Highway Tollgates Traffic Flow Prediction

CUSP-GX 5006.001: Machine Learning for Cities
Final Project Report

Muci Yu, my1826@nyu.edu
Songjian Li, sl4729@nyu.edu
Pengzi Li, pl1840@nyu.edu

1.Introduction

1.1 Motivation

Traffic congestion is a condition on traffic networks which occurred when traffic demand approaches the capacity of a road (or of the intersection along the road). There are many downsides of traffic congestion: stressed and frustrated motorists, encouraging road rage and reduced health of motorists; wasted fuel increases air pollution and carbon dioxide emissions which may contribute global warming due to increased idling, acceleration and breaking; interfere with the passage of ambulance; highly chance of vehicle crashes due to tight spacing and constant stopping-and-going; etc. There are many types of places on the road that could cause traffic congestion, and highway tollgates intersection is one of them ("FACT CHECK: Does This Image Show a Traffic Jam on China's 50-Lane Highway?", 2019).

Highway tollgates are known as bottlenecks in traffic networks. Long queues at toll gates during peak hours can cause severe congestion, which needs effective countermeasures to solve this challenge. Preemptive countermeasure includes accelerating the toll collection process by allocating temporary toll collectors to open more lanes from traffic regulators aspect, and route recommendation for motorists from map application company aspect, etc. However, these countermeasures are valid only if they acknowledge precise predictions for future traffic flow. For example, if heavy traffic flow in the next hour can be precisely predicted, traffic regulators could then immediately deploy additional toll collectors by expanding more lanes so that the traffic flow could divert at upstream intersections. Or the map application can recommend another route that takes less time for motorists to travel to the same destination, which can also eliminate traffic congestion at a certain level. Besides, many factors might affect the pattern of

traffic flow: weather conditions, holidays, peak hours, etc. This various possibility makes future traffic flow and ETA (Estimated Time of Arrival) a known challenge.

1.2 Goal

In this project, we want to design reliable approaches for future traffic flow and ETA prediction by deploying linear regression model and random forest regression model, so that the traffic regulators and map application companies can make their decision based on our prediction model for fewer congestions at tollgates and route recommendations.

2. Related Works

2.1 Similar researches

The framework of our project is based on Knowledge Discovery and Data Mining (KDD) Cup 2017 competition. There are two tasks in this competition: travel time prediction and traffic volume prediction. The tasks are to predict travel time and volume for a given road and tollgate during rush hours, knowing the previous two hours data and several days before. The goal is to find suitable methods for the two predictions and to achieve good prediction performances. Most of the candidate teams are deploying eXtreme Gradient Boosting, some of them also using ARIMA (time series analysis) and SVR (Super Vector machine Regression).

2.2 Differences between our project and similar researches

The differences between our project and the competition are that we not only predict the travel time during rush hours, but we also want to predict travel time of a random vehicle given its starting location, destination, weather condition and starting time. And based on the scope of this course, we want to find suitable parameters using GridSearch to achieve good prediction performances for the models we choose.

3. Data Analysis

The analysis in this paper is based on data provided by Tianchi Datasets. The raw data contains four different datasets: road link properties; vehicle routes from intersections to tollgates; vehicle trajectories along routes; traffic volume through the tollgates and weather data (every three hours) in the target area. Among these four datasets, road link properties, vehicle routes from intersections to toll gates are used for road network visualization; vehicle trajectories, traffic volume and weather data are used for the prediction model.

3.1 Exploratory Data Analysis

The road network (See Figure.1) considered in this project includes three intersections (A, B, C) and three tollgates (1, 2, 3). Vehicles enter Intersection A can exit at tollgates 2 and 3, while vehicles enter Intersections B and C can exit at tollgates 1 and 3. Tollgate 2 only allows traffic entering the highway, while tollgates 1 and 3 allow traffic entry and exit. There are six routes in this dataset.

Vehicle Trajectories Along Routes

Vehicle trajectories data (Table 1) contains time-stamped records of actual vehicles driving from different intersections to different tollgates. The date range of this dataset is between 19th July to 17th October 2016. This dataset includes intersection id, tollgate id, vehicle id, the starting time of each vehicle enters the route, travel sequence (trajectory in the form of a sequence of link traces) and the total time (in seconds) that the vehicle takes to travel from the intersection to the tollgate.

Traffic Volume through the Tollgates

The dataset that contains traffic volume through the tollgates (Table 2) includes the following features: datetime when a vehicle passes the tollgate, tollgate id, direction (entry or

exit), vehicle model which indicates the capacity of the vehicle, dummy variable that indicates whether this vehicle uses electronic toll collection (ETC) or not and the vehicle type (passenger vehicle or cargo vehicle).

Weather Data (every three hours) in the Target Area

The weather data collected every three hours in the target area (Table 3) consists of weather-related measurements, includes: date and hour, air pressure (in hundred Pa), sea level pressure (in hundred Pa), wind direction (in degrees), wind speed (in m/s), temperature (in Celsius degrees), relative humidity and precipitation (in mm).

3.2 Data Cleaning and Feature Engineering

Feature engineering is a process of creating new input features from the existing ones. Feature engineering can help us to improve the model's performance by bringing in prior knowledge to isolate and highlight key information so that the models can "focus" on what is important.

After a rough data exploratory analysis, we decide that the influence factors of travel time include: 1. Weather conditions; 2. Road conditions; 3. Traffic conditions; and 4. Time-related variables.

1. Road conditions

We assumed that the road conditions such as road width and length would influence the travel time.

We have 24 links in our dataset, and they formed six different routes. We created six route dummies to record the trajectories of each car. For example, if a car passes through B_1, the values for this route dummies will be one while values of other route dummies will be 0. We

then calculated the average width and length for each route based on the width and length of the links it contains to capture the road conditions.

2. Traffic conditions

It is reasonable to assume that traffic conditions have a significant impact on travel times. To learn the impact of traffic conditions, we created a "same_route_cars" feature. The defined "same_route-cars" as the number of cars departure from the same origin at the same time. We resampled our dataframe by frequency of 5 minutes and calculated the number of "same_route_cars" based on the time range of 5 minutes.

3. Weather conditions

To capture weather conditions in our model, we first deploy a KMeans clustering model over the weather based on normalized features, including pressure, sea pressure, wind direction, wind speed, temperature, humidity, and precipitation. We adopt the Silhouette score to evaluate clustering performance for different K, and the optimized result is dividing weather into 3 clusters. We then merged the weather label back to our vehicle dataframe based on timestamp.

4. Time-related variables

We first assume that the travel time depends very much on whether it is peak hours or off-peak hours. After visualized the average travel time over hours of the day, we defined peak-hour as 8 a.m. to 10 a.m, 4 p.m. to 6 p.m., and the rest of hours are marked as off-peak, as shown in figure 3.

Moreover, during the exploratory data analysis, we find out that the travel time varies a lot in different weekdays. As shown in figure 4, the average travel time is high on Fridays and Saturdays and is low on Mondays. Also, we figure that the volatility of travel time is different during three periods of the month: the average travel time is higher at the early and end of month

comparing to the mid-month as shown in figure 5. We also inspected the impact of holidays, since our data range covered the National Day holiday from Oct 1st to Oct 7th. We figured that the holidays do not have a significant impact, as shown in the highlight portion in figure 5.

Therefore, the features we input to our models include average width, average length, which are numerical features, and dummies for routes, weather, weekdays, peak or off-peak hours, and part-of-month to capture these variables' impacts on travel time, which is our target variable.

For the target variable, the distribution of travel time has a severe left skew pattern, so we simply drop the data which lower than the first quartile and higher than the third quartile from the training set. After removing outliers, the distribution of travel time becomes less skew than before (See Figure 6).

3.2 Train/Test Split

Since we are dealing with temporal dependent data, particular care should be taken in the train/test split process to prevent data leakage. To simulate the real-world forecasting environment, we should withhold the data that occur chronologically after the data we used to fit our model. Instead of using traditional sci-kit-learn train_test_split package, we sliced the data into 33% - 66% proportion according to the starting time of each vehicle to make sure that the test data comes chronologically after the training data, and the validation set comes chronologically after the training data as well.

4. Methodology

4.1 Input Features

As a recall from section 3.3, we created four types of features --- Weather condition, Road condition, Traffic condition and Time. We first tried road and traffic condition features as

our baseline input features. Then to improve the performance of our model, we added weather conditions and time as input for our second round of training. To sum up, we ran the following two sets of features on each of our models.

Set	Features
1	Road Condition, Traffic Condition
2	Road Condition, Traffic Condition, Weather Condition, Time-related

4.2 Baseline - Central Tendency

There are in total six different routes in our training data. We calculated the mean and median travel time for each route and considered them as our baselines. The means and medians for each route are shown in figure 7. We will compare the results of our models to these baselines. If our model outperforms them, it would suggest our models can produce meaningful results.

4.3 Linear Regression

We first ran our feature sets on ordinary least squared (OLS) model. Linear regression is easy to implement and highly interpretable, so we use it as a quick start to validate our feature engineering process.

4.4 Random Forest Regression

A Random Forest (RF) is an ensemble technique capable of performing regression tasks with the use of multiple decision trees and bagging. This is a much complex model than the OLS model and can capture the non-linear variation in data. If properly tuned, it can have strong predictive power. To mitigate the overfitting problem that is common in tree-based models, we tune the hyperparameters, number of estimators, maximum features and max depth, through cross-validation. After grid search on these hyperparameters, we find that tuning the maximum

features is the most effective way to decrease model errors. The relationship between model performance and max_features is shown in figure 8. The final hyperparameters we select for testing is max_depth = 4, max_features = 8 and n_estimators = 7.

4.5 eXtreme Gradient Boosting

eXtreme Gradient Boosting (XGBoost) is another ensemble method based on decision trees. In general, trees-based methods enumerates over all data instances to find the best splitting point. It is a sparsity aware algorithm, so we think it should have a robust performance over our data. We tune the alpha, learning rate and the number of estimators through GridSearch for better performance. The final hyperparameters we select for XGBoost regression model is objective colsample_bytree = 0.1, learning_rate = 0.4, learning_rate = 0.4, max_depth = 3, alpha = 6, and n_estimators = 18. The predictions are shown as Figure 9.

4.6 Evaluation Metric

To evaluate the performance of our models, we use root mean square error (RMSE), which is defined as,

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}((y_j - \hat{y}_j)^2}$$

where y is the actual value and $\hat{y}$ is the predicted value. We use RMSE and MAE to evaluate the performance of our models. RMSE has the benefit of penalizing large errors. In the real world, users tend to be more negatively impacted by larger errors. For instance, underestimating a traffic jam may lead users to choose a badly congested route, which will not only aggravate the traffic but also waste our users' time. We also include MAE because it can provide us with a direct comparison of the errors to the actual values.

5. Results

The performance of our models is list in table 4. Unfortunately, the baseline prediction using the historical median of each route beats any other model used in this study. The other baseline using the historical mean of each route has a similar performance as the other models. Also, it is important to note that our complex models such as random forest and XGBoost do not perform significantly better than the simple OLS. This suggests that increasing the model complexity does not transfer to capturing more variance in the data.

We attribute the underperformance of our models to the feature engineering process. After putting more features in the feature set 2, the RMSE and MAE of OLS only improve slightly, indicating that the variables we create for the feature set 2 have no predictive power. After investigating what is wrong with our features, we have the following three conclusions.

a. We have too many dummy variables at hourly, daily and even monthly levels while our data are in the second granularity. For instance, we have weather and day of week dummies. The values of these columns are likely to be the same for every instance from the same day, yet the travel time for those instances could vary a lot. Number of cars on the same route might be the only feature we have that may vary from row to row. As a result, the predictions of our model tend to be very uniform. This is proven by the much narrower standard deviation of our random forest prediction as compared to the actual value (see Table 5).
b. We don't have enough features that can serve as proxies for the severity of congestion, which might be the most important predictor for travel time. Number of cars on the same route is the only proxy we have, and it is not accurate. To recall, we define the number of same route cars as the number of cars that start at the same location and have the same

destination within 5-minute time frame. There are several loopholes in the way we design this feature. First of all, we don't take into account how many cars have exited the route from the previous time frame. For instance, say, there is severe congestion on route 1, so at t = 3, most of the vehicles that entered route one at t = 1 are still en route to their destination. However, if there is only one vehicle enters the route at t = 3, our same route cars feature for this car will be 0 and the model will predict a short travel time for this vehicle, even though the route is congested.

c. We fail to take route structure into account when engineering our features. The way we construct out route dummies and same route cars variables assume that routes are independent to each other. However, as shown in Figure 1 and 2, some routes are interconnected. If two routes merge into one, the travel time might slow down after merging. On the opposite, travel time might increase if two routes split. We need to incorporate more features to represent the interconnections between routes.

Conclusion

Accurately measuring travel time can help predict congestions in the road network. Such information could be valuable to traffic operators and policymakers. In this study, our models fail to beat the baseline. While this is discouraging, we do investigate the problem and find ways to improve our models in the future. We believe that we should generate more features that have stronger predictive power. Specifically, more features that can accurately capture traffic condition and route network should be incorporated into the models.

Group Task Distribution

All group members are involved in the whole process of data cleaning, feature engineering, model building, and project writing. However, each group member mainly focused

on one part of the project. Muci Yu primarily focus on model selection, model training, and result interpretation. Songjian Li contributed in feature engineering and xgboost model training while Pengzi Li was responsible for the linear regression model, project write-up.

Figures and Tables

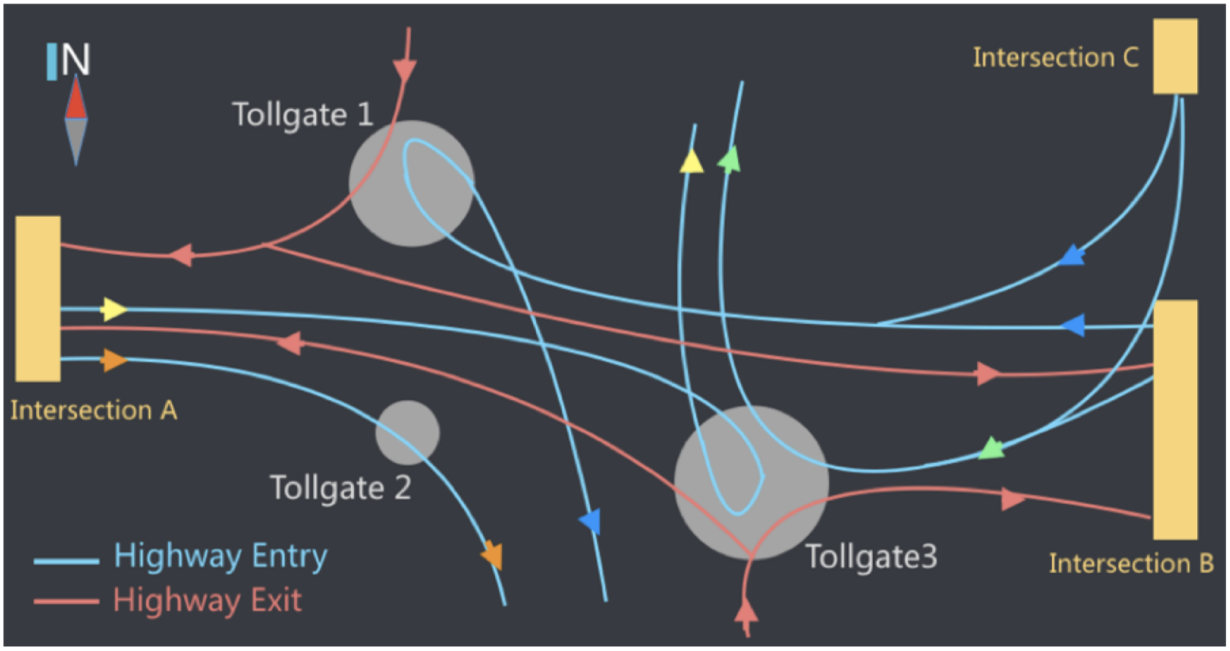

Figure 1. A visualization of the road network (Lin et al., 2018)

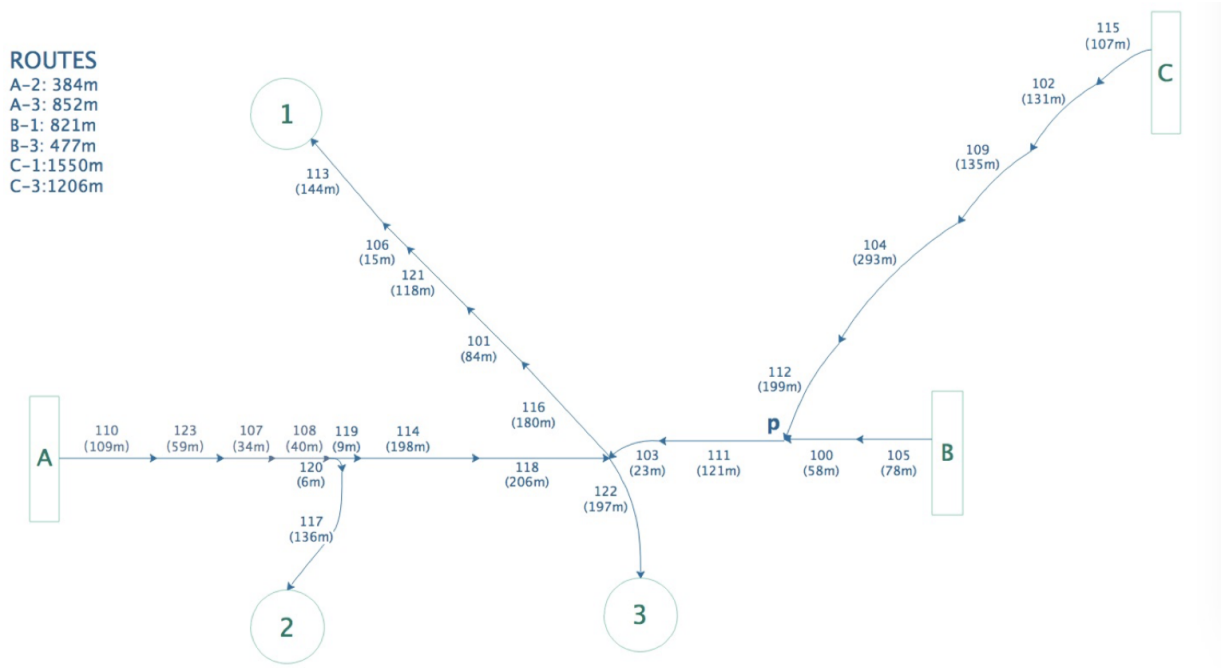

Figure 2. The link-representation of the road network. Each route is composed by a sequence of links; each link is represented by an arrow. The value without parentheses over a link represents the unique id of the link, and the value in parentheses represents the length of the link. The total length of each route is presented at the upper left corner (Lin et al., 2018)

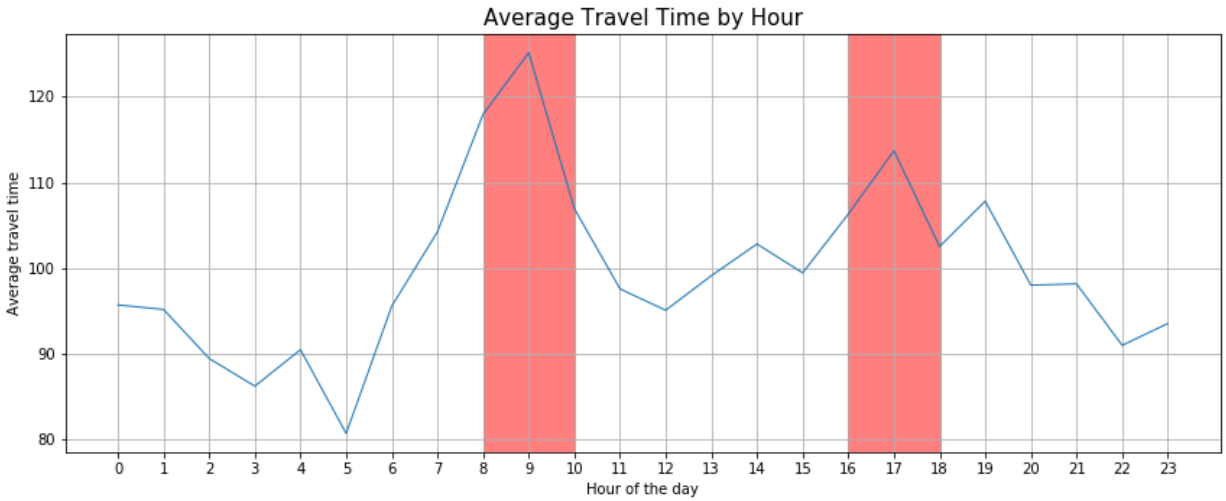

Figure 3. Average travel time in the hour of the day

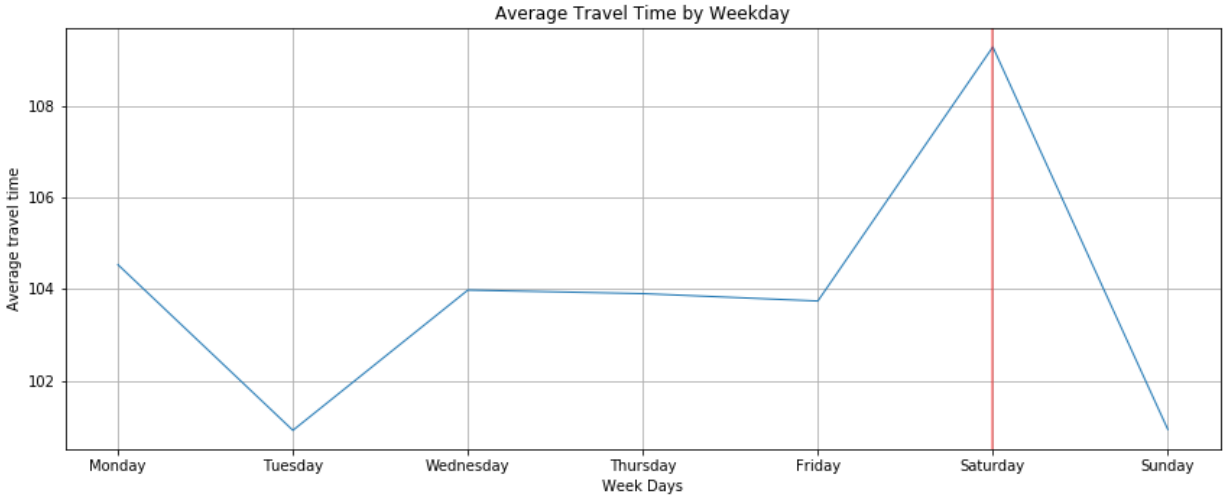

Figure 4. Average travel time in week days

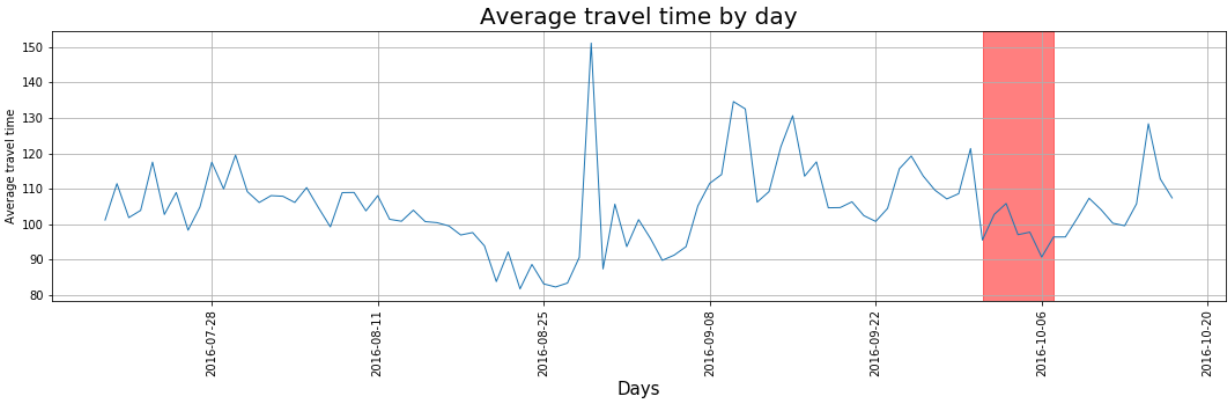

Figure 5. Average travel time by day

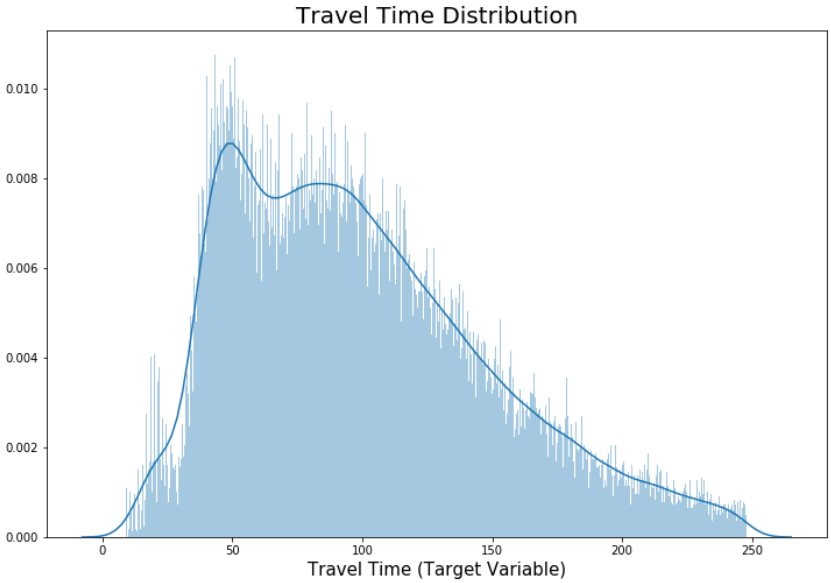

Figure 6. Travel Time Distribution

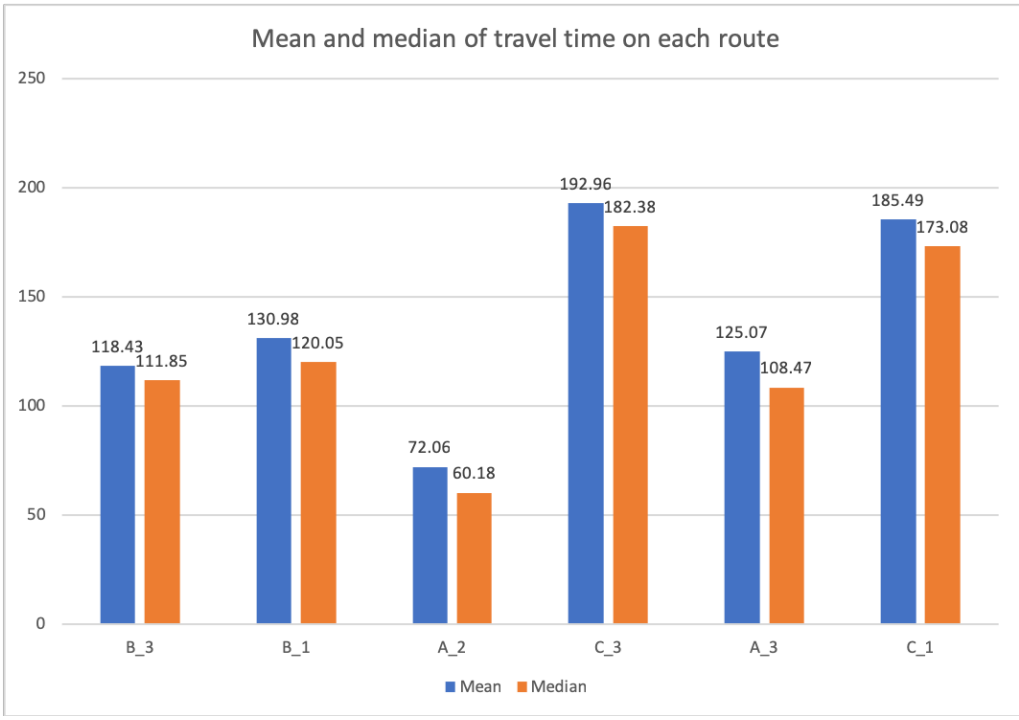

Figure 7. Mean and Median of Travel Time on each Route

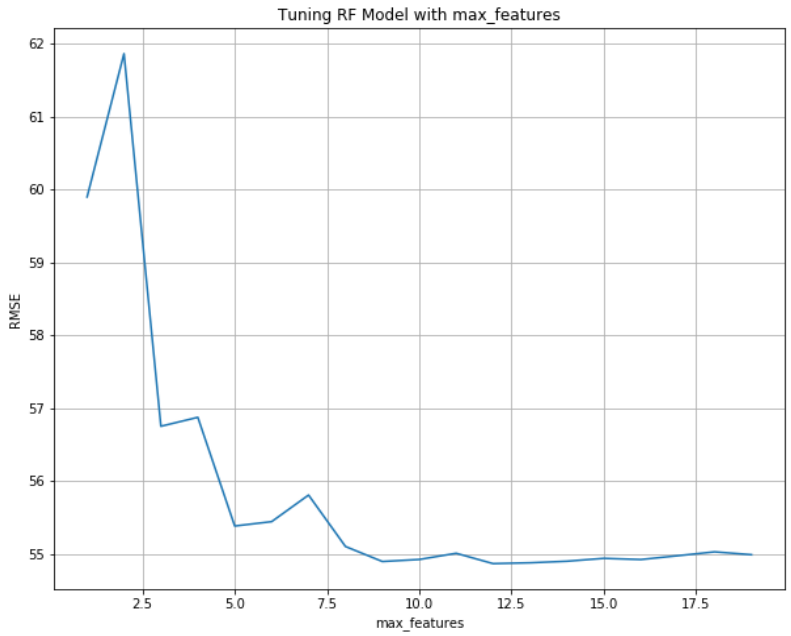

Figure 8. Tuning RF model with max_features

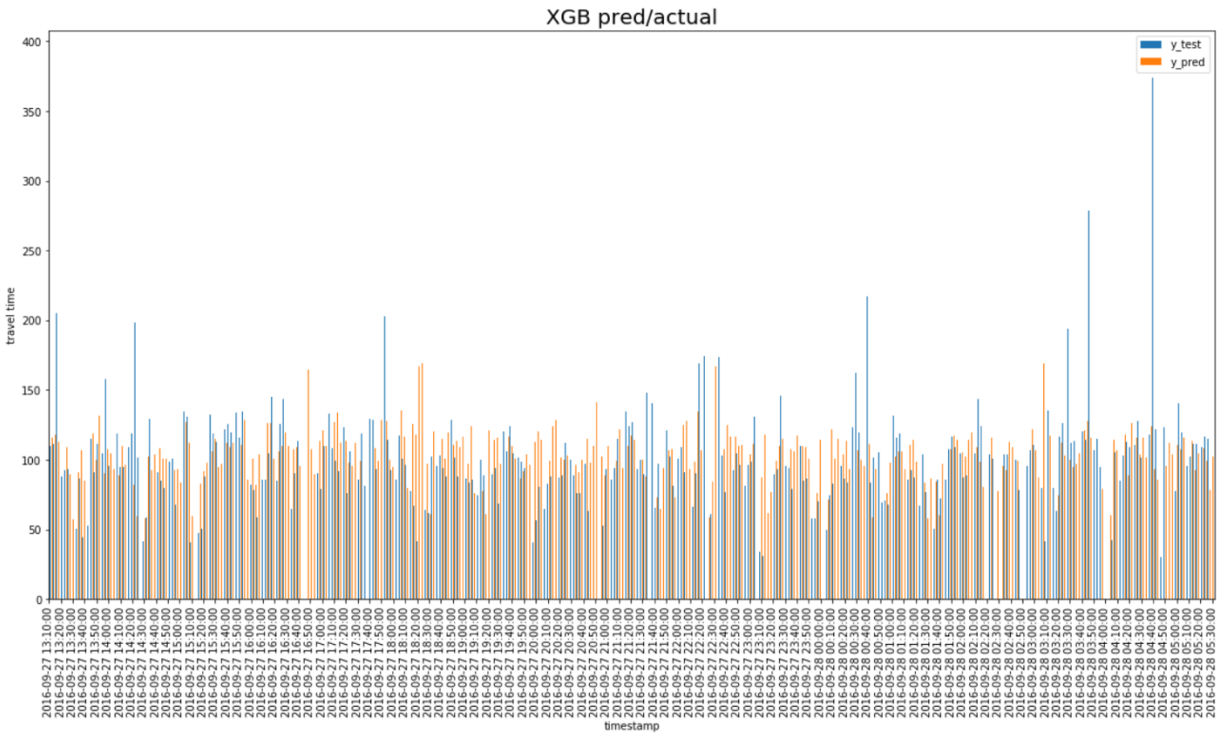

Figure 9. Xgboost prediction visualization

Field	Type	Description
intersection_id	string	Intersection ID
tollgate_id	string	Tollgate ID
vehicle_id	string	Vehicle ID
starting_time	datetime	Time point when the vehicle enters the route
travel_seq	string	Trajectory in the form of a sequence of link traces separated by “;”, each trace consists of link id, enter time, and travel time in seconds, separated by “#”
travel_time	float	The total time (in seconds) that the vehicle takes to travel from the intersection to the tollgate

Table 1. Vehicle Trajectories Along Routes

Field	Type	Description
time	datetime	The time when a vehicle passes the tollgate
tollgate_id	string	ID of the tollgate
direction	string	0: entry, 1:exit
vehicle_model	int	This number ranges from 0 to 7, which indicates the capacity of the vehicle (bigger the higher)
has_etc	string	Does the vehicle use ETC (Electronic Toll Collection) devices? 0:No, 1:Yes
vehicle_type	string	Vehicle type: 0 - passenger vehicle, 1- cargo vehicle

Table 2. Traffic Volume through the Tollgate

Field	Type	Description
date	date	date
hour	int	hour
pressure	float	Air pressure (hPa: Hundred Pa)
sea_pressure	float	Sea level pressure (hPa: Hundred Pa)
wind_direction	float	Wind direction (∘)
wind_speed	float	Wind speed (m/s)
temperature	float	Temperature (∘C)
rel_humidity	float	Relative humidity
precipitation	float	Precipitation (mm)

Table 3. Weather Data (every 3 hours) in the Target Area

	Baseline - Mean	Baseline - Median	OLS - Feature1	OLS- Feature2	Random Forest	XGBoost
MAE	32.15	30.74	33.89	33.61	32.37	32.9
RMSE	55.07	55.5	55.5	55.3	54.99	55.63

Table 4. Model Performance Comparison

	Prediction	Actual Value
Mean	107.59	104.02
Std	33.93	67.11

Table 5. Comparison of Mean and Standard Deviation for Predicted and Actual Value

Reference

"FACT CHECK: Does This Image Show a Traffic Jam on China's 50-Lane Highway?" Snopes.com. Retrieved May 09, 2019. https://www.snopes.com/fact-check/china-50-lane-highway-traffic-jam/.

Lin, Amanda Yan, Mengcheng Zhang, and Selpi. "Using Scaling Methods to Improve Support Vector Regression's Performance for Travel Time and Traffic Volume Predictions." *Time Series Analysis and Forecasting Contributions to Statistics*, 2018, 115-27. doi:10.1007/978-3-319-96944-2_8.

GitHub Link

https://github.com/muciyuallen/TrafficFlowOptimization