Solution Abstract Problem 1: SmartFly

Cindy Lamm

Saturday 17th January, 2015

1 Approach

I identified the given problem as a classification, respectively a class probability estimation and opted for the following steps as general approach:

- analyse file structure
- explore the historic and scheduled data
- prepare the historic data for modeling
- estimate and validate a model
- prepare the scheduled data for prediction
- predict the delay probability of scheduled flights using the estimated model

In the spirit of the Lean Startup by Eric Ries, I tried to keep my mind focused on the "minimal viable solution": I first estimated a basic model on a minor set of data without any validation to get as quick as possible to the prediction stage producing the required output file of ordered flight IDs. I then iterated over each step and improved the solution.

1.1 Analyse file structure

As first step I do a basic structure and content analysis of the given data files on the command line in order to prepare loading the data into any tool/program[1]. The main goal is to get a first impression of data quality: Do all lines of the csv file have the same number of fields? Are there any quotes within the fields that might need escaping?

1.2 Explore the data

As second step I do an exploratory data analysis on the historic as well as the prediction data set. I analyse both sets to avoid potential issues arising when the prediction data is slightly (or strongly) different in structure or content from the historic data that will be used for training a model.

I use the open source statistics software R[2] for the exploratory data analysis since it has all necessary analysis functions and plotting tools already implemented and the package knitr[3] allows for reproducible analysis.

For the exploratory data analysis I used most variables as factors since they're not really on a continuous scale (s.t. standard arithmetics would make sense)[4]. I iterated over the process of loading the historic data in R and looking at it in basic tables and plots until I found out which strings to mark as `NA` (applicable

[1] see `problem1_smartfly/src/00_file_structure_analysis/file_structure_analysis.sh`

[2] `http://www.r-project.org/`

[3] `http://yihui.name/knitr/`

[4] For a list of variable types I used see `problem1_smartfly/src/01_exploratory_data_analysis/resources/raw_variables.csv`

to all variables!) right from the data loading, which data needs cleaning and reformatting. To get the necessary code straight without too long waiting time[5] I switched back and forth between the full set of historic data and a reduced set of 4000 observations (random observations not the first 4000, to get an impression of the data more close to the truth even with a reduced set).

Assumptions I took in these steps:

- "NA" in general and "" (empty string) for variables `cancellation_code` and `tail_number` indicate a missing value
- Scheduled times either start with "00" or with "24" (I decided to remap "24" to "00")
- I truncated scheduled times to the hour for better intelligibility
- I defined the target variable `is_delayed=(departure_delay > 0) & (cancelled == FALSE)` (as per the given definition in the problem description)

The most important statistic from this exploratory analysis of historic data is for me the percentage of delayed flights (based on `is_delayed`), which amounts to 28.6%. If I see that as a base rate, the model I estimate has to have an error rate better than this (because otherwise I could just say at random with a probability of 28.6% that a flight is delayed - without any input data or model).

After I applied the same pre-processing steps (mark missing values, mark as factors, truncate times, reformat labels) I ran the same exploratory analysis on the scheduled flight data with special focus on factor levels:

- The variables seem to be roughly distributed in the same manner as they are in the historic data, except for the values for `year` and `month` (which stems from the problem itself).
- Not all factor variables have the same levels as in the historic flight data[6].

The difference in factor levels mattered to me since I have the experience, that when solving a classification problem with a random forest in R, the training and the prediction data need to contain the same variables with the same levels. Otherwise you can't even predict with the random forest.

1.3 Prepare historic data for modeling

Independent from the model I choose it's better to remove all variables that don't help with the prediction from the data before applying an model estimation algorithm. In this case I drop the variables `taxi_time_in`, `taxi_time_out`, `cancelled`, `cancellation_code` because they are not available in the scheduled flight data - and thus can't be used as input for delay prediction. I also drop `departure_delay` because I condensed its information into a new target variable `is_delayed` already.

Since I opted for a random forest model implemented in the R package `randomForest`[7] I also needed to work around the issue that this implementation throws an error if factor variables with more than 53 levels are included in the data. My first (and only) solution to this was to exclude the concerned variables (`flight_number`, `tail_number`, `origin_airport`, `destination_airport` - note that the scheduled time variables did not fall into this category since I truncated them to the hour).

Note that although I assume that the variable `id` does not have any prediction power regarding the delay of a flight, I don't drop it from the data since I need it for identification in the prediction stage to come up with the required order list of flight IDs.

To work around the issue that a random forest requires the training and the prediction data to contain the same factor levels (and `year` and `month` clearly have different levels in the historic and the scheduled flight data) I transform the date and time related variables to numeric data types.

Furthermore, the algorithm for random forest does by default throw an error when encountering missing values so I check if there are missing values in any of the *remaining* variables - and there are not.

[5] The compilation of the basic plots and summary statistics for the full historic data set of 7.3 million observations took about 4 minutes per run.

[6] Venn diagrams helped a lot to come with a clear picture of this. See `problem1_smartfly/src/01_exploratory_data_analysis/exploratory_data_a`

[7] `http://cran.r-project.org/web/packages/randomForest/index.html`

1.4 Estimate and validate a model

I used a training sample consisting of 7% (i.e. 516.206 observations) of the historic flight data to train a random forest with default parameters:

- number of variables tried at each split: `mtry=floor(sqrt(ncol(df)))=3`
- number of trees in the forest: `ntree=500`

(About model selection criteria: The first model I envisioned was a logistic regression, however I could not get it to work using package glmnet in R. So I switched over to random forests in R. The reason behind both options was that I already had experience estimating these in R. Witin the model I did not select model parameters beyond the default because I did not have enough memory to run crossvalidation which would have picked the best model parameters.)

Anyway, here is the list of variables I included for the my basic random forest estimation:

```
training sample: 516206 obs. of  12 variables:
 $ id                      : chr  "3280125367763225179" "6587250861035043912" ...
 $ year                    : num  2013 2013 2013 2013 2013 ...
 $ month                   : num  8 8 8 8 8 8 8 8 8 8 ...
 $ day_of_month            : num  8 29 10 17 3 13 3 9 17 6 ...
 $ day_of_week             : num  4 4 6 6 6 2 6 5 6 2 ...
 $ scheduled_departure_time: num  7 7 12 12 15 15 18 12 12 6 ...
 $ scheduled_arrival_time  : num  8 8 14 14 17 17 19 15 15 8 ...
 $ airline                 : Factor w/ 17 levels "AA","AS","B6",..: 15 15 15 15 15 ...
 $ plane_model             : Factor w/ 6 levels "737","747","757",..: 6 6 2 1 2 3 4 ...
 $ seat_configuration      : Factor w/ 6 levels "Standard","Three Class",..: 1 3 4 ...
 $ distance_travelled      : num  185 185 508 508 329 329 213 809 809 329 ...
 $ is_delayed              : Factor w/ 2 levels "on_time","delayed": 1 1 1 1 1 1 ...
```

The estimated random forest has an estimated error rate of 27.5% which is pretty bad given the base rate of 28.6%.

In order to improve the model I would check out a plot of accuracy vs model complexity ("fitting graph") for train and test data set to get an impression whether the model overfits. I would double-check the result by having a look at a plot of error rate vs training set size ("learning curve").

1.5 Current Limitations

- I excluded some observations from prediction because of new values in factor variable `airline` → these flights are not contained in the output!
- I only included a small percentage of the massive historic data for training
- An error rate of 26% of the estimated model is not acceptable.

2 Used Software

In general I developed on a Macbook Pro with a 2.3 GHz Intel Core i7 Processor and 16GB Memory.

- git
- LateX
- R in RStudio with packages caret, ggplot, knitr, plyr, randomForest
- shell command line

3 Time Spent

- 5h general setup and basic descriptive statistics
- 4h (failed) logistic regression
- 10h exploring data
- 12h preparing data for random forest modeling & prediction
- 10h modeling and predicting data
- 1h wrting solution abstract
- 3h cloud server setup

Summing up I spend xx hours on this problem.