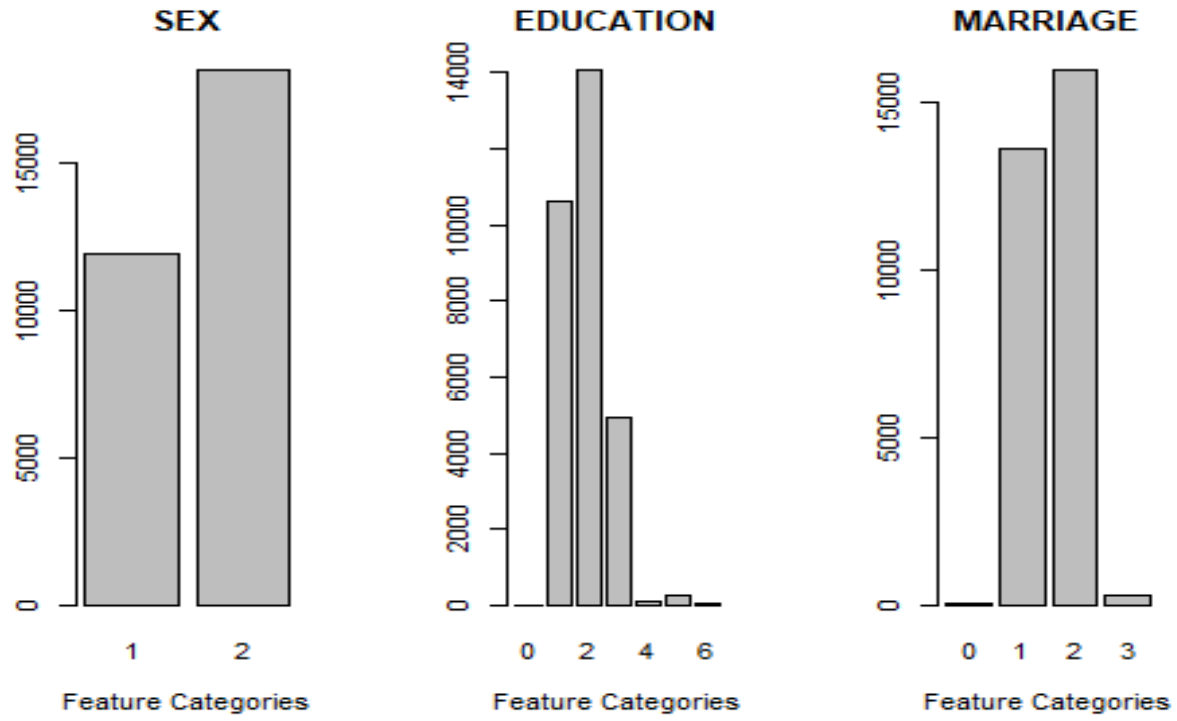

PAY_AMT2
0
300000




```
#Histogram
options(repr.plot.width=10, repr.plot.height=10)
ggplot(data = melt(credit.data[, credit.numericCols]), mapping = aes(x = valu
e)) +
geom_histogram(bins = 10) + facet_wrap(~variable, scales = 'free_x')

## Using  as id variables
```



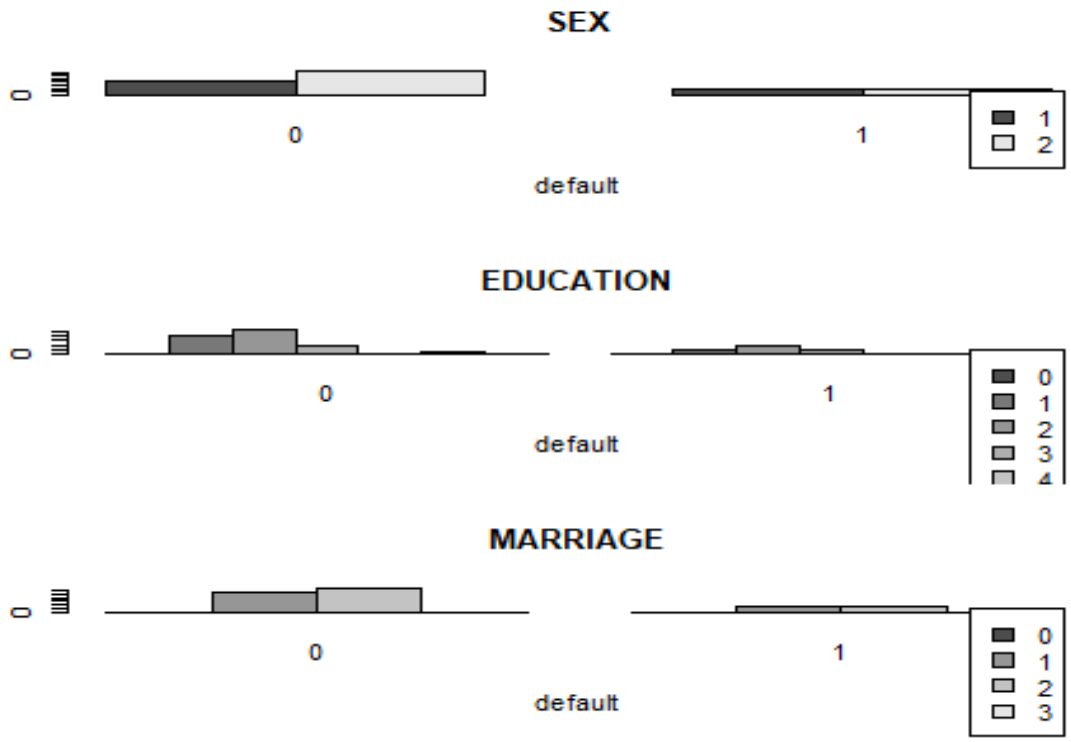

```
#Bar plot
options(repr.plot.width=10, repr.plot.height=5)
par(mfrow=c(1,3))
barPlots <- function(featureVector, dataIN){
  tab <- table(dataIN[featureVector])
  barplot(tab, main=featureVector, xlab="Feature Categories")
}
sapply(credit.nominalCols, FUN=barPlots, credit.data)
```




```
## $SEX
##      [,1]
## [1,]  0.7
## [2,]  1.9
##
## $EDUCATION
##      [,1]
## [1,]  0.7
## [2,]  1.9
## [3,]  3.1
## [4,]  4.3
## [5,]  5.5
## [6,]  6.7
## [7,]  7.9
##
## $MARRIAGE
##      [,1]
## [1,]  0.7
## [2,]  1.9
## [3,]  3.1
## [4,]  4.3
```

```
options(repr.plot.width=10, repr.plot.height=10)
par(mfrow=c(3,1))
crossTab_barplots <- function(featureVector, dataIN, labelCol){
  tab <- table(dataIN[[featureVector]], dataIN[[labelCol]])
  barplot(tab, main=featureVector,
```

```
          xlab=labelCol,
          legend = rownames(tab), beside=TRUE)
}

sapply(credit.nominalCols, FUN=crossTab_barplots, credit.data, 'default')
```




```
## $SEX
##      [,1] [,2]
## [1,]  1.5  4.5
## [2,]  2.5  5.5
##
## $EDUCATION
##      [,1] [,2]
## [1,]  1.5  9.5
## [2,]  2.5 10.5
## [3,]  3.5 11.5
## [4,]  4.5 12.5
## [5,]  5.5 13.5
## [6,]  6.5 14.5
## [7,]  7.5 15.5
##
## $MARRIAGE
##      [,1] [,2]
## [1,]  1.5  6.5
## [2,]  2.5  7.5
## [3,]  3.5  8.5
## [4,]  4.5  9.5
```

6. Scaling data

```
credit.data.scaledNumeric <- scale(credit.data[credit.numericCols])

# Check if the mean is 0 and is unit variance
stopifnot(colMeans(credit.data.scaledNumeric) != 0)
stopifnot(round(apply(credit.data.scaledNumeric, 2, sd)) == 1)

credit.data.scaled <- cbind(credit.data[credit.nominalCols], credit.data.scal
edNumeric, credit.data['default'])

head(credit.data)
```

```
##   LIMIT_BAL SEX EDUCATION MARRIAGE AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6
## 1     20000   2         2        1  24     2     2    -1    -1    -2    -2
## 2    120000   2         2        2  26    -1     2     0     0     0     2
## 3     90000   2         2        2  34     0     0     0     0     0     0
## 4     50000   2         2        1  37     0     0     0     0     0     0
## 5     50000   1         2        1  57    -1     0    -1     0     0     0
## 6     50000   1         1        2  37     0     0     0     0     0     0
##   BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT1
## 1      3913      3102       689         0         0         0        0
## 2      2682      1725      2682      3272      3455      3261        0
## 3     29239     14027     13559     14331     14948     15549     1518
## 4     46990     48233     49291     28314     28959     29547     2000
## 5      8617      5670     35835     20940     19146     19131     2000
## 6     64400     57069     57608     19394     19619     20024     2500
##   PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 default
## 1      689        0        0        0        0       1
## 2     1000     1000     1000        0     2000       1
## 3     1500     1000     1000     1000     5000       0
## 4     2019     1200     1100     1069     1000       0
## 5    36681    10000     9000      689      679       0
## 6     1815      657     1000     1000      800       0
```

```
head(credit.data.scaled)
```

```
##   SEX EDUCATION MARRIAGE  LIMIT_BAL        AGE       PAY_0     PAY_2
## 1   2         2        1 -1.1367012 -1.2459991  1.79453395 1.7823185
## 2   2         2        2 -0.3659744 -1.0290300 -0.87497656 1.7823185
## 3   2         2        2 -0.5971924 -0.1611538  0.01486028 0.1117342
## 4   2         2        1 -0.9054832  0.1642998  0.01486028 0.1117342
## 5   1         2        1 -0.9054832  2.3339904 -0.87497656 0.1117342
## 6   1         1        2 -0.9054832  0.1642998  0.01486028 0.1117342
##        PAY_3      PAY_4      PAY_5      PAY_6   BILL_AMT1   BILL_AMT2
## 1 -0.6966518 -0.6665876 -1.5300205 -1.4860160 -0.64249036 -0.64738844
## 2  0.1388625  0.1887429  0.2349126  1.9922823 -0.65920776 -0.66673546
## 3  0.1388625  0.1887429  0.2349126  0.2531332 -0.29855468 -0.49389088
## 4  0.1388625  0.1887429  0.2349126  0.2531332 -0.05749007 -0.01329247
## 5 -0.6966518  0.1887429  0.2349126  0.2531332 -0.57860845 -0.61130773
## 6  0.1388625  0.1887429  0.2349126  0.2531332  0.17894364  0.11085439
```

```
##     BILL_AMT3  BILL_AMT4  BILL_AMT5  BILL_AMT6   PAY_AMT1   PAY_AMT2
## 1 -0.66798218 -0.6724861 -0.6630475 -0.6527133 -0.3419359 -0.2270819
## 2 -0.63924364 -0.6216256 -0.6062192 -0.5979564 -0.3419359 -0.2135841
## 3 -0.48240015 -0.4497227 -0.4171807 -0.3916230 -0.2502874 -0.1918835
## 4  0.03284593 -0.2323688 -0.1867259 -0.1565763 -0.2211869 -0.1693583
## 5 -0.16118606 -0.3469914 -0.3481314 -0.3314761 -0.2211869  1.3350119
## 6  0.15277489 -0.3710227 -0.3403515 -0.3164813 -0.1909996 -0.1782122
##     PAY_AMT3   PAY_AMT4   PAY_AMT5    PAY_AMT6 default
## 1 -0.2967963 -0.3080574 -0.3141309 -0.29337717       1
## 2 -0.2400006 -0.2442256 -0.3141309 -0.18087519       1
## 3 -0.2400006 -0.2442256 -0.2486786 -0.01212223       0
## 4 -0.2286415 -0.2378424 -0.2441624 -0.23712618       0
## 5  0.2711608  0.2664292 -0.2690343 -0.25518275       0
## 6 -0.2594815 -0.2442256 -0.2486786 -0.24837638       0
```

7. Splitting Data

```
# Splitting data into Training & Testing set

# Get the Null model and the Full model
credit.dataIN <- credit.data.scaled
credit.null.model <- as.formula(paste('default', "~", 1))
credit.full.model <- bindModel(yLabel = 'default',xFeatures = c(credit.nomina
lCols, credit.numericCols))

credit.null.model

## default ~ 1

credit.full.model

## default ~ SEX + EDUCATION + MARRIAGE + LIMIT_BAL + AGE + PAY_0 +
##     PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6 + BILL_AMT1 + BILL_AMT2 +
##     BILL_AMT3 + BILL_AMT4 + BILL_AMT5 + BILL_AMT6 + PAY_AMT1 +
##     PAY_AMT2 + PAY_AMT3 + PAY_AMT4 + PAY_AMT5 + PAY_AMT6
## <environment: 0x0000000013efc1e8>

# Get the Train Test Data
dataOUT <- stratifiedSampling(dataIN=credit.dataIN, sample_on_col='default',
trainPrcnt = 0.8)

credit.trainData.sc <- dataOUT[[1]]
credit.testData.sc <- dataOUT[[2]]
nrow(credit.trainData.sc)

## [1] 24000

nrow(credit.testData.sc)

## [1] 6000

head(credit.trainData.sc)
```

```
##   SEX EDUCATION MARRIAGE  LIMIT_BAL        AGE       PAY_0     PAY_2
## 1   2         2        1 -1.1367012 -1.2459991  1.79453395 1.7823185
## 2   2         2        2 -0.3659744 -1.0290300 -0.87497656 1.7823185
## 3   2         2        2 -0.5971924 -0.1611538  0.01486028 0.1117342
## 4   2         2        1 -0.9054832  0.1642998  0.01486028 0.1117342
## 5   1         2        1 -0.9054832  2.3339904 -0.87497656 0.1117342
## 6   1         1        2 -0.9054832  0.1642998  0.01486028 0.1117342
##        PAY_3      PAY_4      PAY_5      PAY_6   BILL_AMT1   BILL_AMT2
## 1 -0.6966518 -0.6665876 -1.5300205 -1.4860160 -0.64249036 -0.64738844
## 2  0.1388625  0.1887429  0.2349126  1.9922823 -0.65920776 -0.66673546
## 3  0.1388625  0.1887429  0.2349126  0.2531332 -0.29855468 -0.49389088
## 4  0.1388625  0.1887429  0.2349126  0.2531332 -0.05749007 -0.01329247
## 5 -0.6966518  0.1887429  0.2349126  0.2531332 -0.57860845 -0.61130773
## 6  0.1388625  0.1887429  0.2349126  0.2531332  0.17894364  0.11085439
##     BILL_AMT3  BILL_AMT4  BILL_AMT5  BILL_AMT6   PAY_AMT1   PAY_AMT2
## 1 -0.66798218 -0.6724861 -0.6630475 -0.6527133 -0.3419359 -0.2270819
## 2 -0.63924364 -0.6216256 -0.6062192 -0.5979564 -0.3419359 -0.2135841
## 3 -0.48240015 -0.4497227 -0.4171807 -0.3916230 -0.2502874 -0.1918835
## 4  0.03284593 -0.2323688 -0.1867259 -0.1565763 -0.2211869 -0.1693583
## 5 -0.16118606 -0.3469914 -0.3481314 -0.3314761 -0.2211869  1.3350119
## 6  0.15277489 -0.3710227 -0.3403515 -0.3164813 -0.1909996 -0.1782122
##     PAY_AMT3   PAY_AMT4   PAY_AMT5    PAY_AMT6 default
## 1 -0.2967963 -0.3080574 -0.3141309 -0.29337717       1
## 2 -0.2400006 -0.2442256 -0.3141309 -0.18087519       1
## 3 -0.2400006 -0.2442256 -0.2486786 -0.01212223       0
## 4 -0.2286415 -0.2378424 -0.2441624 -0.23712618       0
## 5  0.2711608  0.2664292 -0.2690343 -0.25518275       0
## 6 -0.2594815 -0.2442256 -0.2486786 -0.24837638       0
```

```
# Train and Test for the Dummy variable:

credit.data.dummy <- factorToDummy_DF_Builder(dataFrameIN = credit.data.scale
d,
                                              numericCols = credit.numericCol
s,
                                              factorCols = credit.nominalCols
,
                                              labelCol = credit.labelCol)

# credit.data.dummy
dataOUT <- stratifiedSampling(dataIN = credit.data.dummy, sample_on_col = cre
dit.labelCol, trainPrcnt = 0.8)
credit.trainData.dummy <- dataOUT[[1]]
credit.testData.dummy  <- dataOUT[[2]]

nrow(credit.trainData.dummy)

## [1] 24000

nrow(credit.testData.dummy)
```

```
## [1] 6000
```

```
head(credit.testData.dummy)
```

```
##     LIMIT_BAL        AGE       PAY_0      PAY_2      PAY_3      PAY_4
## 1  -1.1367012 -1.2459991  1.79453395  1.7823185 -0.6966518 -0.6665876
## 6  -0.9054832  0.1642998  0.01486028  0.1117342  0.1388625  0.1887429
## 12  0.7130431  1.6830833 -0.87497656 -0.7235579 -0.6966518 -0.6665876
## 13  3.5647323  0.5982379 -0.87497656  0.1117342 -0.6966518 -0.6665876
## 24  2.1774240  0.4897534 -1.76481340 -1.5588500 -1.5321662 -1.5219182
## 25 -0.5971924 -1.3544836  0.01486028  0.1117342  0.1388625 -0.6665876
##         PAY_5      PAY_6  BILL_AMT1  BILL_AMT2  BILL_AMT3  BILL_AMT4
## 1  -1.5300205 -1.4860160 -0.6424904 -0.6473884 -0.6679822 -0.6724861
## 6   0.2349126  0.2531332  0.1789436  0.1108544  0.1527749 -0.3710227
## 12 -0.6475540  1.9922823 -0.5291217 -0.3865058 -0.5342103 -0.5400965
## 13 -0.6475540 -0.6164414 -0.5308057 -0.5996461 -0.5841891 -0.5714490
## 24 -1.5300205 -1.4860160 -0.6207754 -0.4181186 -0.6566771 -0.6637813
## 25  0.2349126  0.2531332 -0.6312051 -0.5916376 -0.6779174 -0.5885787
##     BILL_AMT5  BILL_AMT6     PAY_AMT1    PAY_AMT2     PAY_AMT3   PAY_AMT4
## 1  -0.6630475 -0.6527133 -0.341935920 -0.22708185 -0.296796327 -0.3080574
## 6  -0.3403515 -0.3164813 -0.190999635 -0.17821217 -0.259481541 -0.2442256
## 12 -0.2964678 -0.4232078  0.975315225  0.17555051  0.190681311  1.1154567
## 13 -0.5561346 -0.6045219 -0.281561406  0.02512216  0.072375833  0.1068496
## 24 -0.6630475 -0.6527133  0.831020136 -0.19305536 -0.264990726 -0.3080574
## 25 -0.5584373 -0.5134786  0.005640157 -0.25698524  0.009786953 -0.2314592
##      PAY_AMT5    PAY_AMT6 SEX2 EDUCATION1 EDUCATION2 EDUCATION3 EDUCATION4
## 1  -0.3141309 -0.29337717    1          0          1          0          0
## 6  -0.2486786 -0.24837638    0          1          0          0          0
## 12 -0.3141309 -0.08862357    1          1          0          0          0
## 13 -0.1262828 -0.29337717    1          0          1          0          0
## 24 -0.3141309 -0.22992605    1          1          0          0          0
## 25 -0.1802810 -0.18087519    0          1          0          0          0
##    EDUCATION5 EDUCATION6 MARRIAGE1 MARRIAGE2 MARRIAGE3 default
## 1           0          0         1         0         0       1
## 6           0          0         0         1         0       0
## 12          0          0         0         1         0       0
## 13          0          0         0         1         0       0
## 24          0          0         1         0         0       1
## 25          0          0         0         1         0       0
```

8. Fitting GLM Model

```
# Fitting Model GLM

credit.glm.null <- glm(formula=credit.null.model, family=binomial(logit), dat
a=credit.trainData.sc)
summary(credit.glm.null)
```

```
##
## Call:
## glm(formula = credit.null.model, family = binomial(logit), data = credit.t
```

```
rainData.sc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7101  -0.7101  -0.7101  -0.7101   1.7328
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.24923    0.01551  -80.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25463  on 23999  degrees of freedom
## Residual deviance: 25463  on 23999  degrees of freedom
## AIC: 25465
##
## Number of Fisher Scoring iterations: 4
```

```
credit.glm.full <- glm(formula=credit.full.model, family=binomial(logit), dat
a=credit.trainData.sc)
summary(credit.glm.full)
```

```
##
## Call:
## glm(formula = credit.full.model, family = binomial(logit), data = credit.t
rainData.sc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1574  -0.7040  -0.5434  -0.2819   3.5997
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.775020  89.321246  -0.154  0.87744
## SEX2         -0.111025   0.034294  -3.237  0.00121 **
## EDUCATION1   10.788936  89.318718   0.121  0.90386
## EDUCATION2   10.720409  89.318719   0.120  0.90446
## EDUCATION3   10.683526  89.318725   0.120  0.90479
## EDUCATION4    9.584683  89.319940   0.107  0.91455
## EDUCATION5    9.411208  89.319149   0.105  0.91609
## EDUCATION6   10.228297  89.320089   0.115  0.90883
## MARRIAGE1     1.758907   0.672454   2.616  0.00891 **
## MARRIAGE2     1.575955   0.672620   2.343  0.01913 *
## MARRIAGE3     1.650761   0.688656   2.397  0.01653 *
## LIMIT_BAL    -0.098612   0.022940  -4.299 1.72e-05 ***
## AGE           0.044183   0.019141   2.308  0.02098 *
## PAY_0         0.662543   0.022232  29.801  < 2e-16 ***
## PAY_2         0.110392   0.026926   4.100 4.13e-05 ***
```

```
## PAY_3         0.084403   0.030234   2.792  0.00524 **
## PAY_4         0.004823   0.033115   0.146  0.88420
## PAY_5         0.025777   0.034312   0.751  0.45250
## PAY_6         0.028238   0.028572   0.988  0.32299
## BILL_AMT1    -0.376028   0.092718  -4.056 5.00e-05 ***
## BILL_AMT2     0.112604   0.119420   0.943  0.34572
## BILL_AMT3     0.161874   0.099436   1.628  0.10354
## BILL_AMT4    -0.025791   0.093983  -0.274  0.78376
## BILL_AMT5     0.013308   0.100991   0.132  0.89517
## BILL_AMT6     0.019713   0.078511   0.251  0.80175
## PAY_AMT1     -0.189869   0.040392  -4.701 2.59e-06 ***
## PAY_AMT2     -0.222374   0.052848  -4.208 2.58e-05 ***
## PAY_AMT3     -0.050922   0.035126  -1.450  0.14715
## PAY_AMT4     -0.057020   0.030238  -1.886  0.05933 .
## PAY_AMT5     -0.061933   0.032222  -1.922  0.05460 .
## PAY_AMT6     -0.015266   0.024257  -0.629  0.52912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25463  on 23999  degrees of freedom
## Residual deviance: 22313  on 23969  degrees of freedom
## AIC: 22375
##
## Number of Fisher Scoring iterations: 11
```

```
# Predict for the Full model
credit.testData.sc$defaultPred <- predict(credit.glm.full, newdata=credit.tes
tData.sc, type="response")
```

```
# Range for cuttoff
cutoffRange <- seq(.01,.99,length=1000)
perfMatrix <- performanceMetric(cutoffRange, credit.testData.sc$default, cred
it.testData.sc$defaultPred)
perfDF <- data.frame(perfMatrix)
names(perfDF) <- c('accuracy', 'precision', 'recall')
head(perfDF)
```

```
##    accuracy precision recall
## 1 0.2185000 0.2154927      1
## 2 0.2193333 0.2156731      1
## 3 0.2198333 0.2157815      1
## 4 0.2203333 0.2158900      1
## 5 0.2206667 0.2159624      1
## 6 0.2218333 0.2162162      1
```

```
# Plot Accuracy, precision and recall
par(mfrow=c(1,1))
options(repr.plot.width=6, repr.plot.height=4)
```

```
p <- plotPerfMetric(perfDF, cutoffRange)
p
```

9. GLM With Forward selection

```
# GLM with Forward Selection

credit.glm.forward = step(credit.glm.null,scope=list(lower=credit.null.model,
upper=formula(credit.full.model)), direction="forward")

## Start:  AIC=25464.59
## default ~ 1

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

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

##             Df Deviance   AIC
## + PAY_0      1    22864 22868
## + PAY_2      1    23821 23825
## + PAY_3      1    24178 24182
## + PAY_4      1    24402 24406
## + PAY_5      1    24499 24503
## + PAY_6      1    24628 24632
## + LIMIT_BAL  1    24816 24820
## + PAY_AMT1   1    25206 25210
## + PAY_AMT2   1    25208 25212
## + PAY_AMT3   1    25302 25306
```

```
## + EDUCATION  6    25311 25325
## + PAY_AMT4   1    25336 25340
## + PAY_AMT5   1    25337 25341
## + PAY_AMT6   1    25376 25380
## + SEX        1    25429 25433
## + MARRIAGE   3    25434 25442
## + BILL_AMT1  1    25454 25458
## + BILL_AMT2  1    25458 25462
## + BILL_AMT3  1    25458 25462
## + BILL_AMT4  1    25460 25464
## <none>             25463 25465
## + AGE        1    25461 25465
## + BILL_AMT5  1    25461 25465
## + BILL_AMT6  1    25462 25466
##
## Step:  AIC=22868.39
## default ~ PAY_0

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

##             Df Deviance   AIC
## + LIMIT_BAL  1    22695 22701
## + PAY_3      1    22747 22753
## + PAY_2      1    22752 22758
## + PAY_AMT1   1    22759 22765
## + PAY_AMT2   1    22760 22766
## + BILL_AMT1  1    22763 22769
## + BILL_AMT2  1    22774 22780
## + BILL_AMT3  1    22779 22785
## + BILL_AMT4  1    22786 22792
## + PAY_4      1    22786 22792
## + BILL_AMT5  1    22791 22797
## + PAY_5      1    22795 22801
## + BILL_AMT6  1    22796 22802
## + PAY_6      1    22805 22811
## + PAY_AMT3   1    22807 22813
## + PAY_AMT5   1    22807 22813
## + EDUCATION  6    22798 22814
## + PAY_AMT4   1    22813 22819
## + PAY_AMT6   1    22828 22834
## + MARRIAGE   3    22832 22842
## + SEX        1    22851 22857
## + AGE        1    22855 22861
## <none>             22864 22868
##
## Step:  AIC=22701.18
## default ~ PAY_0 + LIMIT_BAL
##
##             Df Deviance   AIC
## + PAY_3      1    22622 22630
```

```
## + PAY_2      1    22625 22633
## + PAY_AMT1   1    22641 22649
## + PAY_AMT2   1    22643 22651
## + EDUCATION  6    22638 22656
## + PAY_4      1    22650 22658
## + MARRIAGE   3    22648 22660
## + PAY_5      1    22655 22663
## + PAY_6      1    22661 22669
## + BILL_AMT1  1    22664 22672
## + BILL_AMT2  1    22669 22677
## + BILL_AMT3  1    22673 22681
## + AGE        1    22673 22681
## + PAY_AMT3   1    22673 22681
## + PAY_AMT5   1    22673 22681
## + PAY_AMT4   1    22676 22684
## + BILL_AMT4  1    22677 22685
## + BILL_AMT5  1    22680 22688
## + BILL_AMT6  1    22682 22690
## + SEX        1    22684 22692
## + PAY_AMT6   1    22685 22693
## <none>            22695 22701
##
## Step:  AIC=22629.94
## default ~ PAY_0 + LIMIT_BAL + PAY_3
##
##             Df Deviance   AIC
## + PAY_AMT1   1    22552 22562
## + BILL_AMT1  1    22565 22575
## + BILL_AMT2  1    22567 22577
## + PAY_AMT2   1    22572 22582
## + BILL_AMT3  1    22573 22583
## + MARRIAGE   3    22574 22588
## + EDUCATION  6    22568 22588
## + BILL_AMT4  1    22579 22589
## + BILL_AMT5  1    22584 22594
## + BILL_AMT6  1    22588 22598
## + PAY_AMT5   1    22597 22607
## + AGE        1    22599 22609
## + PAY_AMT3   1    22599 22609
## + PAY_AMT4   1    22600 22610
## + PAY_2      1    22607 22617
## + PAY_AMT6   1    22610 22620
## + SEX        1    22613 22623
## + PAY_5      1    22619 22629
## + PAY_6      1    22619 22629
## <none>            22622 22630
## + PAY_4      1    22621 22631
##
## Step:  AIC=22561.5
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1
```

```
##
##             Df Deviance   AIC
## + MARRIAGE   3    22505 22521
## + EDUCATION  6    22499 22521
## + BILL_AMT1  1    22511 22523
## + PAY_AMT2   1    22521 22533
## + BILL_AMT2  1    22524 22536
## + BILL_AMT3  1    22526 22538
## + AGE        1    22529 22541
## + BILL_AMT4  1    22529 22541
## + BILL_AMT5  1    22531 22543
## + BILL_AMT6  1    22532 22544
## + PAY_AMT5   1    22536 22548
## + PAY_AMT4   1    22538 22550
## + PAY_AMT3   1    22539 22551
## + SEX        1    22542 22554
## + PAY_2      1    22544 22556
## + PAY_AMT6   1    22545 22557
## + PAY_6      1    22548 22560
## + PAY_5      1    22548 22560
## <none>            22552 22562
## + PAY_4      1    22550 22562
##
## Step:  AIC=22520.92
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE
##
##             Df Deviance   AIC
## + EDUCATION  6    22448 22476
## + BILL_AMT1  1    22463 22481
## + PAY_AMT2   1    22475 22493
## + BILL_AMT2  1    22477 22495
## + BILL_AMT3  1    22479 22497
## + BILL_AMT4  1    22482 22500
## + BILL_AMT5  1    22485 22503
## + BILL_AMT6  1    22486 22504
## + PAY_AMT5   1    22490 22508
## + PAY_AMT4   1    22492 22510
## + PAY_AMT3   1    22493 22511
## + SEX        1    22494 22512
## + PAY_2      1    22497 22515
## + PAY_AMT6   1    22499 22517
## + AGE        1    22499 22517
## + PAY_6      1    22501 22519
## + PAY_5      1    22501 22519
## <none>            22505 22521
## + PAY_4      1    22504 22522
##
## Step:  AIC=22476.02
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION
##
```

```
##             Df Deviance   AIC
## + BILL_AMT1  1    22411 22441
## + PAY_AMT2   1    22419 22449
## + BILL_AMT2  1    22424 22454
## + BILL_AMT3  1    22425 22455
## + BILL_AMT4  1    22428 22458
## + BILL_AMT5  1    22430 22460
## + BILL_AMT6  1    22430 22460
## + PAY_AMT5   1    22433 22463
## + PAY_AMT4   1    22436 22466
## + PAY_AMT3   1    22437 22467
## + SEX        1    22437 22467
## + PAY_2      1    22440 22470
## + AGE        1    22441 22471
## + PAY_AMT6   1    22443 22473
## + PAY_5      1    22445 22475
## + PAY_6      1    22445 22475
## <none>            22448 22476
## + PAY_4      1    22447 22477
##
## Step:  AIC=22441.23
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1
##
##             Df Deviance   AIC
## + PAY_AMT2   1    22389 22421
## + PAY_2      1    22396 22428
## + BILL_AMT2  1    22398 22430
## + SEX        1    22399 22431
## + PAY_AMT5   1    22402 22434
## + PAY_AMT4   1    22403 22435
## + AGE        1    22403 22435
## + PAY_AMT3   1    22404 22436
## + PAY_5      1    22405 22437
## + PAY_6      1    22405 22437
## + PAY_4      1    22409 22441
## + PAY_AMT6   1    22409 22441
## + BILL_AMT3  1    22409 22441
## <none>            22411 22441
## + BILL_AMT4  1    22410 22442
## + BILL_AMT5  1    22410 22442
## + BILL_AMT6  1    22411 22443
##
## Step:  AIC=22420.96
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2
##
##             Df Deviance   AIC
## + PAY_2      1    22372 22406
## + BILL_AMT3  1    22372 22406
```

```
## + SEX        1    22377 22411
## + BILL_AMT2  1    22378 22412
## + PAY_5      1    22380 22414
## + PAY_6      1    22381 22415
## + AGE        1    22381 22415
## + PAY_AMT5   1    22382 22416
## + PAY_AMT4   1    22382 22416
## + BILL_AMT4  1    22383 22417
## + PAY_4      1    22384 22418
## + PAY_AMT3   1    22384 22418
## + BILL_AMT5  1    22386 22420
## <none>            22389 22421
## + BILL_AMT6  1    22387 22421
## + PAY_AMT6   1    22388 22422
##
## Step:  AIC=22405.58
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2 + PAY_2
##
##             Df Deviance   AIC
## + BILL_AMT3  1    22353 22389
## + BILL_AMT2  1    22360 22396
## + SEX        1    22360 22396
## + AGE        1    22364 22400
## + PAY_AMT5   1    22364 22400
## + PAY_AMT4   1    22364 22400
## + PAY_5      1    22365 22401
## + BILL_AMT4  1    22365 22401
## + PAY_6      1    22366 22402
## + PAY_AMT3   1    22366 22402
## + PAY_4      1    22367 22403
## + BILL_AMT5  1    22368 22404
## <none>            22372 22406
## + BILL_AMT6  1    22370 22406
## + PAY_AMT6   1    22370 22406
##
## Step:  AIC=22389.28
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2 + PAY_2 + BILL_AMT3
##
##             Df Deviance   AIC
## + SEX        1    22341 22379
## + AGE        1    22346 22384
## + PAY_AMT5   1    22346 22384
## + PAY_AMT4   1    22347 22385
## + PAY_5      1    22349 22387
## + PAY_6      1    22349 22387
## + PAY_AMT3   1    22350 22388
## + PAY_4      1    22351 22389
## <none>            22353 22389
```

```
## + PAY_AMT6   1    22352 22390
## + BILL_AMT2  1    22352 22390
## + BILL_AMT6  1    22353 22391
## + BILL_AMT5  1    22353 22391
## + BILL_AMT4  1    22353 22391
##
## Step:  AIC=22378.77
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2 + PAY_2 + BILL_AMT3 + SEX
##
##             Df Deviance   AIC
## + PAY_AMT5   1    22334 22374
## + PAY_AMT4   1    22335 22375
## + AGE        1    22335 22375
## + PAY_5      1    22336 22376
## + PAY_6      1    22337 22377
## + PAY_AMT3   1    22337 22377
## + PAY_4      1    22338 22378
## <none>             22341 22379
## + PAY_AMT6   1    22340 22380
## + BILL_AMT2  1    22340 22380
## + BILL_AMT6  1    22340 22380
## + BILL_AMT5  1    22341 22381
## + BILL_AMT4  1    22341 22381
##
## Step:  AIC=22373.56
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2 + PAY_2 + BILL_AMT3 + SEX + PAY_AMT5
##
##             Df Deviance   AIC
## + AGE        1    22328 22370
## + PAY_AMT4   1    22329 22371
## + PAY_5      1    22329 22371
## + PAY_6      1    22330 22372
## + PAY_AMT3   1    22331 22373
## + PAY_4      1    22331 22373
## <none>             22334 22374
## + PAY_AMT6   1    22333 22375
## + BILL_AMT2  1    22333 22375
## + BILL_AMT5  1    22333 22375
## + BILL_AMT4  1    22334 22376
## + BILL_AMT6  1    22334 22376
##
## Step:  AIC=22370.15
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2 + PAY_2 + BILL_AMT3 + SEX + PAY_AMT5 +
##     AGE
##
##             Df Deviance   AIC
## + PAY_AMT4   1    22323 22367
```

```
## + PAY_5      1    22324 22368
## + PAY_6      1    22324 22368
## + PAY_AMT3   1    22325 22369
## + PAY_4      1    22326 22370
## <none>            22328 22370
## + PAY_AMT6   1    22327 22371
## + BILL_AMT2  1    22327 22371
## + BILL_AMT5  1    22328 22372
## + BILL_AMT4  1    22328 22372
## + BILL_AMT6  1    22328 22372
##
## Step:  AIC=22367.24
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2 + PAY_2 + BILL_AMT3 + SEX + PAY_AMT5 +
##     AGE + PAY_AMT4
##
##             Df Deviance   AIC
## + PAY_6      1    22319 22365
## + PAY_5      1    22319 22365
## + PAY_AMT3   1    22321 22367
## + PAY_4      1    22321 22367
## <none>            22323 22367
## + BILL_AMT2  1    22323 22369
## + PAY_AMT6   1    22323 22369
## + BILL_AMT6  1    22323 22369
## + BILL_AMT5  1    22323 22369
## + BILL_AMT4  1    22323 22369
##
## Step:  AIC=22364.66
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2 + PAY_2 + BILL_AMT3 + SEX + PAY_AMT5 +
##     AGE + PAY_AMT4 + PAY_6
##
##             Df Deviance   AIC
## + PAY_AMT3   1    22316 22364
## <none>            22319 22365
## + BILL_AMT2  1    22318 22366
## + PAY_AMT6   1    22318 22366
## + PAY_5      1    22318 22366
## + PAY_4      1    22318 22366
## + BILL_AMT4  1    22318 22366
## + BILL_AMT6  1    22319 22367
## + BILL_AMT5  1    22319 22367
##
## Step:  AIC=22363.68
## default ~ PAY_0 + LIMIT_BAL + PAY_3 + PAY_AMT1 + MARRIAGE + EDUCATION +
##     BILL_AMT1 + PAY_AMT2 + PAY_2 + BILL_AMT3 + SEX + PAY_AMT5 +
##     AGE + PAY_AMT4 + PAY_6 + PAY_AMT3
##
##             Df Deviance   AIC
```

```
## <none>             22316 22364
## + BILL_AMT2  1    22315 22365
## + PAY_5      1    22315 22365
## + PAY_AMT6   1    22315 22365
## + BILL_AMT6  1    22315 22365
## + PAY_4      1    22315 22365
## + BILL_AMT5  1    22316 22366
## + BILL_AMT4  1    22316 22366
```

```
credit.glm.fowardbestModel <- formula(credit.glm.forward)

credit.glm.full.forward <- glm(formula=credit.glm.fowardbestModel, family=bin
omial(logit), data=credit.trainData.sc)

# Predict for the Full model
credit.testData.sc$defaultPredForward <- predict(credit.glm.full.forward, new
data=credit.testData.sc, type="response")

# Range for cuttoff
cutoffRange <- seq(.01,.99,length=1000)
perfMatrix <- performanceMetric(cutoffRange, credit.testData.sc$default, cred
it.testData.sc$defaultPredForward)
perfDF <- data.frame(perfMatrix)
names(perfDF) <- c('accuracy', 'precision', 'recall')
head(perfDF)
```

```
##    accuracy precision recall
## 1 0.2183333 0.2154567      1
## 2 0.2188333 0.2155649      1
## 3 0.2195000 0.2157093      1
## 4 0.2200000 0.2158177      1
## 5 0.2206667 0.2159624      1
## 6 0.2215000 0.2161436      1
```

```
# Plot Accuracy, precision and recall
par(mfrow=c(1,1))
options(repr.plot.width=6, repr.plot.height=4)
p <- plotPerfMetric(perfDF, cutoffRange)
p
```

10. Fitting GLM With Backward selection

```
allFeatures <- c(credit.nominalCols, credit.numericCols)
print (length(allFeatures))
```

```
## [1] 23
```

```
bestModel <- backwardSelection(features=allFeatures, label=credit.labelCol, d
ataIN=credit.data)
bestModel
```

```
## default ~ SEX + EDUCATION + MARRIAGE + LIMIT_BAL + AGE + PAY_0 +
##     PAY_2 + PAY_3 + PAY_5 + BILL_AMT1 + BILL_AMT2 + PAY_AMT1 +
##     PAY_AMT2 + PAY_AMT4 + PAY_AMT5
## <environment: 0x0000000025ab7b80>
```

```
credit.glm.backward.manual <- glm(formula=bestModel, family=binomial(logit),
data=credit.trainData.sc)
credit.testData.sc$defaultPredBackward_Manual <- predict(credit.glm.backward.
manual, newdata=credit.testData.sc, type="response")

# Range for cuttoff
cutoffRange <- seq(.01,.99,length=1000)
perfMatrix <- performanceMetric(cutoffRange, credit.testData.sc$default, cred
it.testData.sc$defaultPredBackward_Manual)
perfDF <- data.frame(perfMatrix)
names(perfDF) <- c('accuracy', 'precision', 'recall')
head(perfDF)
```

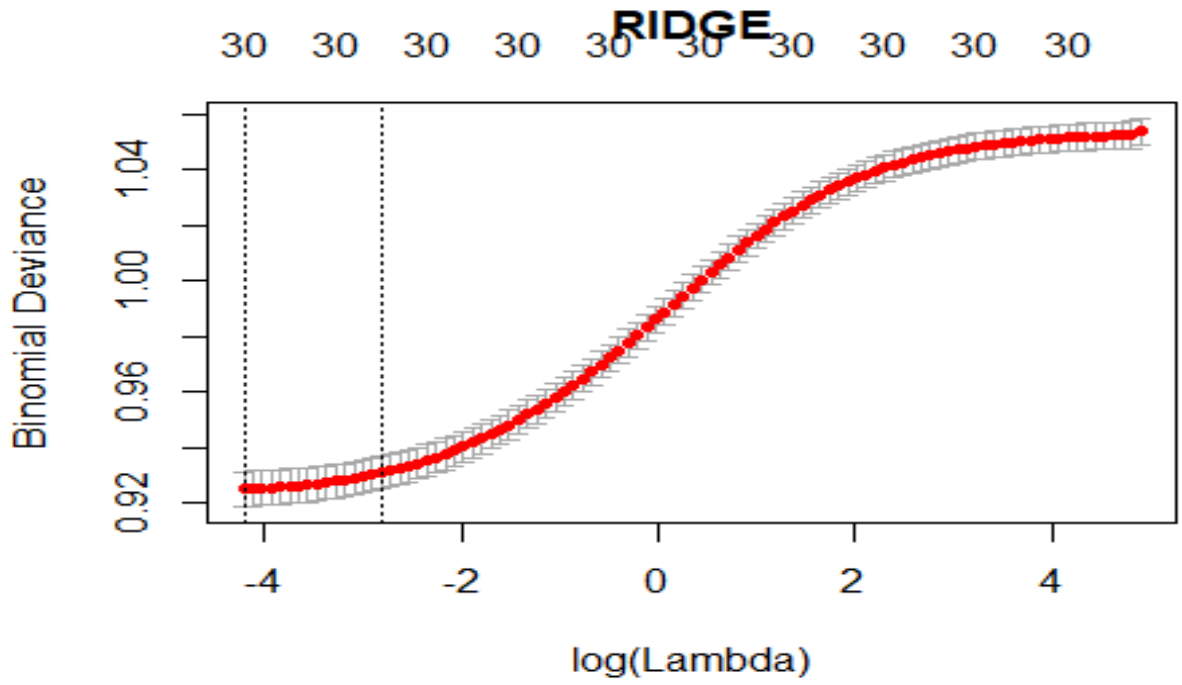
```
##    accuracy precision recall
## 1 0.2178333 0.2153486      1
## 2 0.2183333 0.2154567      1
## 3 0.2188333 0.2155649      1
## 4 0.2191667 0.2156370      1
## 5 0.2198333 0.2157815      1
## 6 0.2203333 0.2158900      1
```

```
# Plot Accuracy, precision and recall
par(mfrow=c(1,1))
options(repr.plot.width=6, repr.plot.height=4)
p <- plotPerfMetric(perfDF, cutoffRange)
p
```

11. Ridge regression

```
# Split the label from the Train and Test Data
xTrainData <- credit.trainData.dummy[, -which(names(credit.trainData.dummy) =
= credit.labelCol)]
yTrainLabel <- credit.trainData.dummy[credit.labelCol]
xTestData <- credit.testData.dummy[, -which(names(credit.testData.dummy) == c
redit.labelCol)]
yTestLabel <- credit.testData.dummy[credit.labelCol]

credit.ridge.full <- lm.ridge(formula=credit.full.model, data=credit.data.sca
led, lambda = seq(0,1,10))
select(credit.ridge.full)
```

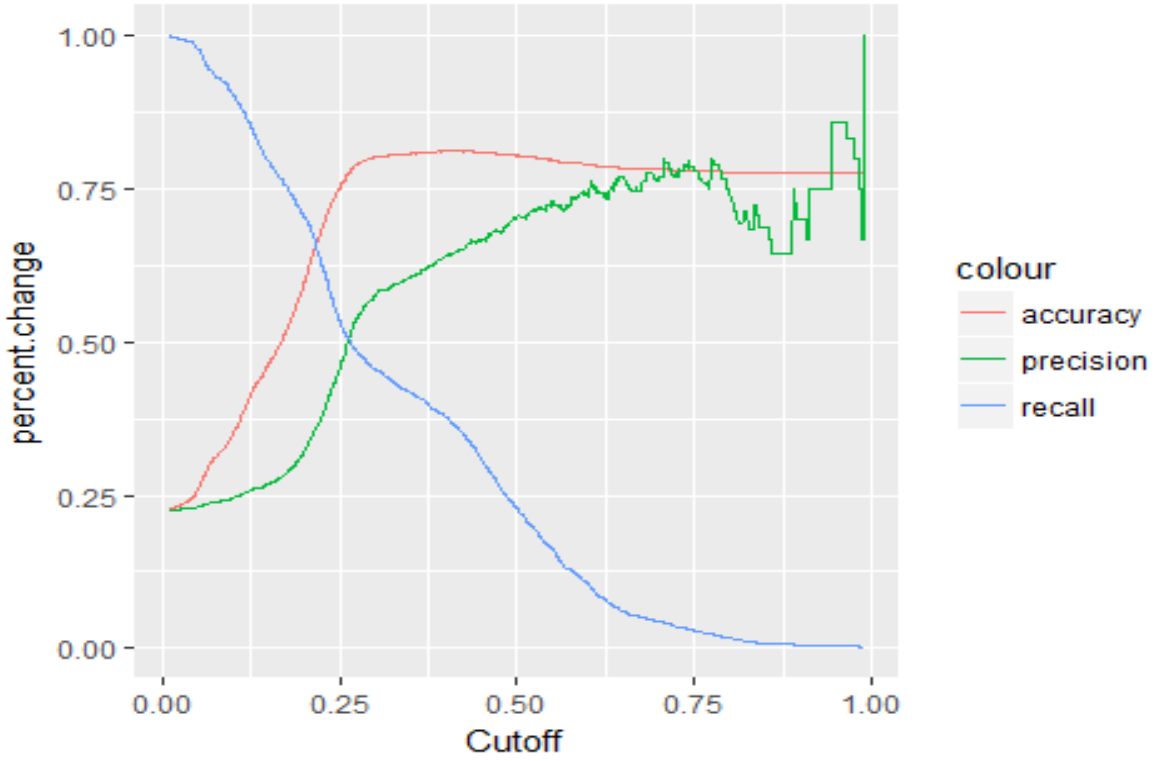
```
## modified HKB estimator is 109.5689
## modified L-W estimator is 196.2137
## smallest value of GCV  at 0

# Find the best lambda and predict on that lambda for the test set.
credit.ridge.cv <- cv.glmnet(x=as.matrix(xTrainData), y=as.matrix(yTrainLabel
), alpha=0, family='binomial')
lambdaBest <- credit.ridge.cv$lambda.min
credit.ridge.fit <- glmnet(x=as.matrix(xTrainData), y=as.matrix(yTrainLabel),
alpha=0, lambda=credit.ridge.cv$lambda.min, family='binomial')
credit.ridge.predict <- predict(credit.ridge.fit, newx = as.matrix(xTestData)
, s = lambdaBest, type = "response")

options(repr.plot.width=10, repr.plot.height=4)
par(mfrow=c(1,1))
plot(credit.ridge.cv, main="RIDGE")
```




```
# Range for cuttoff
cutoffRange <- seq(.01,.99,length=1000)
perfMatrix <- performanceMetric(cutoffRange = cutoffRange,
                                y = yTestLabel$default,
                                y_hat = unlist(credit.ridge.predict))

perfDF <- data.frame(perfMatrix)
names(perfDF) <- c('accuracy', 'precision', 'recall')
head(perfDF)
```

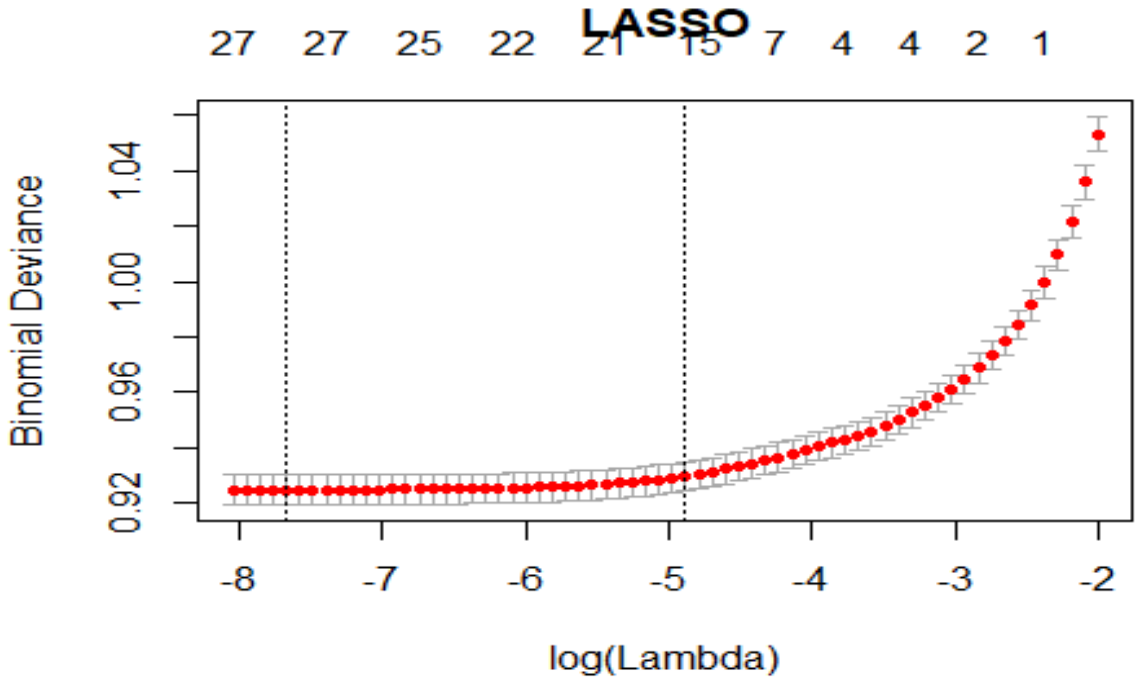
```
##    accuracy precision    recall
## 1 0.2278333 0.2262322 0.9985251
## 2 0.2280000 0.2262701 0.9985251
## 3 0.2280000 0.2261785 0.9977876
## 4 0.2281667 0.2262164 0.9977876
## 5 0.2285000 0.2262004 0.9970501
## 6 0.2288333 0.2262762 0.9970501
```

```
# Plot Accuracy, precision and recall
par(mfrow=c(1,1))
options(repr.plot.width=6, repr.plot.height=4)
p <- plotPerfMetric(perfDF, cutoffRange)
p
```

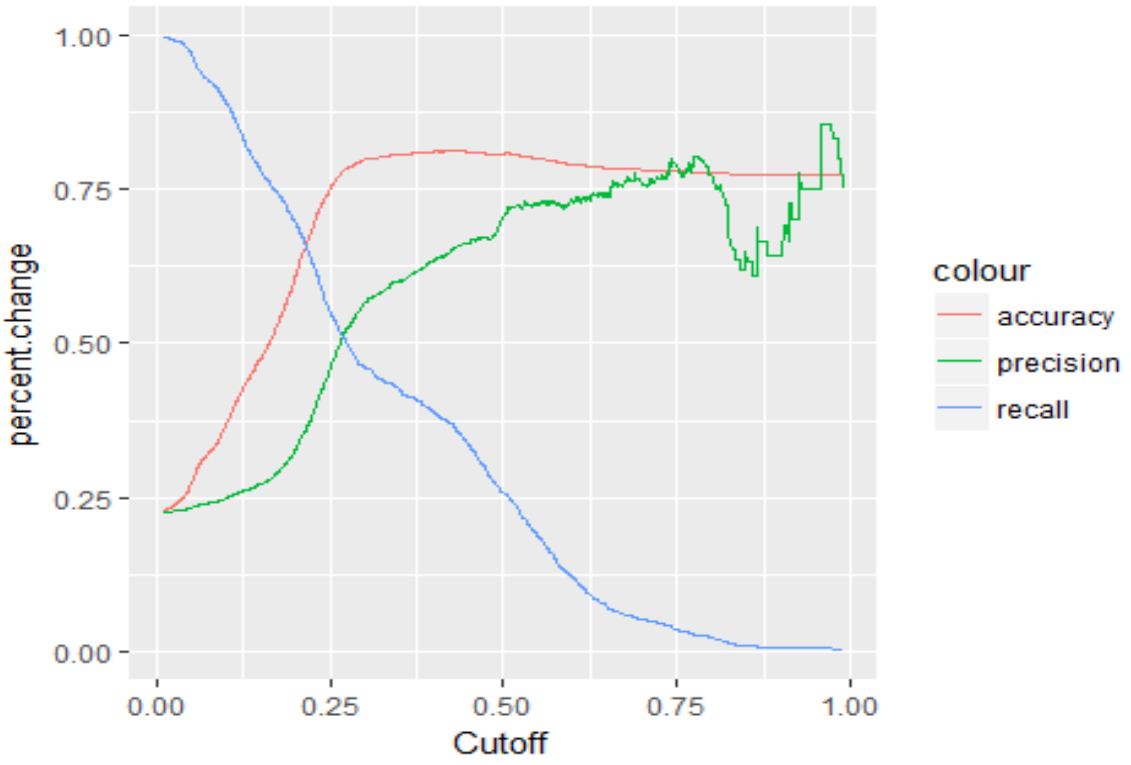


12. LASSO Regression

```
credit.lasso.cv = cv.glmnet(x=as.matrix(xTrainData), y=as.matrix(yTrainLabel)
, alpha=1, family='binomial')
credit.lasso.predict <- predict(credit.lasso.cv, newx = as.matrix(xTestData),
s = "lambda.min", type = "response")
options(repr.plot.width=10, repr.plot.height=4)
par(mfrow=c(1,1))
plot(credit.lasso.cv, main="LASSO")
```




```
# Range for cuttoff
cutoffRange <- seq(.01,.99,length=1000)
perfMatrix <- performanceMetric(cutoffRange = cutoffRange,
                                y = yTestLabel$default,
                                y_hat = unlist(credit.lasso.predict))

perfDF <- data.frame(perfMatrix)
names(perfDF) <- c('accuracy', 'precision', 'recall')
head(perfDF)

##    accuracy precision    recall
## 1 0.2283333 0.2261626 0.9970501
## 2 0.2286667 0.2261466 0.9963127
## 3 0.2296667 0.2262823 0.9955752
## 4 0.2301667 0.2263961 0.9955752
## 5 0.2303333 0.2264341 0.9955752
## 6 0.2306667 0.2265101 0.9955752

# Plot Accuracy, precision and recall
par(mfrow=c(1,1))
options(repr.plot.width=6, repr.plot.height=4)
p <- plotPerfMetric(perfDF, cutoffRange)
p
```



13. Splitting Unscaled data for Decision tree & Random Forest

```
credit.dataIN <- credit.data
credit.null.model <- as.formula(paste('default', "~", 1))
credit.full.model <- bindModel(yLabel = 'default',xFeatures = c(credit.nomina
lCols, credit.numericCols))

credit.null.model

## default ~ 1

credit.full.model

## default ~ SEX + EDUCATION + MARRIAGE + LIMIT_BAL + AGE + PAY_0 +
##     PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6 + BILL_AMT1 + BILL_AMT2 +
##     BILL_AMT3 + BILL_AMT4 + BILL_AMT5 + BILL_AMT6 + PAY_AMT1 +
##     PAY_AMT2 + PAY_AMT3 + PAY_AMT4 + PAY_AMT5 + PAY_AMT6
## <environment: 0x0000000014fba350>

# Get the Train Test Data
dataOUT <- stratifiedSampling(dataIN=credit.dataIN, sample_on_col='default',
trainPrcnt = 0.8)

credit.trainData <- dataOUT[[1]]
credit.testData <- dataOUT[[2]]
nrow(credit.trainData)

## [1] 24000
```

```
nrow(credit.testData)
```

```
## [1] 6000
```

```
head(credit.trainData)
```

```
##   LIMIT_BAL SEX EDUCATION MARRIAGE AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6
## 1     20000   2         2        1  24     2     2    -1    -1    -2    -2
## 2    120000   2         2        2  26    -1     2     0     0     0     2
## 3     90000   2         2        2  34     0     0     0     0     0     0
## 4     50000   2         2        1  37     0     0     0     0     0     0
## 5     50000   1         2        1  57    -1     0    -1     0     0     0
## 6     50000   1         1        2  37     0     0     0     0     0     0
##   BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT1
## 1      3913      3102       689         0         0         0        0
## 2      2682      1725      2682      3272      3455      3261        0
## 3     29239     14027     13559     14331     14948     15549     1518
## 4     46990     48233     49291     28314     28959     29547     2000
## 5      8617      5670     35835     20940     19146     19131     2000
## 6     64400     57069     57608     19394     19619     20024     2500
##   PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 default
## 1      689        0        0        0        0       1
## 2     1000     1000     1000        0     2000       1
## 3     1500     1000     1000     1000     5000       0
## 4     2019     1200     1100     1069     1000       0
## 5    36681    10000     9000      689      679       0
## 6     1815      657     1000     1000      800       0
```

14. Decision Tree

```
credit.dt.fit <- rpart(credit.full.model, data=credit.trainData, method="clas
s")
credit.dt.fit
```

```
## n= 24000
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 24000 5278 0 (0.7800833 0.2199167)
##   2) PAY_0< 1.5 21491 3536 0 (0.8354660 0.1645340) *
##   3) PAY_0>=1.5 2509  767 1 (0.3056995 0.6943005) *
```

```
credit.dt.predict <- predict(credit.dt.fit, credit.testData, type = "class")
credit.testData$defaultPredDT <- credit.dt.predict

CM <- confusionMatrix(reference = credit.testData$default, data = credit.test
Data$defaultPredDT, positive = "1", mode='prec_recall')
CM
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction    0    1
##          0 4456  923
##          1  186  435
##
##                Accuracy : 0.8122
##                  95% CI : (0.8051, 0.8249)
##     No Information Rate : 0.7737
##     P-Value [Acc > NIR] : 2.151e-15
##
##                   Kappa : 0.3468
##  Mcnemar's Test P-Value : < 2.2e-16
##
##               Precision : 0.7005
##                  Recall : 0.3203
##                      F1 : 0.4396
##              Prevalence : 0.2263
##          Detection Rate : 0.0725
##    Detection Prevalence : 0.1035
##       Balanced Accuracy : 0.6401
##
##        'Positive' Class : 1
##
```

15. Random forest

```
x <- subset(credit.trainData, select=-c(default))
y <- as.factor(as.character(credit.trainData$default))


credit.rf.fit <- randomForest(x = x, y = y, importance = TRUE, ntree = 200)

credit.rf.predict <- predict(credit.rf.fit, credit.testData, type = "response
")

credit.testData$defaultPredRF1 <- credit.rf.predict


CM1 = confusionMatrix(reference = credit.testData$default,
                data = credit.testData$defaultPredRF1,
                positive = "1",
                mode='prec_recall')
CM1

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4475  770
##          1  217  538
```

```
##
##                Accuracy : 0.8155
##                  95% CI : (0.802, 0.822)
##     No Information Rate : 0.7737
##     P-Value [Acc > NIR] : 1.88e-13
##
##                   Kappa : 0.3657
##  Mcnemar's Test P-Value : < 2.2e-16
##
##               Precision : 0.6510
##                  Recall : 0.3667
##                      F1 : 0.4691
##              Prevalence : 0.2263
##          Detection Rate : 0.0830
##    Detection Prevalence : 0.1275
##       Balanced Accuracy : 0.6546
##
##        'Positive' Class : 1
##
```

16. CONCLUSION:

Logistic Regression (RIDGE) gives the best model performance at threshold approximately 0.27. The accuracy is seen as approximately 77%.

Decision tree model performs better than logistic regression with accuracy of 81.22%, but worse than random forest and it makes sense because it is prone to both overfitting and under fitting.

Random Forest model on the other hand produces outstanding result with an accuracy of 81.56%, precision of 0.9624 and a recall of 0.8227. This makes sense because random forests average the output from many decision tress which makes it robust to overfitting. Therefore, despite the training error were high the random forest model does an outstanding job in classifying the test data.