Predicting response times of the Paris Fire Brigade vehicles

Introduction

This data subject is extracted from a list of challenges proposed by the ENS.

The goal is to be able to predict the response time of the fire brigade of Paris. With this information the fire brigade can choose better which vehicle to select and where in order to optimize their response time which could save lives.
The goal is then to predict which vehicle to choose and how long it will last to arrive on site.

The data

The data is composed of a training set and a testing set. However, the testing set output is not available since it is used for evaluating the score of the ML algorithm on the ENS platform.

A detailed presentation of the data is available here https://paris-fire-brigade.github.io/data-challenge/challenge.html

The output has four components:

- The ID of the chosen vehicle
- The time between selection of the vehicle and the departure
- The time between departure of the vehicle and the arrival on site
- The time between selection of the vehicle and the arrival on site (sum of the first two)

To predict those outputs, we have a very detailed data set at hand. I will not go through each column in this notebook, but we can divide those columns into four categories:

- The continuous variables, which can be used for regression
- The categorical variables, which can be used for classification
- The labels variables, which are not really useful and are often associated to a categorical variable
- Other types of variables that cannot be directly used (such as latitude and longitude and others...)

There are also some supplementary files that can be used in the study, but I will not go through this since they are more complicated to use. Maybe at the end to enhance the model.

Methodology

The objective is to predict the time of response of the fire brigade. This response time is divided into two different times: the departure time ("delta selection-departure" time between the selection of the vehicle and the departure of the vehicle) and the transit time ("delta departure-presentation" time between departure and arrival on site).

Those two times are different and should each have their own approach. For example, we've seen that the estimated distance travel did not influence the departure time whereas the alert category influenced both times.

Therefore, in a first time we'll try to predict the departure time. Then we'll try to predict the transit time.

But before all that let's prepare out data.

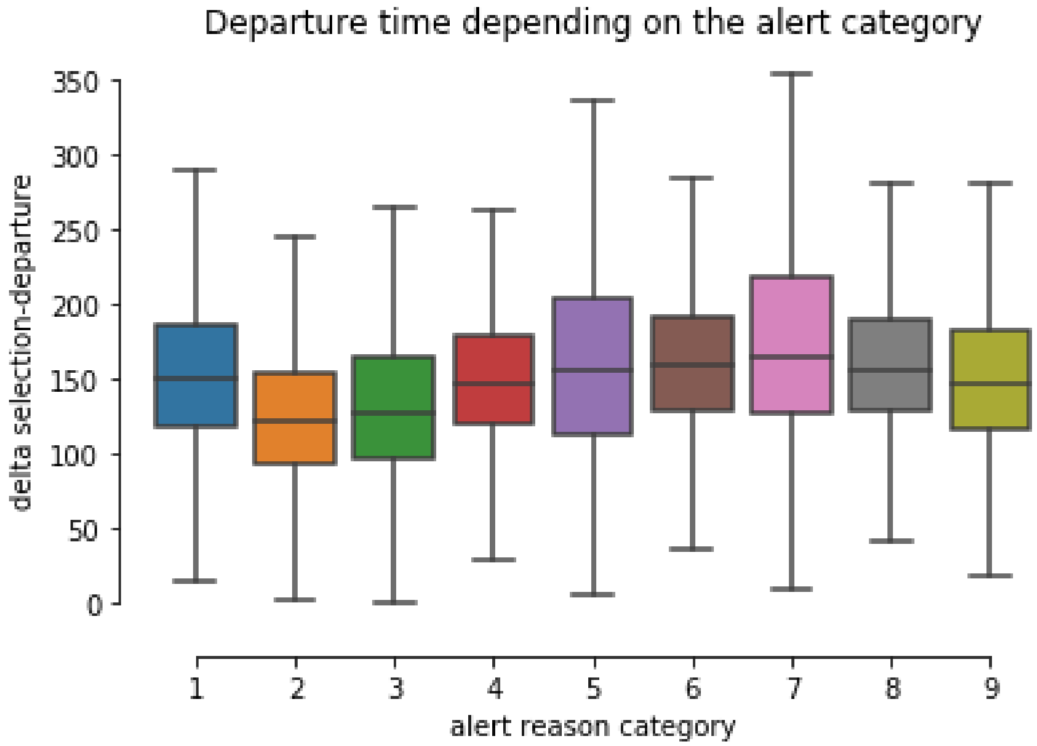
Departure time

So now let's focus on predicting the departure time. To do this we first have to select which features we are going to keep, then which model/techniques we are going to use to predict the departure time.

Feature selection

What we're going to do here is plot the evolution of the departure time of all the features to determine which ones we are going to keep. But first let's make a selection because not all the features are exploitable. They could be but that would require a preprocessing, we'll keep those features for a later enhancing.

For selecting the different features, I had a look at how the feature could influence the output. For example, I plotted this graph which shows that the alert reason category does influence the output. Thus, I will keep this feature.

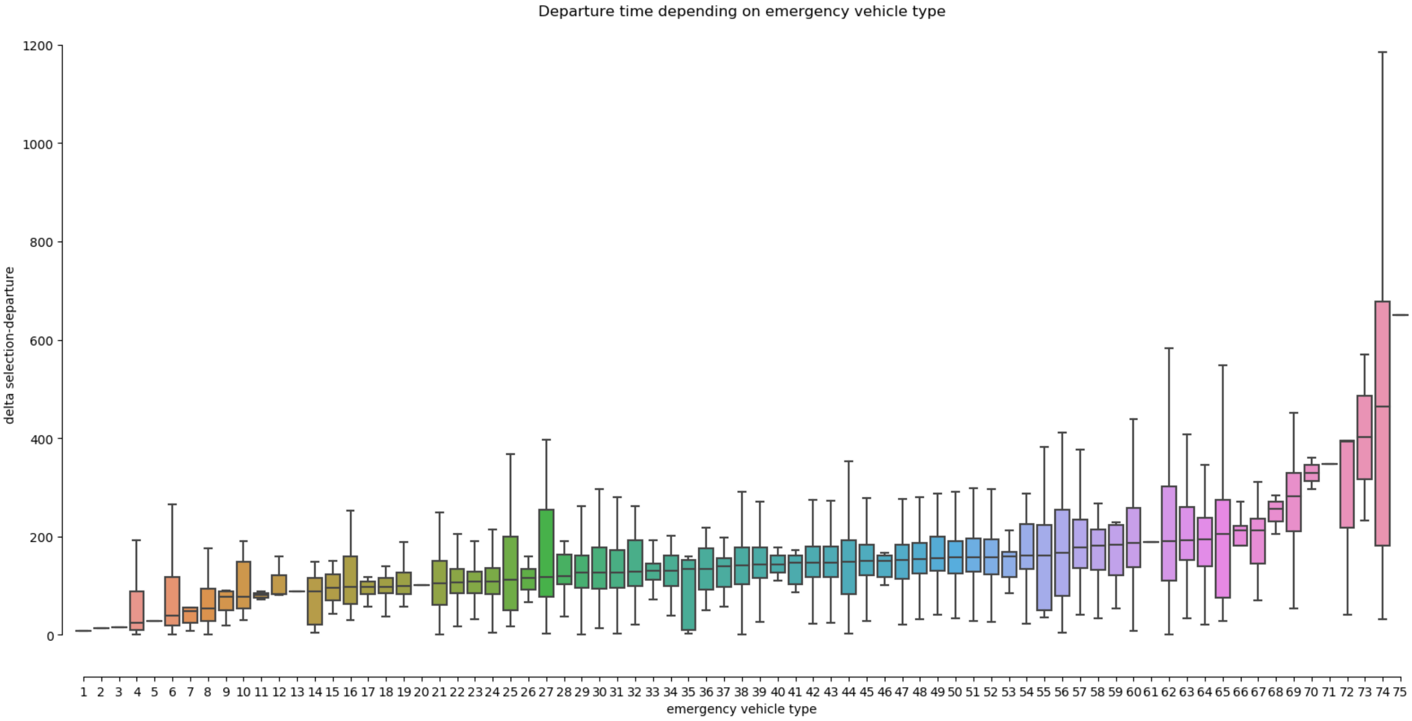


After analysis of all the different features I chose to keep the following ones for the departure time:

- alert reason category
- emergency vehicle type
- time key selection
- date key selection

I would then like to use a linear regression on those features. However, it is hard to apply linear regression directly on categorical variables. To solve this I whill change the labels of each category, ranking them from lowest median departure time to highest median departure time.

After applying such transformation, we obtain this figure on the emergency vehicle type category:



As described, they are ranked in growing median time of departure.

We then apply to these different features a linear regression with a polynomial order of 3.

Transit time

Feature selection

We are also going to select some of the features in order to simplify the dataset.

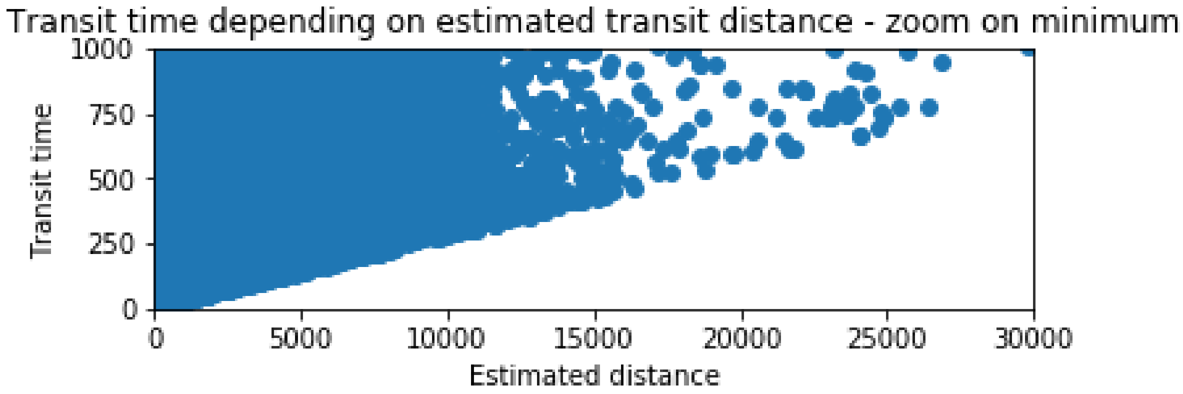
The features selected are

- OSMR estimated distance
- OSMR estimated duration
- Intervention on public roads
- Floor

The first two features are continuous variables and can be directly used into a linear regression. The last two are categorical. However, floor is already ordered (the higher the floor the longer it takes to go) and intervention on public roads is a Boolean and thus require no transformation.

Let’s plot the transit time depending on the OSMR estimated distance. In this figure I have zoomed on the minimum to show the relevant information.

What we can see on this figure is how the OSMR estimated distance imposes a strict linear relation between the minimum of the transit time and the estimated distance. This is interesting to us because it shows that this it can be pertinent to apply a linear regression to this relation.



Modeling

We are going to apply a multi linear regression to the data. This is the simplest way to obtain results.

On the departure time data set I have tried with different type of algorithms (K-means, random forests…) since the features are categorical but they did not have better scores. Maybe due to the noise in the data. The point is it is with linear regression I obtained the best scores on the departure time.

Then on the transit time, since most variables are continuous, linear regression is the natural choice.

For each linear regression I used polynomial degrees of 3. This degree is sufficient, and it impedes the model to overfit.

Moreover, to test our models we split our data into a training set and a test set in order to properly evaluate the performance of our work.

Results

To evaluate the accuracy of our model we will use the R^2 coefficient.

The average score obtained of the local test sets are averaging a little over **38%.** This score is low compared to what a good predictive model should do. However, this data set was provided in the scope of a competition, therefore it is not meant to be easy. The data is very noisy. A score of 38% is thus not that bad. It shows that the model helps to predict the output despite not being so precise.

Moreover, the models used here were simple and did not use any gradient boosting techniques such as XGBoost. This may be implemented in the future.

Conclusion

The Paris fire brigade provided a data set in order to set up a competition in which data scientists or students could try to obtain the best score possible.

This data set contained a lot of noise and was hard to use.

I decided to make a simple model on it and used only the most obvious and easy features. After some transformations on the data, I applied two independent multilinear regressions.

This model got me to a score of 38% which is not that bad given the difficulty of the data set.

However, a lot of improvements can be done. The model could implement the other features, a frequency analysis could be performed on each feature to determine if it is pertinent to use or not, gradient boosting techniques can also be used and much more secrets that data science has yet to reveal to me…