lab7_part2.R

sathvik

2021-03-10

```r
# 171EC146
# Sathvik S Prabhu

# Q.Using this data develop a model based on random forest as done in the book by
# Bret Lanz and report the results with an analysis.
# Also compare the results with that of using the Decision Tree
# for classification on this data. Classification is the aim here.

set.seed(300)

# Loading the dataset into R
library(readxl)
credit_train_raw<-read_excel("/home/sathvik/EC8/ML/Lab/Lab7/Chapter 12 German Credit Rating.xlsx",sheet
credit_val_raw<-read_excel("/home/sathvik/EC8/ML/Lab/Lab7/Chapter 12 German Credit Rating.xlsx",sheet=2

# First col is removed as it has the serial no.s
# Last col is removed as it is based on the classification column
# Target: Credit.classification
credit_train<-data.frame(credit_train_raw[2:15])
credit_val<-data.frame(credit_val_raw[2:15])

credit<-rbind(credit_train,credit_val)

# Checking the structure
str(credit)
```

```
## 'data.frame':    1000 obs. of  14 variables:
##  $ CHK_ACCT              : chr  "0DM" "less-200DM" "no-account" "0DM" ...
##  $ Duration              : num  6 48 12 42 24 36 24 36 12 30 ...
##  $ Credit.History        : chr  "critical" "all-paid-duly" "critical" "all-paid-duly" ...
##  $ Credit.Amount         : num  1169 5951 2096 7882 4870 ...
##  $ Balance.in.Savings.A.C: chr  "unknown" "less100DM" "less100DM" "less100DM" ...
##  $ Employment            : chr  "over-seven" "four-years" "seven-years" "seven-years" ...
##  $ Install_rate          : num  4 2 2 2 3 2 3 2 2 4 ...
##  $ Marital.status        : chr  "Single" "female-divorced" "Single" "Single" ...
##  $ Present.Resident      : num  4 2 3 4 4 4 4 2 4 2 ...
##  $ Age                   : num  67 22 49 45 53 35 53 35 61 28 ...
##  $ Other.installment     : num  1 0 0 0 1 0 0 0 0 1 ...
##  $ Num_Credits           : num  2 1 1 1 2 1 1 1 1 2 ...
##  $ Job                   : chr  "Unskilled" "skilled" "Unskilled" "skilled" ...
##  $ Credit.classification : chr  "good." "bad." "good." "good." ...
```

```r
# Converting columns into factors
col_names<-c(1,3,5,6,7,8,9,11,12,13,14)
credit[col_names] <- lapply(credit[col_names] , factor)

# Checking the structure again
str(credit)
```

```
## 'data.frame':	1000 obs. of  14 variables:
##  $ CHK_ACCT             : Factor w/ 4 levels "0DM","less-200DM",..: 1 2 3 1 1 3 3 2 3 2 ...
##  $ Duration             : num  6 48 12 42 24 36 24 36 12 30 ...
##  $ Credit.History       : Factor w/ 4 levels "all-paid-duly",..: 3 1 3 1 4 1 1 1 1 3 ...
##  $ Credit.Amount        : num  1169 5951 2096 7882 4870 ...
##  $ Balance.in.Savings.A.C: Factor w/ 7 levels "Between 100 and 500 DM",..: 7 4 4 4 4 7 2 4 6 4 ...
##  $ Employment           : Factor w/ 5 levels "four-years","one-year",..: 3 1 4 4 1 1 3 1 4 5 ...
##  $ Install_rate         : Factor w/ 4 levels "1","2","3","4": 4 2 2 2 3 2 3 2 2 4 ...
##  $ Marital.status       : Factor w/ 6 levels "female-divorced",..: 5 1 5 5 5 5 5 5 2 3 ...
##  $ Present.Resident     : Factor w/ 4 levels "1","2","3","4": 4 2 3 4 4 4 4 2 4 2 ...
##  $ Age                  : num  67 22 49 45 53 35 53 35 61 28 ...
##  $ Other.installment    : Factor w/ 2 levels "0","1": 2 1 1 1 2 1 1 1 1 2 ...
##  $ Num_Credits          : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 2 1 1 1 1 2 ...
##  $ Job                  : Factor w/ 6 levels "management","skilled",..: 5 2 5 2 2 5 2 1 5 1 ...
##  $ Credit.classification : Factor w/ 2 levels "bad.","good.": 2 1 2 2 1 2 2 2 2 1 ...
```

```r
table(credit$Job)
```

```
## 
##             management               skilled            Unemployed 
##                    148                   629                    17 
## unemployed-non-resident             Unskilled    unskilled-resident 
##                      5                   162                    39
```

```r
credit_train<-credit[1:800,]
credit_val<-credit[801:1000,]

# Random Forests: bagging with random feature selection

# Selection of model
# Metric: Kappa
library(caret)
```

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

```
## Loading required package: ggplot2
```

```r
ctrl <- trainControl(method = "repeatedcv",
                     number = 10, repeats = 10)

# 1. Random Forest
# mtry defines how many features are randomly selected at each split.
grid_rf <- expand.grid(.mtry = c(2, 4, 8))
m_rf <- train(Credit.classification ~ ., data = credit_train, method = "rf",metric = "Kappa",
              trControl = ctrl, tuneGrid = grid_rf)

# 2. Decision Tree
grid_c50 <- expand.grid(.model = "tree",
                        .trials = c(10, 20, 30),
```

```
                        .winnow = "FALSE")
m_c50 <- train(Credit.classification ~ ., data = credit_train, method = "C5.0",
               metric = "Kappa", trControl = ctrl, tuneGrid = grid_c50)
```

```
## Warning in Ops.factor(x$winnow): '!' not meaningful for factors
```

```
# Comparing RF and C50
m_rf # Best: mtry=8
```

```
## Random Forest
##
## 800 samples
##  13 predictor
##   2 classes: 'bad.', 'good.'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 720, 720, 719, 721, 720, 720, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.7031286  0.01024876
##   4     0.7547686  0.30176172
##   8     0.7533937  0.33924246
##
## Kappa was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 8.
```

```
m_c50 # Best: trials=20
```

```
## C5.0
##
## 800 samples
##  13 predictor
##   2 classes: 'bad.', 'good.'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 720, 720, 720, 721, 720, 720, ...
## Resampling results across tuning parameters:
##
##   trials  Accuracy   Kappa
##   10      0.7238486  0.2995639
##   20      0.7334644  0.3237731
##   30      0.7302537  0.3183349
##
## Tuning parameter 'model' was held constant at a value of tree
## Tuning
##  parameter 'winnow' was held constant at a value of FALSE
## Kappa was used to select the optimal model using the largest value.
## The final values used for the model were trials = 20, model = tree and winnow
##  = FALSE.
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

##
## Attaching package: 'randomForest'

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

```r
# Best model: random forest with mtry=8 which gives the highest kappa
m1<-randomForest(Credit.classification ~ ., data = credit_train, mtry=8)
m1
```

```
##
## Call:
##  randomForest(formula = Credit.classification ~ ., data = credit_train,      mtry = 8)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 25.25%
## Confusion matrix:
##       bad. good. class.error
## bad.   104   135   0.5648536
## good.   67   494   0.1194296
```

```r
library(C50)
m2<-C5.0(Credit.classification ~ ., data = credit_train, trials=20)
m2
```

```
##
## Call:
## C5.0.formula(formula = Credit.classification ~ ., data = credit_train, trials
##  = 20)
##
## Classification Tree
## Number of samples: 800
## Number of predictors: 13
##
## Number of boosting iterations: 20
## Average tree size: 34.2
##
## Non-standard options: attempt to group attributes
```

```r
# Evauluation
# Metric: Kappa
for( i in col_names){
  levels(credit_val[[i]]) <- levels(credit_train[[i]])
}
pred1<-predict(m1,credit_val)
confusionMatrix(pred1,credit_val$Credit.classification)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad. good.
##      bad.    26    17
##      good.   35   122
```

```
##
##                Accuracy : 0.74
##                  95% CI : (0.6734, 0.7993)
##     No Information Rate : 0.695
##     P-Value [Acc > NIR] : 0.09453
##
##                   Kappa : 0.3314
##
##  Mcnemar's Test P-Value : 0.01840
##
##             Sensitivity : 0.4262
##             Specificity : 0.8777
##          Pos Pred Value : 0.6047
##          Neg Pred Value : 0.7771
##              Prevalence : 0.3050
##          Detection Rate : 0.1300
##    Detection Prevalence : 0.2150
##       Balanced Accuracy : 0.6520
##
##        'Positive' Class : bad.
##
```

```
# Kappa: 0.33

pred2<-predict(m2,credit_val)
confusionMatrix(pred2,credit_val$Credit.classification)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad. good.
##      bad.    22    13
##      good.   39   126
##
##                Accuracy : 0.74
##                  95% CI : (0.6734, 0.7993)
##     No Information Rate : 0.695
##     P-Value [Acc > NIR] : 0.0945349
##
##                   Kappa : 0.3034
##
##  Mcnemar's Test P-Value : 0.0005265
##
##             Sensitivity : 0.3607
##             Specificity : 0.9065
##          Pos Pred Value : 0.6286
##          Neg Pred Value : 0.7636
##              Prevalence : 0.3050
##          Detection Rate : 0.1100
##    Detection Prevalence : 0.1750
##       Balanced Accuracy : 0.6336
##
##        'Positive' Class : bad.
##
```

```
# Kappa: 0.30
# Best model upon evaluation: random forest with mtry=8, which gives the highest kappa (0.33)
```