# ML_original_data

Penca Matei

12/23/2021

### Packages & Data

```
library(readr)
library(performance)
library(gmodels)
library(caret)
library(tidyverse)
library(tree)
library(randomForest)
```

### Loading and setting up the data

```
# Load data into the environment and creates partition for 50-50 validation

german_credit <- read_csv("data/german_credit.csv")


Train50 <- read_csv("data/Training50.csv")


Test50 <- read_csv("data/Test50.csv")


attach(Train50)
```

### Logistic regression model with 50:50 Cross-validation

```
LogisticModel50final <- glm(Creditability ~ Account.Balance +
                              Payment.Status.of.Previous.Credit + Purpose +
                              Length.of.current.employment +
                              Sex...Marital.Status, family=binomial,
                            data = Train50)

fit50 <- fitted.values(LogisticModel50final)

Threshold50 <- rep(0,500)
```

```
for (i in 1:500)
  if(fit50[i] >= 0.5) Threshold50[i] <- 1
CrossTable(Train50$Creditability, Threshold50, digits=1, prop.r=F, prop.t=F,
           prop.chisq=F, chisq=F, data=Train50)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##                       | Threshold50
## Train50$Creditability |         0 |         1 | Row Total |
## ----------------------|-----------|-----------|-----------|
##                     0 |        47 |        96 |       143 |
##                       |       0.6 |       0.2 |           |
## ----------------------|-----------|-----------|-----------|
##                     1 |        30 |       327 |       357 |
##                       |       0.4 |       0.8 |           |
## ----------------------|-----------|-----------|-----------|
##          Column Total |        77 |       423 |       500 |
##                       |       0.2 |       0.8 |           |
## ----------------------|-----------|-----------|-----------|
##
##
```

```
for (i in 1:500) {
  if (Threshold50[i] == '1') {
    Threshold50[i] <- 'Creditable'
  }

  if (Threshold50[i] == '0') {
    Threshold50[i] <- 'Non-Creditable'
  }
}

for (i in 1:500) {
  if (Test50$Creditability[i] == '0') {
    Test50$Creditability[i] <- 'Non-Creditable'
  }

  if (Test50$Creditability[i] == '1') {
    Test50$Creditability[i] <- 'Creditable'
  }
}
```

### Confusion matrix for GLM

```
confusion <- confusionMatrix(data = factor(Threshold50),
                             reference = factor(Test50$Creditability))
confusion
```

```
## Confusion Matrix and Statistics
##
##                  Reference
## Prediction       Creditable Non-Creditable
##   Creditable            291            132
##   Non-Creditable         52             25
##
##                Accuracy : 0.632
##                  95% CI : (0.588, 0.6744)
##     No Information Rate : 0.686
##     P-Value [Acc > NIR] : 0.9956
##
##                   Kappa : 0.0089
##
##  Mcnemar’s Test P-Value : 5.747e-09
##
##             Sensitivity : 0.8484
##             Specificity : 0.1592
##          Pos Pred Value : 0.6879
##          Neg Pred Value : 0.3247
##              Prevalence : 0.6860
##          Detection Rate : 0.5820
##    Detection Prevalence : 0.8460
##       Balanced Accuracy : 0.5038
##
##        ’Positive’ Class : Creditable
##
```

### Supervised Tree based method

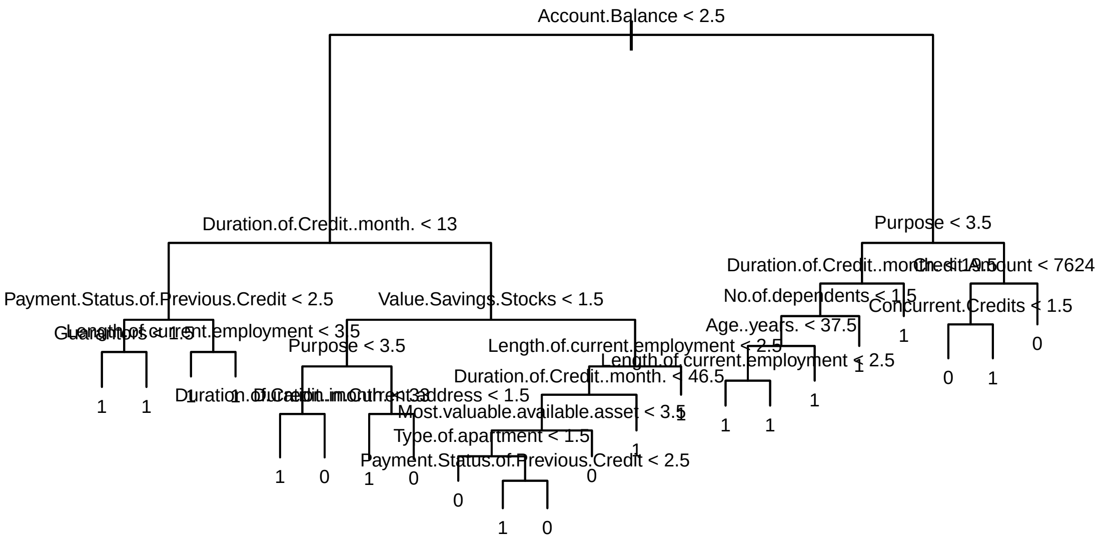
```
# Reload the clean test data set into memory
Test50 <- read_csv("data/Test50.csv")

Train50_tree <- tree(as.factor(Creditability) ~ Account.Balance+
                       Duration.of.Credit..month.+
                       Payment.Status.of.Previous.Credit+Purpose+Credit.Amount+
                       Value.Savings.Stocks+
                       Length.of.current.employment+Instalment.per.cent+
                       Sex...Marital.Status+Guarantors+
                       Duration.in.Current.address+
                       Most.valuable.available.asset+Age..years.+
                       Concurrent.Credits+Type.of.apartment+
                       No.of.Credits.at.this.Bank+Occupation+No.of.dependents+
                       Telephone, data=Train50, method="class")
```

```r
summary(Train50_tree)
```

```
##
## Classification tree:
## tree(formula = as.factor(Creditability) ~ Account.Balance + Duration.of.Credit..month. +
##     Payment.Status.of.Previous.Credit + Purpose + Credit.Amount +
##     Value.Savings.Stocks + Length.of.current.employment + Instalment.per.cent +
##     Sex...Marital.Status + Guarantors + Duration.in.Current.address +
##     Most.valuable.available.asset + Age..years. + Concurrent.Credits +
##     Type.of.apartment + No.of.Credits.at.this.Bank + Occupation +
##     No.of.dependents + Telephone, data = Train50, method = "class")
## Variables actually used in tree construction:
##  [1] "Account.Balance"                   "Duration.of.Credit..month."
##  [3] "Payment.Status.of.Previous.Credit" "Guarantors"
##  [5] "Length.of.current.employment"      "Value.Savings.Stocks"
##  [7] "Purpose"                           "Duration.in.Current.address"
##  [9] "Most.valuable.available.asset"     "Type.of.apartment"
## [11] "No.of.dependents"                  "Age..years."
## [13] "Credit.Amount"                     "Concurrent.Credits"
## Number of terminal nodes:  22
## Residual mean deviance:  0.7682 = 367.2 / 478
## Misclassification error rate: 0.168 = 84 / 500
```

```r
# Tree visual representation
plot(Train50_tree)
text(Train50_tree, pretty=0,cex=0.6)
```


Account.Balance < 2.5
Duration.of.Credit..month. < 13
Purpose < 3.5
Payment.Status.of.Previous.Credit < 2.5
Value.Savings.Stocks < 1.5
Guarantors < 1.5
Length.of.current.employment < 3.5
Purpose < 3.5
Length.of.current.employment < 2.5
Duration.of.Credit..month. < 46.5
Duration.of.Credit..month. < 33
Duration.in.Current.address < 1.5
Most.valuable.available.asset < 3.5
Type.of.apartment < 1.5
Payment.Status.of.Previous.Credit < 2.5
Duration.of.Credit..month. < 19.5
Credit.Amount < 7624
No.of.dependents < 1.5
Concurrent.Credits < 1.5
Age..years. < 37.5
Length.of.current.employment < 2.5
1
1
1
1
1
0
1
0
0
1
0
1
0
1
1
1
1
1
0
1
0

```r
# Simple confusion matrix without pruning
Test50_pred <- predict(Train50_tree, Test50, type="class")
table(Test50_pred, Test50$Creditability)
```

```
##
## Test50_pred   0   1
##           0  46  45
##           1 111 298
```

```r
Train50_prune8 <- prune.misclass(Train50_tree, best=8)
Test50_prune8_pred <- predict(Train50_prune8, Test50, type="class")
```

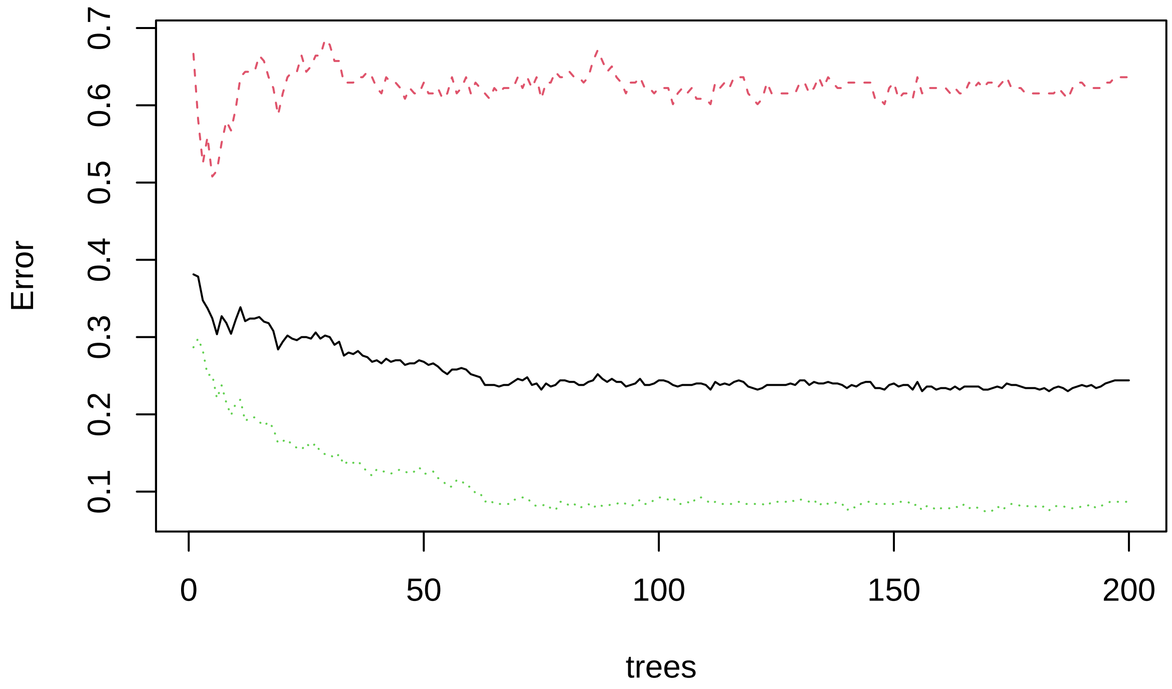
### Confusion matrix for supervised trees (with pruning)

```r
confusionTree <- confusionMatrix(data = factor(Test50_prune8_pred),
                                 reference = factor(Test50$Creditability))
confusionTree
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  38  29
##          1 119 314
##
##                Accuracy : 0.704
##                  95% CI : (0.6619, 0.7437)
##     No Information Rate : 0.686
##     P-Value [Acc > NIR] : 0.207
##
##                   Kappa : 0.1865
##
##  Mcnemar’s Test P-Value : 2.559e-13
##
##             Sensitivity : 0.2420
##             Specificity : 0.9155
##          Pos Pred Value : 0.5672
##          Neg Pred Value : 0.7252
##              Prevalence : 0.3140
##          Detection Rate : 0.0760
##    Detection Prevalence : 0.1340
##       Balanced Accuracy : 0.5787
##
##        ’Positive’ Class : 0
##
```

### Unsupervised Random Forest based method

```
rf50 <- randomForest(as.factor(Creditability) ~., data = Train50, ntree=200, importance=T, proximity=T)
plot(rf50, main="")
```



```
rf50
```

```
##
## Call:
##  randomForest(formula = as.factor(Creditability) ~ ., data = Train50,      ntree = 200, importance =
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 24.4%
## Confusion matrix:
##    0   1 class.error
## 0 52  91  0.63636364
## 1 31 326  0.08683473
```

```
Test50_rf_pred <- predict(rf50, Test50, type="class")
```

### Confusion matrix for unsupervised random forest

```
confusionForest <- confusionMatrix(data = factor(Test50_rf_pred),
                                   reference = factor(Test50$Creditability))
confusionForest
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  52  29
##          1 105 314
##
##                Accuracy : 0.732
##                  95% CI : (0.6909, 0.7704)
##     No Information Rate : 0.686
##     P-Value [Acc > NIR] : 0.01417
##
##                   Kappa : 0.2839
##
##  Mcnemar’s Test P-Value : 9.232e-11
##
##             Sensitivity : 0.3312
##             Specificity : 0.9155
##          Pos Pred Value : 0.6420
##          Neg Pred Value : 0.7494
##              Prevalence : 0.3140
##          Detection Rate : 0.1040
##    Detection Prevalence : 0.1620
##       Balanced Accuracy : 0.6233
##
##        ’Positive’ Class : 0
##
```