ML_synthetic_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


synthetic_sdv <- read.csv("synthetic_data/synthethicHalf.csv", header=T,
                          stringsAsFactors=T)

# Random sample of 50% of row numbers created
indexes = sample(1:nrow(synthetic_sdv), size=0.5*nrow(synthetic_sdv))
# Training data contains created indices
Train50Syn <- synthetic_sdv[indexes,]
# Test data contains the rest
Test50Syn <- synthetic_sdv[-indexes,]

attach(Train50Syn)
```

Logistic regression model with 50:50 Cross-validation

```
LogisticModel50finalSyn <- glm(as.factor(Creditability) ~ Account.Balance +
                                 Payment.Status.of.Previous.Credit + Purpose +
                                 Length.of.current.employment +
                                 Sex...Marital.Status, family=binomial,
                               data = Train50Syn)

fit50Syn <- fitted.values(LogisticModel50finalSyn)
```

```r
Threshold50Syn <- rep(0,500)

for (i in 1:500)
  if(fit50Syn[i] >= 0.5) Threshold50Syn[i] <- 1

for (i in 1:500) {
  if (Threshold50Syn[i] == '1') {
    Threshold50Syn[i] <- 'Creditable'
  }

  if (Threshold50Syn[i] == '0') {
    Threshold50Syn[i] <- 'Non-Creditable'
  }
}

for (i in 1:500) {
  if (Test50Syn$Creditability[i] == '0') {
    Test50Syn$Creditability[i] <- 'Non-Creditable'
  }

  if (Test50Syn$Creditability[i] == '1') {
    Test50Syn$Creditability[i] <- 'Creditable'
  }
}
```

Confusion matrix for GLM

```r
confusionSyn <- confusionMatrix(data = factor(Threshold50Syn), reference = factor(Test50Syn$Creditabili
confusionSyn
```

```
## Confusion Matrix and Statistics
##
##                  Reference
## Prediction       Creditable Non-Creditable
##   Creditable            324            141
##   Non-Creditable         27              8
##
##                Accuracy : 0.664
##                  95% CI : (0.6207, 0.7053)
##     No Information Rate : 0.702
##     P-Value [Acc > NIR] : 0.9707
##
##                   Kappa : -0.0298
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.92308
##             Specificity : 0.05369
##          Pos Pred Value : 0.69677
##          Neg Pred Value : 0.22857
##              Prevalence : 0.70200
##          Detection Rate : 0.64800
##    Detection Prevalence : 0.93000
##       Balanced Accuracy : 0.48838
##
##        'Positive' Class : Creditable
##
```

Supervised Tree based method

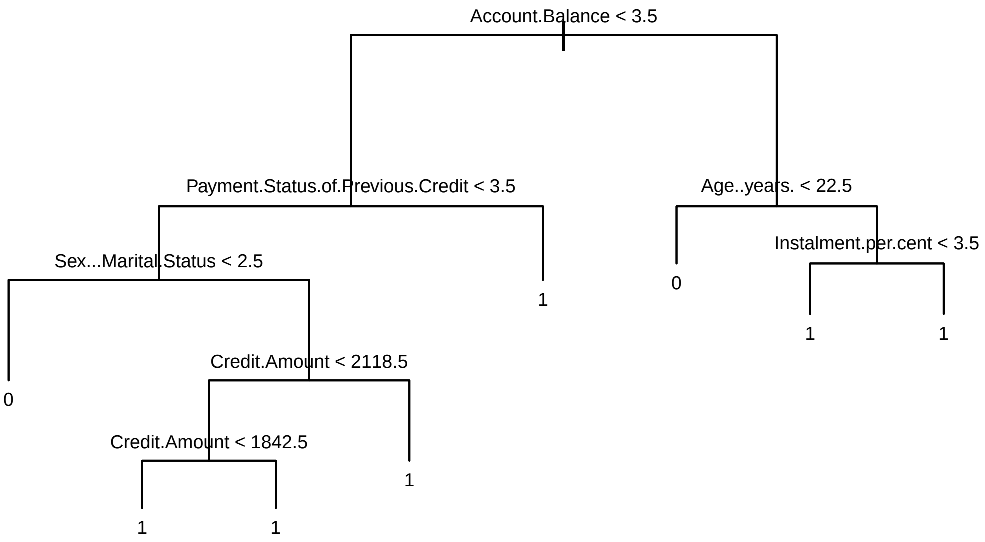
```
# Reload the clean test data set into memory
Test50Syn <- synthetic_sdv[-indexes,]

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=Train50Syn, method="class")

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 = Train50Syn, method = "class")
## Variables actually used in tree construction:
## [1] "Account.Balance"                   "Payment.Status.of.Previous.Credit"
## [3] "Sex...Marital.Status"              "Credit.Amount"
## [5] "Age..years."                       "Instalment.per.cent"
## Number of terminal nodes:  8
## Residual mean deviance:  1.068 = 525.5 / 492
## Misclassification error rate: 0.264 = 132 / 500
```

```
plot(Train50_tree)
text(Train50_tree, pretty=0,cex=0.6)
```



```
Test50_pred <- predict(Train50_tree, Test50Syn, type="class")
table(Test50_pred, Test50Syn$Creditability)
```

```
##
## Test50_pred   0   1
##           0  40  50
##           1 109 301
```

```
Train50_prune8 <- prune.misclass(Train50_tree, best=8)
Test50_prune8_pred <- predict(Train50_prune8, Test50Syn, type="class")
```

Confusion matrix for supervised trees (with pruning)

```
confusionTree <- confusionMatrix(data = factor(Test50_prune8_pred),
                                 reference = factor(Test50Syn$Creditability))
confusionTree
```

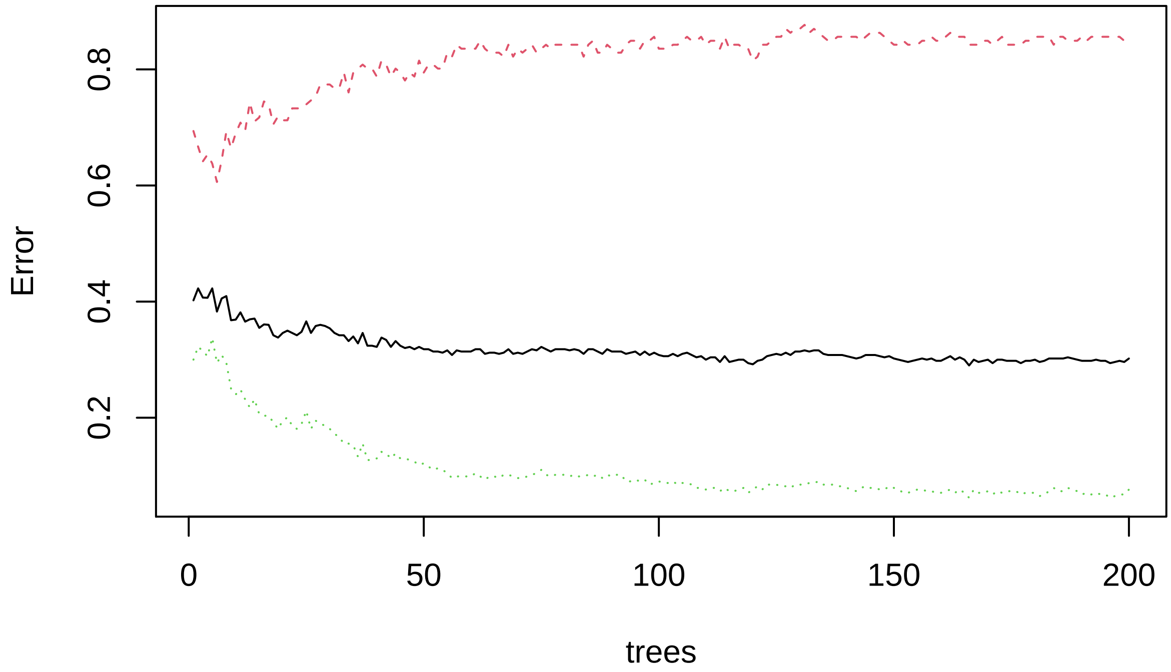
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  40  50
##          1 109 301
```

```
##
##                Accuracy : 0.682
##                  95% CI : (0.6392, 0.7226)
##     No Information Rate : 0.702
##     P-Value [Acc > NIR] : 0.8476
##
##                   Kappa : 0.1422
##
##  Mcnemar's Test P-Value : 4.231e-06
##
##             Sensitivity : 0.2685
##             Specificity : 0.8575
##          Pos Pred Value : 0.4444
##          Neg Pred Value : 0.7341
##              Prevalence : 0.2980
##          Detection Rate : 0.0800
##    Detection Prevalence : 0.1800
##       Balanced Accuracy : 0.5630
##
##        'Positive' Class : 0
##
```

Unsupervised Random Forest based method

```
rf50 <- randomForest(as.factor(Creditability) ~., data = Train50Syn, ntree=200, importance=T, proximity=

plot(rf50, main="")
```



```
rf50
```

```
##
## Call:
##  randomForest(formula = as.factor(Creditability) ~ ., data = Train50Syn,      ntree = 200, importanc
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 30.2%
## Confusion matrix:
##    0   1 class.error
## 0 22 124  0.84931507
## 1 27 327  0.07627119
```

```
Test50_rf_pred <- predict(rf50, Test50Syn, type="class")
```

Confusion matrix for unsupervised random forest

```
confusionForest <- confusionMatrix(data = factor(Test50_rf_pred),
                                   reference = factor(Test50Syn$Creditability))
confusionForest
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  21  18
##          1 128 333
##
##                Accuracy : 0.708
##                  95% CI : (0.666, 0.7475)
##     No Information Rate : 0.702
##     P-Value [Acc > NIR] : 0.4059
##
##                   Kappa : 0.1138
##
##  Mcnemar’s Test P-Value : <2e-16
##
##             Sensitivity : 0.1409
##             Specificity : 0.9487
##          Pos Pred Value : 0.5385
##          Neg Pred Value : 0.7223
##              Prevalence : 0.2980
##          Detection Rate : 0.0420
##    Detection Prevalence : 0.0780
##       Balanced Accuracy : 0.5448
##
##        ’Positive’ Class : 0
##
```