SmartFly: Prepare Data For Modeling

Cindy Lamm

09:39, Friday 16th January, 2015

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

```
rm(list=ls())   #clear memory
load("../01_exploratory_data_analysis/trainDataTyped.Rdata")
modelData <- trainDataTyped
rm(trainDataTyped)
```

1 Analyse & deal with missing values

The variable `cancellation_code` has the most missing values since it is only filled if the flight was not cancelled - which is the case for most flights. I would expect that all non-cancelled flights don't have a `cancellation_code`, however I found 168 non-cancelled flights that do have an entry as `cancellation_code`.

```
weirdIdx <- which(!modelData$cancelled & !is.na(modelData$cancellation_code))
summary(modelData[weirdIdx,])
```

```
##       id              year         month      day_of_month day_of_week
##  Length:168         2013:  0   06     :167   13     : 12   1:34
##  Class :character   2014:168   08     :  1   05     : 11   2:24
##  Mode  :character              01     :  0   04     : 10   3:22
##                                02     :  0   12     : 10   4:30
##                                03     :  0   16     : 10   5:19
##                                04     :  0   24     : 10   6:21
##                                (Other):  0   (Other):105   7:18
##  scheduled_departure_time scheduled_arrival_time    airline    flight_number
##  16     :18               16     :21              WN     :136   152    :  5
##  13     :17               19     :17              AS     : 31   639    :  5
##  14     :15               13     :16              EV     :  1   1622   :  4
##  18     :15               14     :15              AA     :  0   153    :  3
##  15     :14               21     :13              B6     :  0   41     :  3
##  17     :12               17     :11              CO     :  0   66     :  3
##  (Other):77               (Other):75              (Other):  0   (Other):145
##   tail_number  plane_model   seat_configuration departure_delay  origin_airport
##  N740AS :  4   737 :57     Standard   :39       Min.   : -9.00   MDW    :21
##  N80    :  4   747 :39     Three Class:20       1st Qu.:  0.00   PHX    :14
##  N661   :  3   757 :22     Two Class  :16       Median : 18.00   ANC    :13
##  N746AS :  3   777 : 7     V1         :27       Mean   : 36.29   DAL    :13
##  N86    :  3   787 : 9     V2         :33       3rd Qu.: 54.25   HOU    :12
##  N87    :  3   A320:34     V3         :33       Max.   :270.00   ELP    : 7
##  (Other):148                                                     (Other):88
```

```
##  destination_airport distance_travelled  taxi_time_in    taxi_time_out   cancelled
##  HOU    :28          Min.   :  95.0     Min.   :0.0000   Min.   : 3.00   Mode :logical
##  DAL    :16          1st Qu.: 301.2     1st Qu.:0.0000   1st Qu.: 7.00   FALSE:168
##  STL    :11          Median : 562.0     Median :0.0000   Median :10.00   NA's :0
##  BWI    : 9          Mean   : 827.5     Mean   :0.0119   Mean   :11.01
##  OTZ    : 8          3rd Qu.:1271.8     3rd Qu.:0.0000   3rd Qu.:14.00
##  SEA    : 8          Max.   :2724.0     Max.   :2.0000   Max.   :35.00
##  (Other):88
##  cancellation_code   is_delayed
##  A:99              on_time: 54
##  B:51              delayed:114
##  C:18
##  D: 0
##
##
##
```

A quick analysis of the concerned observations reveals that all but one have a `taxi_time_in` of 0 minutes.

I'll exclude those observations from the data set in the following.

```
modelData <- modelData[-weirdIdx,]
```

The variable `tail_number` includes the values "000000" and "0" which are according to Wikipedia[1] not valid registration numbers. However the value "0" appears as well in the scheduled flight data set, so I only mark the values "000000" as `NA`.

```
is_invalid <- modelData$tail_number == "000000"
sum(is_invalid, na.rm=TRUE)

## [1] 10157

naIdx <- which(is_invalid)
modelData$tail_number[naIdx] <- NA
```

Before I do an analysis of how many rows contain any missing values, I remove the variables that I won't use for estimating the model. I exclude the variable `id` because I assume it is randomly assigned to the observation and has no predictive power regarding the delay of a flight. Furthermore I exlcude the variables `departure_delay`, `taxi_time_in`, `taxi_time_out`, `cancelled`, `cancellation_code` because they are not available in the scheduled flight data.

```
nonAvailable <- c("id", "departure_delay", "taxi_time_in", "taxi_time_out",
                  "cancelled", "cancellation_code")
excludeIdx <- sapply(nonAvailable, FUN=function(v, x){ which(v==x)}, v=names(modelData))
modelData <- modelData[,-excludeIdx]
str(modelData)

## 'data.frame': 7374197 obs. of  15 variables:
##  $ year                   : Factor w/ 2 levels "2013","2014": 1 1 1 1 1 1 1 1 1 1 ...
##  $ month                  : Factor w/ 12 levels "01","02","03",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ day_of_month           : Factor w/ 31 levels "01","02","03",..: 11 17 18 24 25 31 1 2 3 4 ...
##  $ day_of_week            : Factor w/ 7 levels "1","2","3","4",..: 7 6 7 6 7 6 4 5 6 7 ...
##  $ scheduled_departure_time: Factor w/ 24 levels "00","01","02",..: 11 11 11 11 11 11 8 8 8 8 ...
```

[1] http://en.wikipedia.org/wiki/Aircraft_registration

```
##  $ scheduled_arrival_time  : Factor w/ 24 levels "00","01","02",..: 12 12 12 12 12 12 9 9 9 ...
##  $ airline                 : Factor w/ 17 levels "AA","AS","B6",..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ flight_number           : Factor w/ 6889 levels "1","10","100",..: 6744 6744 6744 6744 6744 6744 6
##  $ tail_number             : Factor w/ 5035 levels "0","000000","N050AA",..: 3898 3963 3806 3810 4008
##  $ plane_model             : Factor w/ 6 levels "737","747","757",..: 3 3 5 2 5 2 2 3 2 6 ...
##  $ seat_configuration      : Factor w/ 6 levels "Standard","Three Class",..: 2 1 4 5 4 5 2 1 5 2 ...
##  $ origin_airport          : Factor w/ 279 levels "ABE","ABI","ABQ",..: 46 46 46 46 46 46 133 133 133
##  $ destination_airport     : Factor w/ 279 levels "ABE","ABI","ABQ",..: 61 61 61 61 61 61 61 61 61 61
##  $ distance_travelled      : num  361 361 361 361 361 361 185 185 185 185 ...
##  $ is_delayed              : Factor w/ 2 levels "on_time","delayed": 1 2 1 1 1 1 1 1 2 1 ...
```

Often observations with any missing value are excluded from the modeling stage. There would be 7321827 observations left if I would exclude observations that have `NA` for any of the variables (excluding `cancellation_code`). Since most modeling algorithms can't deal with non available values anyway and there is such a vast amount of training data without `NA` values left I remove the 52370 rows with `NA`s from the data.

```
modelData <- modelData[-which(rowHasNa),]
```

Since I will use the `randomForest`[2] package I also remove the factor variables that have more than 53 levels since otherwise an error occurs.

```
modelFactorIdx <- which(sapply(modelData, FUN=class) == "factor")
modelFactorLevels <- sapply(modelData, FUN=levels)
nbLevels <- sapply(modelFactorLevels, FUN=length)
suitable <- which(nbLevels < 53) # condition for this randomForest implementation
rfModelData <- modelData[,suitable]
```

So the data that I use for the estimation of a random forest looks as follows:

```
str(rfModelData)

## 'data.frame': 7321827 obs. of  11 variables:
##  $ year                    : Factor w/ 2 levels "2013","2014": 1 1 1 1 1 1 1 1 1 1 ...
##  $ month                   : Factor w/ 12 levels "01","02","03",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ day_of_month            : Factor w/ 31 levels "01","02","03",..: 11 17 18 24 25 31 1 2 3 4 ...
##  $ day_of_week             : Factor w/ 7 levels "1","2","3","4",..: 7 6 7 6 7 6 4 5 6 7 ...
##  $ scheduled_departure_time: Factor w/ 24 levels "00","01","02",..: 11 11 11 11 11 11 8 8 8 8 ...
##  $ scheduled_arrival_time  : Factor w/ 24 levels "00","01","02",..: 12 12 12 12 12 12 9 9 9 9 ...
##  $ airline                 : Factor w/ 17 levels "AA","AS","B6",..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ plane_model             : Factor w/ 6 levels "737","747","757",..: 3 3 5 2 5 2 2 3 2 6 ...
##  $ seat_configuration      : Factor w/ 6 levels "Standard","Three Class",..: 2 1 4 5 4 5 2 1 5 2 ...
##  $ distance_travelled      : num  361 361 361 361 361 361 185 185 185 185 ...
##  $ is_delayed              : Factor w/ 2 levels "on_time","delayed": 1 2 1 1 1 1 1 1 2 1 ...
```

I save it to use in the next step:

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

[2] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_manual.htm