Data Mining Assignment

Weijie Gao

2 Feb 2017

```
dataPath <- "~/Documents/Chicago2016/Winter/Data Mining/week2"
GermanCredit <- read.table(paste(dataPath,"Germancredit_numertic.csv",sep='/'),header=TRUE)

# represent the good credit as 1, and bad as 0.
GermanCredit$Class <-ifelse(GermanCredit$Class==1,1,0)

# include only numeric independent variables 1,3 through 9 as predictors
GermanCredit <- GermanCredit[,c(5,2,8,11,13,16,18)]
head(GermanCredit)
```

```
##   Credit_Amount Duration Installment_rate Present_residence Age
## 1          1169        6                4                 4  67
## 2          5951       48                2                 2  22
## 3          2096       12                2                 3  49
## 4          7882       42                2                 4  45
## 5          4870       24                3                 4  53
## 6          9055       36                2                 4  35
##   Num_existingcredit Num_maintenance
## 1                  2               1
## 2                  1               1
## 3                  1               2
## 4                  1               2
## 5                  2               2
## 6                  1               2
```

```
# seperate the data into training and test set
set.seed(234)
smp_size <- floor(0.7 * nrow(GermanCredit))
train_ind <- sample(nrow(GermanCredit), size = smp_size)
GermanCredit.train.cw <- GermanCredit[train_ind, ]
GermanCredit.test.cw <- GermanCredit[-train_ind, ]

source(file.path(dataPath,"clustereg.predict.R"))
source(file.path(dataPath,"clustreg.R"))

clustreg.credit.1 <- clustreg(GermanCredit.train.cw,1,1,1234,1)
clustreg.credit.1$results
```

```
## [[1]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5882.7 -1270.7  -320.0   604.6 12125.0
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1779.864    467.531   3.807 0.000153 ***
## Duration             149.361      6.467  23.094  < 2e-16 ***
## Installment_rate    -824.879     72.183 -11.428  < 2e-16 ***
## Present_residence     14.441     76.406   0.189 0.850147
## Age                   14.728      7.527   1.957 0.050778 .
## Num_existingcredit   150.413    141.405   1.064 0.287835
## Num_maintenance       60.803    220.739   0.275 0.783050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2102 on 693 degrees of freedom
## Multiple R-squared:  0.4706, Adjusted R-squared:  0.466
## F-statistic: 102.7 on 6 and 693 DF,  p-value: < 2.2e-16
```

```
table((clustreg.credit.1$cluster))
```

```
##
##   1
## 700
```

```
round(prop.table(table(clustreg.credit.1$cluster)),3)
```

```
##
## 1
## 1
```

```
clustreg.credit.2 <- clustreg(GermanCredit.train.cw,2,30,1234,15)
clustreg.credit.2$results
```

```
## [[1]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2677.5 -1255.2  -525.1   459.0  8989.1
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5776.77    1117.95   5.167 8.79e-07 ***
## Duration              203.17      13.82  14.701  < 2e-16 ***
## Installment_rate    -1263.93     165.26  -7.648 4.14e-12 ***
## Present_residence     364.16     174.45   2.087  0.03882 *
## Age                    57.00      19.61   2.906  0.00431 **
## Num_existingcredit   -776.94     269.89  -2.879  0.00468 **
## Num_maintenance     -1428.43     439.61  -3.249  0.00148 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2091 on 129 degrees of freedom
## Multiple R-squared:  0.7311, Adjusted R-squared:  0.7186
## F-statistic: 58.45 on 6 and 129 DF,  p-value: < 2.2e-16
##
##
```

```
## [[2]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3275.8  -631.9   -41.0   579.3  3235.8
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1653.554    239.260   6.911 1.32e-11 ***
## Duration              114.780      3.384  33.916  < 2e-16 ***
## Installment_rate     -590.078     36.888 -15.996  < 2e-16 ***
## Present_residence      22.694     38.912   0.583   0.5600
## Age                     8.988      3.741   2.402   0.0166 *
## Num_existingcredit    -33.566     77.032  -0.436   0.6632
## Num_maintenance       -94.235    119.955  -0.786   0.4324
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 956.8 on 557 degrees of freedom
## Multiple R-squared:  0.6961, Adjusted R-squared:  0.6928
## F-statistic: 212.6 on 6 and 557 DF,  p-value: < 2.2e-16
```

```r
table((clustreg.credit.2$cluster))
```

```
##
##   1   2
## 136 564
```

```r
round(prop.table(table(clustreg.credit.2$cluster)),3)
```

```
##
##     1     2
## 0.194 0.806
```

```r
clustreg.credit.3 <- clustreg(GermanCredit.train.cw,3,30,1234,15)
clustreg.credit.3$results
```

```
## [[1]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2938.1 -1800.1  -152.5   986.6  6137.4
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           9902.110   1787.182   5.541 1.49e-06 ***
## Duration               124.767     23.285   5.358 2.77e-06 ***
## Installment_rate      -612.414    305.823  -2.003   0.0513 .
## Present_residence        1.221    356.279   0.003   0.9973
## Age                     51.370     30.999   1.657   0.1044
```

```
## Num_existingcredit   576.855    633.476   0.911   0.3673
## Num_maintenance    -3381.491    771.943  -4.380 7.00e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2266 on 45 degrees of freedom
## Multiple R-squared:  0.5772, Adjusted R-squared:  0.5209
## F-statistic: 10.24 on 6 and 45 DF,  p-value: 3.948e-07
##
##
## [[2]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1466.5  -633.4  -185.0   480.8  2742.1
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3422.371    356.917   9.589  < 2e-16 ***
## Duration            134.073      4.472  29.981  < 2e-16 ***
## Installment_rate   -773.065     52.042 -14.855  < 2e-16 ***
## Present_residence    13.753     54.449   0.253    0.801
## Age                   6.270      5.859   1.070    0.286
## Num_existingcredit  524.212    100.788   5.201 4.59e-07 ***
## Num_maintenance    -623.180    143.147  -4.353 2.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 849 on 216 degrees of freedom
## Multiple R-squared:  0.8539, Adjusted R-squared:  0.8498
## F-statistic: 210.4 on 6 and 216 DF,  p-value: < 2.2e-16
##
##
## [[3]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2169.07  -366.44    12.17   416.15  1509.85
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1368.7832   167.0246   8.195 3.09e-15 ***
## Duration             90.1709     2.5670  35.126  < 2e-16 ***
## Installment_rate   -419.9987    26.2534 -15.998  < 2e-16 ***
## Present_residence    13.7027    27.0715   0.506  0.61301
## Age                   0.8184     2.6107   0.313  0.75407
## Num_existingcredit  214.5062    51.2740   4.184 3.50e-05 ***
## Num_maintenance    -258.3763    90.1163  -2.867  0.00435 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 580 on 418 degrees of freedom
## Multiple R-squared:  0.7607, Adjusted R-squared:  0.7573
## F-statistic: 221.5 on 6 and 418 DF,  p-value: < 2.2e-16
```

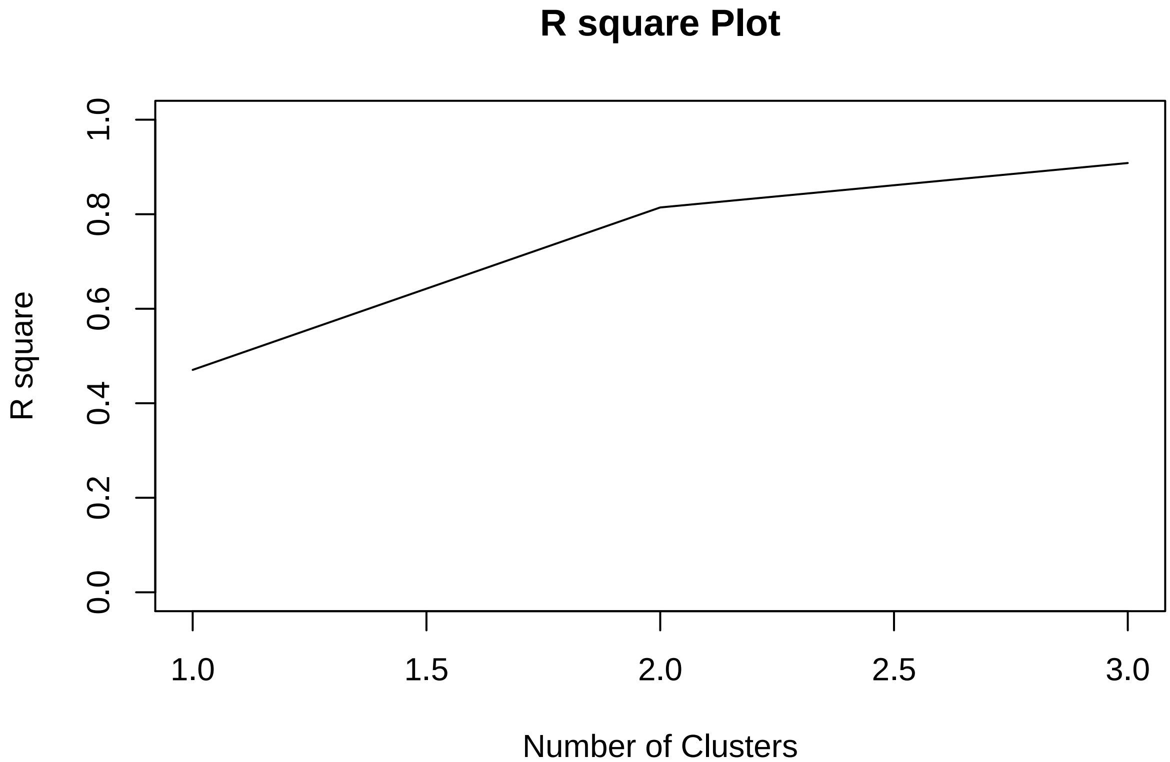
```r
table((clustreg.credit.3$cluster))
```

```
##
##   1   2   3
##  52 223 425
```

```r
round(prop.table(table(clustreg.credit.3$cluster)),3)
```

```
##
##     1     2     3
## 0.074 0.319 0.607
```

```r
plot(c(1,2,3),c(clustreg.credit.1$rsq.best,clustreg.credit.2$rsq.best,clustreg.credit.3$rsq.best),ylim=c
```



The above graph shows that the rsq.best is highest when we group the data into 3 clusters. In this case, cluster 2 and cluster 3 account for the majority of the data, 31.9% and 60.7%, respectively, and cluster 1 only account for 7.4%. From the result of this model clustreg.credit.3, it indicates that in cluster 1 the coefficients of Duration, Num_maintenance are significant, and in cluster 2 and cluster 3 the coefficients of Duration, Installment_rate, Num_existingcredit and Num_maintenance are significant. For the model clustreg.credit.2, its overall R square is slightly lower than the 3 cluster model, with cluster 1 equals to 0.7311, and cluster 2 equals to 0.6961. And in cluster 1, all coefficients except for Present_residence are significant, and for cluster 2, only Duration, Installment_rate and age are significant. And in this case, cluster

1 account for 19.4% of the training samples and cluster 2 account for 80.6% of the training data.

```r
# perform holdout validation
predict.credit.1 <- clustreg.predict(clustreg.credit.1,newdat=GermanCredit.test.cw)
predict.credit.1$rsq
```

```
## [1] 0.5727463
```

```r
round(prop.table(table(predict.credit.1$cluster)),3)
```

```
##
## 1
## 1
```

```r
clustreg.credit.1$results$cluster
```

```
## NULL
```

```r
predict.credit.2 <- clustreg.predict(clustreg.credit.2,newdat=GermanCredit.test.cw)
predict.credit.2$rsq
```

```
## [1] 0.8290967
```

```r
table((predict.credit.2$cluster))
```

```
##
##   1   2
##  50 250
```

```r
round(prop.table(table(predict.credit.2$cluster)),3)
```

```
##
##     1     2
## 0.167 0.833
```

```r
predict.credit.3 <-clustreg.predict(clustreg.credit.3,newdat=GermanCredit.test.cw)
predict.credit.3$rsq
```

```
## [1] 0.8893376
```

```r
table((predict.credit.3$cluster))
```

```
##
##   1   2   3
##  17  92 191
```

```r
round(prop.table(table(predict.credit.3$cluster)),3)
```

```
##
##     1     2     3
## 0.057 0.307 0.637
```

In this part, the best r squre for the first model increased to 0.5727463, and for the second and third model, both r square drop a little bit, but they are still good, with 0.8290967 and 0.8893376 respectively. Therefore, in general, the third model has the best performance. And the size of clusters is relatively stable in model 3, cluster 2 and cluster 3 still account for the majority of the data, 30.7% and 63.7%, respectively, and cluster 1 accounts for 5.7%. Hence, we may choose model 3 in this case, that is we seperate the data into 3 clusters and build the corresponding glm model respectively.

Part 2

Discriminant Analysis

```r
dataPath <- "~/Documents/Chicago2016/Winter/Data Mining/week2"
GermanCredit <- read.table(paste(dataPath,"Germancredit_numertic.csv",sep='/'),header=TRUE)

# represent the good credit as 1, and bad as 0.
GermanCredit$Class <-ifelse(GermanCredit$Class==1,1,0)

# seperate the data into training and test set
set.seed(234)
smp_size <- floor(0.7 * nrow(GermanCredit))
train_ind <- sample(nrow(GermanCredit), size = smp_size)
GermanCredit.train <- GermanCredit[train_ind, ]
GermanCredit.test <- GermanCredit[-train_ind, ]

library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```r
library(MASS)
# Linear Discriminant Analysis
LDA <- lda(GermanCredit.train$Class~., data=GermanCredit.train,CV=FALSE)

# generate confusion matrix for training data
predict_lda_train <- predict(LDA)$class
confusionMatrix(GermanCredit.train$Class,predict_lda_train)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 132  91
##          1  58 419
##
##                Accuracy : 0.7871
##                  95% CI : (0.7549, 0.8169)
##     No Information Rate : 0.7286
##     P-Value [Acc > NIR] : 0.000214
##
##                   Kappa : 0.4896
##  Mcnemar's Test P-Value : 0.008753
##
##             Sensitivity : 0.6947
##             Specificity : 0.8216
##          Pos Pred Value : 0.5919
##          Neg Pred Value : 0.8784
##              Prevalence : 0.2714
##          Detection Rate : 0.1886
##    Detection Prevalence : 0.3186
##       Balanced Accuracy : 0.7582
##
##        'Positive' Class : 0
```

```
##
```

```r
# perform holdout validation test for lda
predict_lda <- predict(LDA,newdata=GermanCredit.test)$class
confusionMatrix(GermanCredit.test$Class,predict_lda)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  39  38
##          1  35 188
##
##                Accuracy : 0.7567
##                  95% CI : (0.704, 0.8041)
##     No Information Rate : 0.7533
##     P-Value [Acc > NIR] : 0.4778
##
##                   Kappa : 0.3541
##  Mcnemar's Test P-Value : 0.8149
##
##             Sensitivity : 0.5270
##             Specificity : 0.8319
##          Pos Pred Value : 0.5065
##          Neg Pred Value : 0.8430
##              Prevalence : 0.2467
##          Detection Rate : 0.1300
##    Detection Prevalence : 0.2567
##       Balanced Accuracy : 0.6794
##
##        'Positive' Class : 0
##
```

```r
# Quadratic Discriminant Analysis
QDA <- qda(GermanCredit.train$Class~., data=GermanCredit.train,CV=FALSE)

# generate confusion matrix for training data
predict_qda_train <- predict(QDA)$class
confusionMatrix(GermanCredit.train$Class,predict_qda_train)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 165  58
##          1  76 401
##
##                Accuracy : 0.8086
##                  95% CI : (0.7774, 0.8371)
##     No Information Rate : 0.6557
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5684
##  Mcnemar's Test P-Value : 0.1419
##
##             Sensitivity : 0.6846
```

```
##             Specificity : 0.8736
##          Pos Pred Value : 0.7399
##          Neg Pred Value : 0.8407
##              Prevalence : 0.3443
##          Detection Rate : 0.2357
##    Detection Prevalence : 0.3186
##       Balanced Accuracy : 0.7791
##
##        'Positive' Class : 0
##
```

```
# perform holdout validation test for qda
predict_qda <- predict(QDA,newdata=GermanCredit.test)$class
confusionMatrix(GermanCredit.test$Class,predict_qda)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  42  35
##          1  49 174
##
##                Accuracy : 0.72
##                  95% CI : (0.6655, 0.7701)
##     No Information Rate : 0.6967
##     P-Value [Acc > NIR] : 0.2079
##
##                   Kappa : 0.3074
##  Mcnemar's Test P-Value : 0.1561
##
##             Sensitivity : 0.4615
##             Specificity : 0.8325
##          Pos Pred Value : 0.5455
##          Neg Pred Value : 0.7803
##              Prevalence : 0.3033
##          Detection Rate : 0.1400
##    Detection Prevalence : 0.2567
##       Balanced Accuracy : 0.6470
##
##        'Positive' Class : 0
##
```

Logistic regression

```
set.seed(123)
# perform Add1 to select important features
full.model <- glm(GermanCredit.train$Class~.,family=binomial(link='logit'),data=GermanCredit.train)
full.model.aic <- full.model$aic

null.model <- glm(GermanCredit.train$Class~1,family=binomial(link='logit'),data=GermanCredit.train)
null.model.aic <- null.model$aic

# perform forward selection
```

```
forwards <- step(null.model,trace=0,scope=list(lower=formula(null.model),upper=formula(full.model)),dire
step.forwards.aic <- forwards$aic

# perform backward elimination on the same data set
# backwards <- step(full.model, data=GermanCredit.train, direction="backward")
# step.backwards.aic <- backwards$aic

best_model <- forwards
summary(best_model)
```

```
##
## Call:
## glm(formula = GermanCredit.train$Class ~ Status + Duration +
##     Credit_history + Savings_Account + Other_guarantors + Employment +
##     Other_installment + Property + Purpose + Num_existingcredit +
##     Foreign_worker + Status_Sex + Installment_rate + Credit_Amount,
##     family = binomial(link = "logit"), data = GermanCredit.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6168  -0.7655   0.4134   0.7108   1.9872
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -4.485e+00  1.069e+00  -4.197 2.71e-05 ***
## Status              6.746e-01  8.483e-02   7.952 1.84e-15 ***
## Duration           -2.765e-02  9.957e-03  -2.777 0.005491 **
## Credit_history      4.150e-01  1.041e-01   3.987 6.70e-05 ***
## Savings_Account     2.561e-01  7.106e-02   3.603 0.000314 ***
## Other_guarantors    3.804e-01  2.084e-01   1.825 0.067971 .
## Employment          2.418e-01  8.221e-02   2.941 0.003273 **
## Other_installment   3.285e-01  1.315e-01   2.499 0.012468 *
## Property           -1.215e-01  1.036e-01  -1.173 0.240836
## Purpose             7.477e-02  3.732e-02   2.004 0.045121 *
## Num_existingcredit -2.989e-01  1.861e-01  -1.606 0.108303
## Foreign_worker      9.740e-01  6.710e-01   1.452 0.146638
## Status_Sex          2.431e-01  1.401e-01   1.735 0.082753 .
## Installment_rate   -2.278e-01  9.752e-02  -2.335 0.019518 *
## Credit_Amount      -8.757e-05  4.441e-05  -1.972 0.048642 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 876.10  on 699  degrees of freedom
## Residual deviance: 660.01  on 685  degrees of freedom
## AIC: 690.01
##
## Number of Fisher Scoring iterations: 5
```

```
best_model$aic
```

```
## [1] 690.0126
```

```r
# generate confusion matrix for training data
predict_logistic_train <- ifelse(predict(best_model,type="response")>0.5,1,0)
confusionMatrix(GermanCredit.train$Class,predict_logistic_train)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 125  98
##          1  55 422
##
##                Accuracy : 0.7814
##                  95% CI : (0.7489, 0.8115)
##     No Information Rate : 0.7429
##     P-Value [Acc > NIR] : 0.010057
##
##                   Kappa : 0.4693
##  Mcnemar's Test P-Value : 0.000685
##
##             Sensitivity : 0.6944
##             Specificity : 0.8115
##          Pos Pred Value : 0.5605
##          Neg Pred Value : 0.8847
##              Prevalence : 0.2571
##          Detection Rate : 0.1786
##    Detection Prevalence : 0.3186
##       Balanced Accuracy : 0.7530
##
##        'Positive' Class : 0
##
```

```r
# generate confusion matrix for test data
predict_logistic <- ifelse(predict(best_model,newdata=GermanCredit.test,type="response")>0.5,1,0)
# fitted_values <- ifelse(best_model$fitted.values>0.5,'Good','Bad')
# GermanCredit.train$Class <- ifelse(GermanCredit.train$Class==1,'Good','Bad')
confusionMatrix(GermanCredit.test$Class,predict_logistic)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  35  42
##          1  30 193
##
##                Accuracy : 0.76
##                  95% CI : (0.7076, 0.8072)
##     No Information Rate : 0.7833
##     P-Value [Acc > NIR] : 0.8531
##
##                   Kappa : 0.3372
##  Mcnemar's Test P-Value : 0.1949
##
##             Sensitivity : 0.5385
##             Specificity : 0.8213
##          Pos Pred Value : 0.4545
```

```
##          Neg Pred Value : 0.8655
##              Prevalence : 0.2167
##          Detection Rate : 0.1167
##    Detection Prevalence : 0.2567
##       Balanced Accuracy : 0.6799
##
##        'Positive' Class : 0
##
```

Decision tree

```r
library(rpart)
library(rpart.plot)
GermanCredit.train$Class <- as.factor(GermanCredit.train$Class)
GermanCredit.test$Class <- as.factor(GermanCredit.test$Class)

set.seed(235)
Credit_tree <- rpart(GermanCredit.train$Class~.,data=GermanCredit.train,control=rpart.control(cp=0,mins

set.seed(345)
printcp(Credit_tree)
```

```
##
## Classification tree:
## rpart(formula = GermanCredit.train$Class ~ ., data = GermanCredit.train,
##     control = rpart.control(cp = 0, minsplit = 30, xval = 10,
##         maxsurrogate = 0))
##
## Variables actually used in tree construction:
##  [1] Age               Credit_Amount     Credit_history
##  [4] Duration          Installment_rate  Other_guarantors
##  [7] Present_residence Property          Savings_Account
## [10] Status            Status_Sex
##
## Root node error: 223/700 = 0.31857
##
## n= 700
##
##          CP nsplit rel error  xerror     xstd
## 1 0.0627803      0   1.00000 1.00000 0.055279
## 2 0.0194320      3   0.81166 0.87444 0.053187
## 3 0.0179372     10   0.62780 0.89686 0.053598
## 4 0.0089686     11   0.60987 0.91031 0.053836
## 5 0.0000000     15   0.57399 0.92825 0.054145
```

```r
num<- which.min(Credit_tree$cptable[,4])
min_cp<- Credit_tree$cptable[num,1]
minimum_xerror <- Credit_tree$cptable[num,4]
cbind(num=num,min_cp=min_cp,minimum_xerror = minimum_xerror)
```

```
##   num     min_cp minimum_xerror
## 2   2 0.01943199      0.8744395
```

```r
set.seed(125)
tree_model<-rpart(GermanCredit.train$Class~.,data=GermanCredit.train,control=rpart.control(cp=min_cp,mi
# generate confusion matrix for training data
predict_tree_train <- predict(tree_model,type="class")
confusionMatrix(GermanCredit.train$Class,predict_tree_train)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 152  71
##          1 110 367
##
##                Accuracy : 0.7414
##                  95% CI : (0.7073, 0.7735)
##     No Information Rate : 0.6257
##     P-Value [Acc > NIR] : 5.439e-11
##
##                   Kappa : 0.4309
##  Mcnemar's Test P-Value : 0.004735
##
##             Sensitivity : 0.5802
##             Specificity : 0.8379
##          Pos Pred Value : 0.6816
##          Neg Pred Value : 0.7694
##              Prevalence : 0.3743
##          Detection Rate : 0.2171
##    Detection Prevalence : 0.3186
##       Balanced Accuracy : 0.7090
##
##        'Positive' Class : 0
##
```

```r
# generate confusion matrix for test data
predict_tree <- predict(tree_model,newdata=GermanCredit.test,type="class")
confusionMatrix(GermanCredit.test$Class,predict_tree)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  42  35
##          1  59 164
##
##                Accuracy : 0.6867
##                  95% CI : (0.6309, 0.7387)
##     No Information Rate : 0.6633
##     P-Value [Acc > NIR] : 0.21430
##
##                   Kappa : 0.2549
##  Mcnemar's Test P-Value : 0.01768
##
##             Sensitivity : 0.4158
##             Specificity : 0.8241
```

```
##          Pos Pred Value : 0.5455
##          Neg Pred Value : 0.7354
##              Prevalence : 0.3367
##          Detection Rate : 0.1400
##    Detection Prevalence : 0.2567
##       Balanced Accuracy : 0.6200
##
##        'Positive' Class : 0
##
```

Ensemble model

```
set.seed(120)
Ensemble_model <- function(results){

ensemble <- rep(NA,nrow(results))
for (i in 1:nrow(results)){
count_1 <- as.numeric(table(results[i,])[names(table(results[i,]))==1])
count_0 <- as.numeric(table(results[i,])[names(table(results[i,]))==0])
if (length(count_1)==0){
  count_1 <- 0
}else if(length(count_0)==0){
  count_0 <-0
 }
if (count_1 > count_0) {
  ensemble[i] <- 1
}else if(count_1 < count_0){
   ensemble[i] <- 0
}
else {
  ensemble[i] <- sample(c(0,1),replace=TRUE,size=1)
   }
 }
return(ensemble)
}

# predict observations in training using ensemble model
predict.results.train <- data.frame(
# LDA
predict_lda=predict_lda_train,
# QDA
predict_qda=predict_qda_train,
# Logistic regression
predict_logistic=predict_logistic_train,
# Decision tree
predict_tree=predict_tree_train
)

predict.results.train <- as.matrix(predict.results.train)
head(predict.results.train)
```

```
##     predict_lda predict_qda predict_logistic predict_tree
## 746 "1"         "1"         "1"              "0"
```

```
## 781 "1"         "1"         "1"              "0"
## 20  "1"         "1"         "1"              "1"
## 774 "1"         "1"         "1"              "1"
## 67  "1"         "1"         "1"              "1"
## 642 "1"         "1"         "1"              "0"
```

```r
predict.ensemble.train <- Ensemble_model(predict.results.train)
confusionMatrix(GermanCredit.train$Class,predict.ensemble.train)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 137  86
##          1  58 419
##
##                Accuracy : 0.7943
##                  95% CI : (0.7624, 0.8237)
##     No Information Rate : 0.7214
##     P-Value [Acc > NIR] : 5.857e-06
##
##                   Kappa : 0.5098
##  Mcnemar's Test P-Value : 0.02445
##
##             Sensitivity : 0.7026
##             Specificity : 0.8297
##          Pos Pred Value : 0.6143
##          Neg Pred Value : 0.8784
##              Prevalence : 0.2786
##          Detection Rate : 0.1957
##    Detection Prevalence : 0.3186
##       Balanced Accuracy : 0.7661
##
##        'Positive' Class : 0
##
```

```r
set.seed(120)
# predict observations in test using ensemble model
predict.results.test <- data.frame(
# LDA
predict_lda=predict_lda,
# QDA
predict_qda=predict_qda,
# Logistic regression
predict_logistic=predict_logistic,
# Decision tree
predict_tree=predict_tree
)

predict.results.test <- as.matrix(predict.results.test)
head(predict.results.test)
```

```
##    predict_lda predict_qda predict_logistic predict_tree
## 1  "1"         "1"         "1"              "1"
## 13 "1"         "1"         "1"              "0"
```

```
## 16 "0"         "0"         "0"              "0"
## 23 "1"         "1"         "1"              "1"
## 24 "1"         "1"         "1"              "0"
## 25 "1"         "1"         "1"              "1"
```

```r
predict.ensemble.test <- Ensemble_model(predict.results.test)
confusionMatrix(GermanCredit.test$Class,predict.ensemble.test)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  40  37
##          1  36 187
##
##                Accuracy : 0.7567
##                  95% CI : (0.704, 0.8041)
##     No Information Rate : 0.7467
##     P-Value [Acc > NIR] : 0.3737
##
##                   Kappa : 0.3596
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.5263
##             Specificity : 0.8348
##          Pos Pred Value : 0.5195
##          Neg Pred Value : 0.8386
##              Prevalence : 0.2533
##          Detection Rate : 0.1333
##    Detection Prevalence : 0.2567
##       Balanced Accuracy : 0.6806
##
##        'Positive' Class : 0
##
```

From the resluts of previous three models, we could see that the Quadratic Discriminant Analysis has the best performance with the overall accuracy equal to 0.8086 for trianing data, and 0.6846 for test data. In the ensemble model the overall accuracy for training data is 0.7943, which is slightly worse than the Quadratic Discriminant Analysis. But the overall accuracy for test data is 0.7567, slightly better than the Quadratic Discriminant model. To choose the model with the best prediction of "bad", we refer to the result of sensitivity, Among the previous three models, logistic regression has the best performance to predict "bad", with the sensitivity value equal to 0.6944 for training set and 0.5385 for test set. And in ensemble model the sensitivity value for training and test samples are 0.7026 and 0.5263 respectively. Therefore, the ensemble model does not have a significant better performance as we expected, and to improve this model we may further consider replace the worst performance model such as decision tree with better model such as logistic regression.

```r
# predict observations in training using ensemble model
predict.results.train <- data.frame(
# QDA
predict_lda=predict_qda_train,
# QDA
predict_qda=predict_qda_train,
```

```r
# Logistic regression
predict_logistic=predict_logistic_train,
# Logistic regression
predict_logistic=predict_logistic_train
)

predict.results.train.enhance <- as.matrix(predict.results.train)

predict.ensemble.train.enhance <- Ensemble_model(predict.results.train.enhance)
confusionMatrix(GermanCredit.train$Class,predict.ensemble.train.enhance)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 146  77
##          1  61 416
##
##                Accuracy : 0.8029
##                  95% CI : (0.7714, 0.8317)
##     No Information Rate : 0.7043
##     P-Value [Acc > NIR] : 1.908e-09
##
##                   Kappa : 0.5371
##  Mcnemar's Test P-Value : 0.2016
##
##             Sensitivity : 0.7053
##             Specificity : 0.8438
##          Pos Pred Value : 0.6547
##          Neg Pred Value : 0.8721
##              Prevalence : 0.2957
##          Detection Rate : 0.2086
##    Detection Prevalence : 0.3186
##       Balanced Accuracy : 0.7746
##
##        'Positive' Class : 0
##
```

```r
set.seed(120)
# predict observations in test using ensemble model
predict.results.test <- data.frame(
# QDA
predict_lda=predict_qda,
# QDA
predict_qda=predict_qda,
# Logistic regression
predict_logistic=predict_logistic,
# Logistic regression
predict_tree=predict_logistic
)

predict.results.test <- as.matrix(predict.results.test)
head(predict.results.test)
```

```
##    predict_lda predict_qda predict_logistic predict_tree
## 1  "1"         "1"         "1"              "1"
## 13 "1"         "1"         "1"              "1"
## 16 "0"         "0"         "0"              "0"
## 23 "1"         "1"         "1"              "1"
## 24 "1"         "1"         "1"              "1"
## 25 "1"         "1"         "1"              "1"
```

```
predict.ensemble.test <- Ensemble_model(predict.results.test)
confusionMatrix(GermanCredit.test$Class,predict.ensemble.test)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  39  38
##          1  38 185
##
##                Accuracy : 0.7467
##                  95% CI : (0.6935, 0.7949)
##     No Information Rate : 0.7433
##     P-Value [Acc > NIR] : 0.4779
##
##                   Kappa : 0.3361
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.5065
##             Specificity : 0.8296
##          Pos Pred Value : 0.5065
##          Neg Pred Value : 0.8296
##              Prevalence : 0.2567
##          Detection Rate : 0.1300
##    Detection Prevalence : 0.2567
##       Balanced Accuracy : 0.6680
##
##        'Positive' Class : 0
##
```

After replacing the Linear Discriminant Analysis with Quadratic Discriminant Analysis, and decision tree with logistic regression, the accuracy and sensitivity of ensemble model has increased to 0.8014 and 0.7019 respectively for training samples, and the accuracy for test samples increased to 0.7467, but the sensitivity is still low, with only 0.5065.