SmartFly: Train model and validate via cross-validation

Cindy Lamm

17:21, Sunday 18th January, 2015

Load prepared data from the previous step "Prepare Data For Modeling"

```
rm(list=ls())   #clear memory
load("../02_prepare_data_for_modeling/rfModelData.RData")
```

Split the train data based on simple bootstrap resampling into a train and test set

```
library(caret)
set.seed(998)
PERCENTAGE <- 0.03
inTraining <- createDataPartition(rfModelData$is_delayed, p=PERCENTAGE, list=FALSE)
length(inTraining)

## [1] 221231

training <- rfModelData[inTraining,]
testing <- rfModelData[-inTraining,]
testing <- testing[1:100,]
```

```
str(training)

## 'data.frame': 221231 obs. of  12 variables:
##  $ id                       : chr  "3280125367763225179" "7292790258672752809" "3535895314207389905" "
##  $ year                     : num  2013 2013 2013 2013 2013 ...
##  $ month                    : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ day_of_month             : num  8 17 3 9 17 25 11 28 29 15 ...
##  $ day_of_week              : num  4 6 6 5 6 7 7 3 4 4 ...
##  $ scheduled_departure_time: num  7 12 15 12 12 6 7 7 7 7 ...
##  $ scheduled_arrival_time   : num  8 14 17 15 15 8 9 8 8 8 ...
##  $ airline                  : Factor w/ 19 levels "AA","AS","B6",..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ plane_model              : Factor w/ 6 levels "737","747","757",..: 6 1 2 3 3 3 1 1 1 2 ...
##  $ seat_configuration       : Factor w/ 6 levels "Standard","Three Class",..: 1 6 5 2 4 6 2 3 2 5 ...
##  $ distance_travelled       : num  185 508 329 809 809 329 728 370 370 508 ...
##  $ is_delayed               : Factor w/ 2 levels "on_time","delayed": 1 1 1 1 1 1 1 1 1 1 ...
```

Estimate a random forest using 3% of the data - without crossvalidation:

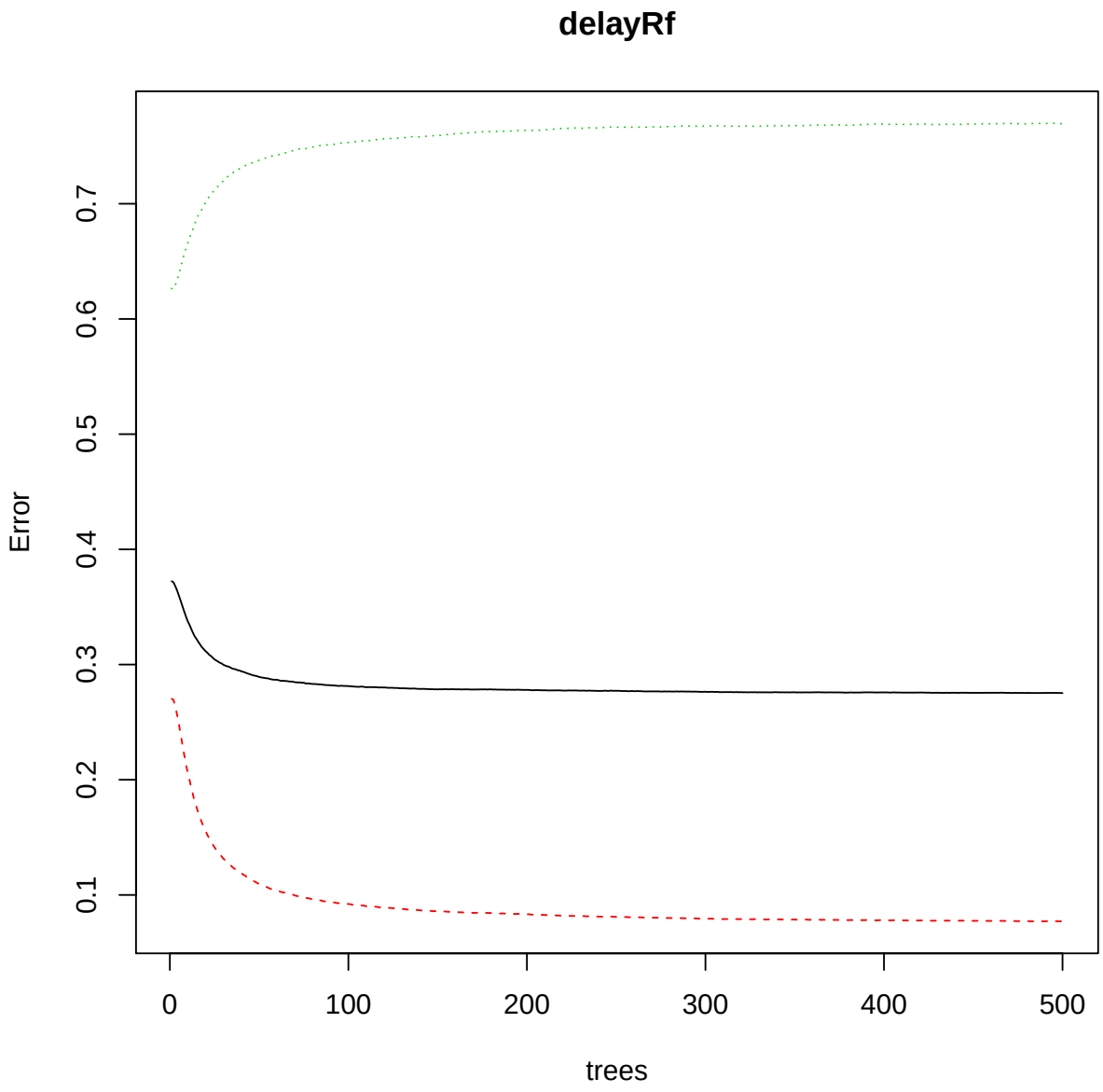
```
library(randomForest)
delayRf <- randomForest(is_delayed ~ . - id, data=rfModelData, subset=inTraining,
                        importance=TRUE, proximity=FALSE)
```

Check out the model result:

```
delayRf

##
## Call:
##  randomForest(formula = is_delayed ~ . - id, data = rfModelData,      importance = TRUE, proximity =
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 27.97%
## Confusion matrix:
##         on_time delayed class.error
## on_time  145972   11944  0.07563515
## delayed   49942   13373  0.78878623

plot(delayRf)
```



Save the model result:

```
save(delayRf, file="../03_train_model/delayRf.RData")
```

Predict on test data:

```
testPrediction <- predict(delayRf, newdata=testing, type="prob")
classPrediction <- factor(ifelse(testPrediction[,"delayed"]<0.5,"on_time","delayed"),
                          levels=c("on_time","delayed"))
confusionMatrix(classPrediction, reference=testing$is_delayed)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction on_time delayed
```

```
##    on_time      75      23
##    delayed       1       1
##
##               Accuracy : 0.76
##                 95% CI : (0.6643, 0.8398)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 0.5545
##
##                  Kappa : 0.0415
##  Mcnemar's Test P-Value : 1.814e-05
##
##             Sensitivity : 0.98684
##             Specificity : 0.04167
##          Pos Pred Value : 0.76531
##          Neg Pred Value : 0.50000
##              Prevalence : 0.76000
##          Detection Rate : 0.75000
##    Detection Prevalence : 0.98000
##       Balanced Accuracy : 0.51425
##
##        'Positive' Class : on_time
##
```

Save the prediction result:

```
save(testPrediction, file="../03_train_model/testPrediction.RData")
```