SmartFly: Prepare Data For Prediction

Cindy Lamm

12:19, Friday 16th January, 2015

Load preprocessed data from the previous step "Exploratory Analysis For Scheduled Flight Data"

```
rm(list=ls())   #clear memory
load("../01_exploratory_data_analysis/predictDataTyped.RData")
rfPredictData <- predictDataTyped
rm(predictDataTyped)
```

1 Set variables as used in modeling data

Load the variables names that were used for the modeling stage

```
load("../02_prepare_data_for_modeling/modelVariables.RData")
modelVariables
```

```
##  [1] "year"                   "month"                  "day_of_month"
##  [4] "day_of_week"            "scheduled_departure_time" "scheduled_arrival_time"
##  [7] "airline"                "plane_model"            "seat_configuration"
## [10] "distance_travelled"     "is_delayed"
```

and remove all non-used variables from the scheduled flight data:

```
removeNames <- setdiff(names(rfPredictData), modelVariables)
excludeIdx <- sapply(removeNames, FUN=function(v, x){ which(v==x)}, v=names(rfPredictData))
rfPredictData <- rfPredictData[,-excludeIdx]
str(rfPredictData)
```

```
## 'data.frame': 566376 obs. of  10 variables:
##  $ year                   : Factor w/ 1 level "2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ month                  : Factor w/ 1 level "01": 1 1 1 1 1 1 1 1 1 1 ...
##  $ day_of_month           : Factor w/ 31 levels "01","02","03",..: 12 13 14 15 16 17 19 20 21 22 ...
##  $ day_of_week            : Factor w/ 7 levels "1","2","3","4",..: 1 2 3 4 5 6 1 2 3 4 ...
##  $ scheduled_departure_time: Factor w/ 24 levels "00","01","02",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ scheduled_arrival_time  : Factor w/ 23 levels "00","01","02",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ airline                : Factor w/ 19 levels "AA","AS","B6",..: 16 16 16 16 16 16 16 16 16 16 ...
##  $ plane_model            : Factor w/ 6 levels "737","747","757",..: 2 2 1 3 5 6 2 3 3 2 ...
##  $ seat_configuration     : Factor w/ 6 levels "Standard","Three Class",..: 6 2 4 4 2 4 4 4 6 4 ...
##  $ distance_travelled     : num  599 599 599 599 599 599 599 599 599 599 ...
```

2 Convert date and time related variables from factors to numbers

For the scheduled flight data we use for date and time related variables as well numeric values instead of factor levels.

```
str(rfPredictData)
```

```
## 'data.frame': 566376 obs. of  10 variables:
##  $ year                    : num  2015 2015 2015 2015 2015 ...
##  $ month                   : num  1 1 1 1 1 1 1 1 1 ...
##  $ day_of_month            : num  12 13 14 15 16 17 19 20 21 22 ...
##  $ day_of_week             : num  1 2 3 4 5 6 1 2 3 4 ...
##  $ scheduled_departure_time: num  6 6 6 6 6 6 6 6 6 6 ...
##  $ scheduled_arrival_time  : num  9 9 9 9 9 9 9 9 9 9 ...
##  $ airline                 : Factor w/ 19 levels "AA","AS","B6",..: 16 16 16 16 16 16 16 16 16 16 ...
##  $ plane_model             : Factor w/ 6 levels "737","747","757",..: 2 2 1 3 5 6 2 3 3 2 ...
##  $ seat_configuration      : Factor w/ 6 levels "Standard","Three Class",..: 6 2 4 4 2 4 4 4 6 4 ...
##  $ distance_travelled      : num  599 599 599 599 599 599 599 599 599 599 ...
```

3 Remove unknown factor levels

The comparison of historic and scheduled flight data furthermore showed that there are two airlines in the scheduled data that are not listed in the historic data. In order to be able to predict using the estimated random forest from the historic data I need to exclude the observations listed for these two airlines:

```
removeNames <- c("HA","OH")
removeIdxs <- union(which(rfPredictData$airline=="HA"), which(rfPredictData$airline=="OH"))
length(removeIdxs)
```

```
## [1] 33398
```

```
rfPredictData <- rfPredictData[-removeIdxs,]
lvlIdx <- c(which(levels(rfPredictData$airline)=="HA"), which(levels(rfPredictData$airline)=="OH"))
levels(rfPredictData$airline)[lvlIdx] <- NA
```

```
sapply(rfPredictData, FUN=levels)
```

```
## $year
## NULL
##
## $month
## NULL
##
## $day_of_month
## NULL
##
## $day_of_week
## NULL
##
## $scheduled_departure_time
## NULL
##
```

```
## $scheduled_arrival_time
## NULL
##
## $airline
##  [1] "AA" "AS" "B6" "CO" "DH" "DL" "EV" "FL" "HP" "MQ" "NW" "OO" "TZ" "UA" "US" "WN" "XE"
##
## $plane_model
## [1] "737"  "747"  "757"  "777"  "787"  "A320"
##
## $seat_configuration
## [1] "Standard"    "Three Class" "Two Class"   "V1"          "V2"          "V3"
##
## $distance_travelled
## NULL
```

There is no need to adapt the levels for the other variables, since they are not included in the modeling (or prediction) process anyway.

4 Analyse & deal with missing values

```
nbRows <- dim(rfPredictData)[1]
rowHasNa <- apply(rfPredictData, MARGIN=1, FUN=function(row){ any(is.na(row)) })
nbRowsWithNa <- sum(rowHasNa)
nbRowsLeft <- nbRows - nbRowsWithNa
# proportion of NA rows:
nbRowsWithNa / nbRows

## [1] 0
```

There are no missing values in any of the remaining varaibles.

So the data that I use for the predicting the probability of delay for the scheduled flights using the estimated random forest looks as follows:

```
str(rfPredictData)

## 'data.frame': 532978 obs. of  10 variables:
##  $ year                   : num  2015 2015 2015 2015 2015 ...
##  $ month                  : num  1 1 1 1 1 1 1 1 1 ...
##  $ day_of_month           : num  12 13 14 15 16 17 19 20 21 22 ...
##  $ day_of_week            : num  1 2 3 4 5 6 1 2 3 4 ...
##  $ scheduled_departure_time: num  6 6 6 6 6 6 6 6 6 6 ...
##  $ scheduled_arrival_time  : num  9 9 9 9 9 9 9 9 9 9 ...
##  $ airline                : Factor w/ 17 levels "AA","AS","B6",..: 14 14 14 14 14 14 14 14 14 14 ...
##  $ plane_model            : Factor w/ 6 levels "737","747","757",..: 2 2 1 3 5 6 2 3 3 2 ...
##  $ seat_configuration     : Factor w/ 6 levels "Standard","Three Class",..: 6 2 4 4 2 4 4 4 6 4 ...
##  $ distance_travelled     : num  599 599 599 599 599 599 599 599 599 599 ...
```

I save the data for the next step:

```
save(rfPredictData, file="../04_prepare_data_for_prediction/rfPredictData.RData")
```