Assignment 3

Arthur Junges Schmidt

23 July 2020

Problem 1

(a) The file ps3-1.csv contains a data set with 34 features (x1, x2, ... x34) and 1 target variable (Y). Estimate a classifier model using Support Vector Machine and Random Forest algorithm in R, respectively, with the first 14628 rows of the data. Optimize your model so that a False Positive Rate is less than 10% for Y = 0 (actual Y = 1 cases falsely classified as Y = 0). Particularly, you are required to review relevant literature, use the k-fold cross-validation method to train the Random Forest model. Use grid search to find hyper-parameter setting: the best number of trees and features, maximum leaf nodes. Assess importance of each feature based on two criteria: Mean Decrease Accuracy and Mean Decrease Gini. Compare two estimation methods with confusion matrices

```
Crude_Data <- read.csv("ps3-1.csv");
Crude_Data <- Crude_Data[,-1];
Crude_Data$Y <- factor(Crude_Data$Y, levels = c(0,1));
Training_Data <- Crude_Data[1:14628, ];
Test_Data <- Crude_Data[-(1:14628), ];
```

```
# SVM_Model <- svm(Y ~ ., data = Training_Data)
# #plot(SVM_Model, data = Training_Data, Y~.)
# SVM_Predict <- predict(object = SVM_Model, newdata = Test_Data[,-35]);
# table(Predicted = Test_Data[,35], True = SVM_Predict)
# test <- tune.svm(Y ~ ., data = Training_Data, gamma = c(0.01, 0.1, 1), cost = 5, kernel = "radial")
# beepr::beep(sound = 9)
```

Random Forest

According to Friedman, Hastie, and Tibshirani (2001) and James et al. (2013), the usual number of predictors candidates m from the full set of predictors p is

$$m \approx \sqrt{p}$$

. So with our data, $m \approx \sqrt{(35)} \approx 6$.

```
tunegrid <- expand.grid(.mtry = c(15), .splitrule = c("gini", "extratrees"),
                        .min.node.size = c(5))
fitControl <- trainControl(method = "cv", number = 5, search = "grid", verboseIter = TRUE)
modellist <- list()
Start_Time <- Sys.time()
for (ntree in c(1000, 2000)) {
    set.seed(42)
  cat(ntree)
    fit <- train(Y ~ ., data = Training_Data,
                              method = "ranger", trControl = fitControl,
                              tuneGrid = tunegrid, num.trees = ntree,
                              )
```

```
    key <- toString(ntree)
    modellist[[key]] <- fit
    running_time <- Sys.time() - Start_Time
}
```

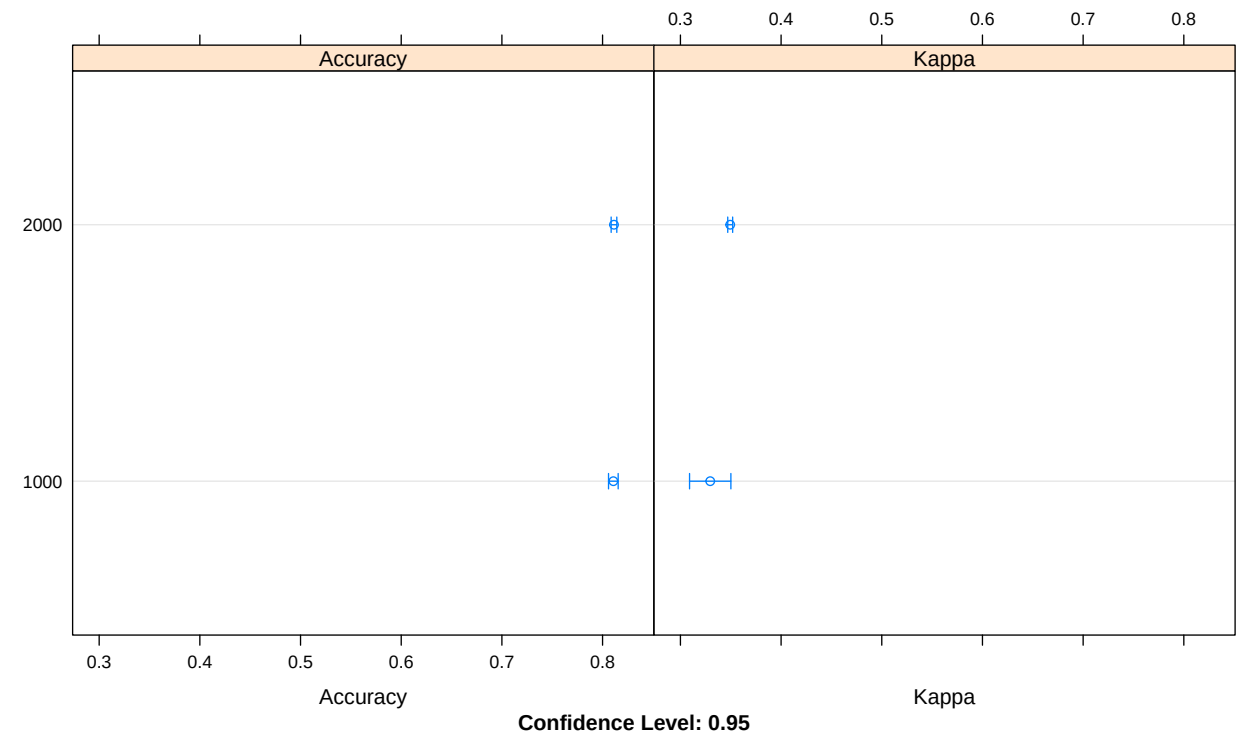
```
## 1000+ Fold1: mtry=15, splitrule=gini, min.node.size=5
## - Fold1: mtry=15, splitrule=gini, min.node.size=5
## + Fold1: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold1: mtry=15, splitrule=extratrees, min.node.size=5
## + Fold2: mtry=15, splitrule=gini, min.node.size=5
## - Fold2: mtry=15, splitrule=gini, min.node.size=5
## + Fold2: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold2: mtry=15, splitrule=extratrees, min.node.size=5
## + Fold3: mtry=15, splitrule=gini, min.node.size=5
## - Fold3: mtry=15, splitrule=gini, min.node.size=5
## + Fold3: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold3: mtry=15, splitrule=extratrees, min.node.size=5
## + Fold4: mtry=15, splitrule=gini, min.node.size=5
## - Fold4: mtry=15, splitrule=gini, min.node.size=5
## + Fold4: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold4: mtry=15, splitrule=extratrees, min.node.size=5
## + Fold5: mtry=15, splitrule=gini, min.node.size=5
## - Fold5: mtry=15, splitrule=gini, min.node.size=5
## + Fold5: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold5: mtry=15, splitrule=extratrees, min.node.size=5
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 15, splitrule = extratrees, min.node.size = 5 on full training set
## 2000+ Fold1: mtry=15, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 57%. Estimated remaining time: 23 seconds.
## - Fold1: mtry=15, splitrule=gini, min.node.size=5
## + Fold1: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold1: mtry=15, splitrule=extratrees, min.node.size=5
## + Fold2: mtry=15, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 65%. Estimated remaining time: 16 seconds.
## - Fold2: mtry=15, splitrule=gini, min.node.size=5
## + Fold2: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold2: mtry=15, splitrule=extratrees, min.node.size=5
## + Fold3: mtry=15, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 70%. Estimated remaining time: 13 seconds.
## - Fold3: mtry=15, splitrule=gini, min.node.size=5
## + Fold3: mtry=15, splitrule=extratrees, min.node.size=5
## Growing trees.. Progress: 83%. Estimated remaining time: 6 seconds.
## - Fold3: mtry=15, splitrule=extratrees, min.node.size=5
## + Fold4: mtry=15, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 70%. Estimated remaining time: 13 seconds.
## - Fold4: mtry=15, splitrule=gini, min.node.size=5
## + Fold4: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold4: mtry=15, splitrule=extratrees, min.node.size=5
## + Fold5: mtry=15, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 71%. Estimated remaining time: 12 seconds.
## - Fold5: mtry=15, splitrule=gini, min.node.size=5
## + Fold5: mtry=15, splitrule=extratrees, min.node.size=5
## - Fold5: mtry=15, splitrule=extratrees, min.node.size=5
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 15, splitrule = gini, min.node.size = 5 on full training set
## Growing trees.. Progress: 51%. Estimated remaining time: 30 seconds.
```

```
# Random_Forest_Model <- train(Y ~ ., data = Training_Data,
#                              method = "rf", trControl = fitControl,
#                              tuneGrid = tunegrid
#                              )

# compare results
results <- resamples(modellist)
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: 1000, 2000
## Number of resamples: 5
##
## Accuracy
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## 1000 0.8062201 0.8078632 0.8103213 0.8107056 0.8137389 0.8153846    0
## 2000 0.8086124 0.8092308 0.8123718 0.8113207 0.8126496 0.8137389    0
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## 1000 0.3034182 0.3279342 0.3290327 0.3295994 0.3430061 0.3446057    0
## 2000 0.3476909 0.3481543 0.3484079 0.3494439 0.3506562 0.3523104    0
```

```
dotplot(results)
```




```
beepr::beep(sound = 9)
```


Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. Vol. 1. 10. Springer series in statistics New York.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.