SmartFly: Train model and validate via cross-validation

Cindy Lamm

15:35, Friday 16th January, 2015

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

```
rm(list=ls())   #clear memory
load("../03_train_model/rfModelData.RData")
```

Split the train data based on simple bootstrap resampling into a series of train and test sets

```
library(caret)
set.seed(998)
inTraining <- createDataPartition(rfModelData$is_delayed, times=1, p = .05, list = FALSE)
length(inTraining)

## [1] 366092

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

Estimate a random forest using 5% of the data (as test run) - without validation:

```
library(randomForest)
delayRF360k <- randomForest(is_delayed ~ ., data=rfModelData, subset=inTraining,
                            importance=TRUE, proximity=FALSE)
```

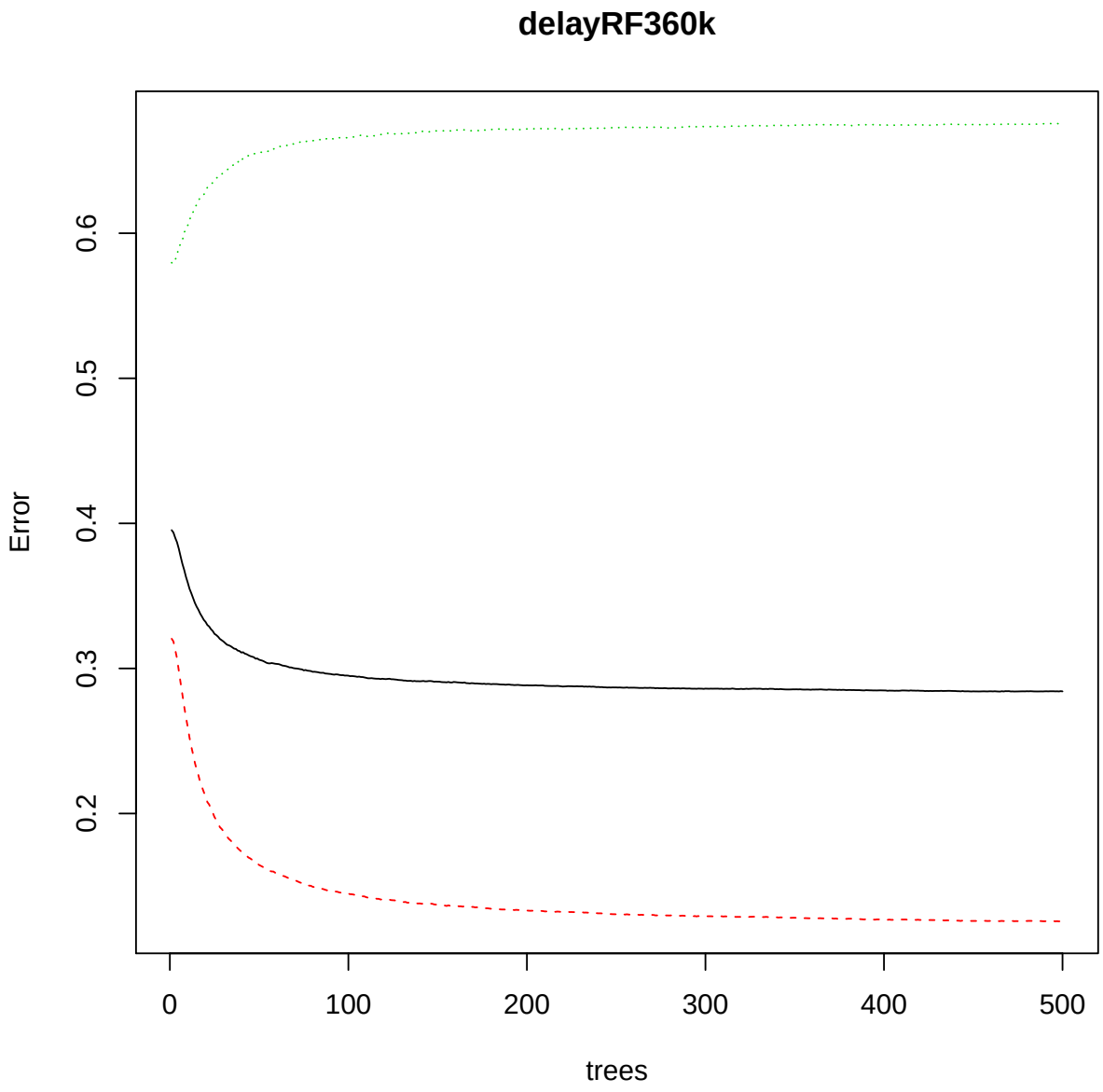
Note: On a Macbook with 16GB RAM it takes 6 minutes for sample size of 100.000 and 30 minutes for sample size of 360.000.

Check out the model result:

```
delayRF360k

##
## Call:
##  randomForest(formula = is_delayed ~ ., data = rfModelData, importance = TRUE,      proximity = FALSE
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 28.42%
## Confusion matrix:
##         on_time delayed class.error
## on_time  227841   32733   0.1256188
## delayed   71306   34212   0.6757710

plot(delayRF360k)
```



Save the model result:

```
save(delayRF360k, inTraining, file="../03_train_model/delayRF360k.RData")
```