10504172 Fossati Nicola

10640083 Espedito Zaccaro

10544315 Montesi Daniele

POLITECNICO
MILANO 1863

4 giugno 2019

Forecasting Traffic Variations in Presence of Events

0. The objective

Our goal was to create a model able to predict the average speeds in case of events, for every road provided in the dataset, at the requested hours, from when an event occurs up to an hour later. We were given datasets about the speeds measured by each sensor every 15 minutes, the events, the weather recorded by the nearby stations and the roads characteristics.

1. Preprocessing

The first step consisted in extracting from the Speeds dataset only the values registered in presence of an event, or within an hour from its beginning. (File 1-preprocess)

In the next group of scripts (2-enrich-data) we did the main preprocessing steps: we added the column DISTANCE_FROM_POINT (Distance between the sensor and the event), the event length in km. From DATETIME_UTC we extracted the day of the week and the time of the day, plus the time delta from the event beginning. The script also retrieves the road characteristics using the sensors dataset. Eventually the script finds the four nearest weather stations for every sensor using the Distances dataset, when present.

Other three scripts (3-parallel-attach-weather, 3bis-rejoin-data and 4-attach-weather) use the Weather dataset in order to retrieve and join the weather conditions for every KEY, KM and DATETIME. If the nearest Station does not have valid data (more than 12 Hours of temporal distance from the last measurement, or missing value), it uses the second nearest one, and so on.

The subsequent script (5-clean-weather) reduces the number of values in the WEATHER column, joining some values with the same meaning or that could have similar consequences on the traffic (e.g. DebolePioggia and Rovescio con Debole Pioggia), plus assigning an empty category to unseen event types.

The script (6-Assign order-to-read) adds a column in order to discriminate measurements made within 15, 30, 45, or 60 minutes from the event beginning.

Given that in a realistic setting the last speed recorded by the sensors before an event happens would be available for the model, we extracted and set them as a feature for the events (PREC_SPEED). In the 7th Document they were fetched from the original dataset, in the 8th we used the speeds already obtained with the previous script for assigning the data in the subsequent time intervals. (7.assign-initial-speed & 8.assign-initial-speed2).

Considering also that speeds and standard deviations vary according to the day -and the hours of the day- in an extremely periodic way, we added two more columns for accounting this periodicity: AVG_ALL_DATASET and STD_DEV_ALL_DATASET. (9.superfluos-data & 9bis attach-superfluos-data)

2. Creating the model

2.1 Visualisation and exploration

At first we decided to simply drop the rows with the missing values, after assuming that they were missing at random. In fact, almost all of them were missing information in the columns related to the weather stations. Compared to the dataset size, they were quite a few.

Then we tried to use the information we had about the hourly average speeds and number of vehicles, in search for a periodicity that could have helped us in creating the model. We found out that even in presence of events, the speeds maintain a certain daily periodicity according to the hour of the day (speeds lowers significantly at rush hours), and a weekly periodicity (Speeds tend to be higher during the weekends).

Then we plotted a histogram for every numerical column, in order to check the distribution of the values, possible outliers, and in order to decide whether or not to apply a scaler. With a posteriori reasoning, we decided that using a scaler was not the case.

We also visualised some scatter plots, searching for any possible variable would have had a correlation with the target one. The most promising variables were used for visualising a correlation matrix and furthermore checked.

2.2 Feature selection

In order to confirm our reasoning about the features, we trained a quick and simple Random Forest and plotted its Feature importances, in order to see if the feature added by us would have helped us in improving the model.

2.3 Fitting the model

After some tries we decided to settle with a CatBoost algorithm. Our model has been tuned with a max depth of 9, 1000 predictors, a learning rate of 0.04 and a regularization lambda of 3.68. These values were the ones that gave us the best score, according to our tests.

We tried two approaches and selected the most satisfying one: in both approaches the first prediction step consists in training the model using the selected features, including the last available speed before the event's beginning (column PREC_SPEED).
In the first approach the model is trained only once, and the subsequent prediction steps are made putting the predicted values in the column PREC_SPEED corresponding to the next quarter of hour. So basically we use the same model four times.

The second approach requires training four different models, adding the results column as a new feature before training the subsequent model.
In our tests we found out that the model with the best performances used the first approach. So we used it for predicting the speeds required by the project task.

3 Results

The evaluation metric requested was the mean average error. The model reaches its maximum performance with an error of 5.75 Km/h if we consider only the first prediction step. Its performance decreases with the following steps, but it is to be considered normal. The total mean average error on the whole validation set settles around 6.11 Km/h.

4 Summary

The main challenge at the beginning was separating the useful data from all the others. In fact we had received a dataset containing almost 11 million rows and ended up with around 1 million after the preprocessing. In order to avoid problems of size with our computers' RAM memories we had to split the initial dataset in chunks and process them separately before re-joining them back in. We also decided to split the workflow in more than one file for avoiding possible confusion during the elaboration; and also for recovering easily the elaborated data in case of errors. The model could be further improved by collecting data covering at least a whole year. In this way it could be possible making a more accurate prediction as we exploit the seasonality of the speeds during the whole year.