# Predicting response times of the Paris Fire Brigade vehicles

Antoine Moulin
Télécom Paris & ENS Paris-Saclay
antoine.moulin@ens-paris-saclay.fr

March 25, 2020

## 1 Introduction

The response time is one of the most important factors for emergency services because their ability to save lives and rescue people depends on it. A non-optimal choice of an emergency vehicle for a rescue request may lengthen the arrival time of the rescuers and impact the future of the victim. This choice is therefore highly critical for emergency services and directly relies on their ability to predict precisely the arrival time of the different units available.

Given a dataset of interventions done by the Paris Fire Brigade in 2018, the goal of this challenge is to predict accurately this latter response time. As shown in figure 1, it is divided into two parts: (i) the time between the selection and the departure of the unit (i.e. the time firefighters take to prepare), and (ii) the time between the departure and the arrival of the unit (i.e. the time they spend on the road). Note that the vehicle does not have to be chosen.

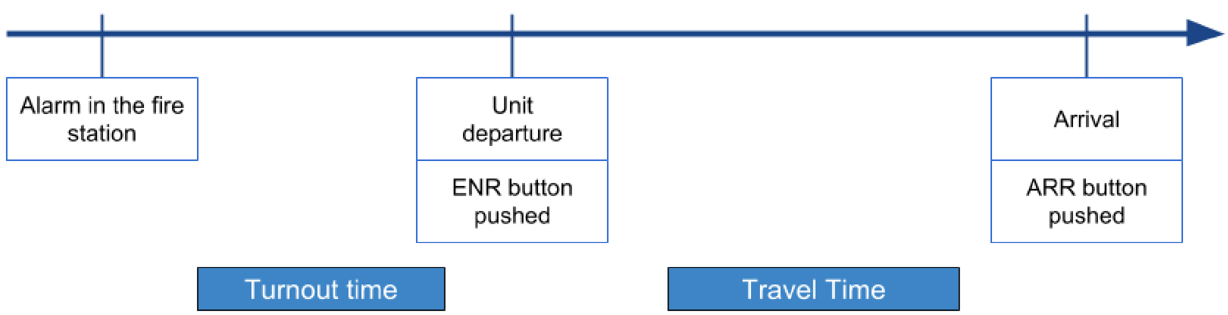


Figure 1: Definition of the response time. Illustration taken from [1] which shows how the measurement is done in Montréal.

In section 2, we quickly present the dataset used. In section 3, we use the knowledge we have about the problem to design new features that help to better predict the response time. In section 4, we describe the models we used, as well as the fine-tuning done. Finally, in section 5, we conclude and try to provide ideas to explore.

## 2 Data Analysis

In total, 327k interventions are available: 219k for training, and 108k for testing. There are 31 initial features (that are not always all available). The features can be grouped into four categories:

- the characteristic features with which we can identify the different components of the intervention,
- the vehicle features which give information on the vehicle and its center,
- the intervention features which give information on the intervention's nature,
- the spatial-temporal features which describe where and when the trip to the intervention took place.

**Characteristic features**. Those 4 features identify the vehicle selected, its rescue center and the intervention with 4 IDs: `emergency vehicle selection` (int), `rescue center` (int), `intervention` (int), and `emergency vehicle` (int).

**Vehicle features**. Those 3 features give information on the vehicle and its situation (the names are quite explicit): `emergency vehicle type` (category), `status preceding selection` (boolean), and `departed from its rescue center` (boolean).

**Intervention features**. Those 5 features give information on the nature of the intervention, i.e. if there is a fire, an accident or something else with the two features `alert reason category` (category) and `alert reason` (category) which has a finer granularity (9 categories for the first feature, 126 for the second), and where the intervention takes place with the features `intervention on public roads` (boolean), `floor` (int) and `location of the event` (category). The last one gives the place of the intervention, e.g. entrance hall, motorway, etc., and takes 210 different values.

**Spatial-temporal features**. Those 19 features give information on the position of the departure or the arrival – with `longitude intervention` (float), `latitude intervention` (float), `longitude before departure` (float), `latitude before departure` (float) –, the time of the selection – with `selection time` (date), `date key selection` (date), `time key selection` (date), `delta status preceding selection - selection` (float) –, the GPS track when available (only 30% of the data) – with `delta position gps previous departure-departure` (float), `GPS tracks departure - presentation` (list of longitudes/latitudes), `GPS tracks datetime departure - presentation` (list of dates) – and finally, the estimate provided by OSRM, a service route response with `OSRM response` (dictionary), `OSRM estimate distance` (float), `OSRM estimate duration` (float).

The 5 last features are provided (for only 30% of the data) by another dataset, which updates the OSRM data with the features `OSRM estimate from last observed GPS position` (dictionary), `OSRM estimated distance from last observed GPS position` (float), `OSRM estimated duration from last observed GPS position` (float), `time elapsed between selection and last observed GPS position` (float) and `updated OSRM estimated duration` (float).

**The response time**. With those features, we have to predict the turnout time and the travel time, which are shown in figure 2. We can easily imagine that it does not take the firefighters 10 minutes to get ready, or 30 minutes to arrive at the destination. Hence, there are outliers to remove. After the cleaning – during which we removed less than 2k examples –, we better see the distributions, as shown in figure 2. On average, it takes about 2:18 minutes to prepare and 5:48 minutes to arrive.

Now, let us analyze the different features and see if we can find some insights.

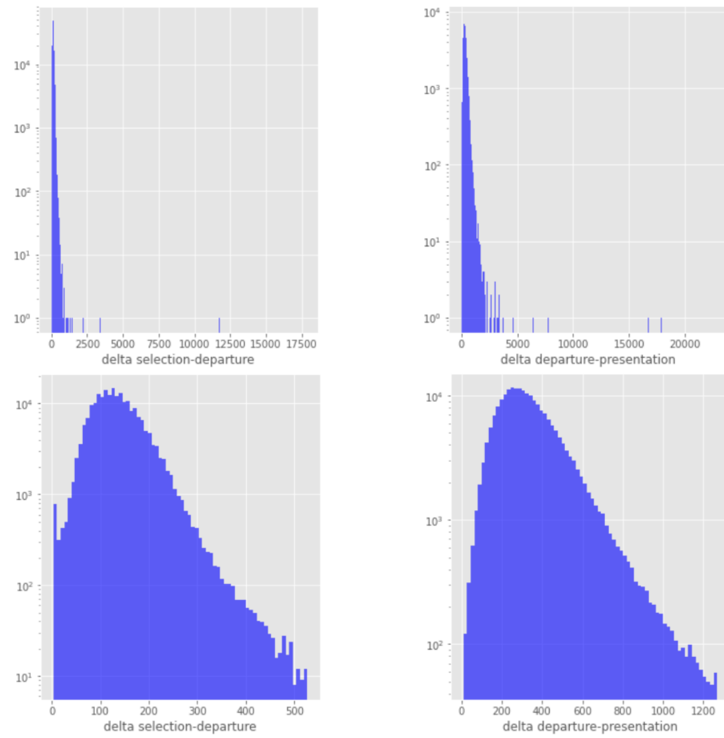


Figure 2: At the top, distributions of the turnout and travel times before cleaning, respectively. At the bottom, distributions after removing the outliers.

## 2.1 Vehicle features

The feature `status preceding selection` can take two values: "available" or "returned". We could expect the "available" vehicles to be faster than the others, but as shown in figure 3, except the proportions, we do not notice a big difference. The same thing applies to the feature `departed from its rescue center`.

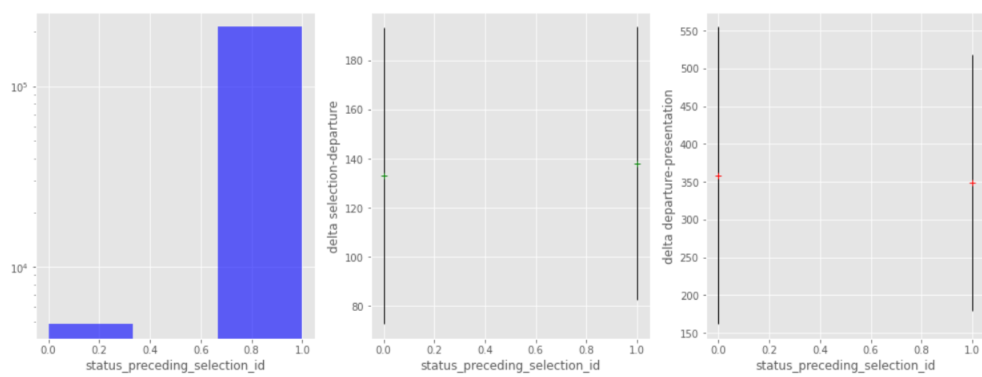


Figure 3: Distribution of the `status preceding selection` feature and the associated times.

Regarding the `emergency vehicle type`, there are 70 different types of vehicles. Among them, we can notice several trucks, cars or even boats, as shown on this website: https://www.pompiersparis.fr/fr/operationnelle/engin. Hence, when plotting the average times as a function of the types, one could expect to see emerge two groups, one with slow vehicles and one with fast vehicles. The initial distribution is shown in figure 4. What we see is that some types are used only once, making them useless for learning. Some others have a small standard deviation but do not appear enough to be statistically relevant, and other types differ greatly from the average but have a huge standard deviation.


Figure 4: Distribution of the different vehicle types for the travel time.

Hence let us try to reduce the number of types to see what we get. To do so, we notice that a vehicle type has the format (at least we suppose it is the case for some types) `<acronym> <place>`, such as `UMH 94` or `UMH GARC`, which is why I regrouped the types using the first word of the name, giving 51 different groups. The result is shown in figure 5.

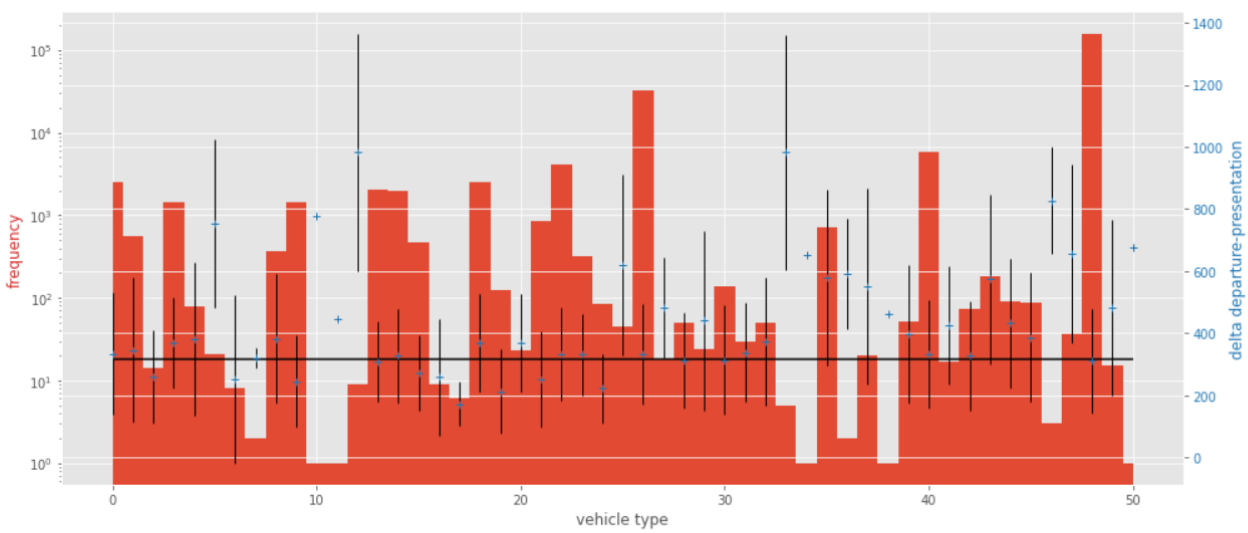


Figure 5: Distribution of the vehicle types after regrouping.

Unfortunately, we see that we still have similar issues. After reducing the number of groups to 30, 20 or 10 (by iteratively merging the groups whose means/medians were close), noticing that it is hard to distinguish two groups (and hence find the slow and fast vehicles), I stopped working on this feature. This could be due to the fact that the speed is influenced by the traffic on the road, which does not depend on the vehicle type. Not using this feature also reduces the dimension of our training set by more than 50, which is not negligible.

## 2.2 Intervention features

As mentioned before, `alert reason category` and `alert reason` both describe the nature of the intervention but on two different levels: the first takes 9 values, while the second takes 126 values. As we could expect for the latter, we could hardly extract informative subgroups, hence we focus on `alert reason category`. This feature appears to be quite useful as the samples are well distributed among the categories and that despite huge standard deviations, we can notice differences in time between those categories, as shown in figure 6. This may come from the fact that depending on the reason of the intervention, different equipment is required, e.g. if there is a fire or if someone had a heart attack leading to different turnout times, or if different vehicles are required and that some of them are slower or faster (hence, it could provide us the information we could not get with the vehicle features).

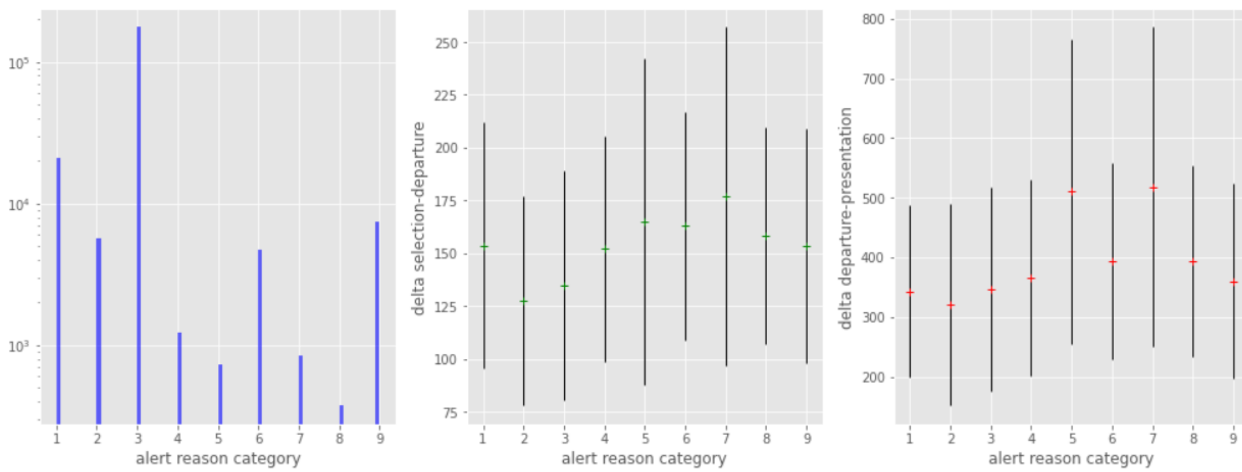


Figure 6: Distribution of the `alert reason category` feature and the associated times.

The `intervention on public roads` feature is interesting as well as we can see a difference (we do not plot it here) on the travel time between the two categories, which is normal as it is faster to get to an intervention that takes place on a public road.

There are still two features to mention: `location of the event` and `floor`. For the same reasons I did not use `alert reason`, I did not use `location of the event` as well, as it can take 126 different values. Regarding the `floor` feature, it initially ranged from -10 to 100. As there is no building higher than 60 floors in Ile-de-France, there remained outliers, which is why this feature was manually upper bounded to 20. Similarly, it was lower bounded to -3 (to group the samples that had a lower floor than -3).

## 2.3 Spatial-temporal features

Now comes the most valuable part of the features. Let us first analyze the geographical positions of the departures and arrivals using the longitudes and latitudes available. We create a graph whose nodes are formed by the departure and arrivals points and whose edges are created between a departure and an arrival. The result is shown in figure 7.

What we notice is that, except for two zones, the interventions span most of the region Ile-de-France and that the rescue centers are well distributed (the test set is very similar). Even though it is hard to see on the figure, we notice by looking at the edges that some centers intervene relatively far from their position, i.e. that not every intervention is managed by the closest rescue center. This could be due to the fact that some specialized units are only located in some centers, or that no vehicle is available in the closest center at the

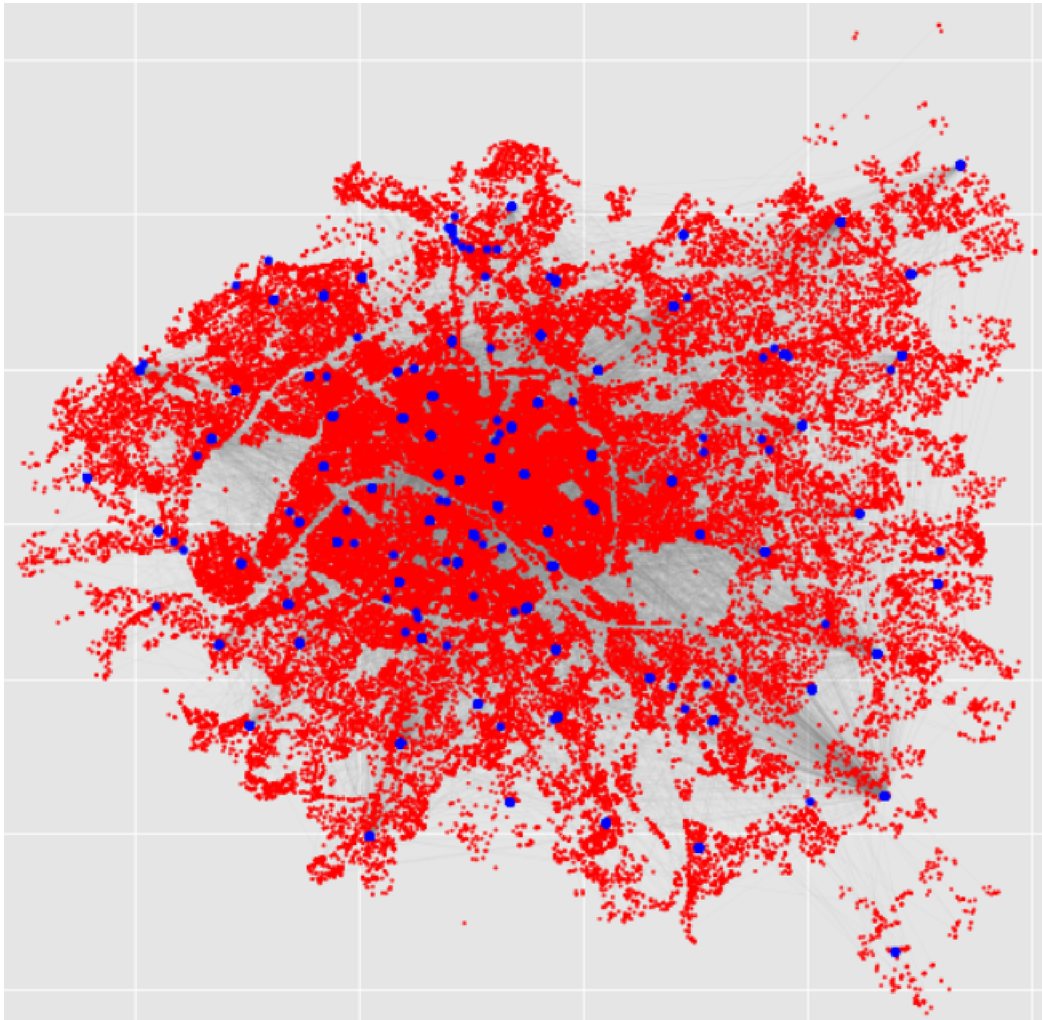

Figure 7: Graph of the interventions on the training set. Red points are destinations, blue points are rescue centers. It is not plotted here but the test set spans the same area.

moment of the intervention. Those features are very interesting as they can be used to estimate a distance, a direction or a speed between two points, which will be discussed in section 3.

Regarding the dates, the features we have enable to extract some useful information thanks to the month, day or hour of the intervention. This will be discussed in section 3.

There are still two kinds of features to describe: the GPS tracks and the OSRM response. The first is available for only 30% of the data but will be used to refine the estimate of the response time as explained in section 3. The second, with the OSRM response, is very interesting as it already estimates both the length and the duration of the trip. We plot the estimated duration and distance with the travel time in figure 8.

We see that for both features, we have a positive correlation, which is hopefully what we expected. Those features are going to be the most useful during the training. However, we can see that for many points, the estimate is not very accurate. We will refine it, as shown in section 3. This will be done not only with the GPS tracks mentioned above but also with the additional data provided, especially with the feature `updated OSRM duration`.

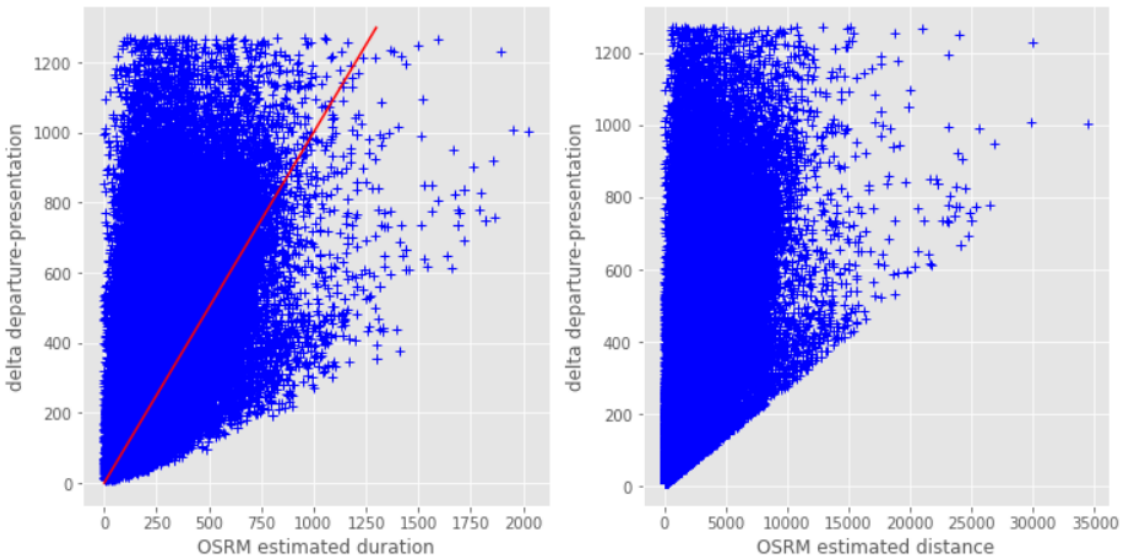


Figure 8: On the left, plot of the OSRM estimated duration and the travel time. On the right, plot of the OSRM estimated distance and the travel time.

# 3 Feature engineering

Now that we analyzed the interesting features, it is time to use the information we have on the problem to obtain a better representation. Note that regarding the categorical features, we use one-hot encoding.

## 3.1 Capture seasonality with dates

The first thing one can think about is the variation of traffic density as a function of time. Indeed, in the morning, everyone goes to work (e.g. from the suburbs to Paris), which creates traffic jams and slows down every vehicle. Similarly, at the end of the day, everyone goes home, and the same thing happens. If firefighters go on an intervention during these periods, it is obvious that it will take more time to travel. Besides, it also depends on the day of the week, as the traffic changes during the weekend, and during the different months. Hence, we extract these pieces of information thanks to the dates available in the dataset. The results are shown in figure 9.

As expected we notice some differences. For instance, the turnout time is higher during the early morning (before 5 am) or on Sunday, probably because fewer firefighters are working during this period. Similarly, it is faster to get to the intervention place during a weekend.

## 3.2 Meteorological data

It is fair to assume that the weather influences the response time. Indeed, if it is very cold, the firefighters would have to take more clothes, hence increasing the turnout time, and if it rains, the vehicle has to move slowly. Several features concerning the weather can be found on this website: https://www.historique-meteo.net/france/ile-de-france/paris/2018/. Thanks to a script that retrieves automatically these data, it is possible to have: the maximum temperature, the minimum temperature, the wind speed, the wind temperature, the rain level, the humidity, the visibility, the cloud coverage, the heat index, the dew point temperature, the pressure, the date of the sunrise, the date of the sunset, the length of the day and a review (e.g. very favorable). For instance, figure 10 confirms what we expected regarding the rain level.


Figure 9: Turnout and travel times as a function of the hour, day or month.

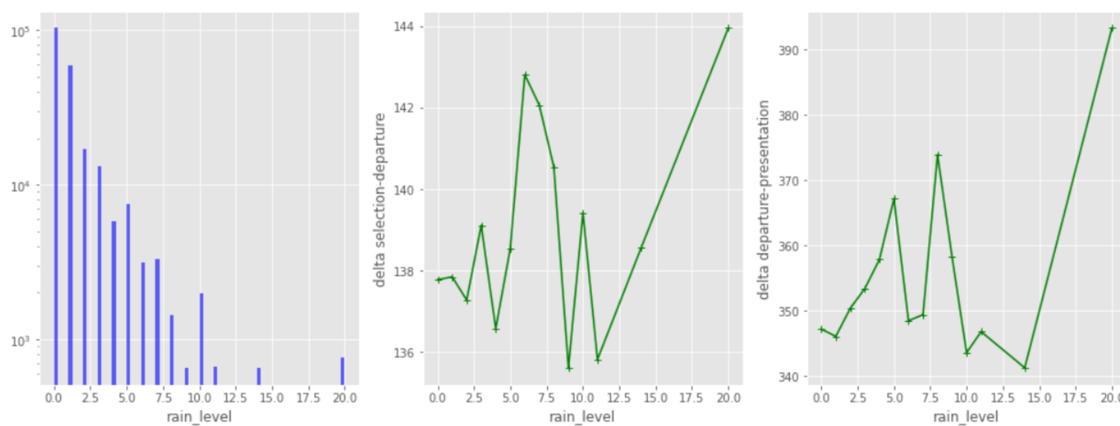


Figure 10: Response time as a function of the rain level.

Hence, we use some of these features during the training, especially the rain level, the wind speed, and the review.

## 3.3 Taking advantage of the longitudes and latitudes

As mentioned above, it is possible to use the longitudes and latitudes available to estimate a distance, a direction, and an average speed. A code can be found in [2] to compute the Haversine distance, the Manhattan distance, and the direction. The average speed is computed using the OSRM estimations (distance divided by the time). We do not plot it here, but for instance, with the average speed, we can notice a similar pattern than the ones observed in figure 9.

Another information we can extract with the longitudes and latitudes is about the neighborhoods. Indeed, we can assume that depending on the neighborhoods of the departure and the arrival, the response times will be different. The result is shown in figure 11.

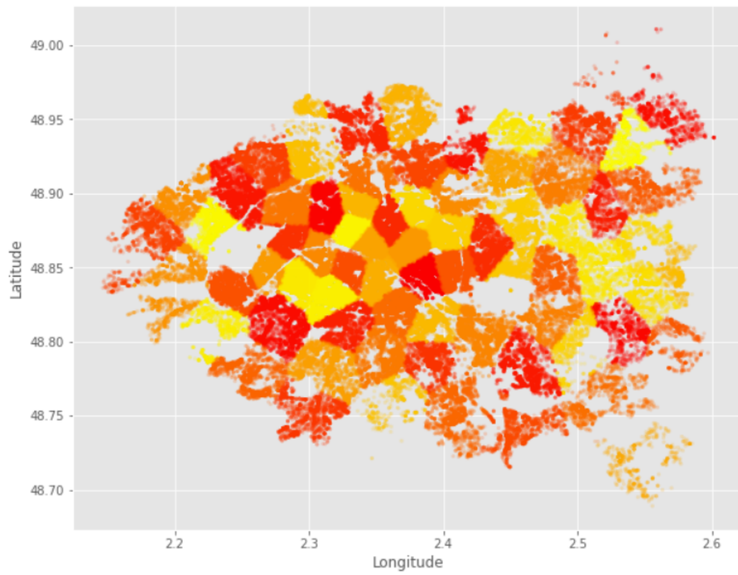


Figure 11: Neighborhoods obtained with a KMeans on the longitudes and latitudes.

## 3.4 Refine the OSRM estimation

As said above, even though the OSRM estimated duration is very useful, it is sometimes quite inaccurate. Hence, let us try to refine it. There are two things we can do:

- First, we can use the additional data and the feature `updated OSRM estimated duration` when it is available (30% of the data) to simply replace it with the previous value.
- Second, we can use the GPS data when available to retrieve the date of the first and the last positions and compute the difference to have an estimation of the travel time.

We can see the impact it has on the estimation in figure 12.

We see that some points are now much closer to the diagonal, i.e. better estimate the travel time. The correlation seems lower when using the additional data, but this is only due to the few outliers it induces. We also notice that most of the points with the additional data lie below the diagonal, while the points with the GPS tracks lie above the diagonal. Let us take a closer look at this.

When removing the outliers and considering only the points for which the additional data and GPS tracks are available, we plot the travel time as a function of the estimation in figure 13. We notice that the additional data tend to almost systematically underestimate the travel time, whereas the GPS tracks always overestimate it. In both cases, it correlates well with the travel time, but an idea to improve it is to average those both estimations to get a better result.

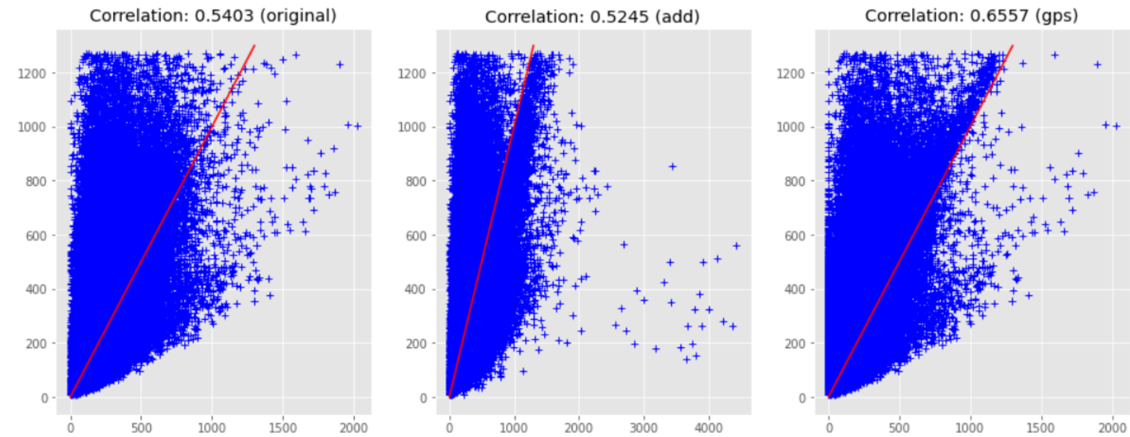


Figure 12: On the left, original OSRM estimation. On the middle, the updated OSRM estimation has been used. On the right, the GPS tracks have been used. The y-axis corresponds to the travel time.

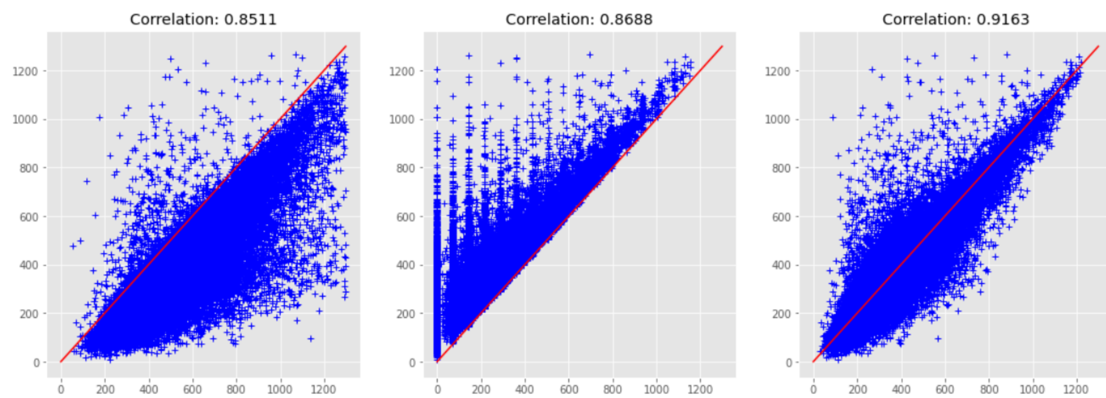


Figure 13: Estimation when using the additional data (on the left), the GPS tracks (in the middle), or the mean of both (right).

# 4 Model

Now, let us talk about the models considered. First, we apply a logarithm function to both the turnout and travel times.

Regarding the basic models, I first tried a linear regression with L1 or L2 regularization (i.e. Ridge regression or LASSO), as well as a polynomial regression. However, these models did not yield very good performances.

The problem probably not comes from the fact that some of these models are linear, as we can see in figure 13 that linear regression can estimate relatively efficiently the travel time. The problem likely comes from the fact that a lot of the features used are categorical. A thing we would naturally do as human beings would be to first identify some groups of interventions by looking at all the categorical features (type of intervention, type of vehicle, which day, etc. as explained in sections 2 and 3) and then try to make a regression on this subgroup using the continuous features (e.g. by looking at the means of medians). Hence, we first have to arbitrarily decide which samples to consider together. This is not possible with the models mentioned above. We could think about kernel methods but it would not solve the problem, and given the size of the dataset, it would take too much time to train.

Hence, a natural direction to take is to use decision trees, and more precisely, I decided to use a boosting method, with the library `XGBoost`. This is the model that yielded the best performances, which could be explained by the previous remark. The different parameters such as the maximum depth, the minimum children weight, the sampling policy or the regularization parameters were fine-tuned, and the loss function used was the squared error (used on the logarithmic times). To evaluate the models, I used 3-fold cross-validation. A plot giving a feature importance ranking is shown in figure 14. The features designed in section 3 were the most useful.

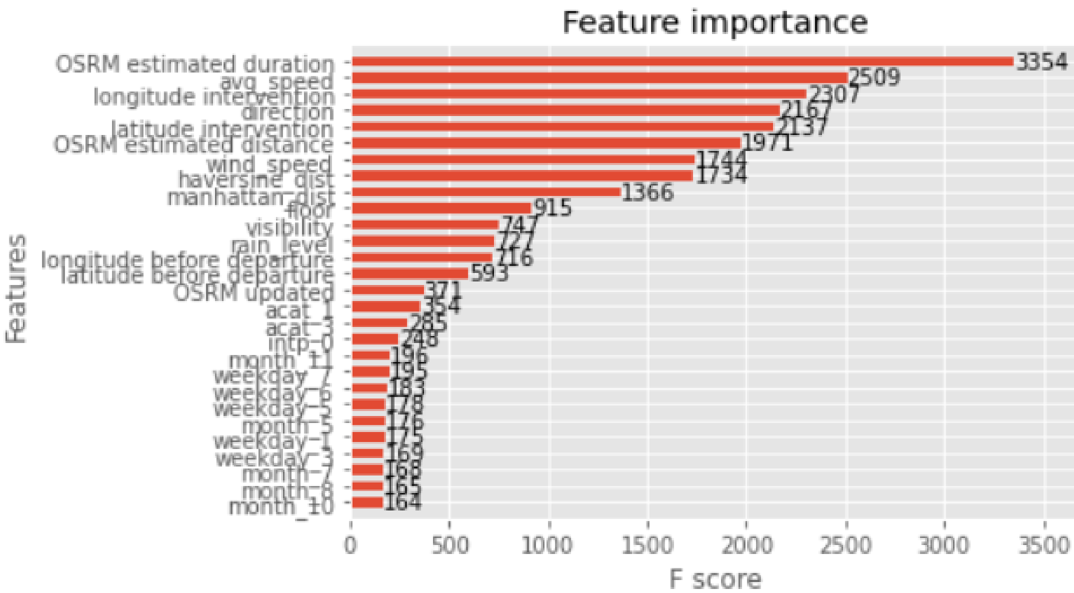


Figure 14: Feature importance of a XGB model.

One could also think about using a neural network. However, the challenge is not really designed for deep learning models. Indeed, because of these categorical features, a neural network is likely to quickly overfit and hence yields poor performances on the test set, which is why I did not keep such a model. Because of a lack of time, I did not have the time this idea but it could be interesting to create an attributed multi-graph (similar to the one shown in figure 7) and train a graph neural network, to capture some local interactions that we could not get with decision trees (e.g. two interventions that have close departure and arrival are likely to share most of the path from one point to another, and hence have similar response times, but it is not directly taken into account with the other models).

# 5 Conclusion

Finally, we showed that using the information we had on the problem, it was possible to design relevant features and obtain a good representation of the different interventions. The best performances were obtained with a boosting method using decision trees, but as mentioned in section 4, it would be also interesting to try using a graph neural network to solve this task, as it encodes interactions that are not directly captured in a boosting model.

# References


[1] Cyril Pecoraro. Using data science to predict response times of firefighters. *Medium post*, 2018.

[2] Kaggle challenge: Nyc taxi trip duration, https://www.kaggle.com/c/nyc-taxi-trip-duration/notebooks.