Class 6 - Classification Analysis

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(mice)
```

```
##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```r
data = read.csv("G:/My Drive/Master-Data-Science/Semester_1/Business_Analytics/Data/index.csv", header=
str(data)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ Creditability                    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Account.Balance                  : int  1 1 2 1 1 1 1 1 4 2 ...
##  $ Duration.of.Credit..month.       : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ Payment.Status.of.Previous.Credit: int  4 4 2 4 4 4 4 4 4 2 ...
##  $ Purpose                          : int  2 0 9 0 0 0 0 0 3 3 ...
##  $ Credit.Amount                    : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
##  $ Value.Savings.Stocks             : int  1 1 2 1 1 1 1 1 1 3 ...
##  $ Length.of.current.employment     : int  2 3 4 3 3 2 4 2 1 1 ...
##  $ Instalment.per.cent              : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ Sex...Marital.Status             : int  2 3 2 3 3 3 3 3 2 2 ...
##  $ Guarantors                       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Duration.in.Current.address      : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ Most.valuable.available.asset    : int  2 1 1 1 2 1 1 1 3 4 ...
##  $ Age..years.                      : int  21 36 23 39 38 48 39 40 65 23 ...
```

```
##  $ Concurrent.Credits              : int  3 3 3 3 1 3 3 3 3 3 ...
##  $ Type.of.apartment               : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ No.of.Credits.at.this.Bank      : int  1 2 1 2 2 2 2 1 2 1 ...
##  $ Occupation                      : int  3 3 2 2 2 2 2 2 1 1 ...
##  $ No.of.dependents                : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ Telephone                       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Foreign.Worker                  : int  1 1 1 2 2 2 2 2 1 1 ...
```

```
unique(data$No.of.Credits.at.this.Bank)
```

```
## [1] 1 2 3 4
```

```
table(data$Purpose)/1000*100
```

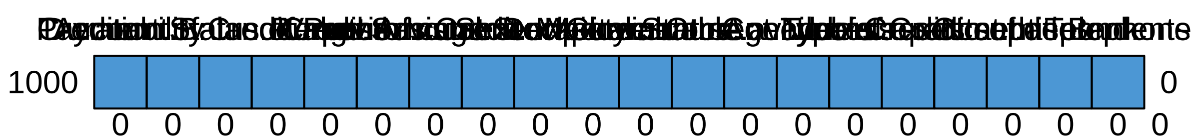
```
##
##    0    1    2    3    4    5    6    8    9   10
## 23.4 10.3 18.1 28.0  1.2  2.2  5.0  0.9  9.7  1.2
```

```
data$Creditability = as.factor(data$Creditability)
data$Account.Balance <- replace(data$Account.Balance, data$Account.Balance==4, 3)
data$Account.Balance = factor(data$Account.Balance, levels = seq(1,3), labels = c('No Account', 'No bala

data$Payment.Status.of.Previous.Credit[data$Payment.Status.of.Previous.Credit <=1] =1
data$Payment.Status.of.Previous.Credit[data$Payment.Status.of.Previous.Credit ==2] = 2
data$Payment.Status.of.Previous.Credit[data$Payment.Status.of.Previous.Credit >=3] = 3
data$Payment.Status.of.Previous.Credit = factor(data$Payment.Status.of.Previous.Credit, levels = seq(1,3

data$Value.Savings.Stocks[data$Value.Savings.Stocks == 4] = 3
data$Value.Savings.Stocks[data$Value.Savings.Stocks == 5] = 4
data$Value.Savings.Stocks = factor(data$Value.Savings.Stocks, levels = seq(1,4), labels = c('None','Bel

data$Length.of.current.employment[data$Length.of.current.employment == 2] = 1
data$Length.of.current.employment[data$Length.of.current.employment == 3] = 2
data$Length.of.current.employment[data$Length.of.current.employment == 4] = 3
data$Length.of.current.employment[data$Length.of.current.employment == 5] = 4
data$Length.of.current.employment = factor(data$Length.of.current.employment, levels = seq(1,4), labels

data$Sex...Marital.Status[data$Sex...Marital.Status <=2] = 1
data$Sex...Marital.Status[data$Sex...Marital.Status ==3] = 2
data$Sex...Marital.Status[data$Sex...Marital.Status ==4] = 3
data$Sex...Marital.Status = factor(data$Sex...Marital.Status, levels = seq(1,3), labels = c('Male Divor

data$No.of.Credits.at.this.Bank[data$No.of.Credits.at.this.Bank >= 3] = 2
data$No.of.Credits.at.this.Bank = factor(data$No.of.Credits.at.this.Bank, levels = seq(1,2), labels = c

data$Guarantors[data$Guarantors >= 2] = 2
data$Guarantors = factor(data$Guarantors, levels = seq(1,2), labels = c('None','Yes'))

data$Concurrent.Credits[data$Concurrent.Credits <=2] = 1
data$Concurrent.Credits[data$Concurrent.Credits ==3] = 2
data$Concurrent.Credits = factor(data$Concurrent.Credits, levels = seq(1,2), labels = c('Other Banks or
```

```r
data = data[-21]

data$Purpose[data$Purpose ==1] = 1
data$Purpose[data$Purpose ==2] = 2
data$Purpose[data$Purpose %in% c(3,4,5,6)] = 3
data$Purpose[data$Purpose %in% c(8,9,10,0)] = 4
data$Purpose = factor(data$Purpose, levels = seq(1,4), labels = c('New Car','Used Car','Home Related','
```

```r
md.pattern(data)
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```



```
##      Creditability Account.Balance Duration.of.Credit..month.
## 1000             1               1                          1
##                  0               0                          0
##      Payment.Status.of.Previous.Credit Purpose Credit.Amount
## 1000                                 1       1             1
##                                      0       0             0
##      Value.Savings.Stocks Length.of.current.employment Instalment.per.cent
```

```
## 1000                    1                             1                   1
##                         0                             0                   0
##      Sex...Marital.Status Guarantors Duration.in.Current.address
## 1000                    1          1                           1
##                         0          0                           0
##      Most.valuable.available.asset Age..years. Concurrent.Credits
## 1000                             1           1                  1
##                                  0           0                  0
##      Type.of.apartment No.of.Credits.at.this.Bank Occupation No.of.dependents
## 1000                 1                          1          1                1
##                      0                          0          0                0
##      Telephone
## 1000         1 0
##              0 0
```

```
str(data)
```

```
## 'data.frame':    1000 obs. of  20 variables:
##  $ Creditability                    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Account.Balance                  : Factor w/ 3 levels "No Account","No balance",..: 1 1 2 1 1 1 1
##  $ Duration.of.Credit..month.       : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ Payment.Status.of.Previous.Credit: Factor w/ 3 levels "Some Problems",..: 3 3 2 3 3 3 3 3 3 2 ...
##  $ Purpose                          : Factor w/ 4 levels "New Car","Used Car",..: 2 4 4 4 4 4 4 4 3 3
##  $ Credit.Amount                    : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
##  $ Value.Savings.Stocks             : Factor w/ 4 levels "None","Below 100 DM",..: 1 1 2 1 1 1 1 1 1
##  $ Length.of.current.employment     : Factor w/ 4 levels "Below 1 year (including unemployed)",..: 1
##  $ Instalment.per.cent              : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ Sex...Marital.Status             : Factor w/ 3 levels "Male Divorces/Single",..: 1 2 1 2 2 2 2 2
##  $ Guarantors                       : Factor w/ 2 levels "None","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Duration.in.Current.address      : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ Most.valuable.available.asset    : int  2 1 1 1 2 1 1 1 3 4 ...
##  $ Age..years.                      : int  21 36 23 39 38 48 39 40 65 23 ...
##  $ Concurrent.Credits               : Factor w/ 2 levels "Other Banks or Dept Stores",..: 2 2 2 2 1 2
##  $ Type.of.apartment                : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ No.of.Credits.at.this.Bank       : Factor w/ 2 levels "1","More than 1": 1 2 1 2 2 2 2 1 2 1 ...
##  $ Occupation                       : int  3 3 2 2 2 2 2 2 1 1 ...
##  $ No.of.dependents                 : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ Telephone                        : int  1 1 1 1 1 1 1 1 1 1 ...
```

Statistical Testing

Chi-square for

```
Categorical.Table = data.frame(
    'Variable' = character(),
  'p-value' = numeric()
)

for (i in colnames(data[,-c(1,3,6,14)])){
  test = chisq.test(table(data$Creditability,data[,i]))
  test2 = data.frame(i,test$p.value)
  Categorical.Table = rbind(Categorical.Table, test2)
```

```
}
Categorical.Table
```

```
##                                     i test.p.value
## 1                     Account.Balance 5.742621e-27
## 2  Payment.Status.of.Previous.Credit 1.557328e-12
## 3                             Purpose 2.760708e-04
## 4                Value.Savings.Stocks 8.335937e-08
## 5         Length.of.current.employment 4.220685e-04
## 6                  Instalment.per.cent 1.400333e-01
## 7                Sex...Marital.Status 1.043498e-02
## 8                          Guarantors 1.000000e+00
## 9          Duration.in.Current.address 8.615521e-01
## 10      Most.valuable.available.asset 2.858442e-05
## 11                 Concurrent.Credits 4.763431e-04
## 12                  Type.of.apartment 8.810311e-05
## 13         No.of.Credits.at.this.Bank 1.693042e-01
## 14                         Occupation 5.965816e-01
## 15                   No.of.dependents 1.000000e+00
## 16                          Telephone 2.788762e-01
```

Train test split

```
indexes = sample(1:1000, size = 500)
Train = data[indexes,]
Test = data[-indexes,]
```

Logistic Regression

generalized linear model = glm()

- when y is discrete/binary

$$H_0 \ : \ B_j = 0 H_1 : B_j \neq 0$$

Create initial model

```
logisticmodel50 = glm(Creditability~Account.Balance+Payment.Status.of.Previous.Credit+Purpose+Value.Savi
summary(logisticmodel50)
```

```
##
## Call:
## glm(formula = Creditability ~ Account.Balance + Payment.Status.of.Previous.Credit +
##     Purpose + Value.Savings.Stocks + Length.of.current.employment +
```

```
##     Sex...Marital.Status + Most.valuable.available.asset + Type.of.apartment +
##     Concurrent.Credits + Duration.in.Current.address + Credit.Amount +
##     Age..years., family = "binomial", data = Train)
##
## Coefficients:
##                                                           Estimate
## (Intercept)                                              1.006e-01
## Account.BalanceNo balance                                3.098e-01
## Account.BalanceSome balance                              1.615e+00
## Payment.Status.of.Previous.CreditPaid Up                 7.344e-01
## Payment.Status.of.Previous.CreditNo Problems(in this bank)  1.253e+00
## PurposeUsed Car                                         -1.556e-01
## PurposeHome Related                                     -4.363e-01
## PurposeOther                                            -9.007e-01
## Value.Savings.StocksBelow 100 DM                         1.149e-01
## Value.Savings.Stocks[100, 1000)                          8.636e-01
## Value.Savings.StocksAbove 1000 DM                        1.336e+00
## Length.of.current.employment[1,4)                       -4.843e-01
## Length.of.current.employment[4,7)                        1.024e+00
## Length.of.current.employmentAbove 7                      2.128e-01
## Sex...Marital.StatusMale Married/Widowed                 5.081e-01
## Sex...Marital.StatusFemale                               3.474e-01
## Most.valuable.available.asset                           -2.706e-01
## Type.of.apartment                                       -2.999e-02
## Concurrent.CreditsNone                                   1.839e-01
## Duration.in.Current.address                             -1.644e-01
## Credit.Amount                                           -8.544e-05
## Age..years.                                              1.613e-02
##                                                         Std. Error z value
## (Intercept)                                              8.890e-01   0.113
## Account.BalanceNo balance                                2.914e-01   1.063
## Account.BalanceSome balance                              2.963e-01   5.453
## Payment.Status.of.Previous.CreditPaid Up                 3.754e-01   1.956
## Payment.Status.of.Previous.CreditNo Problems(in this bank)  3.889e-01   3.221
## PurposeUsed Car                                          5.251e-01  -0.296
## PurposeHome Related                                      4.972e-01  -0.877
## PurposeOther                                             4.841e-01  -1.861
## Value.Savings.StocksBelow 100 DM                         3.578e-01   0.321
## Value.Savings.Stocks[100, 1000)                          4.459e-01   1.937
## Value.Savings.StocksAbove 1000 DM                        4.030e-01   3.315
## Length.of.current.employment[1,4)                        3.016e-01  -1.606
## Length.of.current.employment[4,7)                        4.035e-01   2.537
## Length.of.current.employmentAbove 7                      3.613e-01   0.589
## Sex...Marital.StatusMale Married/Widowed                 2.561e-01   1.984
## Sex...Marital.StatusFemale                               4.416e-01   0.787
## Most.valuable.available.asset                            1.270e-01  -2.131
## Type.of.apartment                                        2.299e-01  -0.130
## Concurrent.CreditsNone                                   2.922e-01   0.629
## Duration.in.Current.address                              1.143e-01  -1.439
## Credit.Amount                                            4.542e-05  -1.881
## Age..years.                                              1.212e-02   1.331
##                                                         Pr(>|z|)
## (Intercept)                                             0.909897
## Account.BalanceNo balance                               0.287789
```

```
## Account.BalanceSome balance                                4.97e-08 ***
## Payment.Status.of.Previous.CreditPaid Up                   0.050442 .
## Payment.Status.of.Previous.CreditNo Problems(in this bank) 0.001278 **
## PurposeUsed Car                                            0.766946
## PurposeHome Related                                        0.380218
## PurposeOther                                               0.062814 .
## Value.Savings.StocksBelow 100 DM                           0.748123
## Value.Savings.Stocks[100, 1000)                            0.052771 .
## Value.Savings.StocksAbove 1000 DM                          0.000917 ***
## Length.of.current.employment[1,4)                          0.108293
## Length.of.current.employment[4,7)                          0.011183 *
## Length.of.current.employmentAbove 7                        0.555811
## Sex...Marital.StatusMale Married/Widowed                   0.047300 *
## Sex...Marital.StatusFemale                                 0.431437
## Most.valuable.available.asset                              0.033086 *
## Type.of.apartment                                          0.896195
## Concurrent.CreditsNone                                     0.529045
## Duration.in.Current.address                                0.150292
## Credit.Amount                                              0.059933 .
## Age..years.                                                0.183337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.16  on 499  degrees of freedom
## Residual deviance: 470.72  on 478  degrees of freedom
## AIC: 514.72
##
## Number of Fisher Scoring iterations: 5
```

Optimize model

```r
logisticmodel50final = glm(Creditability~Account.Balance + Payment.Status.of.Previous.Credit + Purpose
summary(logisticmodel50final)
```

```
##
## Call:
## glm(formula = Creditability ~ Account.Balance + Payment.Status.of.Previous.Credit +
##     Purpose + Length.of.current.employment + Sex...Marital.Status,
##     family = "binomial", data = Train)
##
## Coefficients:
##                                                            Estimate Std. Error
## (Intercept)                                                -0.87234    0.54749
## Account.BalanceNo balance                                   0.35787    0.26145
## Account.BalanceSome balance                                 1.83244    0.27878
## Payment.Status.of.Previous.CreditPaid Up                    0.95727    0.34069
## Payment.Status.of.Previous.CreditNo Problems(in this bank)  1.38639    0.36325
## PurposeUsed Car                                            -0.08772    0.47511
## PurposeHome Related                                        -0.25652    0.44019
## PurposeOther                                               -0.64564    0.43185
```

```
## Length.of.current.employment[1,4)                            -0.35418    0.28674
## Length.of.current.employment[4,7)                             0.92450    0.38003
## Length.of.current.employmentAbove 7                           0.26054    0.32096
## Sex...Marital.StatusMale Married/Widowed                      0.42585    0.23592
## Sex...Marital.StatusFemale                                    0.57170    0.42675
##                                                             z value Pr(>|z|)
## (Intercept)                                                  -1.593 0.111081
## Account.BalanceNo balance                                     1.369 0.171060
## Account.BalanceSome balance                                   6.573 4.93e-11 ***
## Payment.Status.of.Previous.CreditPaid Up                      2.810 0.004957 **
## Payment.Status.of.Previous.CreditNo Problems(in this bank)   3.817 0.000135 ***
## PurposeUsed Car                                              -0.185 0.853526
## PurposeHome Related                                          -0.583 0.560057
## PurposeOther                                                 -1.495 0.134904
## Length.of.current.employment[1,4)                            -1.235 0.216757
## Length.of.current.employment[4,7)                             2.433 0.014987 *
## Length.of.current.employmentAbove 7                           0.812 0.416934
## Sex...Marital.StatusMale Married/Widowed                      1.805 0.071072 .
## Sex...Marital.StatusFemale                                    1.340 0.180355
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.16  on 499  degrees of freedom
## Residual deviance: 503.43  on 487  degrees of freedom
## AIC: 529.43
##
## Number of Fisher Scoring iterations: 4
```

Obtain fitted values

```
fit50 = fitted.values(logisticmodel50final)
head(fit50)
```

```
##       542       873       156       587       818       656
## 0.8229738 0.8403520 0.8967508 0.6352489 0.6877547 0.7135468
```

Change binary response

```
thres = rep(0,500)
for (i in 1:500) {
  if(fit50[i]>0.5) {
    thres[i] = 1
  }
  else {
    thres[i] = 0
  }
}
str(thres)
```

```
##  num [1:500] 1 1 1 1 1 1 1 1 1 1 ...
```

```
str(Train$Creditability)
```

```
##  Factor w/ 2 levels "0","1": 2 1 2 2 1 2 2 2 2 2 ...
```

Create cross table

```
conf.mat = table(Train$Creditability, thres)
conf.mat
```

```
##    thres
##       0   1
##   0  55  94
##   1  40 311
```

Compute accuracy

```
LR_train_acc = sum(diag(conf.mat))/500*100
```

Perform on testing data

print(sum(diag(ct)))

```
library(gmodels)
# Perform modeling on testing data
prr = predict(logisticmodel50final, data=Test,type='response')

# Set threshold
thres_pred=rep(0,500)
for (i in 1:500) {
  if(prr[i]>0.5) {
    thres_pred[i] = 1
  }
  else {
    thres_pred[i] = 0
  }
}
str(thres_pred)
```

```
##  num [1:500] 1 1 1 1 1 1 1 1 1 1 ...
```

```
ct = CrossTable(Test$Creditability, thres_pred, digits=1, prop.r=F, prop.t=F,prop.chisq = F, chisq = F)
```

```
##
##
```

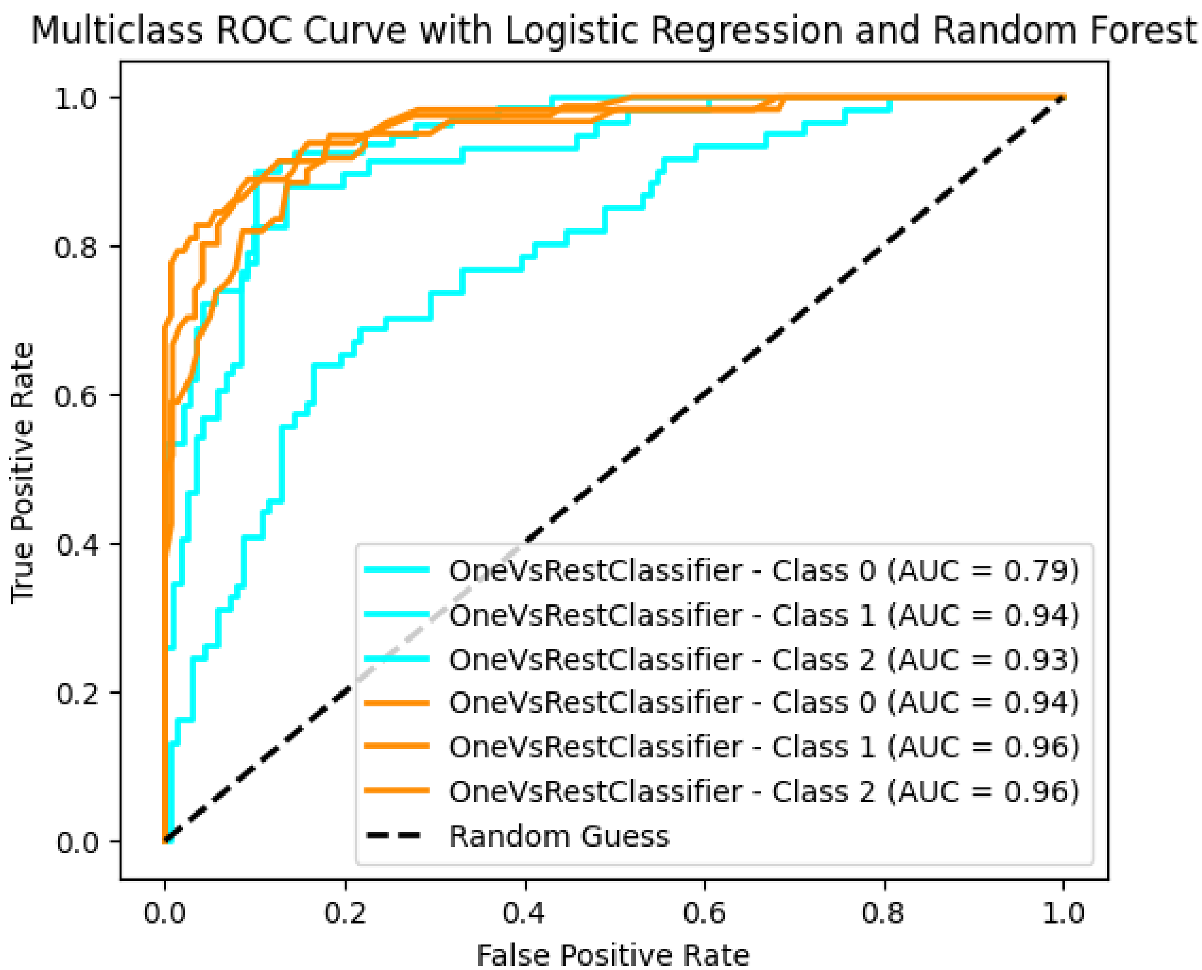
```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##                    | thres_pred
## Test$Creditability |         0 |         1 | Row Total |
## -------------------|-----------|-----------|-----------|
##                  0 |        31 |       120 |       151 |
##                    |       0.3 |       0.3 |           |
## -------------------|-----------|-----------|-----------|
##                  1 |        64 |       285 |       349 |
##                    |       0.7 |       0.7 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        95 |       405 |       500 |
##                    |       0.2 |       0.8 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

```
print(ct)
```

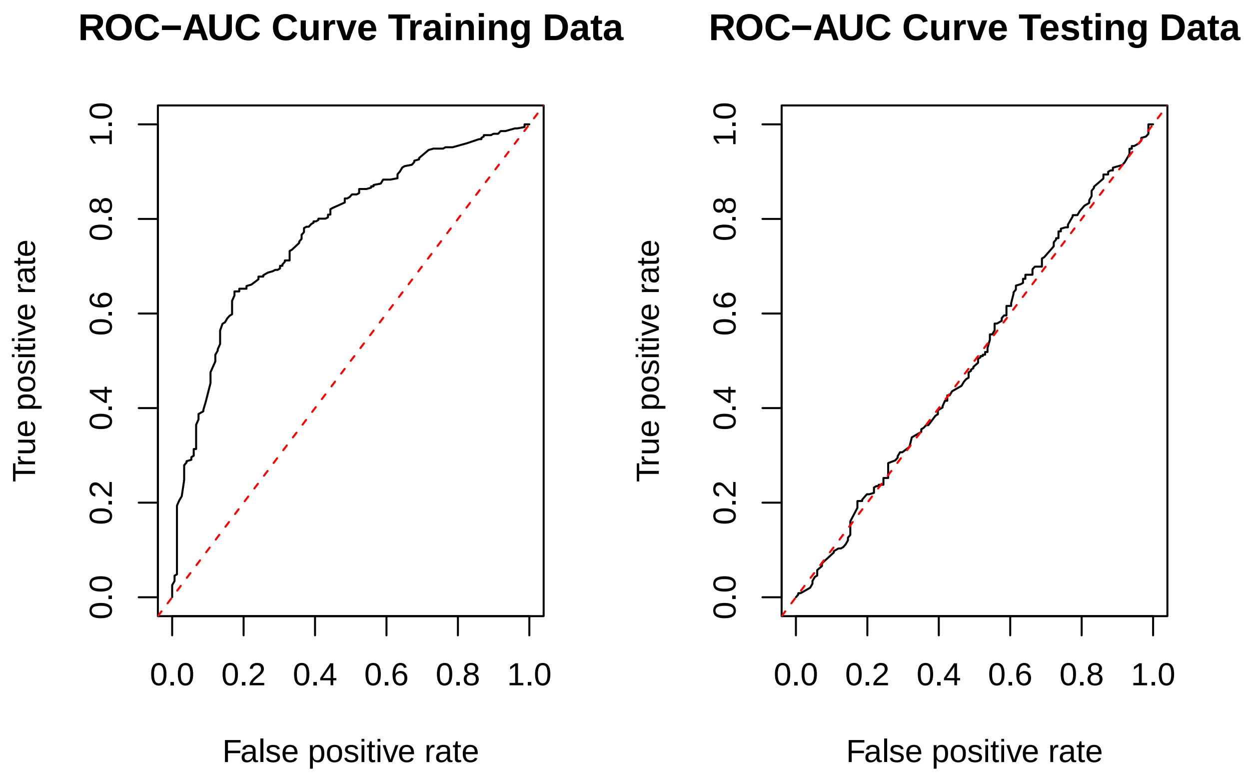
```
## $t
##    y
## x     0   1
##   0  31 120
##   1  64 285
##
## $prop.row
##    y
## x           0         1
##   0 0.2052980 0.7947020
##   1 0.1833811 0.8166189
##
## $prop.col
##    y
## x           0         1
##   0 0.3263158 0.2962963
##   1 0.6736842 0.7037037
##
## $prop.tbl
##    y
## x       0     1
##   0 0.062 0.240
##   1 0.128 0.570
```

```
conf.mat2 = table(Test$Creditability, thres_pred)
LR_test_acc = sum(diag(conf.mat2))/500*100
```

Plot ROC AUC Curve




```r
library(ROCR)


par(mfrow=c(1,2))
#Training Data
prod_pred = prediction(fit50, Train$Creditability)
perf = performance(prod_pred,'tpr','fpr')
plot(perf, main='ROC-AUC Curve Training Data');abline(a = 0, b = 1, col = "red", lty = 2)

#Testing Data
prod_pred = prediction(prr, Test$Creditability)
perf = performance(prod_pred,'tpr','fpr')
plot(perf, main='ROC-AUC Curve Testing Data');abline(a = 0, b = 1, col = "red", lty = 2)
```



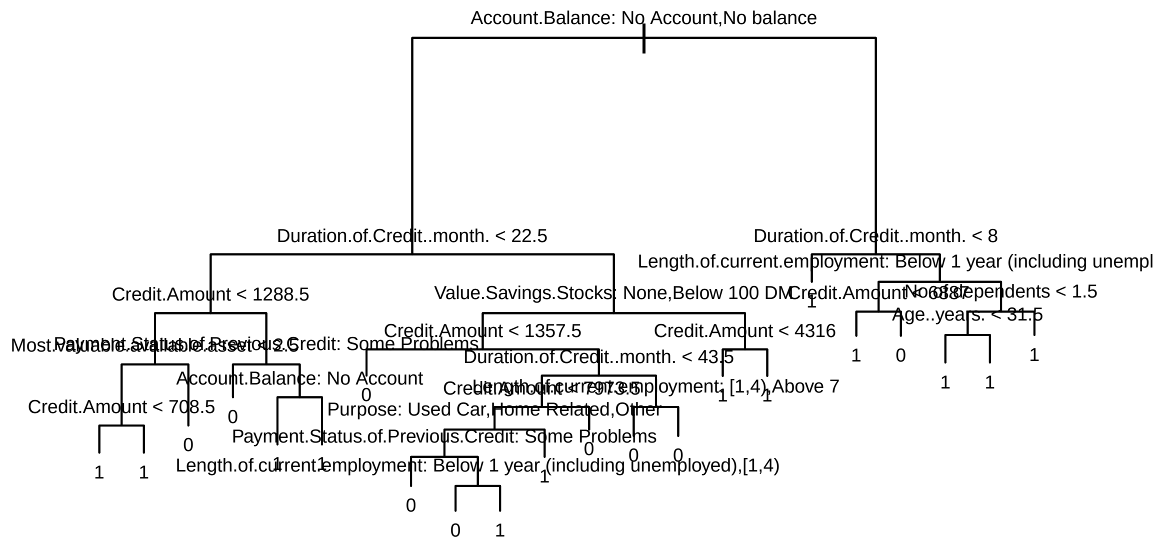
Tree Based Index

Gini Index - recursive partition

Model Building

```
library(tree)

tree_model = tree(Creditability ~ Account.Balance+Duration.of.Credit..month.+Payment.Status.of.Previous
Guarantors+Duration.in.Current.address+Most.valuable.available.asset+Age..years.+Concurrent.Credits+Typ
```

```
plot(tree_model);text(tree_model, pretty=0, cex=0.6)
```



Evaluate Train Set

```r
train_pred = predict(tree_model,Train, type='class')
ct1 = table(Train$Creditability, train_pred)
T_Train_acc = sum(diag(ct1))/500*100
```

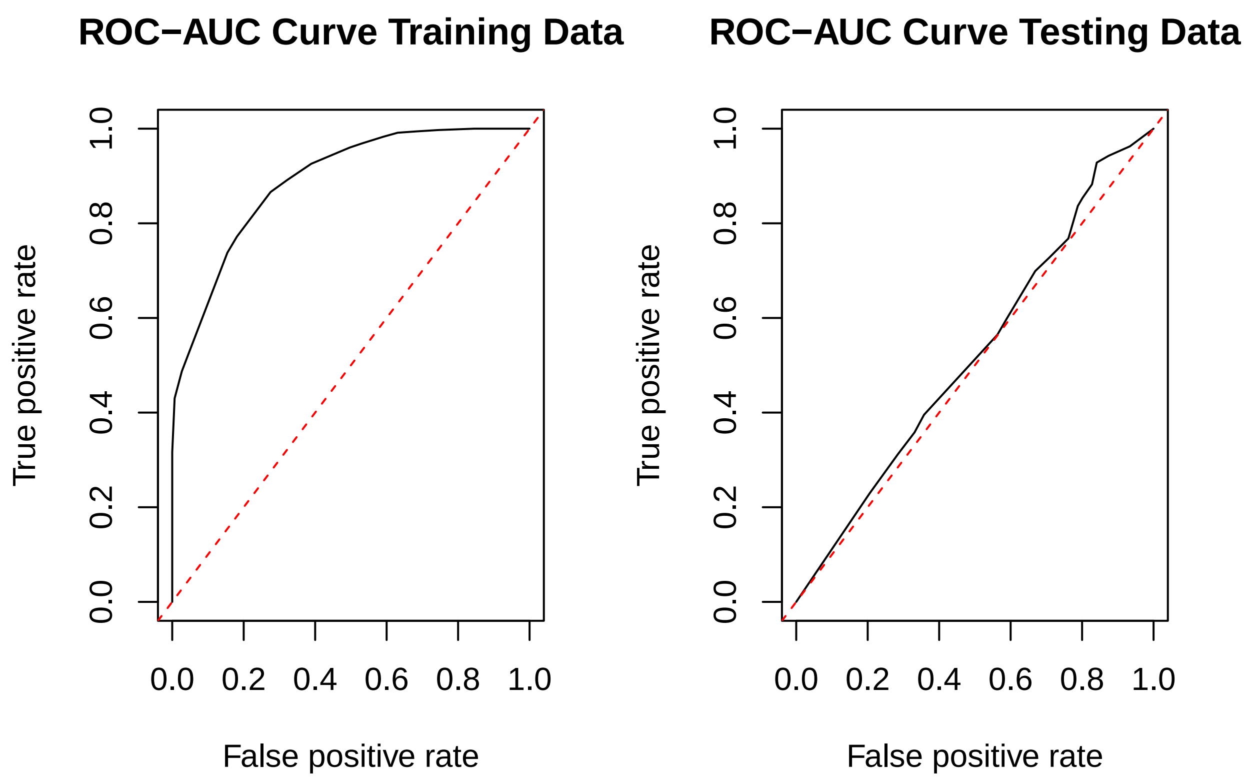
Evaluate Test Set

```r
test_pred = predict(tree_model,Test, type='class')
ct2 = table(Train$Creditability, test_pred)
T_Test_acc = sum(diag(ct2))/500*100
```

ROC AUC Curve

```r
# Training Data
fit_tree_train = predict(tree_model, data=Train)

prod_pred = prediction(fit_tree_train[,2], Train$Creditability)
perf1 = performance(prod_pred,'tpr','fpr')
```

```
# Testing Data
fit_tree_test = predict(tree_model, data=Test)

prod_pred = prediction(fit_tree_test[,2], Test$Creditability)
perf2 = performance(prod_pred,'tpr','fpr')

par(mfrow=c(1,2))
plot(perf1, main='ROC-AUC Curve Training Data');abline(a = 0, b = 1, col = "red", lty = 2)
plot(perf2, main='ROC-AUC Curve Testing Data');abline(a = 0, b = 1, col = "red", lty = 2)
```



Pruning

```
tree_model_prune = prune.misclass(tree_model, best=8)
```

Evaluate Train Set

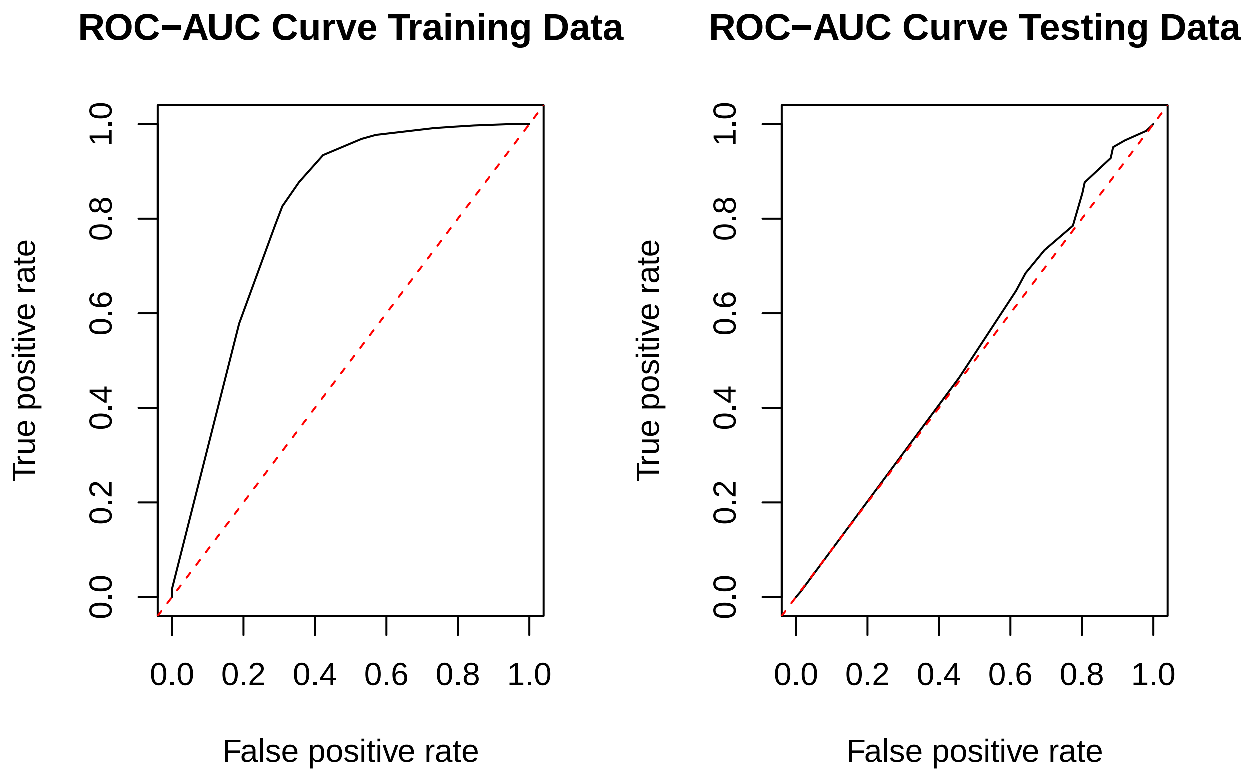
```
train_prune_pred = predict(tree_model_prune,Train, type='class')
ct3 = table(Train$Creditability, train_prune_pred)
TP_Train_acc= sum(diag(ct3))/500*100
```

Evaluate Test Set

```
test_prune_pred = predict(tree_model_prune,Test, type='class')
ct4 = table(Train$Creditability, test_prune_pred)
TP_Test_acc = sum(diag(ct4))/500*100
```

ROC AUC Curve

```
# Train Dataset

fit_tree_prune_train = predict(tree_model_prune, data=Train)

prod_pred = prediction(fit_tree_prune_train[,2], Train$Creditability)
perf1 = performance(prod_pred,'tpr','fpr')


# Test Dataset
fit_tree_prune_test = predict(tree_model_prune, data=Test)

prod_pred = prediction(fit_tree_prune_test[,2], Test$Creditability)
perf2 = performance(prod_pred,'tpr','fpr')

par(mfrow=c(1,2))
plot(perf1, main='ROC-AUC Curve Training Data');abline(a = 0, b = 1, col = "red", lty = 2)
plot(perf2, main='ROC-AUC Curve Testing Data');abline(a = 0, b = 1, col = "red", lty = 2)
```



Combine accuracy

```
data.frame(
  Models = c('Logistic Regression','Tree','Tree Pruned'),
  Train = c(LR_train_acc,T_Train_acc,TP_Train_acc),
  Test = c(LR_test_acc,T_Test_acc,TP_Test_acc)
)
```

```
##                Models Train Test
## 1 Logistic Regression  73.2 63.2
## 2                Tree  83.2 62.8
## 3         Tree Pruned  82.8 64.2
```