Springboard Data Science Intensive Program

Capstone Project 1:

Paris Road Traffic Prediction

By: Julien Laks

June 2020

Table of Contents

I. Introduction and Data Description

While I was working as a cultural tour guide in Paris, cycling German and American tourists around the city and showing them the hidden spots, I realized that car traffic had a huge impact on the well-being and the satisfaction of my clients. I therefore became curious to know if there is a way to understand traffic trends and predict how much cars will be at a certain place, a certain time, and a certain day of the year in Paris.

This is, of course, also a problem of general interest. Being able to understand and to forecast the intensity of traffic at certain locations into the future may help, for example , to choose better itineraries for driving, or to reorganize city planning. The results likely to generalize to other problems involving multiple time series, such as forecast of customer demand or of supply chain.

This following report contains the results of my investigation, which I conducted with all the more pleasure and interest that I love the city Paris, and am very familiar with the places and

The raw data I used was collected from the official Pairs OpenData website:

https://opendata.paris.fr/explore/dataset/comptages-routiers-permanents-historique/information/.

It contains the hourly records of traffic rates at 1700 different locations, (identified by their location ID) within Paris over a period of three years. As the data was divided into several files each corresponding approximately to one week of past records, I first had to concatenate these files into a workable dataset.

Pieced together, the original dataset was of medium size (25GB). I therefore had to be careful to use efficient computing methods, so as to limit the time and make the analysis possible on my own computer. Moreover, the dataset containing the geographical information of each location was stored in a different file, which I had to download separately before merging with the numerical data. After merging and removing unimportant information, each row of the dataset now contains :

- **`Timestamp:`** the exact time at which the traffic rate was recorded

- **Location_ID:** unique key identifying the location
- **Road_name :** the name of the road where recording station is located
- **Latitude :** location's latitude
- **Longitude :** location's longitude
- **Occupation_rate:** the acutal traffic occupation rate recording

A quick review of the dataset soon showed that many locations had no traffic records at all, So I just erased them, and kept only locations with a maximum of 5% of missing records. For the sake of predictive analysis, I later filled these missing values using an time interpolation method.

The dataset also contained some location IDs that corresponded to multiple locations. By taking a closer look I found out that such locations were always either the beginning or the end of the road segment corresponding to that location ID, so I just reassigned the different locations to the location with the maximum latitude.Lastly, the dataset also contained some duplicate time series, which I checked by comparing their means, and got rid of.

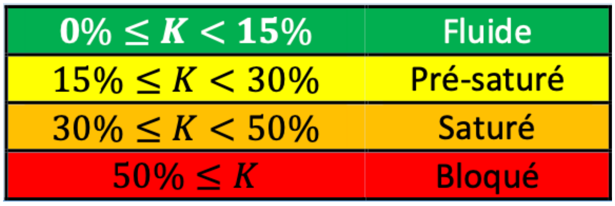
The cleaned dataset remains quite big, with more 43M rows. Each row represents one traffic rate record at a given place and a given hour.

II. Exploratory Data Analysis

a/ Defining the traffic occupation rate target variable

The magnitude of traffic at each recording station is measured using the traffic occupation rate, noted K. The occupation rate measures the percentage amount of time during which vehicles occupied the road segment where the sensor is located over 1 hour.

Globally, the occupation rate varies on a scale of 0 to around 60. The table below classifies the traffic occupation rate into 4 catogeries : Fluid, pre-saturated, saturated, and blocked. For the purpose of this study, we define a jam situation as K>30%



$0\% \leq K < 15\%$	Fluide
$15\% \leq K < 30\%$	Pré-saturé
$30\% \leq K < 50\%$	Saturé
$50\% \leq K$	Bloqué

b/ Global traffic rate patterns and trends

Not surprisingly, we observe a triple seasonality in the global patterns : yearly, weekly, and daily. Our task will be to a model that is able to capture such seasonal trends. The three graphs below capture this seasonality.

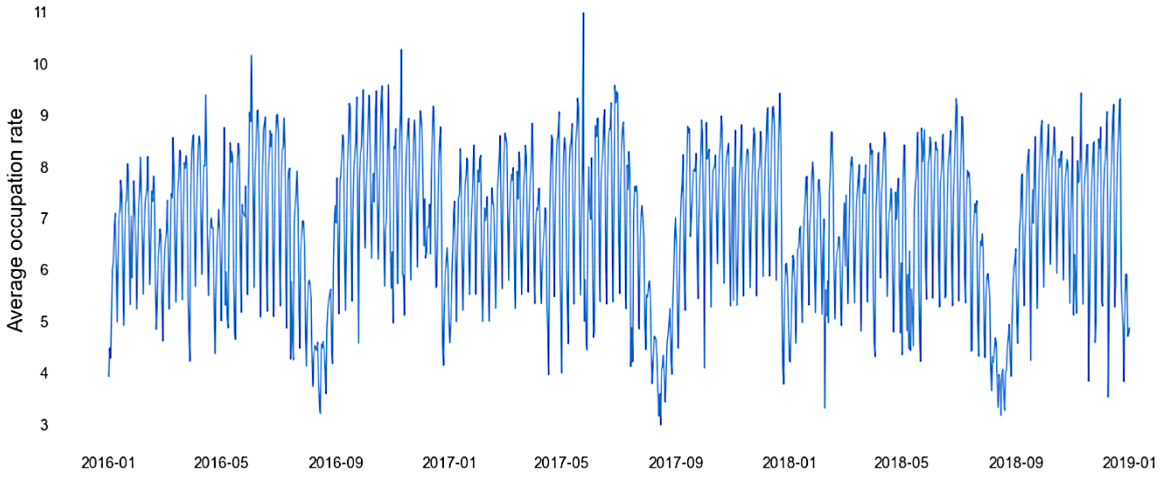


Figure 1. Daily Average Traffic Rates over 3 years

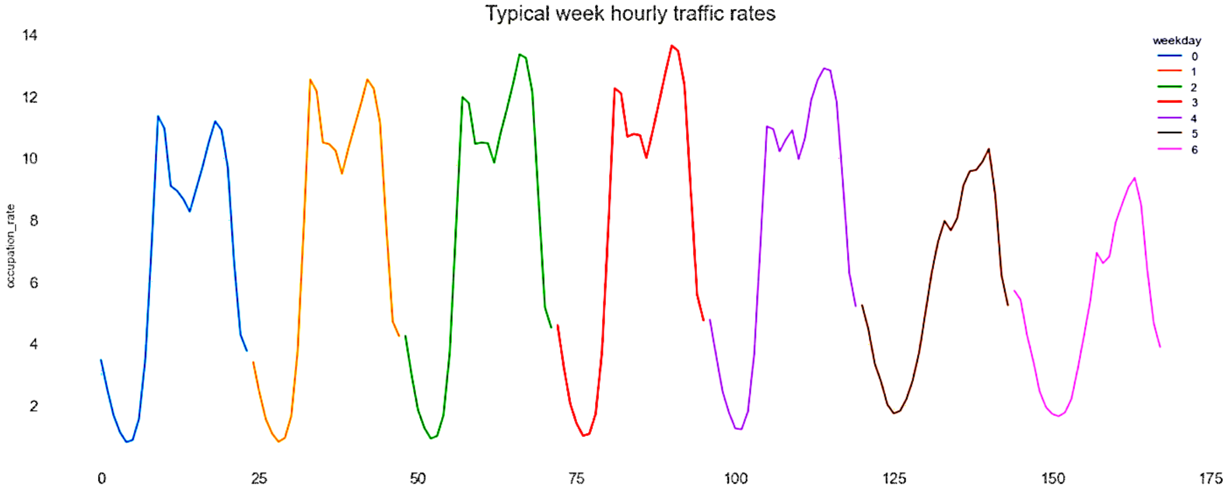


Figure 2. Hourly Average Traffic Rate over a typical Week

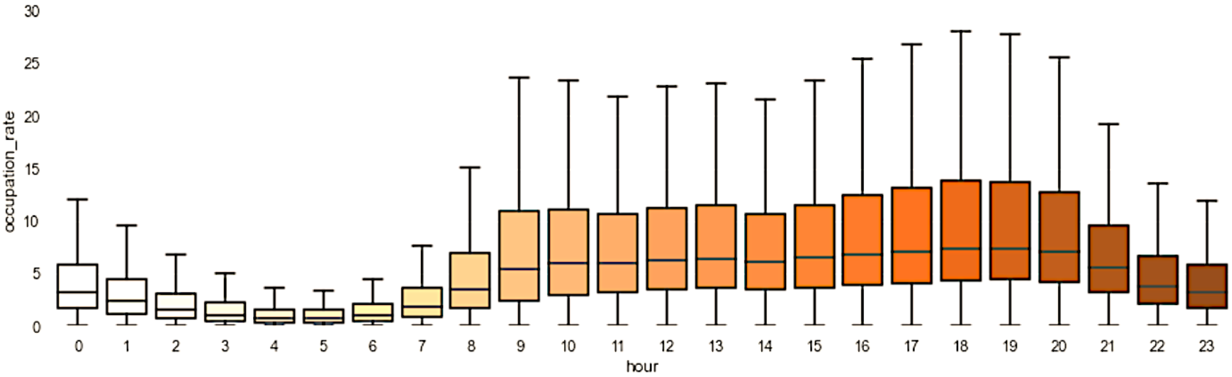


Figure 3. Hourly Average Traffic Rate over a typical Day

c/ Hourly Traffic rates at specific locations

The map below shows the average traffic rates at each location. The plots above clearly show that traffic rates are unevenly distributed. The traffic is globally fluid in most locations, but tends to be very high and staturated at some specific places. These locations obiously correspond to the main crossroads, where cars typically need to stop for longer times, leading to a cumulative phenomenon

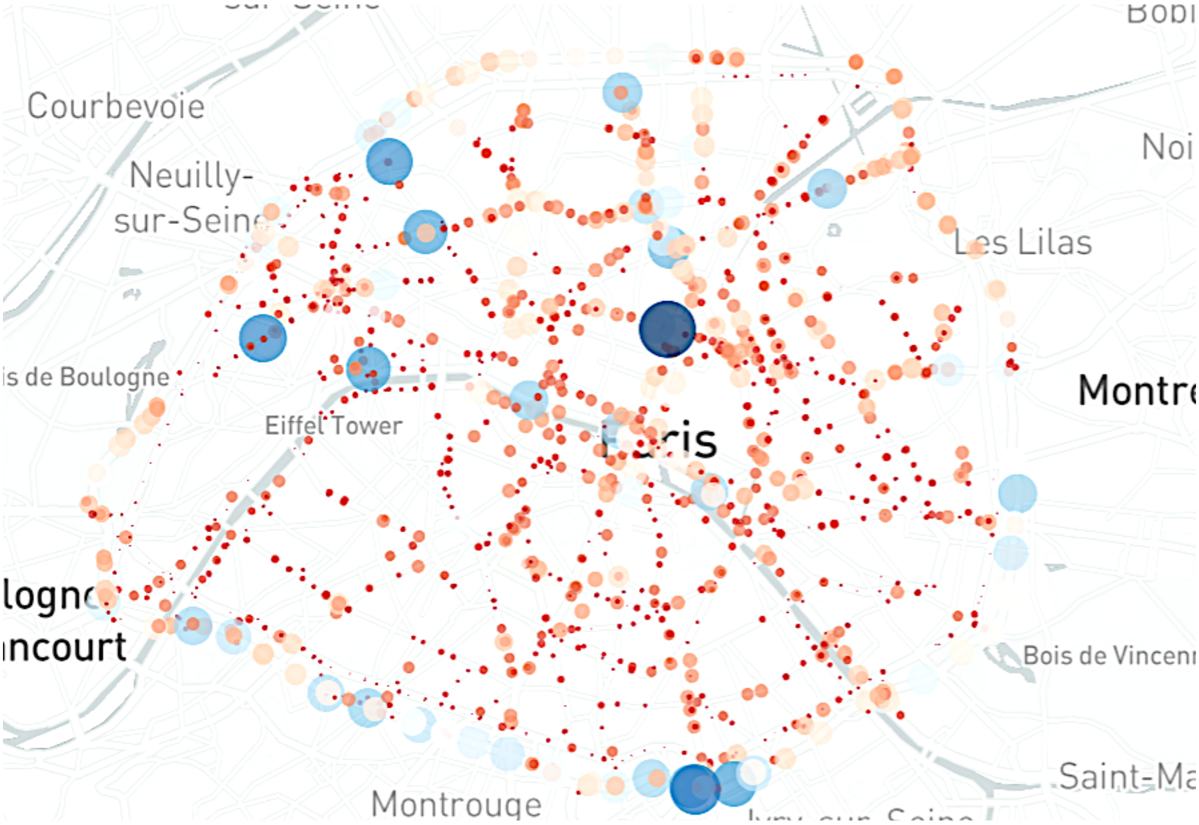


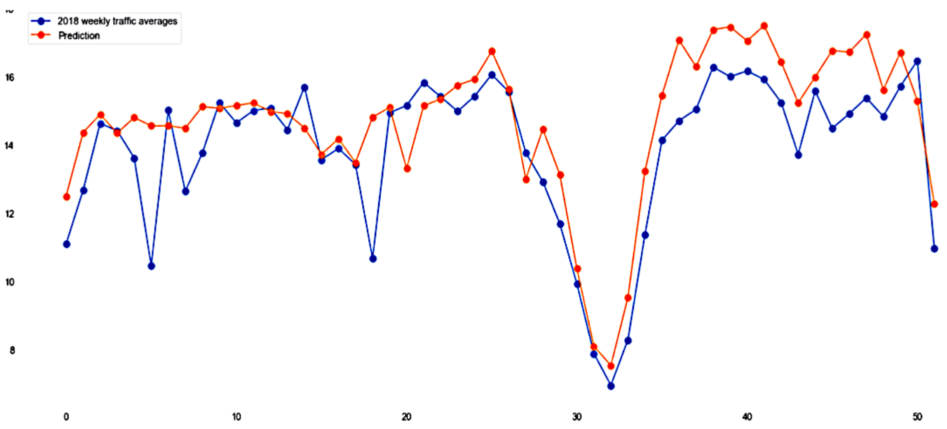
Figure 4. Average traffic rates at specific locations

III. Global Traffic Rate Prediction

The metrics I used to test the performance of the models is MAE, mean average error, which is typical for time series analysis.

a/ Using a simplified AR model to predict weekly average traffic rates

The problem with forecasting average traffic rates on a weekly basis is that we actually do not possess much historical data (3 years), even though we have plenty of records over these three years. Hover, when aggregated, they reduce to a very short time series(circa 150 values). Given the fact that the data shows a strong annual seasonality, this gives only 2 (or 3 without testing) main values to reference to in order to look into the future. About the best we can do, therefore, is to take the average of preceding years to predict future years. The following plot shows us the true vs predicted weekly rates in 2018. Predictively, the model is not able to capture local variations and also unable to adapt. However, it is satisfactory at a descriptive level and could certainly be enhanced with more historical data

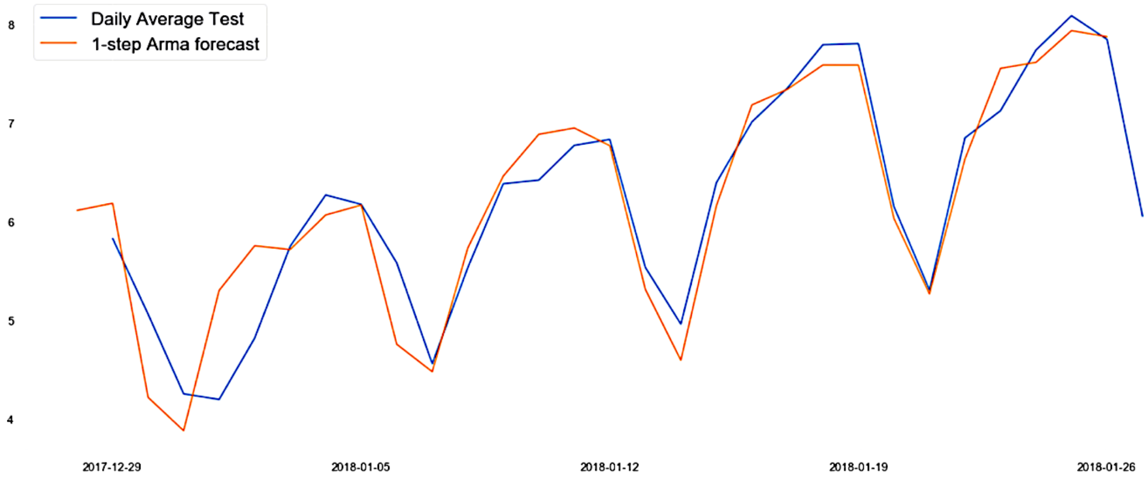


b/ Using a SARIMA model on differenced time series to predict daily average traffic rates

Next we look at a lower level of granularity and predict average traffic rates on a daily basis.

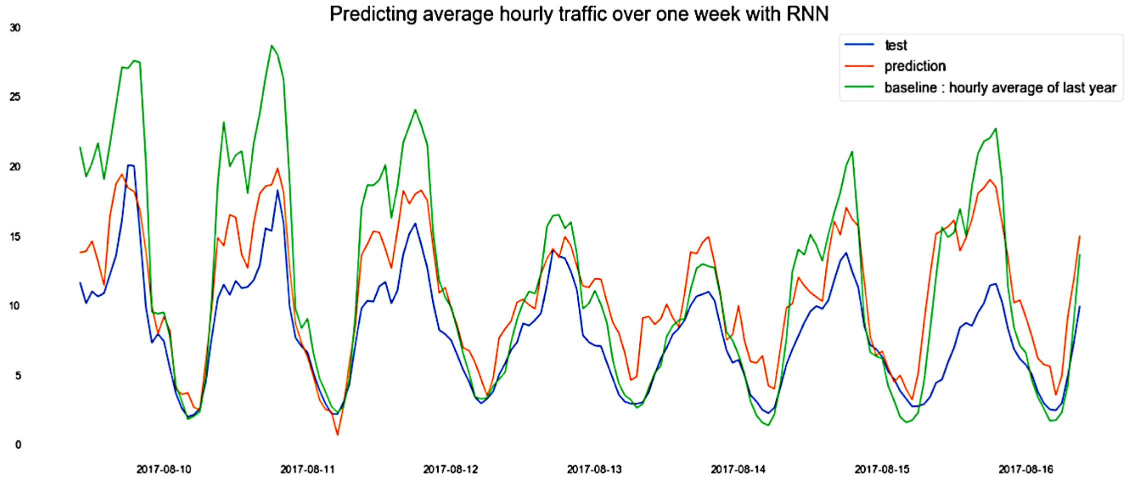
My first attempt was to use the traffic averages of 2017 with an added error term computed using a SARIMA model in order to predict the traffic averages of 2018. This turned out not working well, however, because the differenced time series was too noisy to be captured by a SARIMA model. I therefore decided to ignore the yearly seasonality and to adapt a SARIMA model with a weekly seasonality to the time series. This model did good job predicting one step into the the future, but was not able to capture long trends

1-step forward daily average prediction with SARIMA



c/ Predicting average hourly traffic rates one week into the future with RNN (LSTM)

Next we go one further step down in the level of granularity. Using a LSTM model with one layer of 120 units and feeding in 366 hours at a time, I was able to predict hourly traffic rates 168 (one week) hours into the future with an MAE of 2.97. This is a much better performance than the baseline, the hourly average of last years, which has an MAE of 3.94

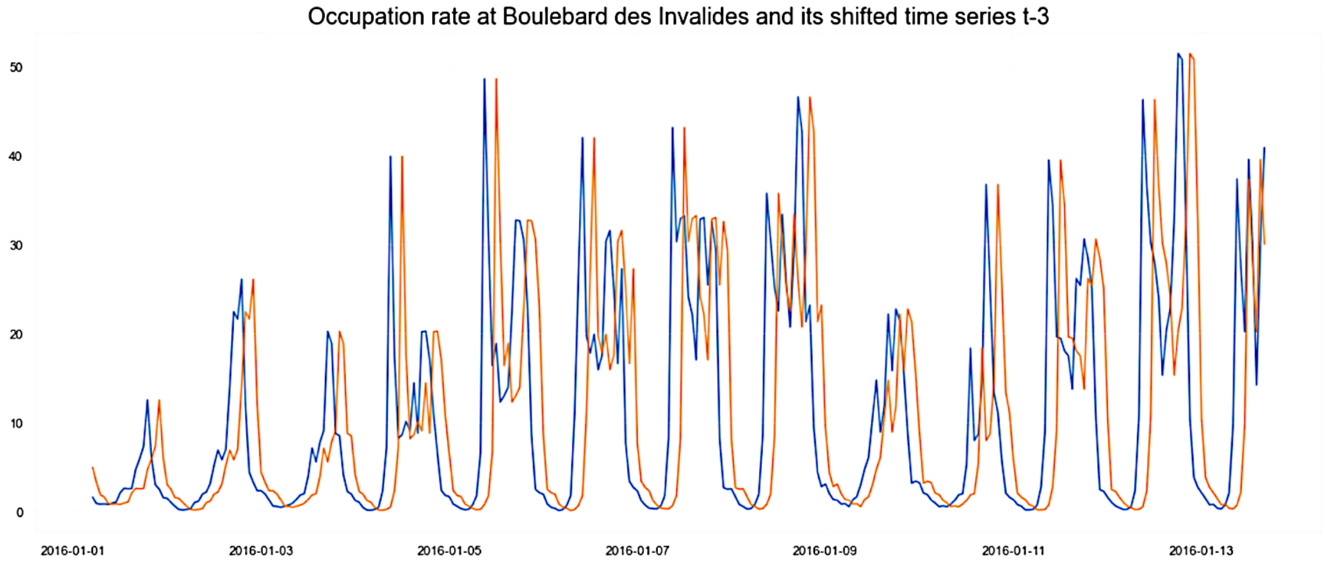


IV. Predicting hourly traffic rates at specific locations

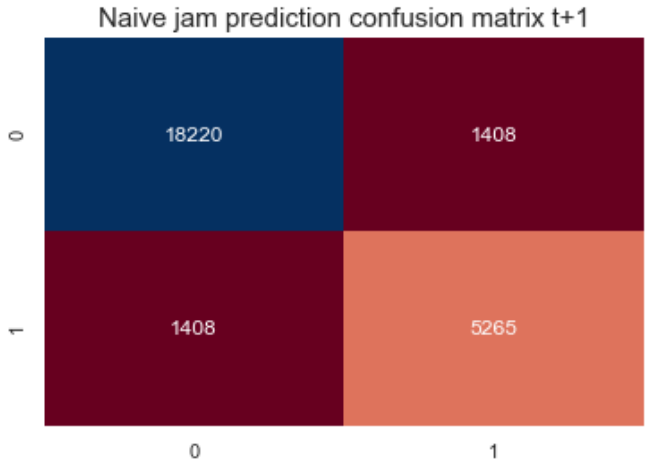
We now get at the core of this investigation : predicting hourly traffic rates at specific locations with the highest possible accuracy. I picked such locations among those with highest traffic rates because , as we have seen before such, locations correspond to the critical crossroads that are responsible for traffic jams. I performed the analysis at several locations and the results were always quite similar. For the sake of clarity, I limited the presentation of the findings to Boulevard des Invalides, one of the most frequented roads in Paris.

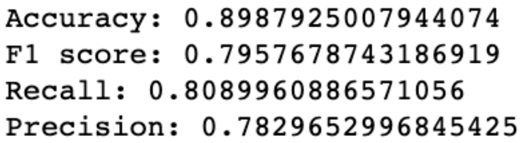
Baseline – traffic rates predictions at Boulevard des Invalides

Here again, LSTMs proved to be by far the most efficient method. In order to assess the predictive power of the LSTM model, I used a consistent naïve baseline as a forecast. That is to say, the predicted hourly traffic rate at time t+delta is equal to the traffic rate at time t



In quantitative terms, this method yields results that are not too aweful at time t+1, with an MAE of 5.28. However this kind of prediction has two main disadvantages. First of all, it becomes increasingly unreliable and times t+2, t+3 etc. For example at time t+3 the MAE becomes greater than ten. Also, it will always fail to capture sudden changes in traffic, which is what is our concern. For this very reason it does a very bad job predicting traffic jams, as shown in the confusion matrix below





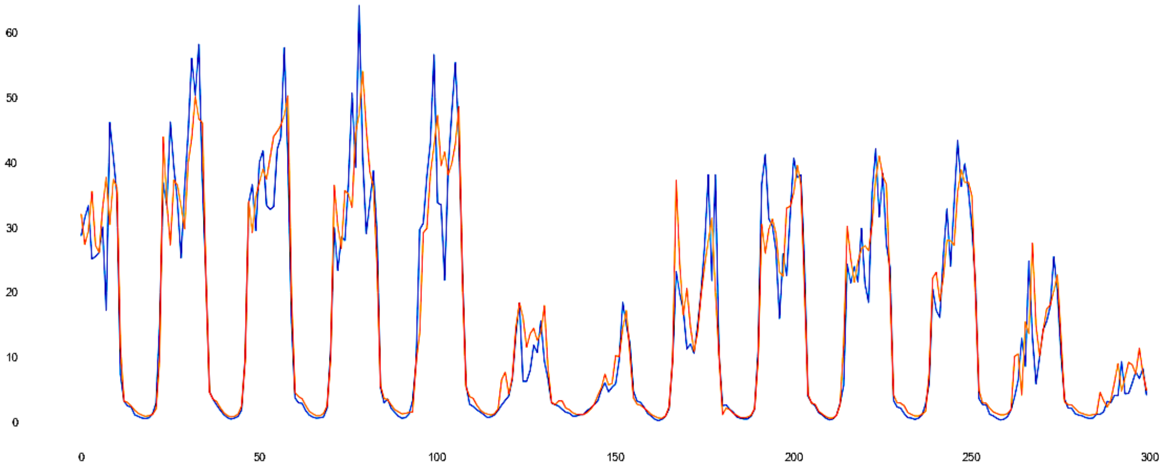
```
Accuracy: 0.8987925007944074
F1 score: 0.7957678743186919
Recall: 0.8089960886571056
Precision: 0.7829652996845425
```

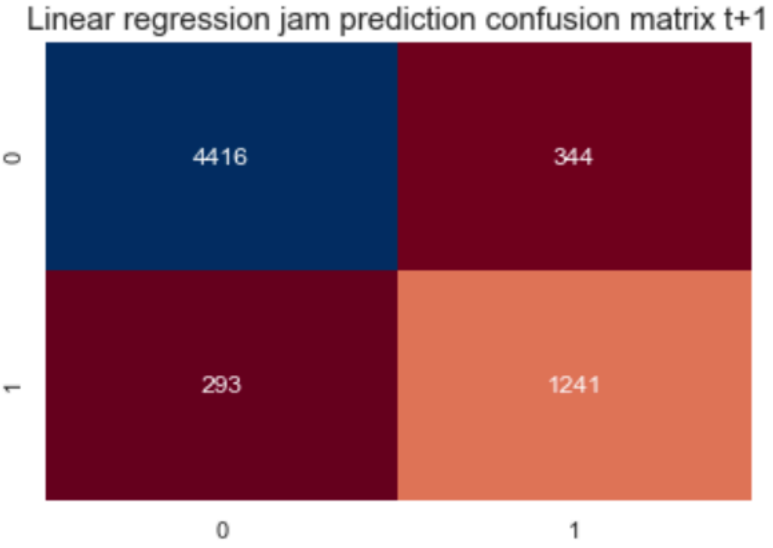
Best model : LSTM model on univariate time series -1 hour into the future, Boulevard des Invalides

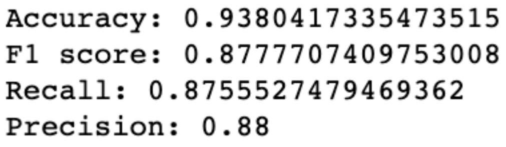
Next I used an LSTM with one layer of 120 units and 72 entries to predict to forecast traffic rates on hour into the future. This predictive model shows a significative improvement over the baseline, with an MAE of 3.15. The accuracy score for jam prediction has also significantly higher

I tried to use more complex models, for example by adding a hidden unit, or increasing the entries, decreasing the number of entries, but more complex models had the tendency to overfit, whilst simpler ones did not achieve as good a performance.

Interestingly, I was hoping that traffic rates at other stations might add valuable information to the model and increase performance. However , the model did not perform better on a multivariate training test



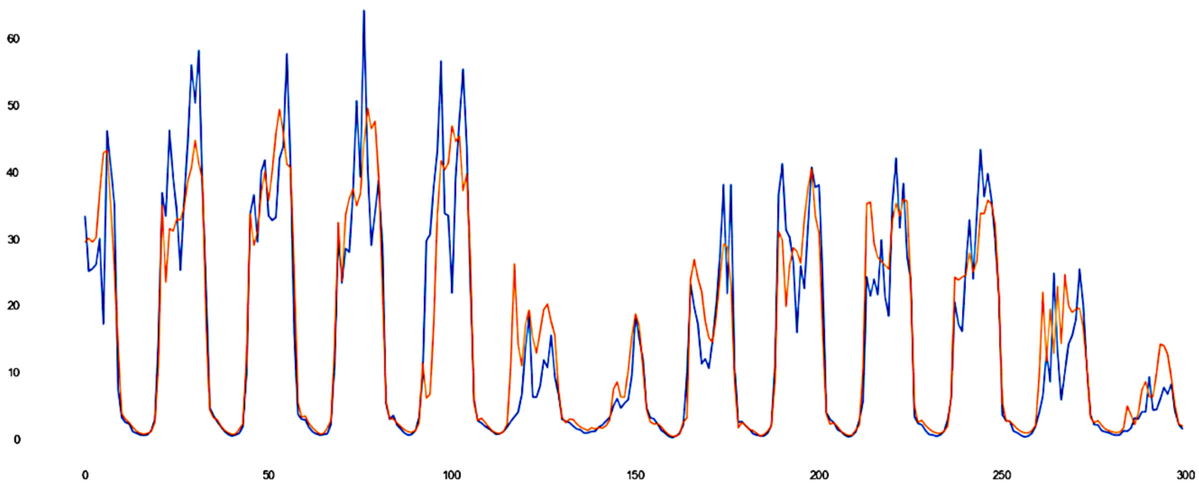


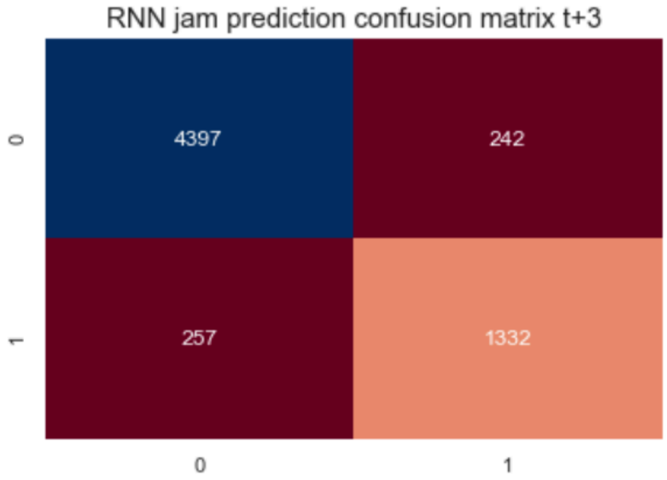


```
Accuracy: 0.9380417335473515
F1 score: 0.8777707409753008
Recall: 0.8755527479469362
Precision: 0.88
```

LSTM model on univariate time series, 3 hours into the future, Boulevard des Invalides

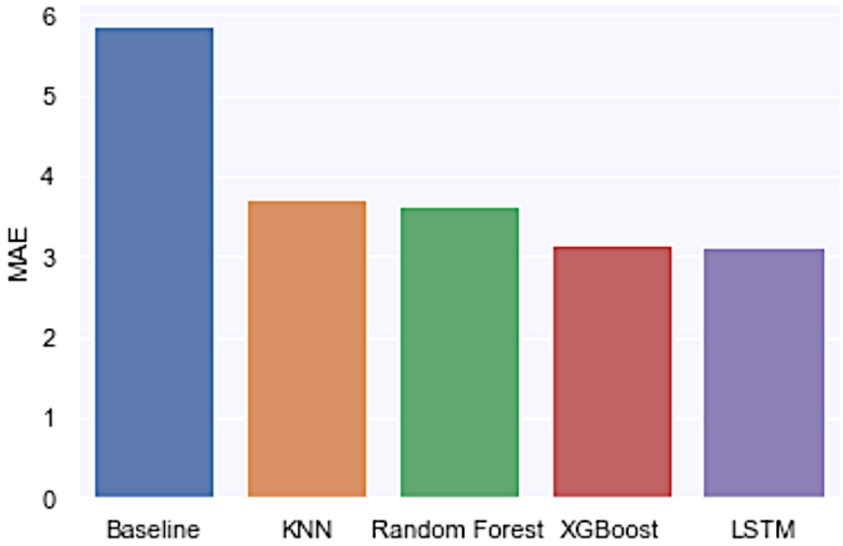
The LSTM model performance at time t+3 is quite impressive as well. The MAE increases to just above 4, which is still better as the baseline at time t+1. The predictive power for jam forecasting, too, is quite satisfactory





```
Accuracy: 0.9198779704560052
F1 score: 0.8422383812835914
Recall: 0.8382630585273757
Precision: 0.8462515883100381
```

Comparing LSTM approach with simpler models



Simpler models, such as KNN, Random Forest and XGBoost also perform significantly better than the Baseline. In particular, XGBoost performs almost as well. Therefore, depending on the desired level of accuracy, one might also prefer to choose a simpler model.

Conclusion

Although the amount of data was quite huge, because It contained multiple parallel time series, I was unfortunately not quite able to extract much valuable information about the relationship between these time series other than global statistical trends.

At best, exploratory data analysis also showed the interesting result that when the threshold of about 12% in average traffic rates is reached, the traffic jams tend to become much more frequent.

In the end, I was therefore reduced to analyze each time series separately. On this subject, however, I was able to achieve good forecasting results.

A simple LSTM neural network proved very effective to forecast traffic rates both on a global and individual scale. As a consequence, I was also able to predict whether or not a jam is likely to occur at any specific location in the next few hours.

If we want to turn our quantitative forecasting into a measure of jam probability, all we need to do is to transform the last dense unit of our LSTM into a softmax unit.

All in all, this investigation was an excellent way for me to deepen my understanding of time series manipulation ,analysis and forecasting using multiple tools and methods, in particular ARIMA, Ensemble models and RNNs, as well as visualization tools such as Plotly and Dash.