SmartFly: Train model and validate via cross-validation

Cindy Lamm

18:08, Friday 16th 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 series of train and test sets

```
library(caret)
set.seed(998)
PERCENTAGE <- 0.07
inTraining <- createDataPartition(rfModelData$is_delayed, times=1, p = PERCENTAGE, list = FALSE)
length(inTraining)

## [1] 516206

training <- rfModelData[inTraining,]
# testing  <- rfModelData[-inTraining,]
```

Estimate a random forest using 7% of the data - without crossvalidation:

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

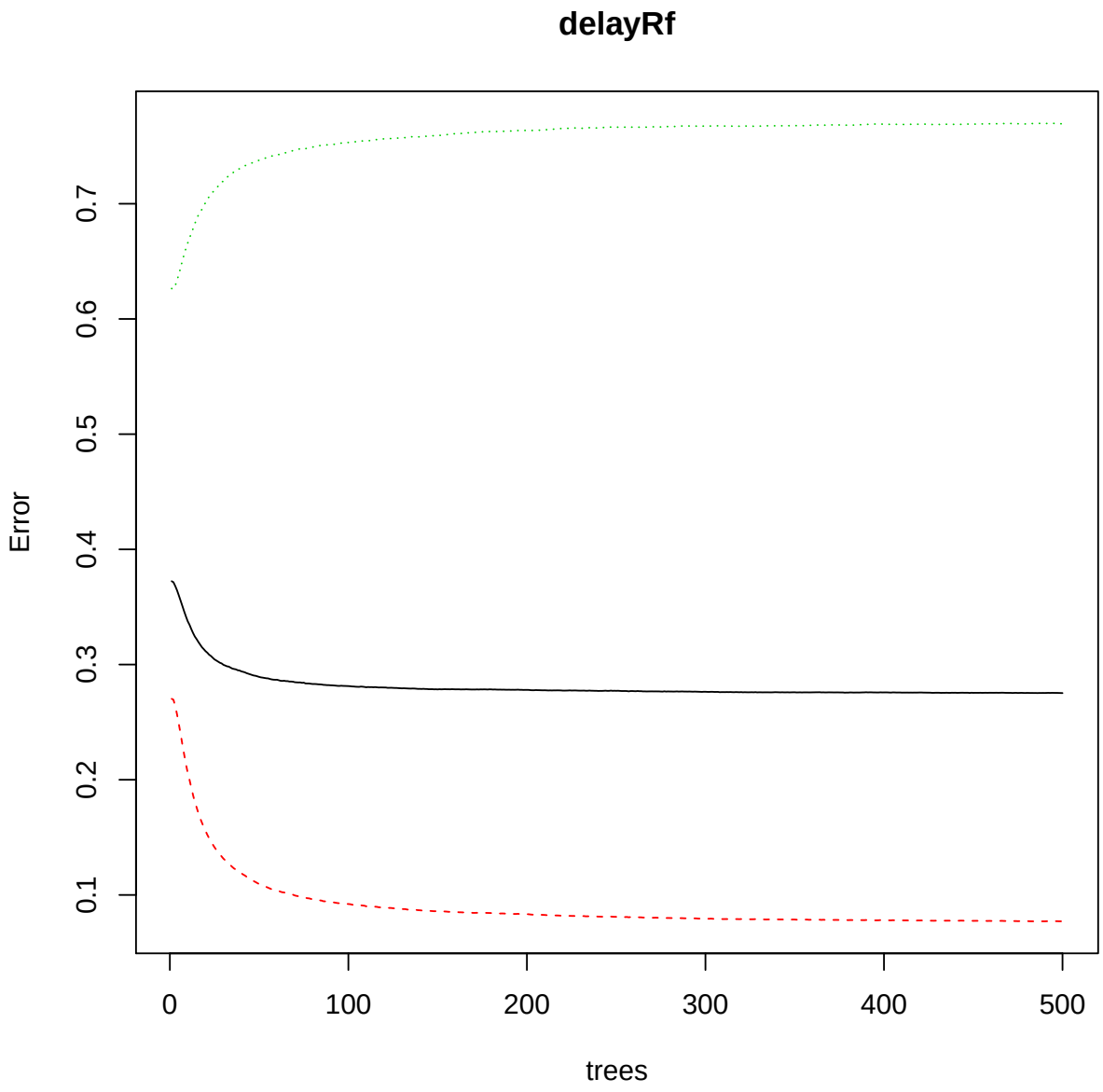
Note: On a Macbook with 16GB RAM it takes 6 minutes for sample size of 100.000 and 30 minutes fortraining sample size of 5% (about 360.000 obs).

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: 27.52%
## Confusion matrix:
##         on_time delayed class.error
## on_time  340077   28394  0.07705898
## delayed  113682   34053  0.76949944

plot(delayRf)
```



Save the model result:

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