RAMP PROJECT: PREDICTING CYCLIST TRAFFIC IN PARIS

Simon Mack
X-HEC
Paris, France

Adrian Tan
X-HEC
Paris, France

December 6, 2021

ABSTRACT

There are numerous bike counter machines located throughout Paris. These counters record how many individual bikes pass by the specific counter on an hourly basis. The objective of our project was whether we could accurately predict daily traffic at each specific counter for the upcoming months and whether we could accurately model any rush hour traffic spikes. The performance of our final model was ultimately decided by an unknown data-set that we could not train or test on.

1 INTRODUCTION

As Paris is moving away from a car-centric model, it has been investing in expanding its bikelanes and making the city more friendly towards bikers. The bike counters, which measure the amount of bikers heading in both street directions, help keep track of bike traffic on specific Parisian roads. Determining the amount of a bikers at any given moment or location can be considered a challenging task. However, there are many variables that can help make a more accurate predication model. Our project therefore aims to develop a prediction model using any model of our choosing (which we will elaborate on).

2 DATA AND ITS PREPROCESSING

The data source for this project is open source data provided by the city of Paris, consisting of countless bike counters. For our project, we will only consider 30 specific locations. The original data set for the project contained 455163 rows

counter_name	site_id	site_name	bike_count	date	counter_installation_date	counter_technical_id	latitude	longitude	log_bike_count
28 boulevard Diderot E-O	100007049	28 boulevard Diderot	0.0	2020-09-01 02:00:00	2013-01-18	Y2H15027244	48.846028	2.375429	0.000000
28 boulevard Diderot E-O	100007049	28 boulevard Diderot	1.0	2020-09-01 03:00:00	2013-01-18	Y2H15027244	48.846028	2.375429	0.693147
28 boulevard Diderot E-O	100007049	28 boulevard Diderot	0.0	2020-09-01 04:00:00	2013-01-18	Y2H15027244	48.846028	2.375429	0.000000
28 boulevard Diderot E-O	100007049	28 boulevard Diderot	4.0	2020-09-01 15:00:00	2013-01-18	Y2H15027244	48.846028	2.375429	1.609438
28 boulevard Diderot E-O	100007049	28 boulevard Diderot	9.0	2020-09-01 18:00:00	2013-01-18	Y2H15027244	48.846028	2.375429	2.302585

Figure 1: First 5 rows of the original data set.

and 11 different columns. It is immediately clear that not all columns would help in constructing our prediction model; meaning that we were eventually left with the following variables to build our model on:

- site_name.
- counter_name
- date

The variables bike_count and log_bike_count were the variables that we had to predict, and therefore could not be considered part of our prediction model.

3 INITIAL DATA ANALYSIS

To get a better understanding of the provided data, we attempted to visualize the data with respect to a couple different time values and bike counters.

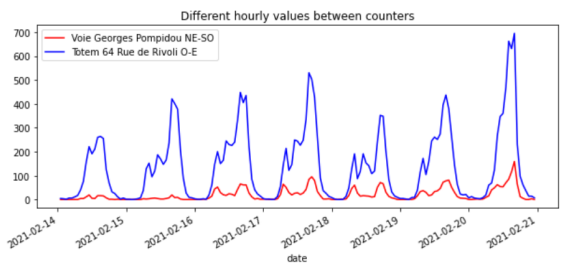

Figure 2: Hourly values

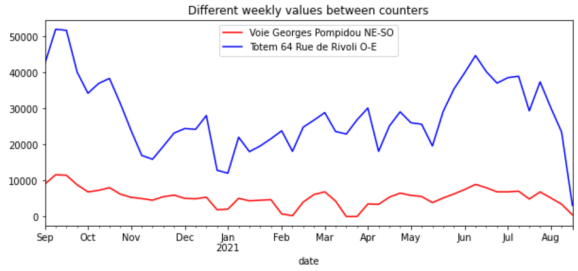

Figure 3: Weekly values

From these two graphs we can immediately see that there is a large discrepancy between different bike counters but also between different timestamps. This confirms our assumption that we must consider both the different bike counters in our model, as well as timestamps. While these initial variables will get us quite far for our prediction model, it is likely that further external data must be utilized to help explain the variation in bike counts across different weeks and hours.

4 MODEL CONSTRUCTION

4.1 Initial Data Modeling

To get started on our model selection, we began by creating a model using solely the data provided in the original data set. The rationale behind taking this approach first was that we believed it helped us understand the limitations of the data in the original data set and would set a baseline on which the external data would have to improve on. The "strength" of a model would be determined by the RMSE value obtained. The lower the RMSE, the "stronger" the model.

4.1.1 Model selection

To begin, we tested out some of the more popular prediction models, as well as models that are claimed to be especially strong for time series forecasting. The models that we tested were: (1) Linear regression, (2) Random forest regression, (3) Histogram-based boosting regression, (4) Extreme gradient Boosting, and (5) Light gradient boosting method.

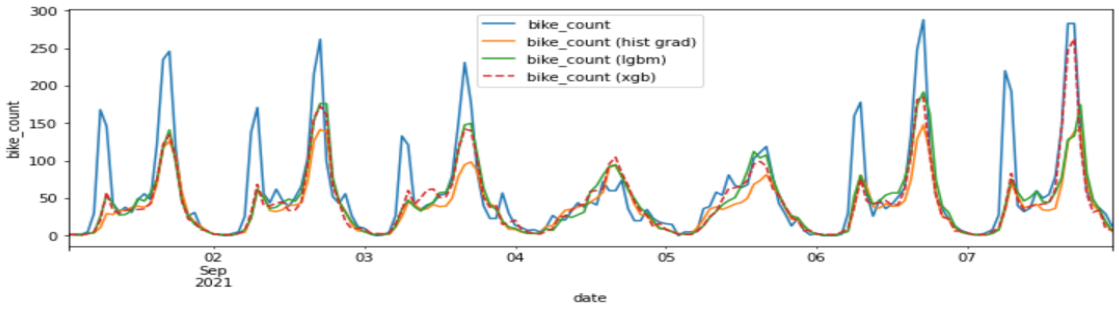

Figure 4: Comparing different predictive models

Since we are trying to map an output that has a continuous (real) value, we only need to consider regression functions. Upon running a couple simple tests with the previously listed models, we decided to continue with the HistGrad, XGBoost, and LGBM models. Our rationale behind dropping the linear and random forest regressors was that they had the lowest RMSE and did not work properly with the NAn values that the data set contained.

4.1.2 Cyclical dates

Once we selected the optimal model to predict our data, we attempted to decrease the RMSE of the model by redefining the dates of the data set. For the initial model selection, the date column was originally broken up into the following columns:

- year => (2020, 2021)
- month => (1,...,12)
- weekday => (0,...,6)
- day => (1,...,31)
- hour => (0,...,23)

While ordinally encoding the time in such a manner helps differentiate between different days of the week and hours of the day, the regressor function isn't aware of the fact that all of the variables are cyclical (e.g. weekday 6 is followed by weekday 0, hour 23 is followed by hour 0). The model would consider hour 0 and hour 23 as 23 hours apart, whereas they are in fact only 1. To combat this issue, we decided to cyclical encode all the time variables. By converting all our time variables as cosine and sine functions, we provide the machine with the hint that the variables are cyclical. We use both cosine and sine functions because there are duplicate values on a 2π sinusoidal wave and adding the sine functions prevents this from becoming an issue with the model.

	weekday_cos	weekday_sin	weekday	norm
48339	0.5	0.866025	1	1.047198
48342	-0.5	0.866025	2	2.094395

Figure 5: Avoiding duplicate value confusion

We did not end up converting all time variables to a cyclical equivalent, however. After conducting several tests to find the most stable model and strongest RMSE, we determined that the day and year values would remain integers while the weekday, month, and hour variables would become cyclically encoded.

The fact cyclically encoding the day variable had an adverse effect on our model was surprising, but it can likely be explained by the fact that there are little patterns to be observed by the day variables (e.g. days can fall on any weekday and are influenced by the month).

Following this change, we recorded an improvement on our RMSE from **0.805 -> 0.757**, $6\% \uparrow$.

4.2 Leveraging external data

After cleaning and improving the original data-set, we moved on to finding external data that could help explain certain spikes and troughs that we observed in the data visuals. The original data set came accompanied with an external CSV file containing various weather variables.

Having analyzed our initial data visualization, it was immediately clear that there were external factors contributing to the spikes and decreases in bike count during certain hours, days, weeks, and months. To help explain these anomalies, we decided to try out the following external variables for our model:

- Rush hour: $Binary : x_i = \{0, 1\}$
- Curfew: $Binary : x_i = \{0, 1\}$
- Holidays: $Binary : x_i = \{0, 1\}$
- Bike rentals: $Discrete : x_i = [0, \infty]$
- Brent crude price: $Continuous : x_i = [0, \infty]$
- Car traffic: $Discrete : x_i = [0, \infty]$
- Temperature: $Continuous : x_i = [0, \infty]$
- Wind speed: $Continuous : x_i = [0, \infty]$
- Humidity: $Continuous : x_i = [0, \infty]$

To determine whether they had any influence on our model, we graphed the feature importance of the 3 different regressors to determine which variables to keep.

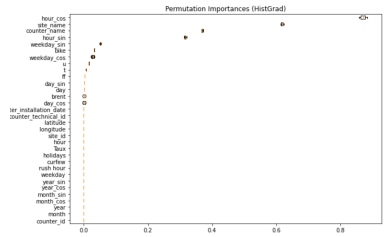

Figure 6: Hist-Gradient

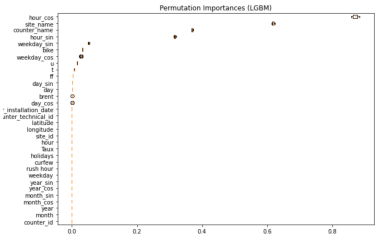

Figure 7: Light GBM

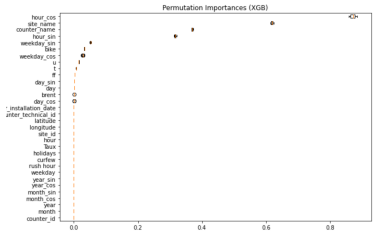

Figure 8: XGBoost

All three different models allot the same importance to all the different variables in the external data and in the original data set. Based on this information, we ultimately decided to select the variables highlighted in yellow (from the previous list) as variables that we included in our regressor preprocessor.

4.3 Hyper parameter tuning

After determining the external variables we were going to use and confirming that they had a positive effect on the RMSE, we started adjusting the hyper parameters of our model. To find the best parameters, we used to TimeSeriesCV package to help in selecting the optimal criterion. It was decided to use TimeSeriesCV instead of GridSearchCV because the former did not use future data (i.e. data points whose datetime would be after the test set). This seemed a more appropriate approach to finding the right parameters and prevented the CV from using future data to predict past data, as a GridSearchCV might have done. We tested the following parameters for our regressors:

Histogram-based boosting regression:

- random_state:
- max_leaf_nodes
- max_iter

Extreme gradient Boosting:

- random_state
- num_leaves
- n_estimators
- importance_type

Light gradient boosting method:

- max_depth
- max_leaves
- random_state

For any of the random_state in our regressors, we kept the value = 0 to ensure that our results would be replicable across different attempts.

5 Results

6 Conclusion

7 Acknowledgement

References

[1] One

[2] Two