LogisticRegression

From the ROC plot we can conclude that Logistic regression is not a best model for this problem. The Balance Accuracy is 69.50 with sensitivity of 88.65207 and specificity of 50.34.

```
library(caret)
library(tidyr)
library(MASS)
library(e1071)
library(pROC)
```

Reading the data

```
data1 <- read.table(file = "C://Users/cs_mo/Downloads/ISYE7406/ProjectCreditCard/creditcards.csv", head
names(data1)[25] <- 'default'
head(data1)
```

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

Removing 167 outliers as identified in the data exploration part

```
out <- boxplot.stats(data1$LIMIT_BAL)$out
out_ind <- which(data1$LIMIT_BAL %in% c(out))
mydata1 <- data1[-out_ind,]
dim(mydata1)
```

```
## [1] 29833    25
```

Cleaning up Marriage and Education feature

```
mydata1$MARRIAGE[mydata1$MARRIAGE == "0"] <- "3"
mydata1$EDUCATION[mydata1$EDUCATION== "6"]<-"4"
mydata1$EDUCATION[mydata1$EDUCATION== "5"]<-"4"
mydata1$EDUCATION[mydata1$EDUCATION== "0"]<-"4"
```

```
mydata1$default[mydata1$default=="0"] <- "ND"
mydata1$default[mydata1$default=="1"] <- "DEF"
```

Removing the ID column...

```
mydata <- mydata1[,2:25]
head(mydata)
```

```
##   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 PAY_AMT2
## 1      3913      3102       689         0         0         0        0      689
## 2      2682      1725      2682      3272      3455      3261        0     1000
## 3     29239     14027     13559     14331     14948     15549     1518     1500
## 4     46990     48233     49291     28314     28959     29547     2000     2019
## 5      8617      5670     35835     20940     19146     19131     2000    36681
## 6     64400     57069     57608     19394     19619     20024     2500     1815
##   PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 default
## 1        0        0        0        0     DEF
## 2     1000     1000        0     2000     DEF
## 3     1000     1000     1000     5000      ND
## 4     1200     1100     1069     1000      ND
## 5    10000     9000      689      679      ND
## 6      657     1000     1000      800      ND
```

```
dim(mydata)
```

```
## [1] 29833    24
```

Splitting the data....

```
mydata$SEX <- as.numeric(mydata$SEX)
mydata$EDUCATION <- as.numeric(mydata$EDUCATION)
mydata$MARRIAGE <- as.numeric(mydata$MARRIAGE)

set.seed(7406)
flag<- sort(sample(1:29833,4475))
data_train  <- mydata[-flag,]
data_test <- mydata[flag,]
dim(data_train)
```

```
## [1] 25358    24
```

```
dim(data_test)
```

```
## [1] 4475   24
```

```
head(data_train)
```

```
##   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
## 7    500000   1         1        2  29     0     0     0     0     0     0
##   BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT1 PAY_AMT2
## 1      3913      3102       689         0         0         0        0      689
## 2      2682      1725      2682      3272      3455      3261        0     1000
## 3     29239     14027     13559     14331     14948     15549     1518     1500
## 4     46990     48233     49291     28314     28959     29547     2000     2019
## 5      8617      5670     35835     20940     19146     19131     2000    36681
## 7    367965    412023    445007    542653    483003    473944    55000    40000
##   PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6 default
## 1        0        0        0        0     DEF
## 2     1000     1000        0     2000     DEF
## 3     1000     1000     1000     5000      ND
## 4     1200     1100     1069     1000      ND
## 5    10000     9000      689      679      ND
## 7    38000    20239    13750    13770      ND
```

Logistic Regression Full model

```
fullmod <- glm(as.factor(default)~., data = data_train, family = binomial)
summary(fullmod)
```

```
##
## Call:
## glm(formula = as.factor(default) ~ ., family = binomial, data = data_train)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1893   0.2799   0.5458   0.7021   3.1280
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.539e-01  1.293e-01   4.286 1.82e-05 ***
## LIMIT_BAL    6.857e-07  1.754e-07   3.910 9.24e-05 ***
## SEX          1.341e-01  3.334e-02   4.023 5.75e-05 ***
## EDUCATION    1.129e-01  2.399e-02   4.707 2.52e-06 ***
## MARRIAGE     1.712e-01  3.449e-02   4.965 6.89e-07 ***
## AGE         -6.498e-03  1.939e-03  -3.351 0.000806 ***
## PAY_0       -5.706e-01  1.919e-02 -29.730  < 2e-16 ***
## PAY_2       -7.864e-02  2.195e-02  -3.583 0.000340 ***
## PAY_3       -5.602e-02  2.465e-02  -2.272 0.023068 *
## PAY_4       -3.578e-02  2.715e-02  -1.318 0.187549
## PAY_5       -3.926e-02  2.914e-02  -1.347 0.177978
## PAY_6       -9.842e-03  2.419e-02  -0.407 0.684103
## BILL_AMT1    6.947e-06  1.289e-06   5.388 7.12e-08 ***
## BILL_AMT2   -4.436e-06  1.641e-06  -2.703 0.006864 **
## BILL_AMT3   -1.385e-06  1.447e-06  -0.957 0.338415
## BILL_AMT4    1.988e-06  1.540e-06   1.291 0.196850
## BILL_AMT5   -1.860e-06  1.759e-06  -1.057 0.290288
## BILL_AMT6   -5.675e-07  1.348e-06  -0.421 0.673750
## PAY_AMT1     1.616e-05  2.578e-06   6.269 3.62e-10 ***
## PAY_AMT2     9.674e-06  2.334e-06   4.144 3.41e-05 ***
## PAY_AMT3     1.764e-06  1.858e-06   0.950 0.342218
## PAY_AMT4     6.571e-06  2.161e-06   3.041 0.002360 **
## PAY_AMT5     5.575e-06  2.125e-06   2.624 0.008690 **
## PAY_AMT6     3.560e-06  1.492e-06   2.385 0.017060 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26812  on 25357  degrees of freedom
## Residual deviance: 23567  on 25334  degrees of freedom
## AIC: 23615
##
## Number of Fisher Scoring iterations: 6
```

Stepwise regression

As we can see in below results "backward" and "both" selection processes give the lowest AIC score.And also they selected same variables. We can use features to train our model.

```
forward <- stepAIC(fullmod, trace = FALSE, direction = "forward")
forward
```

```
##
## Call:  glm(formula = as.factor(default) ~ LIMIT_BAL + SEX + EDUCATION +
##     MARRIAGE + 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, family = binomial, data = data_train)
##
## Coefficients:
## (Intercept)    LIMIT_BAL          SEX    EDUCATION     MARRIAGE          AGE
##   5.539e-01    6.857e-07    1.341e-01    1.129e-01    1.712e-01   -6.498e-03
##       PAY_0        PAY_2        PAY_3        PAY_4        PAY_5        PAY_6
##  -5.706e-01   -7.864e-02   -5.602e-02   -3.578e-02   -3.926e-02   -9.842e-03
##   BILL_AMT1    BILL_AMT2    BILL_AMT3    BILL_AMT4    BILL_AMT5    BILL_AMT6
##   6.947e-06   -4.436e-06   -1.385e-06    1.988e-06   -1.860e-06   -5.675e-07
##    PAY_AMT1     PAY_AMT2     PAY_AMT3     PAY_AMT4     PAY_AMT5     PAY_AMT6
##   1.616e-05    9.674e-06    1.764e-06    6.571e-06    5.575e-06    3.560e-06
##
## Degrees of Freedom: 25357 Total (i.e. Null);  25334 Residual
## Null Deviance:       26810
## Residual Deviance: 23570     AIC: 23620
```

```
bac <- stepAIC(fullmod, trace = FALSE, direction = "backward")
bac
```

```
##
## Call:  glm(formula = as.factor(default) ~ LIMIT_BAL + SEX + EDUCATION +
##     MARRIAGE + AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + BILL_AMT1 +
##     BILL_AMT2 + BILL_AMT5 + PAY_AMT1 + PAY_AMT2 + PAY_AMT3 +
##     PAY_AMT4 + PAY_AMT5 + PAY_AMT6, family = binomial, data = data_train)
##
## Coefficients:
## (Intercept)    LIMIT_BAL          SEX    EDUCATION     MARRIAGE          AGE
##   5.560e-01    7.028e-07    1.347e-01    1.127e-01    1.709e-01   -6.541e-03
##       PAY_0        PAY_2        PAY_3        PAY_5    BILL_AMT1    BILL_AMT2
##  -5.723e-01   -7.890e-02   -7.123e-02   -6.443e-02    6.950e-06   -4.751e-06
##   BILL_AMT5     PAY_AMT1     PAY_AMT2     PAY_AMT3     PAY_AMT4     PAY_AMT5
##  -1.524e-06    1.624e-05    8.866e-06    3.294e-06    5.734e-06    5.224e-06
##    PAY_AMT6
##   3.615e-06
##
## Degrees of Freedom: 25357 Total (i.e. Null);  25339 Residual
## Null Deviance:       26810
## Residual Deviance: 23570     AIC: 23610
```

```
both <- stepAIC(fullmod, trace = FALSE, direction = "both")
both
```

```
##
## Call:  glm(formula = as.factor(default) ~ LIMIT_BAL + SEX + EDUCATION +
##     MARRIAGE + AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + BILL_AMT1 +
##     BILL_AMT2 + BILL_AMT5 + PAY_AMT1 + PAY_AMT2 + PAY_AMT3 +
##     PAY_AMT4 + PAY_AMT5 + PAY_AMT6, family = binomial, data = data_train)
##
## Coefficients:
## (Intercept)    LIMIT_BAL          SEX    EDUCATION     MARRIAGE          AGE
##   5.560e-01    7.028e-07    1.347e-01    1.127e-01    1.709e-01   -6.541e-03
##       PAY_0        PAY_2        PAY_3        PAY_5    BILL_AMT1    BILL_AMT2
```

```
##  -5.723e-01   -7.890e-02   -7.123e-02   -6.443e-02    6.950e-06   -4.751e-06
##   BILL_AMT5     PAY_AMT1     PAY_AMT2     PAY_AMT3     PAY_AMT4     PAY_AMT5
##  -1.524e-06    1.624e-05    8.866e-06    3.294e-06    5.734e-06    5.224e-06
##    PAY_AMT6
##   3.615e-06
##
## Degrees of Freedom: 25357 Total (i.e. Null);  25339 Residual
## Null Deviance:       26810
## Residual Deviance: 23570     AIC: 23610
```

Buidling model using features selected by Stepwise regression.

```
submodel <- glm(as.factor(default)~LIMIT_BAL+SEX+EDUCATION+MARRIAGE+AGE+PAY_0+PAY_2+PAY_3+PAY_5+BILL_AM
summary(submodel)
```

```
##
## Call:
## glm(formula = as.factor(default) ~ LIMIT_BAL + SEX + EDUCATION +
##     MARRIAGE + AGE + PAY_0 + PAY_2 + PAY_3 + PAY_5 + BILL_AMT1 +
##     BILL_AMT2 + BILL_AMT5 + PAY_AMT1 + PAY_AMT2 + PAY_AMT3 +
##     PAY_AMT4 + PAY_AMT5 + PAY_AMT6, family = binomial, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2205   0.2807   0.5459   0.7020   3.1278
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.560e-01  1.292e-01   4.303 1.69e-05 ***
## LIMIT_BAL    7.028e-07  1.749e-07   4.019 5.84e-05 ***
## SEX          1.347e-01  3.333e-02   4.041 5.32e-05 ***
## EDUCATION    1.127e-01  2.398e-02   4.700 2.60e-06 ***
## MARRIAGE     1.709e-01  3.448e-02   4.955 7.23e-07 ***
## AGE         -6.541e-03  1.939e-03  -3.374 0.000742 ***
## PAY_0       -5.723e-01  1.915e-02 -29.889  < 2e-16 ***
## PAY_2       -7.890e-02  2.191e-02  -3.601 0.000317 ***
## PAY_3       -7.123e-02  2.209e-02  -3.224 0.001264 **
## PAY_5       -6.443e-02  1.951e-02  -3.302 0.000960 ***
## BILL_AMT1    6.950e-06  1.283e-06   5.416 6.10e-08 ***
## BILL_AMT2   -4.751e-06  1.443e-06  -3.293 0.000992 ***
## BILL_AMT5   -1.524e-06  7.399e-07  -2.060 0.039444 *
## PAY_AMT1     1.624e-05  2.575e-06   6.307 2.85e-10 ***
## PAY_AMT2     8.866e-06  2.078e-06   4.266 1.99e-05 ***
## PAY_AMT3     3.294e-06  1.645e-06   2.003 0.045213 *
## PAY_AMT4     5.734e-06  1.956e-06   2.931 0.003379 **
## PAY_AMT5     5.224e-06  1.845e-06   2.831 0.004642 **
## PAY_AMT6     3.615e-06  1.475e-06   2.452 0.014222 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

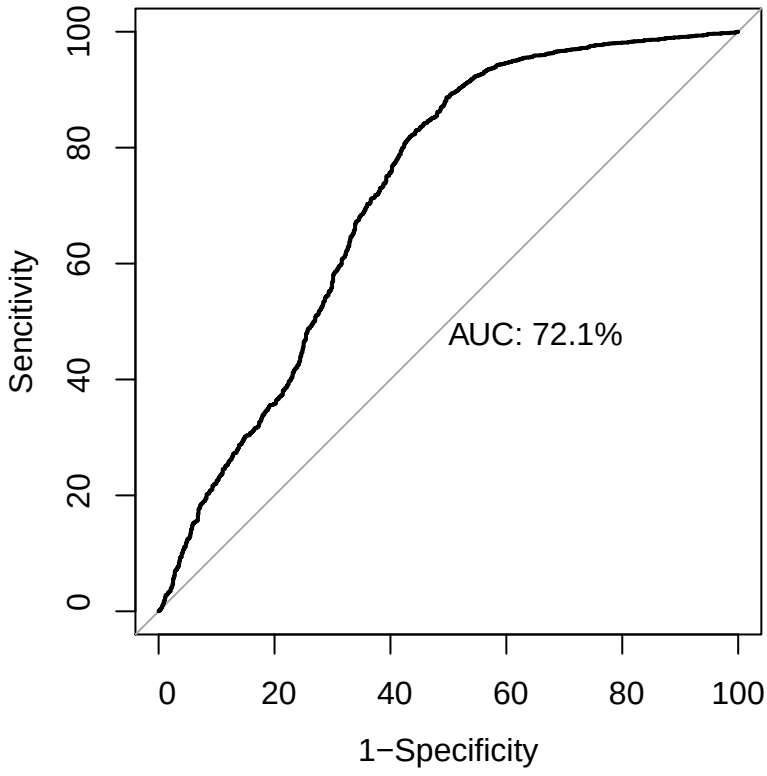
```
##     Null deviance: 26812  on 25357  degrees of freedom
## Residual deviance: 23571  on 25339  degrees of freedom
## AIC: 23609
##
## Number of Fisher Scoring iterations: 6
```

```
pred_glm <- predict(submodel, data_test, type = "response")
head(pred_glm)
```

```
##         6        14        19        25        28        32
## 0.7608835 0.6004588 0.7106056 0.7656185 0.8098012 0.5066689
```

```
par(pty = "s")
roc(as.factor(data_test[,24]), pred_glm, plot = TRUE, legacy.axes = T, percent = TRUE,
    print.auc =TRUE,

    #auc.polygon = TRUE,
    xlab= "1-Specificity",
    ylab= "Sencitivity"
    #xlab ="False Positive Percentage",
    #ylab =" True positive Percentage"
    )
```




```
##
```

```
## Call:
## roc.default(response = as.factor(data_test[, 24]), predictor = pred_glm,     percent = TRUE, plot = T
##
## Data: pred_glm in 1003 controls (as.factor(data_test[, 24]) DEF) < 3472 cases (as.factor(data_test[,
## Area under the curve: 72.07%
```

```
roc.infoglm <- roc(as.factor(data_test[,24]), pred_glm, plot = FALSE, legacy.axes = TRUE)
auc(roc.infoglm)
```

```
## Area under the curve: 0.7207
```

```
roc.dfglm <- data.frame(sensitivity = roc.infoglm$sensitivities*100,
                        specificity =(roc.infoglm$specificities)*100,
                        thresholds = roc.infoglm$thresholds)
```

```
roc.infoglm
```

```
##
## Call:
## roc.default(response = as.factor(data_test[, 24]), predictor = pred_glm,     plot = FALSE, legacy.axe
##
## Data: pred_glm in 1003 controls (as.factor(data_test[, 24]) DEF) < 3472 cases (as.factor(data_test[,
## Area under the curve: 0.7207
```

```
roc.dfglm$Balance <- ((roc.dfglm$sensitivity + roc.dfglm$specificity)/2)
head(roc.dfglm)
```

```
##   sensitivity specificity  thresholds  Balance
## 1    100.0000   0.0000000        -Inf 50.00000
## 2    100.0000   0.0997009 0.007771642 50.04985
## 3     99.9712   0.0997009 0.015457444 50.03545
## 4     99.9712   0.1994018 0.032691027 50.08530
## 5     99.9712   0.2991027 0.068560704 50.13515
## 6     99.9424   0.2991027 0.094402549 50.12075
```

Printing the top 10 records with the highest Balance accuracy.

```
dfglm <- roc.dfglm[with(roc.dfglm,order(-Balance)),]
head(dfglm)
```

```
##     sensitivity specificity thresholds  Balance
## 900    88.65207    50.34895  0.7105493 69.50051
## 901    88.62327    50.34895  0.7107179 69.48611
## 902    88.59447    50.34895  0.7108903 69.47171
## 903    88.56567    50.34895  0.7109541 69.45731
## 899    88.65207    50.24925  0.7104873 69.45066
## 877    89.14171    49.75075  0.7076620 69.44623
```