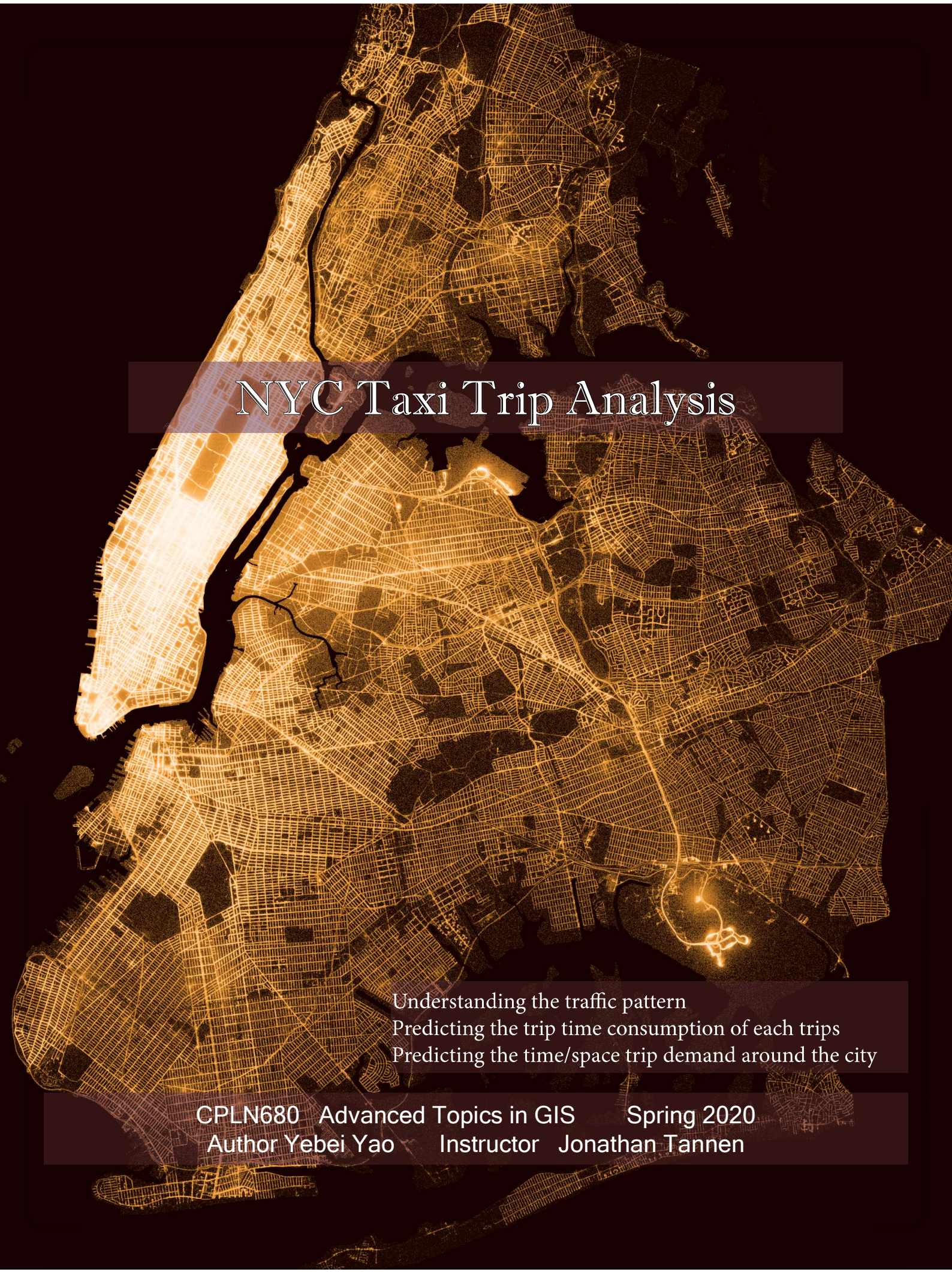


NYC Taxi Trip Analysis

Understanding the traffic pattern
Predicting the trip time consumption of each trips
Predicting the time/space trip demand around the city

CPLN680 Advanced Topics in GIS Spring 2020

Author Yebei Yao Instructor Jonathan Tannen

TABLE OF CONTENTS

Introduction

You are not the only one going your way.

Taking taxi is a both convenient and indispensable mode of transit in megacities around world. Despite the effect of public transportation during daytime, taxi works more flexible and possess the feature of door-to-door and in time service to the public. That's also the main reason why there are plenty of riding share company have been risen up these years. With the concept of "Sharing Economy", vehicle service can be updated to match drivers with passengers through mobile apps or websites and arrange fastest and the most economical service near them, and take passengers from the start point to a certain destination.

However, as the situation growing complicated, issues like how to allocate limited taxi resources so as to balance the passenger demand and how to accurately show passengers the potential time consumption of upcoming trip has emerged. Those problems can be addressed with the past experience we gained from history records, which is an important use case of taxi trip data nowadays.

The in progress Intelligent Transportation System offers more and more opportunities helping researchers, companies and planners to identify daily travel habits, daily routines of local populations. In addition, thanks to the development of GPS technics and Internet data server, we are able to grab those biliions of records on information to do "big data" analytical work. The The New York City Taxi & Limousine Commission has released a staggeringly detailed historical dataset covering over 1.1 billion individual taxi trips in the city from January 2009 through June 2015.[1]

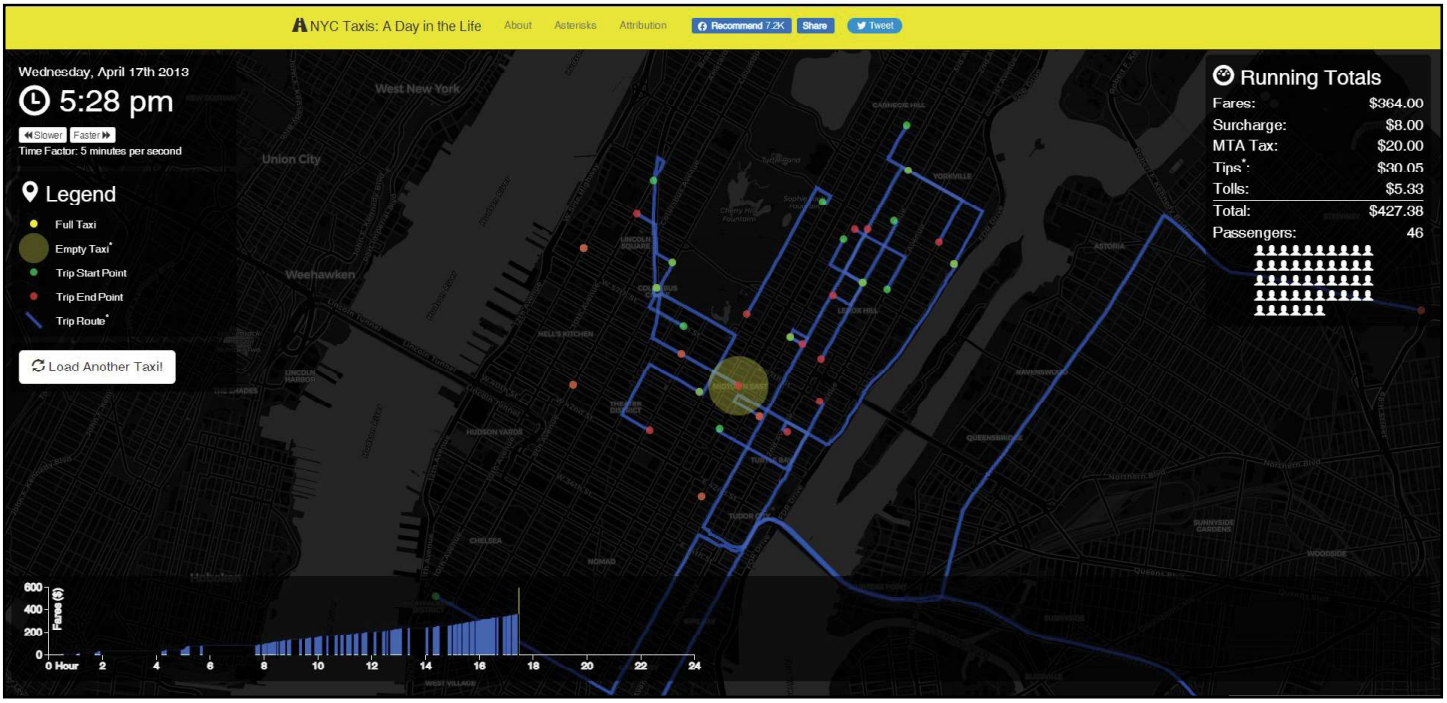
In this project, I'm going to analyze the data grabed from Kaggle with over 1.5 million taxi trip observasions which are sampled and cleaned from NYC Taxi and Limousine Commision (TLC) in year of 2016, with the help of temporal and spatial analysis method, I would want to identify the hotspot area and time where taxis are mostly in need, to predict the trip demand and time consuption given assumed situation.The analysis results can not only work as a decision support tool to determine where investments in transit infrastructure or where and when to allocate taxi resources appropriately, but also works to identify the taxi mobility pattern spatiotemporally. Drivers will be able to foresee the most in dimand district and head there in advance, which could decrease overall waiting time for both drivers and passengers.All these analytical work routine can be duplicated and applied into other similar cities or districts, which will be more convenient for people to understand the traffic story underlying the taxis in the city.

1 Schneider, T. (2015, November 17). Analyzing 1.1 billion NYC taxi and Uber Trips, with a vengeance. toddwschneider.com. Retrieved April 27, 2022, from https://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/

Literature Review

Taxi trip analysis works to help explore so much underlying pattern in a city. With up to billions of records and the big data analytical methods, we can get to know the city's neighborhoods, daily routines, airport traffic and so on through the lens of publicly available taxi data.

In the field of taxi trip data visualization, Mark Litwintschik has conducted a famous visualization called "NYC Taxis: A Day in the Life", which is an amazing interactive dashboard telling the mobility story of a random taxi in a day. His work make use of various relational database and big data technologies with a dataset of moderate 400GB size.[1]



Pic 1 Website that tells story of each taxi in a day

In order to improve the cost effect of implementing taxis into city, there are plenty of regulating strategies that match supply to demand (Posner, 1974),which essentially requires the understanding of taxi ridership.

Won Kyung Lee from Yonsei University While the vacancy rate of taxi is around 40%, passengers still have to wait for considerable length of time to catch taxis. so they conducted parametric models to focus on taxi vacancy duration analysis, considered variables in weather, landuse, demographics, socioeconomic variables and accessibility of public transportation, so as to find out the determinants of taxi vacancy duration in urban areas.

Because NYC has largest taxi demand and has large-scale GPS taxi data available publically, there are also bunch of relevent studies. Camerer et al. (1997) worked on trip sheets written by taxi drivers, which shows drivers are more likely to work longer time if their daily wage is low. Yang and Gonzales(2014) used 147 million NYC taxi trips t o find the relationship between the number of taxi trips and other factors, such as demographic and socioeconomic variables. By conducting multiple regressions and calculate the trip count generated by hour in each census tract, considering factors like education, age, income, population, transit access time and total jobs etc,. Corpuz(2007) found high-income workers are more likely to use private cars instead of public transportation system. Xinwu in Purdue University(2015) explored the spatial variation of urban taxi ridership in NYC, they found that most trips are more sparsely distributed in other boroughs than Manhattan. Although OLS regression model is widely used to make overwelmed assumption of all variables, taxi ridership is more sensitive to urban forms and varies across space.

1 NYC taxis. A Day in the Life. (n.d.). Retrieved April 27, 2022, from http://chriswhong.github.io/nyctaxi/

Data

For my overall analysis, I used four types of data from seperate sources in total, which are taxi trip records in NYC 2016 from Jan to June, weather records, ACS demographic data regarding to social-demographic and transit features, amenity infrastructures from Open Data NYC. The details of the data set are listed below:

- **Taxi Trip Data in NYC, 2016**

Although The **New York City Taxi & Limousine Commission** has already released a staggeringly detailed historical data set covering over 1.1 billion individual taxi trips in the city from January 2009 till now.[1] , I grab a more cleaned and well organized data with specific column of trip duration of each trip records from Kaggle. The data was sampled and cleaned for the purposes of a playground competition. which shares code and data to collaboratively create more optimized model to predict taxi ride time consumption. In return, the data set is kind of old in 2022, but the overall analytical method can be easily duplicate to other cities, or taxi trip dataset of other time.

The variables from the data are as follows:

Id	a unique identifier for each trip
vendor id	a code indicating the provider associated with the trip record
pickup datetime	date and time when the meter was engaged
drop off datetime	date and time when the meter was disengaged
passenger count	the number of passengers in the vehicle (driver entered value)
pickup longitude	the longitude where the meter was engaged
pickup latitude	the latitude where the meter was engaged
drop off longitude	the longitude where the meter was engaged
drop off latitude	the latitude where the meter was disengaged
store and fwd flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
trip duration	duration of the trip in seconds

Pic 1 Taxi trip data set detail

- **American Community Survey Data**

The American Community Survey (ACS) releases new data every year through a variety of data tables that you can access with different data tools. From which, I grabbed features in three main fields that looks more correlated to taxi trip demand. Those three fields are public commuting and transit pattern, socio-economic features and races.

The original features are:

a) Means of travel for commuting:

whether residents in each tracts use public transit, driving car or van, or taking taxi.

b) Travel time in minutes

the travel time to work with four categories, which are 30-35 min, 35-45 min, 45-60 min and above 60min.

c) Socio-economical features

1. Median Household Income in the past 12 months.

---

1 Www1.nyc.gov. 2022. TLC Trip Record Data - TLC. [online] Available at: <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>.

2. Median Age in each tracts.
3. The number of people who has a bachelor degree or above.
4. Total populations in each census tract.

d) Races

The number of White people, Black or African Americans alone, American Indian and Alaska Native alone and Asian in each census tract.

In addition, with the data info underlying in each census tracts, I also chose census tract as my spatial unit of analysis. In this way, I can not only visualize them with GEOID, identify the start and end point of trip in a larger perspective, and calculate the trip demand in each tract for further analysis.

- **Weather Records**

Since it is obvious that adverse weathers condition tend to have impact on transportation system and people's preference on whether taking taxi or not, I grabbed weather data with riem package from Automated Surface Observing System (ASOS) NWS stations (airports) on Iowa Environment Mesonet website[2]. ASOS is designed to support weather forecast activities and aviation operations and, at the same time, support the needs of the meteorological, hydrological, and climatological research communities.

After that, I aggregated weather features such as precipitation, temperature and wind speed with census tract across time.

- **Amenity/Built Environment features in NYC**

Explored from open data NYC, I imported several point of interest as amenity features that could influence the preference of taking taxi in each tracts. Those data sets are listed as follows:

a) Restaurants : Open Restaurant Applications
b) Stores : City Store - The Official Store of the City of New York
c) Hotels : Hotels Properties Citywide
d) Subway Stations : City Subway Stations
e) WIFI-Hotspots : NYC Wi-Fi Hotspot Locations

2 Akrherz@iastate.edu, D. H. (n.d.). Iowa environmental mesonet. IEM News + Notes. Retrieved April 29, 2022, from https://mesonet.agron.iastate.edu/

Methods

The main purpose of my project is listed below:

1) To introduce and visualize a real-world dataset abount NYC taxi trips.
With maps, bar plots and charts,etc,By conducting Exploratory Data Analysis to visualize the hidden pattern inside the dataset and identify the most appropriate features for model building.

2) To build two set of Machine Learning Models that predicts the taxitrip time duration and the taxi demand in each tracts across time.

3) After the process of feature selection and model building, I will conduct Algorithm Development which builds multiple models and make comparison, to apply the best model that performs to show the most accurate and generalized prediction.

The overall routine can be simplified in the following framework:

Process

- **Data Wrangling and Feature Engineering**

ACS demographic features:

Considering taxi ride is a kind of commuting method, the commuting pattern of public transit should be potential factors to taxi trip time consumption and taxi demand around city. Under the unit of Census Tract in NYC, there are overall 2194 tracts, which means there are overall 21940 observations of ACS features.

The cleaned feature table is as follows, which has 25 columns.

pub_traveltime_60 <dbl>	Total_Pop <dbl>	Median_income <dbl>	Median_age <dbl>	upper_rent <dbl>	lower_rent <dbl>	GEOID <chr>	NAME <chr>
0	1136	*NA*	35.8	*NA*	*NA*	36051031000	Census Tract 310, Livingston County, New York
0	4904	40924	39.6	626	439	36053030102	Census Tract 301.02, Madison County, New York
27	2335	12977	25.3	728	500	36055000200	Census Tract 2, Monroe County, New York
0	3522	41750	31.6	824	562	36055001000	Census Tract 10, Monroe County, New York
87	4023	43846	28.1	892	615	36055002900	Census Tract 29, Monroe County, New York
0	1136	19375	27.3	644	486	36055004100	Census Tract 41, Monroe County, New York

Pct_publictrans_towork <dbl>	Pct_ride_towork <dbl>	pub_traveltime_30_60 <dbl>	ride_traveltime_30_60 <dbl>	ride_traveltime_60 <dbl>	pct_bachelor <dbl>	pct_white <dbl>	pct_black <dbl>
0.00000000	0.2307692	0	1	0	0.03609155	0.4779930	0.43309859
0.02973790	0.8805444	0	408	41	0.13988581	0.9588091	0.01937194
0.34018265	0.5821918	25	37	0	0.07623126	0.1841542	0.50149893
0.04462294	0.7425257	77	195	143	0.40601931	0.7572402	0.17035775
0.05891980	0.8193126	67	192	100	0.41884166	0.8429033	0.09594830
0.02848101	0.8386076	9	0	9	0.05721831	0.3653169	0.37940141

pct_indian <dbl>	pct_asian <dbl>	pct_poverty <dbl>	pct_onemployment <dbl>	pct_pubtrans_native <dbl>	pct_pubtrans_foreign <dbl>	geometry <S3: sfc_MULTIPOLYGON>	rownames <lgl>	options <list>
0.010563380	0.0008802817	0.004401408	0.008802817	NaN	NaN	<S3: sfc_MULTIPOLYGON>	FALSE	<list [1]>
0.013254486	0.0085644372	0.133972268	0.150693312	1.0000000	0.0000000	<S3: sfc_MULTIPOLYGON>	FALSE	<dbl [1]>
0.010278373	0.1982869379	0.431263383	0.120342612	0.5436242	0.4563758	<S3: sfc_MULTIPOLYGON>	FALSE	<list [1]>
0.000000000	0.0289608177	0.128052243	0.071550256	0.8300000	0.1700000	<S3: sfc_MULTIPOLYGON>	FALSE	<dbl [1]>
0.008699975	0.0454884415	0.195625155	0.059656972	1.0000000	0.0000000	<S3: sfc_MULTIPOLYGON>	FALSE	<list [1]>
0.000000000	0.1716549296	0.316021127	0.127640845	1.0000000	0.0000000	<S3: sfc_MULTIPOLYGON>	FALSE	<dbl [1]>

6 rows | 17-25 of 25 columns

Table 1 Feature selected from ACS

Taxi Trip features:

The taxi trip dataset I used collects various set of information related to taxi trip, including pickup and drop off time and location. Grabbed from Kaggle, the data set is much clean than raw data on TLC already, however, there are still some problems with the data. As real-world data, it has missing data, unimportant columns, baked-in biases, etc. , which requires processing operations.

1) First, I transferred the data column property.

At beginning, all the columns are in character type, which means although the value shows in numeric or time form, it can not be recognized in specific form.(such as latitude and longitude, although it shows the explicit features I can not transfer the raw form into geometries). So I transferred longitude, latitude from character into numeric, and the pickup and dropoff time into "timestamp" form.

In this way, I can easily map the start and end location in point geometry form, and extract detailed time property like hour in a day, or weekday in a week for further analysis.

2) Second, I created several new features based on current value in some columns.

Some columns contains multiple information that should be extracted for better analysis.

- I calculated distance of each trip by points of pick up and drop off location.
- I extracted features like week, hour, time in a day, weekday, is the date weekend or not, etc. In this way, each trip will be more correlated to specific time slots, making prediction more accurate.
- I parsed a binary variable called "check" to calculate the time interval between pickup time and dropoff time manually so as to make comparison to the existing trip duration, in this way, it will be clear if the existing trip duration match the calculation result or not. Learning from the checking result, it shows all the duration equals the calculation between pick up and drop off interval..This step is necessary because the data is provided by a third party, where the trip time duration is provided di-

rectly, but since this feature is in a way a dependent variable for future analysis, its accuracy is essential to the performance of the model. So, create a checking variable to fillter out the trip data that is not reliable will save more potential trouble.

id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration	RowID	distance
Length:1458644	Min. :1.000	Min. :2016-01-01 00:00:17	Min. :2016-01-01 00:03:31	Min. :0.000	Min. :-121.93	Min. :34.36	Min. :-121.93	Min. :32.18	Length:1458644	Min. : 1	Min. : 1	Min. : 0
Class :character	1st Qu.:1.000	1st Qu.:2016-02-17 16:46:04	1st Qu.:2016-02-17 17:05:32	1st Qu.:1.000	1st Qu.: -73.99	1st Qu.:40.74	1st Qu.: -73.99	1st Qu.:40.74	Class :character	1st Qu.: 397	1st Qu.: 364662	1st Qu.: 1233
Mode :character	Median :2.000	Median :2016-04-01 17:19:40	Median :2016-04-01 17:35:12	Median :1.000	Median : -73.98	Median :40.75	Median : -73.98	Median :40.75	Mode :character	Median : 662	Median : 729323	Median : 2096
NA	Mean :1.535	Mean :2016-04-01 10:10:24	Mean :2016-04-01 10:26:24	Mean :1.665	Mean : -73.97	Mean :40.75	Mean : -73.97	Mean :40.75	NA	Mean : 959	Mean : 729323	Mean : 3445
NA	3rd Qu.:2.000	3rd Qu.:2016-05-15 03:56:08	3rd Qu.:2016-05-15 04:10:51	3rd Qu.:2.000	3rd Qu.: -73.97	3rd Qu.:40.77	3rd Qu.: -73.96	3rd Qu.:40.77	NA	3rd Qu.: 1075	3rd Qu.:1093983	3rd Qu.: 3880
NA	Max. :2.000	Max. :2016-06-30 23:59:39	Max. :2016-07-01 23:02:03	Max. :9.000	Max. : -61.34	Max. :51.88	Max. : -61.34	Max. :43.92	NA	Max. :3526282	Max. :1458644	Max. :1242299
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

interval60	week	hour	WeekDay	weekend	duration	check
Min. :2016-01-01 00:00:00	Min. : 1.0	Min. : 0.00	周日:195366	Length:1458644	Min. : 1	Mode:logical
1st Qu.:2016-02-17 16:00:00	1st Qu.: 7.0	1st Qu.: 9.00	周一:187418	Class :character	1st Qu.: 397	TRUE:1458644
Median :2016-04-01 17:00:00	Median :14.0	Median :14.00	周二:202749	Mode :character	Median : 662	NA
Mean :2016-04-01 09:40:20	Mean :13.7	Mean :13.61	周三:210136	NA	Mean : 959	NA
3rd Qu.:2016-05-15 03:00:00	3rd Qu.:20.0	3rd Qu.:19.00	周四:218574	NA	3rd Qu.: 1075	NA
Max. :2016-06-30 23:00:00	Max. :27.0	Max. :23.00	周五:223533	NA	Max. :3526282	NA
NA	NA	NA	周六:220868	NA	NA	NA

Table 2 Summary of taxi trip before data clean process

3) Third, I cleaned the dataset by filter out strange outliers and unreliable trip records following several rules.

- At beginning, let's firstly see the current stage of the data table with summary function.

- I processed the data to each necessary columns accordingly, the rule of cleaning and the results are as follows:
  a. **id:** original id is a set of distinct character that represent each individual trips. which doesn't need operation.
  b. **vendor_id:** There are 2 unique values, which is 1 and 2, represents the vendor of trips. It's okay no matter the column form is in factor or character.
  c. **passenger_count:** There are 9 unique passenger counts. However, the minimum value of passenger count is 0 while the maximum value of passenger count is 10, which is impossible for taxi trip! Firstly, all the trip should have at least one passenger, or the trip will not count, so I should remove any trip that has none passenger, because there must be something strange. Second, according to the NYC Taxi&Limousine Commission, the maximum number of people by law is five

passengers in one taxicab. Considering the condition when there are passengers under the age of 7 and doesn't take any space and prefer sitting on adults' leg. The maximum passenger of each trip should be 7. So I have to remove any trip record that has passengers higher than 7.

d. **Longitude and Latitude:** No matter the long or lat of pick-up and drop-off location, the trip should be happened inside the boundary of New York City , So I have to remove any trips that locates outside NYC boundaries.

e. **store_and_fwd_flag:** This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server : Y=store and forward; N=not a store and forward trip, which doesn't make any difference to the target variable.

f. **trip_duration:** duration of the trip in seconds, which is the dependent variable in our analysis. According to the summary,the maximum duration of trip is 3526282 seconds,converting to hours is 980 hours. If this time is really the trip time consumption, it could be ridiculous. Maybe it's because the owner of the taxi forgot to turn off the meter before he leave the car for a long time. In this case, it will be helpful to r**emove any trip that has time duration more than two standard deviations away from the mean duration time**, because according to statistical theory, about 95% of values will be within 2 standard deviations of the mean.

After all the cleaning work, by comparison, it shows that I removed overall 3500 observations from the original taxitrip records, which takes over 0.24% of the dataset. Now the cleaned dataset contains 1455087 observasions.

Weather features:

interval60	Temperature	Precipitation	Wind_Speed
Min. :2016-01-01 00:00:00	Min. :-0.94	Min. :0.000000	Min. : 0.00
1st Qu.:2016-02-14 22:30:00	1st Qu.:37.94	1st Qu.:0.000000	1st Qu.: 3.00
Median :2016-03-30 05:00:00	Median :50.00	Median :0.000000	Median : 5.00
Mean :2016-03-31 04:50:14	Mean :50.98	Mean :0.008357	Mean : 4.98
3rd Qu.:2016-05-14 23:30:00	3rd Qu.:64.04	3rd Qu.:0.000000	3rd Qu.: 7.00
Max. :2016-06-29 23:00:00	Max. :89.96	Max. :1.850000	Max. :74.00

Table 3 Cleaned weather feature table

The weather records extracted Iowa Environment Mesonet lies between January 1, 2016 and June 1, 2016, after the basic aggregation and cleaning work, the key features includes precipitation, temperature and wind speed. As for how to combine the records with the taxi trip dataset? With the timestamp in taxitrip data, we can easily join the two table and label the weather condition on each date, so that the weather panel would give each taxitrip a clear background of weather condition.

Amenity/Built Environment features:

Since the dataset from open data NYC are all in from of point geometry, I combined those geometries with each taxi trip records based on the which census tract is the passenger of trip records picked up. The original amenity datasets are listed below:

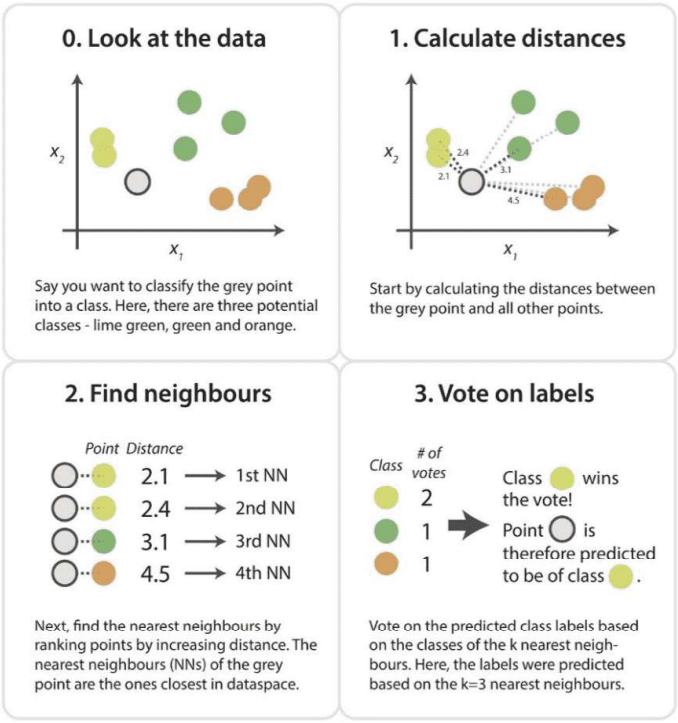
a. **Restaurants :** Open Restaurant Applications
b. **Stores :** City Store - The Official Store of the City of New York
c. **Hotels** : Hotels Properties Citywide
d. **Subway Stations :** City Subway Stations
e. **WIFI-Hotspots :** NYC Wi-Fi Hotspot Locations

k nearest neighbor

In order to combine the points to the taxi trip table, to transfer geometries into features in the panel, I used a supervised machine learning algorithms called "k nearest neighbor" [1]to do this job. "k-nearest neighbor" defines neighbor based on the logic "similar things tend to be near to each other", it defines "similarity" by distance, closeness or proximity, in this case, we can calculate the average distance between the pick up location to the point geometris on the map.

K is the number of nearest neighbors to retrieve for making predictions. there are two choices of weighting method, which are uniform and inverse distance weighting. With uniform weighting, we do not take into account the distance between the new

1 K-Nearest Neighbor. (2021). Antony Christopher. https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4



Pic3 The theory of K nearest neighbor

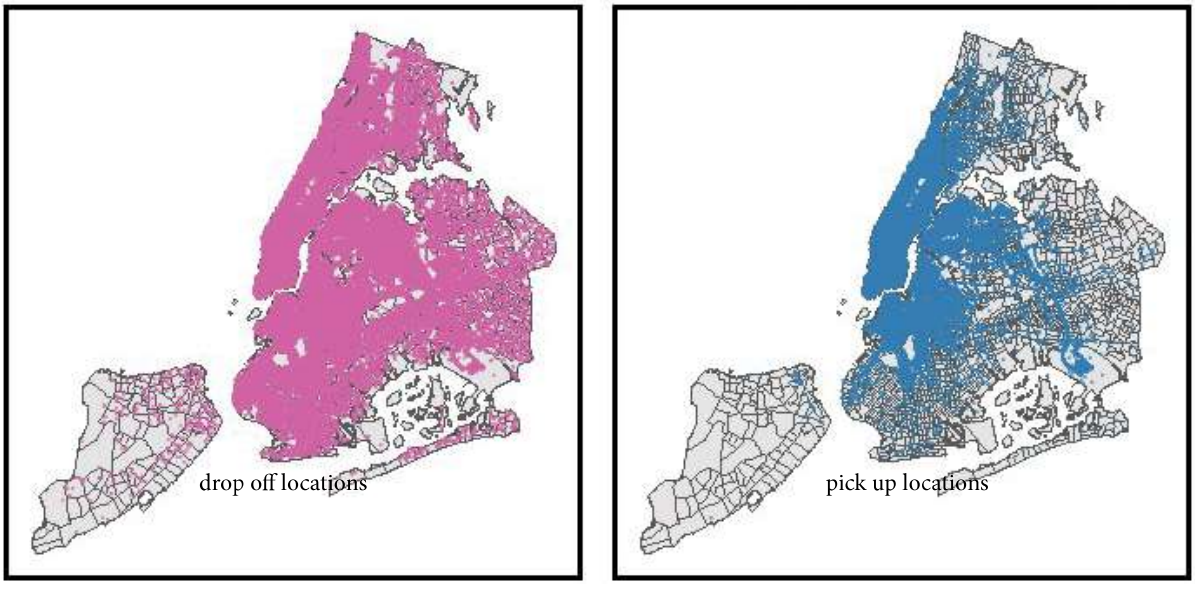
data point and its k nearest neighbors, because they all have equal influence over the prediction. With inverse distance weighting, we can assign higher weights to the closer training examples.

In my project, the way I select the value of k is by the number of each point of interest and I used uniform weighting method so as to get the average distance to the neighbor aminties to each trip

- **Exploratory Data Analysis**

After I wrangled all the features for the model building, I firstly made several plots, charts and maps to firstly get to know the variables briefly. The exploratory work are mainly univariate analysis, multivariate analysis and correlation analysis.

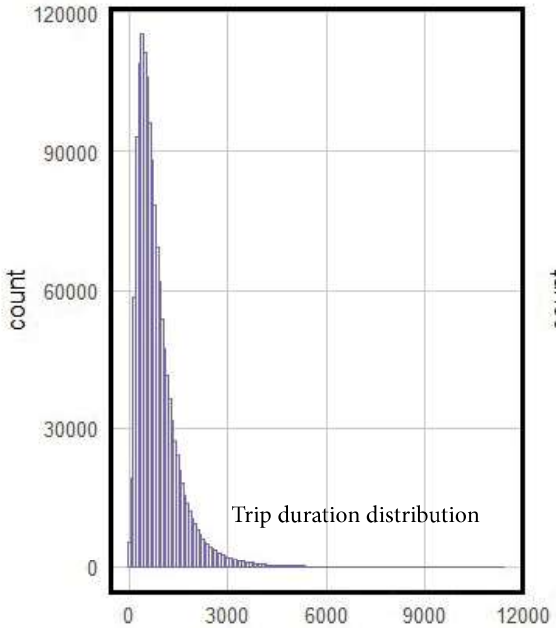
1. pick up and drop off location maps

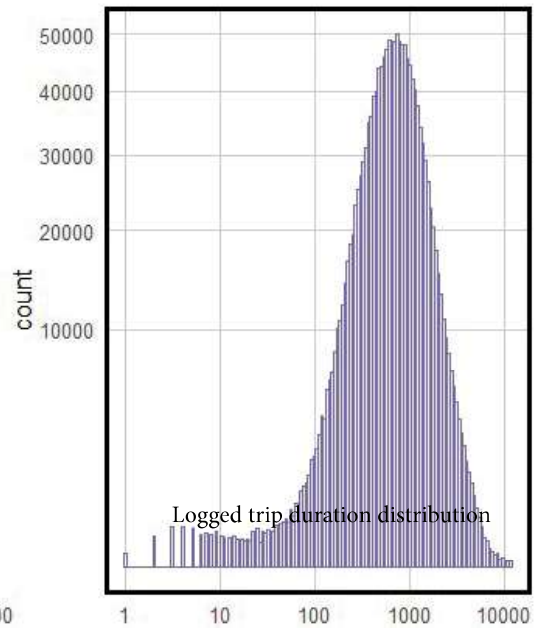


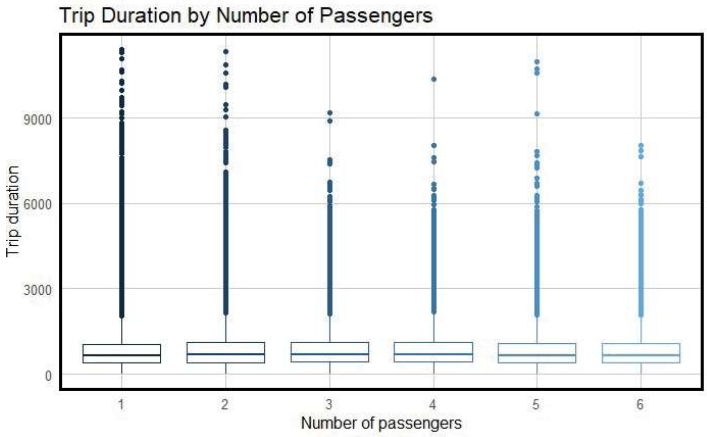
Observing from the map,it shows point geometry scattered around city, it is obvious that Manhattan Island and Brooklyn of NYC burdens pretty much higher taxi ride demand. And drop off locations clearly spreads farther than pick up locations, which means most passengers taking taxis inside center city and goes outer boroughts.

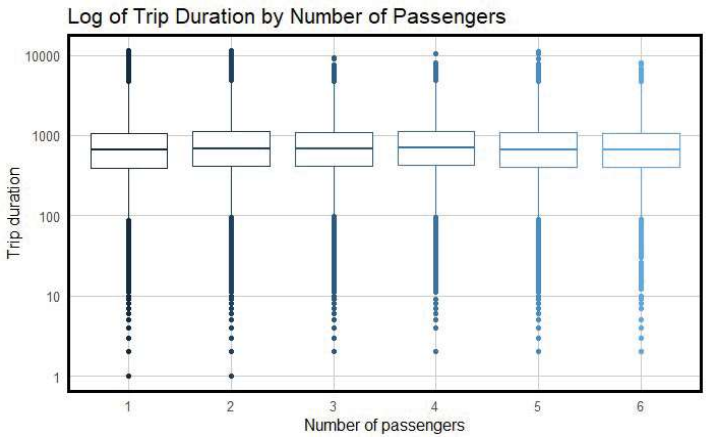
2. For target variable "Trip duration"

Learning from the histogram, by count the distribution about time length of each trip, the count of trip duration shows a long skew as the time consumption goes up. But if I use log transfermation to the trip duration, it shows a normal distribution pattern.







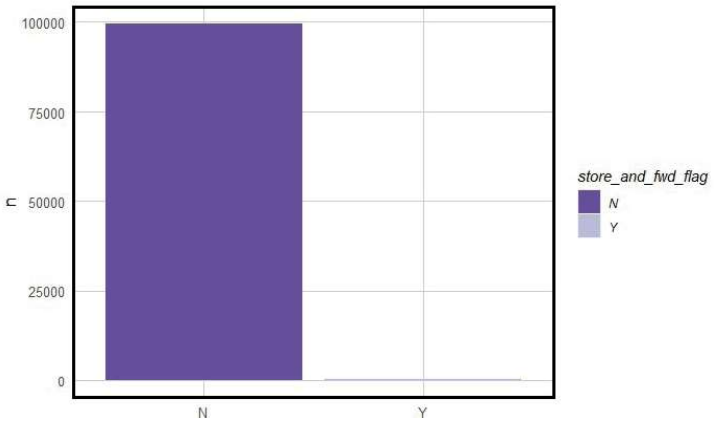


3. For target variable "passenger_count"

I firstly made a box plot chart for each category of passenger count, to show the distribution of trip duration with regard to the passenger count, but the plot shows little difference and there are bunch of outliers, so I log transferred the trip duration and made the log of trip duration box plot, the chart shows the same pattern between the passenger count to the trip duration. which means passenger count doesn't do any difference to the trip duration.

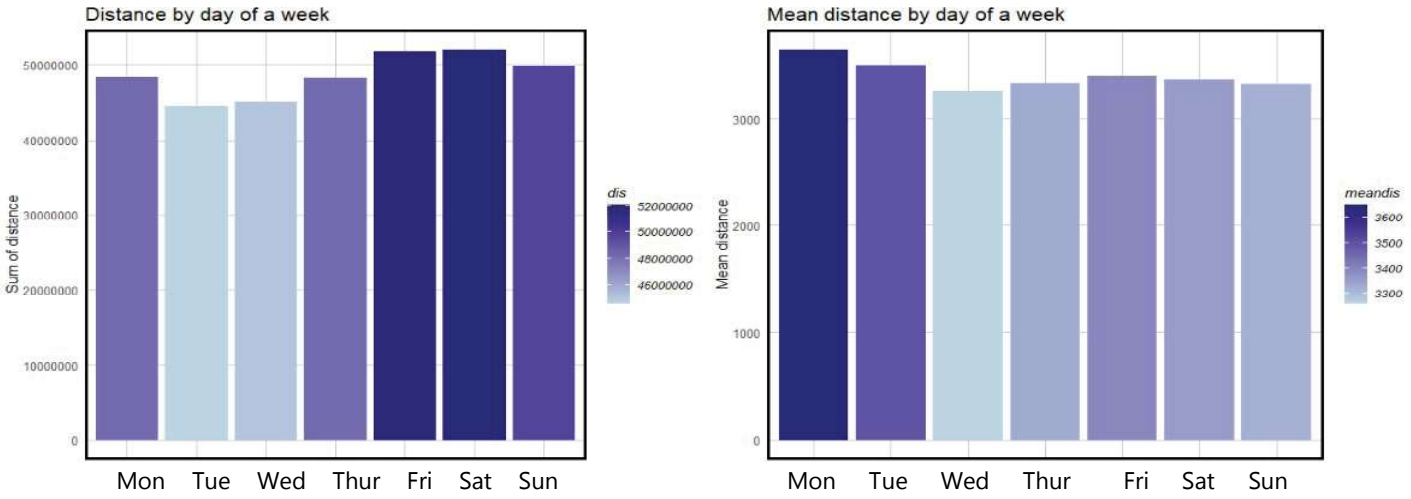
4. For target variable "store_and_fwd_flag"

This is a binary variable indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server,where Y=store and forward; N=not a store and forward trip. As it is shown in the chart, nearly almost observasion returns the value of N.



5. For target variable "distance" and its relationship between time.

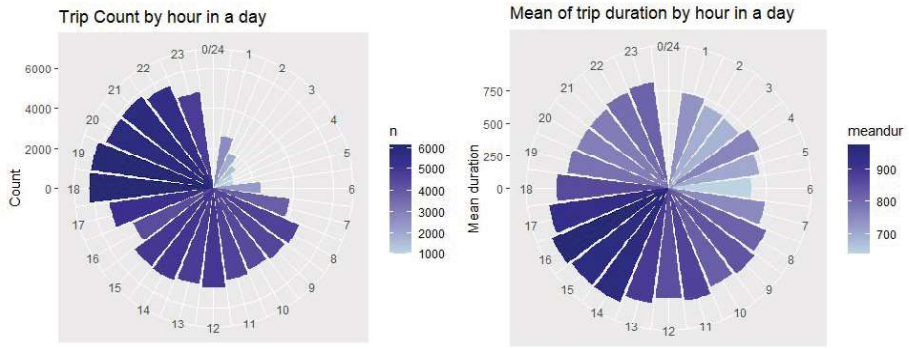
Here I plot both the sum of distance across the week day, and the mean value of distance of a week.I thought the pattern of two plots should look similar, but result is surprising! As plots shown below, especially in the cases of Monday, Thursday and Friday. On Monday, the sum of distance is the lowerest, but the mean value of the distance is second highest, which means the trip count is least but people tend to take taxi in longer ridership, maybe take taxi from outside town to center city, etc.
As on Thursday and Friday, the mean value of distance is in the middle, while the sum of distance are the highest. Which means during these days, people tend to take taxi to relatively near place, with lower price but more frequent call. Maybe it's becuase



people tend to hang out on these days.In other words, the distribution pattern of distance and trip count looks similar all the same.

6. For target variable "Hour in a day"

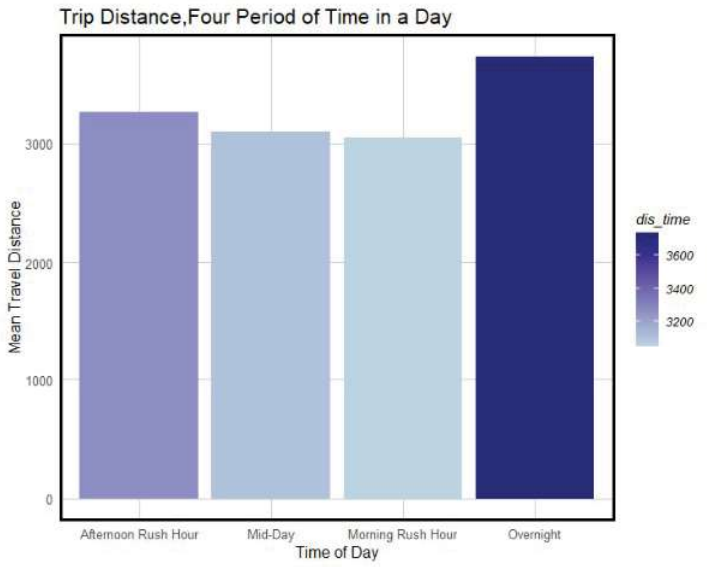
Here I plot both the mean trip duration and trip count across each 24 hours in a day.We can see the peak of trip count is at noon to

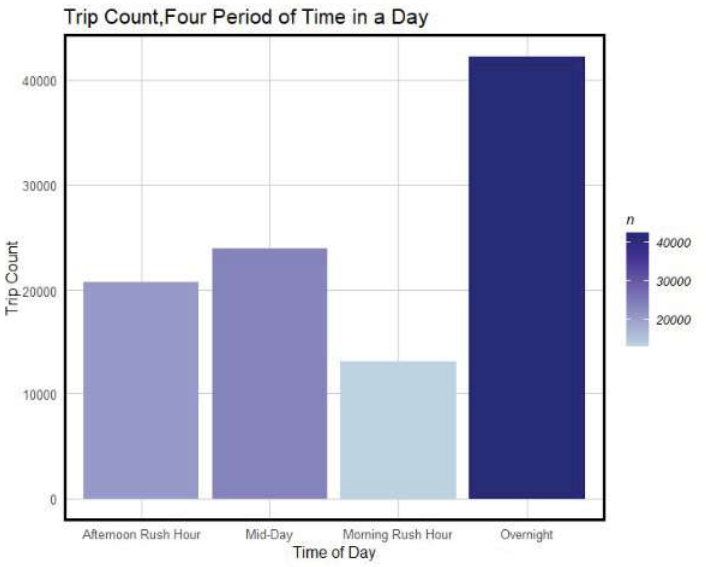


afternoon, while the peak of trip distance is in late afternoon. Although in midnight, the trip count is low, the average trip duration is similar to other times.

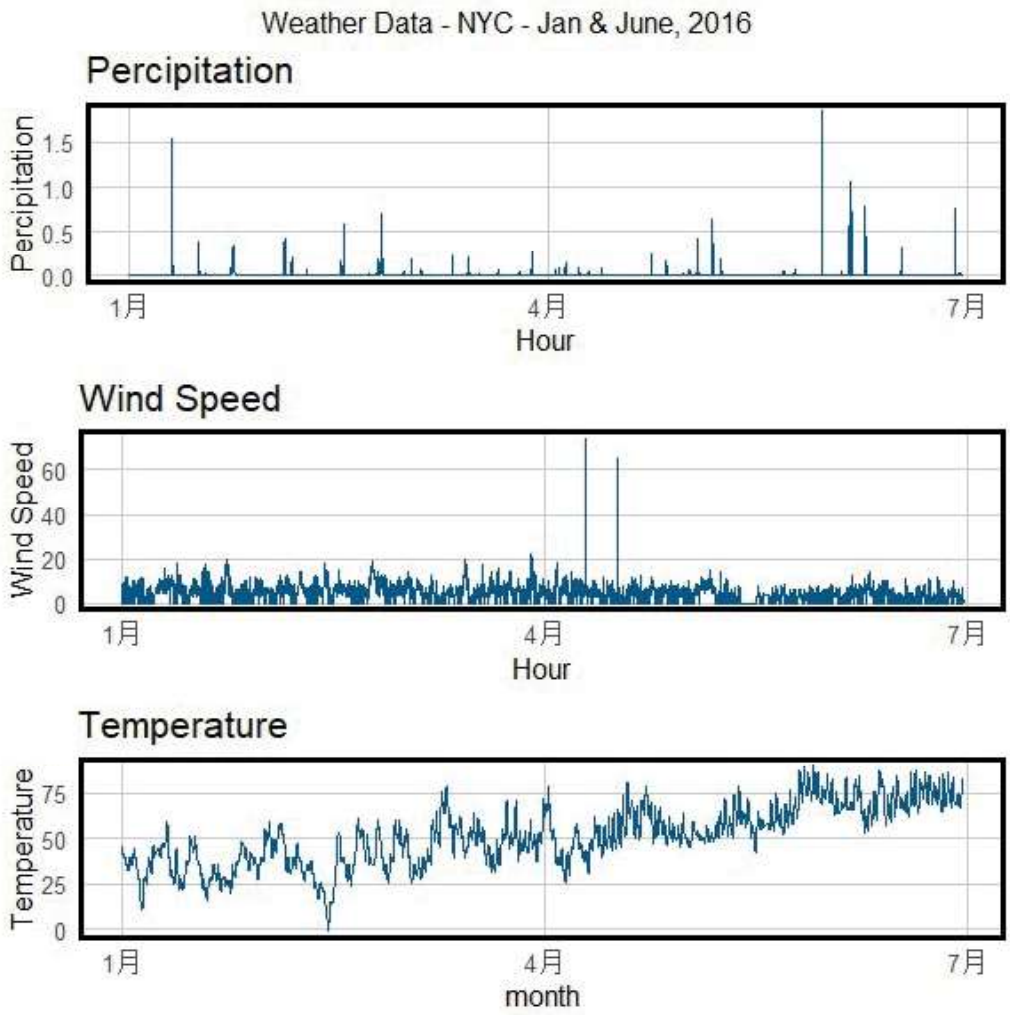
7. For target variable "time of day"

By classifying one day into four main periods,the distance and trip count distribution across day is clear. It is surprising that





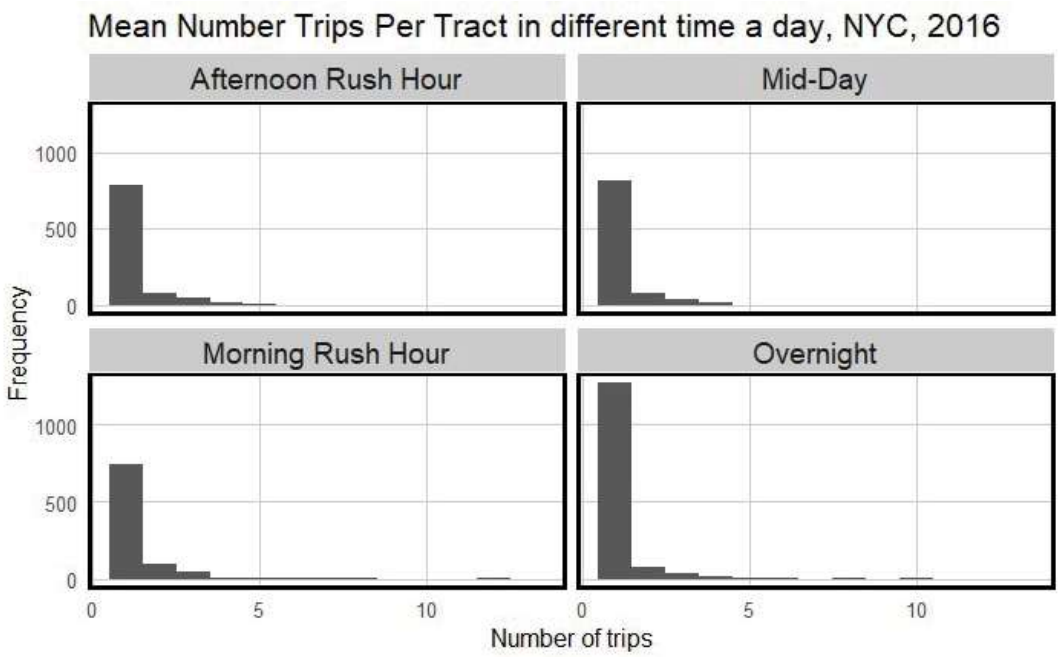
during both morning and afternoon rush hour,trip distance is lower than other time. Also, the trip count during morning rush hour is the lowest one! Which means during rush hour, people choose to commute to work in other ways instead of taking taxi.

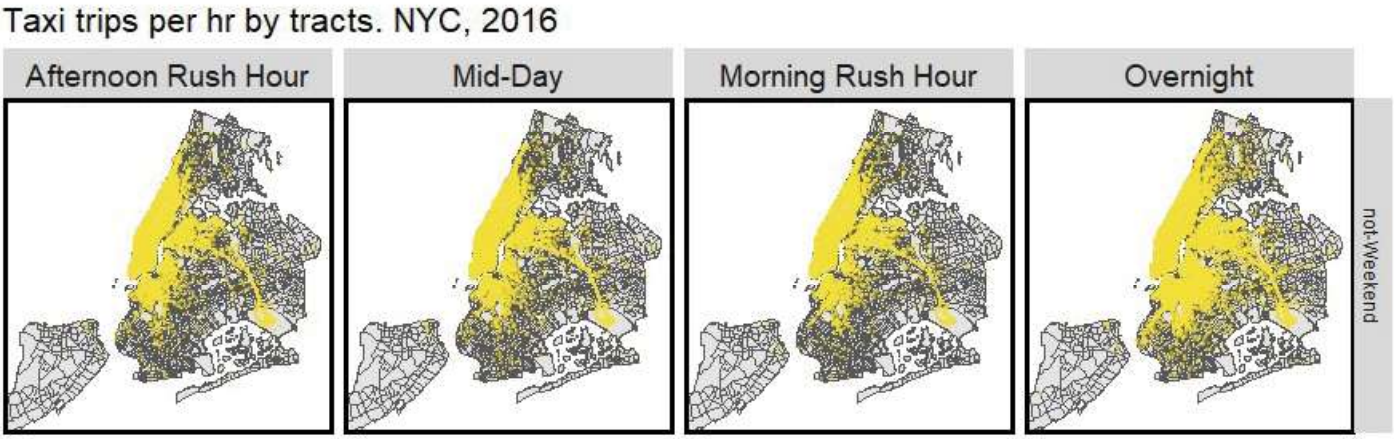


8. For weather features

9. For number of trips across different time of a day.

After I classify one day into four periods of time, and count the total number of trip in each periods, with the histogram and maps at same time, the spatial and temporal distribution of the trip is shown clearly.





Time Lag Correlation

Variable	correlation
lagHour	0.02
lag2Hours	0.02
lag3Hours	0.00
lag4Hours	0.00
lag12Hours	-0.01
lag1day	0.03

10. Create time lag features

To test for serial (temporal) correlation, I also creates time lags to explore the shift of taxi trip demand and time duration backward in the time(sequence). Lag features are values at prior timesteps that are considered useful because they are created on the assumption that what happened in the past can influence or contain a sort of intrinsic information about the future. For example, it can be beneficial to generate features for sales that happened in previous days at 4:00 p.m. if you want to predict similar sales at 4:00 p.m. the next day.

An interesting category of lag features is called nested lag features. In order to create nested lag features, data scientists must identify a fixed time period in the past and then group feature values by that time period, for example, the number of items sold in the previous two hours, previous three days, and previous week.[2]

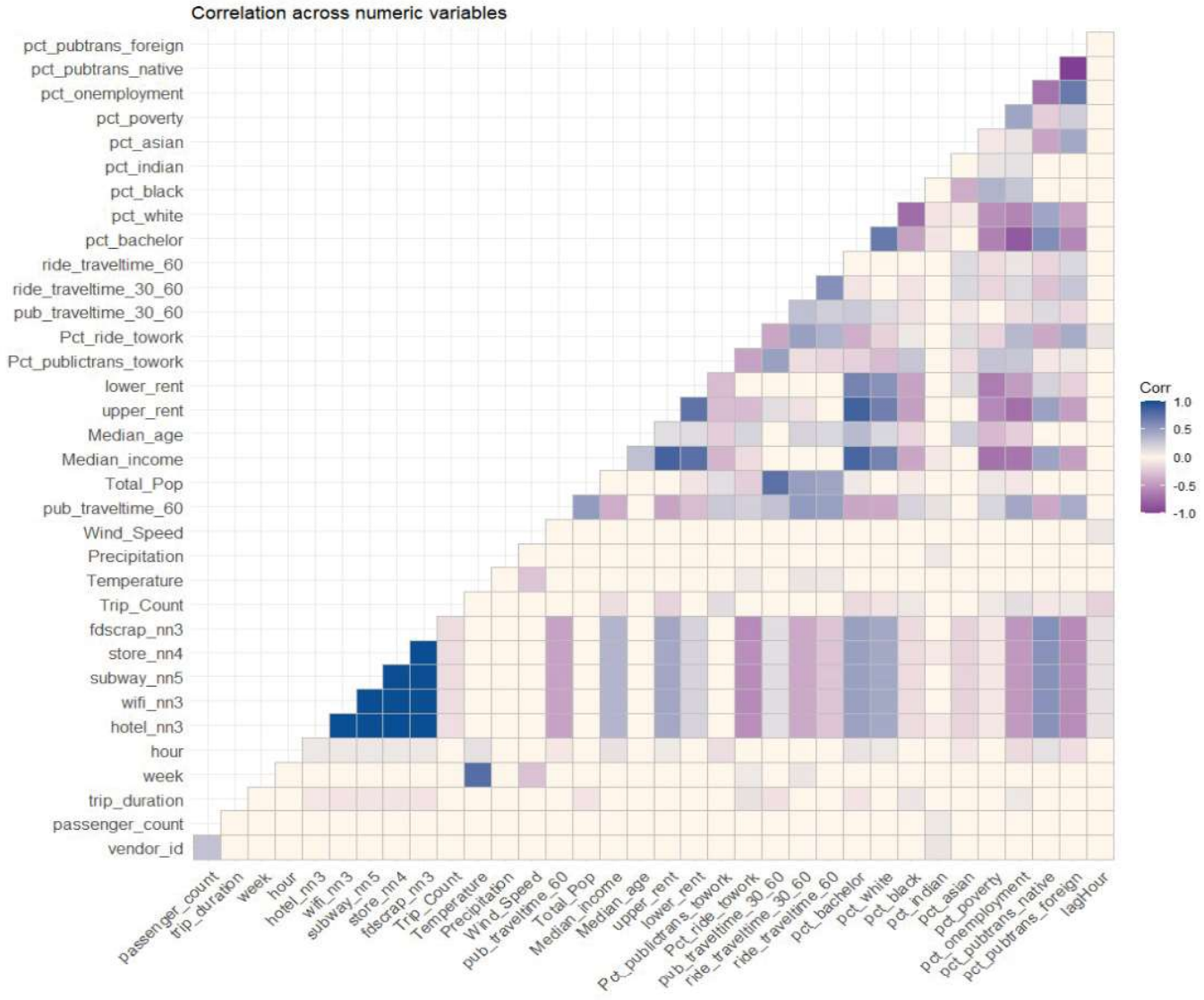
2 Introduction to feature engineering for time series forecasting. (2021). Francesca Lazzeri. https://medium.com/data-science-at-microsoft/introduction-to-feature-engineering-for-time-series-forecasting-620aa55fcab0

11. Sum up the taxi trip in each census tract

In order to calculate taxi demand in the unit of census tract, I created another feature called "Trip count" in each trip records. After group the trips with census tract id, and count the trip number in each census tract, I will get an aggregate value of trips in each census tract across time, both in weekday in a week or hours in a day.

12. Correlation Analysis of numeric features

After all the correlation work, I aggregated all features created above and get the final panel with overall 55 variables. The panel table is as follows and the correlation paring those numeric features are plotted in the matrix below. Learning from the matrix, the pattern of collinearity, it shows that some travel time period are correlated with each other, and percentage of people that is



id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	store_and_fwd_flag	trip_duration	interval60	week	hour	WeekDay	weekend	time_of_day	Origin.Tract	Destination.Tract	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	hotel_nn3
id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	N	455	2016-03-14 17:00:00	11	17	周一	not-Weekend	Afternoon Rush Hour	36061014500	36061012000	-73.98215	40.76794	-73.96463	40.76560	304738.4
id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	N	663	2016-06-12 00:00:00	24	0	周日	not-Weekend	Overnight	36061006600	36061006500	-73.98042	40.73856	-73.99948	40.73115	304738.5
id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	N	2124	2016-01-19 11:00:00	3	11	周二	not-Weekend	Mid-Day	36061013700	36061001501	-73.97903	40.76394	-74.00533	40.71009	304738.4
id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	N	429	2016-04-06 19:00:00	14	19	周三	not-Weekend	Overnight	36061003900	36061000700	-74.01004	40.71997	-74.01227	40.70672	304738.5
id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	N	435	2016-03-26 13:00:00	13	13	周六	not-Weekend	Mid-Day	36061017900	36061016500	-73.97305	40.79321	-73.97292	40.78252	304738.4
id0801584	2	2016-01-30 22:01:40	2016-01-30 22:09:03	6	N	443	2016-01-30 22:00:00	5	22	周六	not-Weekend	Overnight	36061006800	36061010100	-73.98286	40.74220	-73.99208	40.74918	304738.5

wifi_nn3	subway_nn5	store_nn4	fdscrap_nn3	Trip_Count	Temperature	Precipitation	Wind_Speed	pub_traveltime_60	Total_Pop	Median_income	Median_age	upper_rent	lower_rent	NAME	Pct_publictrans_towork	Pct_ride_towork	pub_traveltime_30_60	ride_traveltime_30_60
289594.2	302328	309952.8	285171.6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
289594.2	302328	309952.8	285171.6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
289594.2	302328	309952.8	285171.6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
289594.3	302328	309952.8	285171.7	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
289594.2	302328	309952.7	285171.6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
289594.2	302328	309952.8	285171.6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

on employment and peple who taking public transit are correlated.

ride_traveltime_60	pct_bachelor	pct_white	pct_black	pct_indian	pct_asian	pct_poverty	pct_onemployment	pct_pubtrans_native	pct_pubtrans_foreign	lagHour	lag2Hours	lag3Hours	lag4Hours	lag12Hours	lag1day
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

- ## Model Building

With the existing panel with overall features, I firstly created subset sample with 100000 observasions, split the sample into 75% train data and 25% test data, and conducted multiple machine learning models to trying to find the most fitted model. For the next step I'm going to make comparison based on the prediction accuracy and generalizability performance of each model.

As I illustrated before, in order to find the pattern of in both spatial and temporal field, I want to make model to predict "trip duration" which is the time consumption of taxi trip, and "trip count" which is the total number of trips happened in different time in the unit of census tracts. I will seperately conduct those two set of models.

- OLS Regression: Ordinary Least Squares regression (OLS) is a common technique for estimating coefficients of linear regression equations which describe the relationship between one or more independent quantitative variables and a dependent variable (simple or multiple linear regression). Least squares stands for the minimum squares error.

- Poisson Regression: Poisson regression is used to predict a dependent variable that consists of "count data" given one or more independent variables. The variable we want to predict is called the dependent variable (or sometimes the response, outcome, target or criterion variable). The variables we are using to predict the value of the dependent variable are called the independent variables (or sometimes the predictor, explanatory or regressor variables).

- Random Forest Model: Random forest is a kind of ensemble classifier which is using a decision tree algorithm in a randomized fashion and in a randomized way. It combines multiple decision trees and come to the final result. Some decision trees may not give it correctly, but because for random forest, we can randomly define the number of trees we want to build and how many variables we need at each node ,so the the output will be balanced together to be correct.

I. For Trip Duration

I overall conduct four kinds of models, which are , by with the 75% of data to train model, and predict trip time consumption of the last 25% of trip records, to see the performance of the model. For feature selection, I created three set of selections for model building.

1) Model1: Features related to weather condition, taxi trip itself and demographic features.

2) Model2: Features related to amenities or environmental buildings, weather and timelags.

3) Model3: Selected features related to weather condition, taxi trip itself, demographic features, amenities and time lag.

The summary of OLS regression three models is as follows:

OLS Regression First Model

```
OLS Regression Results
==========================================================
                                Dependent variable:
                            ---------------------------
                                    trip_duration
----------------------------------------------------------
store_and_fwd_flagY                   142.960***
                                       (7.843)
hour                                   1.387***
                                       (0.098)
WeekDay.L                             63.912***
                                       (1.485)
WeekDay.Q                            -92.663***
                                       (1.510)
WeekDay.C                            -36.651***
                                       (1.488)
WeekDay4                              9.463***
                                       (1.499)
WeekDay5                             -7.163***
                                       (1.489)
WeekDay6                              9.425***
                                       (1.482)
week                                  3.468***
                                       (0.129)
passenger_count                       5.410***
                                       (0.426)
weekendWeekend
time_of_dayMid-Day                     -0.952
                                       (1.781)
time_of_dayMorning Rush Hour         -65.802***
                                       (2.198)
time_of_dayOvernight                -106.015***
                                       (1.535)
Temperature                           0.468***
                                       (0.059)
Precipitation                        -26.035***
                                       (9.949)
Wind_Speed                            0.417***
                                       (0.149)
pub_traveltime_60                     -0.051***
                                       (0.005)
Total_Pop                             -0.004***
                                       (0.0003)
Median_income                         -0.00004*
                                       (0.00002)
Median_age                            -2.956***
                                       (0.107)
upper_rent                            0.025***
                                       (0.002)
Pct_publictrans_towork               -42.973***
                                       (4.923)
pct_bachelor                          29.315***
                                       (7.899)
pct_white                            -122.724***
                                       (6.174)
pct_onemployment                      505.946***
                                       (26.086)
Constant                              861.268***
                                       (8.294)
----------------------------------------------------------
Observations                             946,169
R2                                         0.023
Adjusted R2                                0.023
Residual Std. Error                  544.022 (df = 946143)
F Statistic                      897.701*** (df = 25; 946143)
==========================================================
Note:                          *p<0.1; **p<0.05; ***p<0.01
```

```
Call:
lm(formula = trip_duration ~ store_and_fwd_flag + hour + WeekDay +
    week + passenger_count + weekend + time_of_day + Temperature +
    Precipitation + Wind_Speed + pub_traveltime_60 + Total_Pop +
    Median_income + Median_age + upper_rent + Pct_publictrans_towork
    pct_bachelor + pct_white + pct_onemployment, data = trip_train)

Residuals:
    Min      1Q  Median      3Q     Max
-1104.8  -373.3  -128.5   230.6 10650.1
```

```
Coefficients: (1 not defined because of singularities)
                                Estimate    Std. Error
(Intercept)                   861.26811082    8.29426154
store_and_fwd_flagY           142.96010379    7.84282118
hour                            1.38670312    0.09803405
WeekDay.L                      63.91219958    1.48508513
WeekDay.Q                     -92.66318309    1.50964014
WeekDay.C                     -36.65106620    1.48832214
WeekDay^4                       9.46253313    1.49942854
WeekDay^5                      -7.16322602    1.48909903
WeekDay^6                       9.42485290    1.48202757
week                            3.46763544    0.12905642
passenger_count                 5.40961298    0.42642581
weekendWeekend                          NA            NA
time_of_dayMid-Day             -0.95226004    1.78147782
time_of_dayMorning Rush Hour  -65.80182270    2.19791918
time_of_dayOvernight         -106.01487086    1.53509332
Temperature                     0.46778284    0.05891861
Precipitation                 -26.03530597    9.94911711
Wind_Speed                      0.41684192    0.14931549
pub_traveltime_60              -0.05133054    0.00468803
Total_Pop                      -0.00354283    0.00025269
Median_income                  -0.00003950    0.00002161
Median_age                     -2.95551320    0.10708200
upper_rent                      0.02523414    0.00163221
Pct_publictrans_towork        -42.97288629    4.92295679
pct_bachelor                   29.31467948    7.89909296
pct_white                    -122.72351237    6.17388543
pct_onemployment              505.94605777   26.08642764
                               t value             Pr(>|t|)
(Intercept)                   103.839 < 0.0000000000000002 ***
store_and_fwd_flagY            18.228 < 0.0000000000000002 ***
hour                           14.145 < 0.0000000000000002 ***
WeekDay.L                      43.036 < 0.0000000000000002 ***
WeekDay.Q                     -61.381 < 0.0000000000000002 ***
WeekDay.C                     -24.626 < 0.0000000000000002 ***
WeekDay^4                       6.311  0.00000000027779131 ***
WeekDay^5                      -4.810  0.00000150619270854 ***
WeekDay^6                       6.359  0.00000000020259358 ***
week                           26.869 < 0.0000000000000002 ***
passenger_count                12.686 < 0.0000000000000002 ***
weekendWeekend                     NA                   NA
time_of_dayMid-Day             -0.535              0.592972
time_of_dayMorning Rush Hour  -29.938 < 0.0000000000000002 ***
time_of_dayOvernight          -69.061 < 0.0000000000000002 ***
Temperature                     7.939  0.00000000000000203 ***
Precipitation                  -2.617              0.008875 **
Wind_Speed                      2.792              0.005244 **
pub_traveltime_60             -10.949 < 0.0000000000000002 ***
Total_Pop                     -14.020 < 0.0000000000000002 ***
Median_income                  -1.828              0.067539 .
Median_age                    -27.600 < 0.0000000000000002 ***
upper_rent                     15.460 < 0.0000000000000002 ***
Pct_publictrans_towork         -8.729 < 0.0000000000000002 ***
pct_bachelor                    3.711              0.000206 ***
pct_white                     -19.878 < 0.0000000000000002 ***
pct_onemployment               19.395 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 544 on 946143 degrees of freedom
Multiple R-squared:  0.02317,   Adjusted R-squared:  0.02314
F-statistic: 897.7 on 25 and 946143 DF,  p-value: < 0.00000000000000022
```

OLS Regression First Model Error Matrix

MAE	MAPE
397.0962	1.155846

OLS Regression Second Model

```
OLS Regression Results Second Model
==========================================================
                                Dependent variable:
                            ---------------------------
                                    trip_duration
----------------------------------------------------------
store_and_fwd_flagY                    150.100***
                                        (7.863)
hour                                    1.003***
                                        (0.098)
WeekDay.L                              63.885***
                                        (1.491)
WeekDay.Q                             -92.189***
                                        (1.511)
WeekDay.C                             -35.410***
                                        (1.494)
WeekDay4                               7.887***
                                        (1.504)
WeekDay5                              -7.817***
                                        (1.495)
WeekDay6                               7.819***
                                        (1.488)
week                                   3.452***
                                        (0.130)
passenger_count                        5.551***
                                        (0.428)
weekendWeekend
time_of_dayMid-Day                      -1.906
                                        (1.786)
time_of_dayMorning Rush Hour          -70.296***
                                        (2.205)
time_of_dayOvernight                  -96.177***
                                        (1.531)
Temperature                            0.477***
                                        (0.059)
Precipitation                         -21.959**
                                        (9.994)
Wind_Speed                             0.401***
                                        (0.149)
hotel_nn3
wifi_nn3
subway_nn5
fdscrap_nn3
store_nn4
lagHour                                76.333***
                                        (29.629)
lag2Hours                               15.901
                                        (29.587)
lag3Hours                             -55.362*
                                        (31.596)
lag4Hours                               -8.329
                                        (31.972)
lag12Hours                              -1.430
                                        (37.346)
Constant                              719.052***
                                        (3.125)
----------------------------------------------------------
Observations                            946,169
R2                                        0.017
Adjusted R2                               0.017
Residual Std. Error                 546.229 (df = 946147)
F Statistic                      783.149*** (df = 21; 946147)
==========================================================
Note:                          *p<0.1; **p<0.05; ***p<0.01

Call:
lm(formula = trip_duration ~ store_and_fwd_flag + hour + WeekDay +
    week + passenger_count + weekend + time_of_day + Temperature +
    Precipitation + Wind_Speed + hotel_nn3 + wifi_nn3 + subway_nn5 +
    fdscrap_nn3 + store_nn4 + lagHour + lag2Hours + lag3Hours +
    lag4Hours + lag12Hours, data = trip_train)

Residuals:
    Min      1Q  Median      3Q     Max
-1048.4  -375.1  -128.3   232.9 10595.9
```

```
Coefficients: (6 not defined because of singularities)
                               Estimate Std. Error t value
(Intercept)                  719.05151    3.12532 230.073
store_and_fwd_flagY          150.09952    7.86331  19.089
hour                           1.00288    0.09799  10.235
WeekDay.L                     63.88483    1.49068  42.856
WeekDay.Q                    -92.18865    1.51061 -61.027
WeekDay.C                    -35.41024    1.49379 -23.705
WeekDay^4                      7.88719    1.50436   5.243
WeekDay^5                     -7.81693    1.49488  -5.229
WeekDay^6                      7.81926    1.48842   5.253
week                           3.45159    0.12959  26.636
passenger_count                5.55091    0.42817  12.964
weekendWeekend                      NA         NA      NA
time_of_dayMid-Day            -1.90627    1.78646  -1.067
time_of_dayMorning Rush Hour -70.29580    2.20522 -31.877
time_of_dayOvernight         -96.17730    1.53147 -62.800
Temperature                    0.47674    0.05913   8.062
Precipitation                -21.95948    9.99408  -2.197
Wind_Speed                     0.40054    0.14940   2.681
hotel_nn3                           NA         NA      NA
wifi_nn3                            NA         NA      NA
subway_nn5                          NA         NA      NA
fdscrap_nn3                         NA         NA      NA
store_nn4                           NA         NA      NA
lagHour                       76.33326   29.62950   2.576
lag2Hours                     15.90074   29.58660   0.537
lag3Hours                    -55.36167   31.59552  -1.752
lag4Hours                     -8.32880   31.97167  -0.261
lag12Hours                    -1.43044   37.34592  -0.038
                                Pr(>|t|)
(Intercept)                  < 0.0000000000000002 ***
store_and_fwd_flagY          < 0.0000000000000002 ***
hour                         < 0.0000000000000002 ***
WeekDay.L                    < 0.0000000000000002 ***
WeekDay.Q                    < 0.0000000000000002 ***
WeekDay.C                    < 0.0000000000000002 ***
WeekDay^4                    0.000000158111659185 ***
WeekDay^5                    0.000000170330922310 ***
WeekDay^6                    0.000000149343093983 ***
week                         < 0.0000000000000002 ***
passenger_count              < 0.0000000000000002 ***
weekendWeekend                                  NA
time_of_dayMid-Day                         0.28594
time_of_dayMorning Rush Hour < 0.0000000000000002 ***
time_of_dayOvernight         < 0.0000000000000002 ***
Temperature                  0.000000000000000749 ***
Precipitation                              0.02800 *
Wind_Speed                                 0.00734 **
hotel_nn3                                       NA
wifi_nn3                                        NA
subway_nn5                                      NA
fdscrap_nn3                                     NA
store_nn4                                       NA
lagHour                                    0.00999 **
lag2Hours                                  0.59097
lag3Hours                                  0.07974 .
lag4Hours                                  0.79447
lag12Hours                                 0.96945
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 546.2 on 946147 degrees of freedom
Multiple R-squared:  0.01709,   Adjusted R-squared:  0.01706
F-statistic: 783.1 on 21 and 946147 DF,  p-value: < 0.00000000000000022
```

OLS Regression Second Model Error Matrix

MAE_OLSreg2	MAPE_OLSreg2
398.4847	1.156887

OLS Regression Third Model

OLS Regression Results Third Model

	Dependent variable:
	trip_duration
store_and_fwd_flagY	147.745***
	(7.863)
vendor_id	-0.735
	(1.177)
passenger_count	5.559***
	(0.446)
hour	1.078***
	(0.098)
WeekDay.L	65.035***
	(1.487)
WeekDay.Q	-92.163***
	(1.509)
WeekDay.C	-35.751***
	(1.490)
WeekDay4	6.274***
	(1.500)
WeekDay5	-7.934***
	(1.491)
WeekDay6	9.556***
	(1.483)
weekendWeekend	
time_of_dayMid-Day	6.741***
	(1.763)
time_of_dayMorning Rush Hour	-57.927***
	(2.179)
time_of_dayOvernight	-103.081***
	(1.535)
hotel_nn3	
wifi_nn3	
subway_nn5	
fdscrap_nn3	
pub_traveltime_60	-0.054***
	(0.005)
Total_Pop	-0.003***
	(0.0003)
Median_income	-0.0001***
	(0.00002)
Median_age	-2.930***
	(0.107)
upper_rent	0.025***
	(0.002)
Pct_publictrans_towork	-45.211***
	(4.929)
pct_bachelor	15.810**
	(7.908)
pct_white	-123.127***
	(6.188)
pct_onemployment	473.361***
	(26.160)
lag2Hours	12.301
	(29.507)
lag3Hours	-55.309*
	(31.510)
lag4Hours	-10.680
	(31.885)
Temperature	1.721***
	(0.036)
Precipitation	-30.428***
	(9.961)
Wind_Speed	0.261*
	(0.149)
Constant	857.135***
	(8.431)
Observations	946,169
R2	0.022
Adjusted R2	0.022
Residual Std. Error	544.748 (df = 946140)
F Statistic	774.821*** (df = 28; 946140)
Note:	*p<0.1; **p<0.05; ***p<0.01

```
Call:
lm(formula = trip_duration ~ store_and_fwd_flag + vendor_id +
    passenger_count + hour + WeekDay + weekend + time_of_day +
    hotel_nn3 + wifi_nn3 + subway_nn5 + fdscrap_nn3 + pub_traveltime_60 +
    Total_Pop + Median_income + Median_age + upper_rent + Pct_publictrans_towor
    pct_bachelor + pct_white + pct_onemployment + lag2Hours +
    lag3Hours + lag4Hours + Temperature + Precipitation + Wind_Speed,
    data = trip_train)

Residuals:
    Min      1Q  Median      3Q     Max
-1097.2  -373.5  -128.8   230.8 10645.8

Coefficients: (1 not defined because of singularities)
                               Estimate    Std. Error
(Intercept)                861.26811082    8.29426154
store_and_fwd_flagY        142.96010379    7.84282118
hour                         1.38670312    0.09803405
WeekDay.L                   63.91219958    1.48508513
WeekDay.Q                  -92.66318309    1.50964014
WeekDay.C                  -36.65106620    1.48832214
WeekDay^4                    9.46253313    1.49942854
WeekDay^5                   -7.16322602    1.48909903
WeekDay^6                    9.42485290    1.48202757
week                         3.46763544    0.12905642
passenger_count              5.40961298    0.42642581
weekendWeekend                       NA            NA
time_of_dayMid-Day          -0.95226004    1.78147782
time_of_dayMorning Rush Hour -65.80182270    2.19791918
time_of_dayOvernight        -106.01487086    1.53509332
Temperature                  0.46778284    0.05891861
Precipitation              -26.03530597    9.94911711
Wind_Speed                   0.41684192    0.14931549
pub_traveltime_60           -0.05133054    0.00468803
Total_Pop                   -0.00354283    0.00025269
Median_income               -0.00003950    0.00002161
Median_age                  -2.95551320    0.10708200
upper_rent                   0.02523414    0.00163221
Pct_publictrans_towork     -42.97288629    4.92295679
pct_bachelor                29.31467948    7.89909296
pct_white                 -122.72351237    6.17388543
pct_onemployment           505.94605777   26.08642764
                           t value             Pr(>|t|)
(Intercept)                103.839 < 0.0000000000000002 ***
store_and_fwd_flagY         18.228 < 0.0000000000000002 ***
hour                        14.145 < 0.0000000000000002 ***
WeekDay.L                   43.036 < 0.0000000000000002 ***
WeekDay.Q                  -61.381 < 0.0000000000000002 ***
WeekDay.C                  -24.626 < 0.0000000000000002 ***
WeekDay^4                    6.311  0.00000000027779131 ***
WeekDay^5                   -4.810  0.00000150619270854 ***
WeekDay^6                    6.359  0.00000000020259358 ***
week                        26.869 < 0.0000000000000002 ***
passenger_count             12.686 < 0.0000000000000002 ***
weekendWeekend                  NA                   NA
time_of_dayMid-Day          -0.535             0.592972
time_of_dayMorning Rush Hour -29.938 < 0.0000000000000002 ***
time_of_dayOvernight        -69.061 < 0.0000000000000002 ***
Temperature                   7.939  0.00000000000000203 ***
Precipitation                -2.617             0.008875 **
Wind_Speed                    2.792             0.005244 **
pub_traveltime_60           -10.949 < 0.0000000000000002 ***
Total_Pop                   -14.020 < 0.0000000000000002 ***
Median_income                -1.828             0.067539 .
Median_age                  -27.600 < 0.0000000000000002 ***
upper_rent                   15.460 < 0.0000000000000002 ***
Pct_publictrans_towork       -8.729 < 0.0000000000000002 ***
pct_bachelor                  3.711             0.000206 ***
pct_white                   -19.878 < 0.0000000000000002 ***
pct_onemployment             19.395 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 544 on 946143 degrees of freedom
Multiple R-squared:  0.02317,   Adjusted R-squared:  0.02314
F-statistic: 897.7 on 25 and 946143 DF,  p-value: < 0.00000000000000022
```

OLS Regression Third Model Error Matrix

MAE_OLSreg3	MAPE_OLSreg3
397.2045	1.156175

Random forest model is a kind of tree model which consists of many decisions trees. Because the number of trees require large computation system, I created subset of 20000 observations from overall taxi trip record, and made three models, with ntree=50, selected features = 20. With different feature selection, the results of three model is as follows:

Random Forest Regression First Model

```
Call:
 randomForest(formula = trip_duration ~ store_and_fwd_flag + hour +      WeekDay + week +
passenger_count + weekend + time_of_day +      Temperature + Precipitation + Wind_Speed +
pub_traveltime_60 +      Total_Pop + Median_income + Median_age + upper_rent +
Pct_publictrans_towork +      pct_bachelor + pct_white + pct_onemployment, data = trip_train,
ntree = 50, mtry = 20, bootstrap = T, na.action = na.exclude)
               Type of random forest: regression
                     Number of trees: 50
No. of variables tried at each split: 19

          Mean of squared residuals: 317138.2
                    % Var explained: -5.68
```

Random Forest Error Metrics - second model

rf2_MAE	rf2_MAPE
410.2859	1.134579

	IncNodePurity
store_and_fwd_flag	41604411
hour	907734119
WeekDay	501286351
week	1226709268
passenger_count	585581472
weekend	98158121
time_of_day	167845078
Temperature	1617087658
Precipitation	223718204
Wind_Speed	1011911182
pub_traveltime_60	376161291
Total_Pop	404113896
Median_income	385465120
Median_age	415088331
upper_rent	381371984
Pct_publictrans_towork	467944091
pct_bachelor	375854346
pct_white	469970092
pct_onemployment	418307605

Random Forest Regression Second Model

```
Call:
 randomForest(formula = trip_duration ~ store_and_fwd_flag + hour +      WeekDay + week +
passenger_count + weekend + time_of_day +      Temperature + Precipitation + Wind_Speed + hotel_nn3
+ wifi_nn3 +      subway_nn5 + fdscrap_nn3 + store_nn4 + lagHour + lag2Hours +      lag3Hours +
lag4Hours + lag12Hours, data = trip_train, ntree = 50,      mtry = 20, bootstrap = T, na.action =
na.exclude)
               Type of random forest: regression
                     Number of trees: 50
No. of variables tried at each split: 20

          Mean of squared residuals: 317884.6
                    % Var explained: -5.93
```

Random Forest Error Metrics - third model

rf3_MAE	rf3_MAPE
407.5295	1.125704

	IncNodePurity
store_and_fwd_flag	35699702.1
hour	702667734.9
WeekDay	416589009.1
week	897790384.0
passenger_count	504835957.9
weekend	95604468.0
time_of_day	156168998.3
Temperature	1240464245.4
Precipitation	185296243.6
Wind_Speed	767299222.3
hotel_nn3	1254149605.5
wifi_nn3	935660550.4
subway_nn5	972613026.9
fdscrap_nn3	1125256125.8
store_nn4	997958602.8
lagHour	1425248.0
lag2Hours	1917475.5
lag3Hours	2323923.0
lag4Hours	342477.4
lag12Hours	1283172.6

Random Forest Regression Third Model

```
Call:
 randomForest(formula = trip_duration ~ store_and_fwd_flag + vendor_id +      passenger_count +
hour + WeekDay + weekend + time_of_day +      hotel_nn3 + wifi_nn3 + subway_nn5 + fdscrap_nn3 +
pub_traveltime_60 +      Total_Pop + Median_income + Median_age + upper_rent +
Pct_publictrans_towork +      pct_bachelor + pct_white + pct_onemployment + lag2Hours +
lag3Hours + lag4Hours + Temperature + Precipitation + Wind_Speed,      data = trip_train, ntree =
50, mtry = 20, bootstrap = T,      na.action = na.exclude)
               Type of random forest: regression
                     Number of trees: 50
No. of variables tried at each split: 20

          Mean of squared residuals: 313691.8
                    % Var explained: -4.54
```

Random Forest Error Metrics - third model

rf3_MAE	rf3_MAPE
407.5295	1.125704

	IncNodePurity
store_and_fwd_flag	38949720.4
vendor_id	203389704.4
passenger_count	440552242.6
hour	761797943.5
WeekDay	466229133.1
weekend	99557500.7
time_of_day	179521251.7
hotel_nn3	948360306.2
wifi_nn3	704123572.1
subway_nn5	764409889.6
fdscrap_nn3	751605209.7
pub_traveltime_60	264049344.8
Total_Pop	288558978.1
Median_income	255630854.5
Median_age	265471265.1
upper_rent	234757229.4
Pct_publictrans_towork	287062169.9
pct_bachelor	255335940.3
pct_white	269238759.0
pct_onemployment	272115591.7
lag2Hours	2015323.6
lag3Hours	1878279.2
lag4Hours	271382.3
Temperature	1461968203.9
Precipitation	211798887.6

Poisson Regression Model Performance

Poisson Regression First Model

Poisson Regression Results

	Dependent variable:
	trip_duration
store_and_fwd_flagY	0.131*** (0.003)
hour	0.002*** (0.00003)
WeekDay.L	0.089*** (0.001)
WeekDay.Q	-0.134*** (0.001)
WeekDay.C	-0.027*** (0.001)
WeekDay4	0.040*** (0.001)
WeekDay5	-0.001** (0.0005)
WeekDay6	0.005*** (0.0005)
week	0.005*** (0.00004)
passenger_count	0.011*** (0.0001)
weekendWeekend	
time_of_dayMid-Day	-0.008*** (0.001)
time_of_dayMorning Rush Hour	-0.078*** (0.001)
time_of_dayOvernight	-0.144*** (0.001)
Temperature	-0.0001*** (0.00002)
Precipitation	0.060*** (0.003)
Wind_Speed	0.001*** (0.00005)
hotel_nn3	57.186*** (0.227)
wifi_nn3	-56.223*** (0.234)
subway_nn5	
fdscrap_nn3	
store_nn4	
lagHour	-0.082*** (0.010)
lag2Hours	0.139*** (0.012)
lag3Hours	0.248*** (0.012)
lag4Hours	-0.092*** (0.010)
lag12Hours	-0.248*** (0.012)
Constant	-1,145,074.000*** (3,066.342)
Observations	37,497
Log Likelihood	-6,256,550.000
Akaike Inf. Crit.	12,513,148.000
Note:	*p<0.1; **p<0.05; ***p<0.01

Poisson Regression Second Model

Poisson Regression Results - second model

	Dependent variable:
	trip_duration
store_and_fwd_flagY	0.131*** (0.003)
hour	0.002*** (0.00003)
WeekDay.L	0.089*** (0.001)
WeekDay.Q	-0.134*** (0.001)
WeekDay.C	-0.027*** (0.001)
WeekDay4	0.040*** (0.001)
WeekDay5	-0.001** (0.0005)
WeekDay6	0.005*** (0.0005)
week	0.005*** (0.00004)
passenger_count	0.011*** (0.0001)
weekendWeekend	
time_of_dayMid-Day	-0.008*** (0.001)
time_of_dayMorning Rush Hour	-0.078*** (0.001)
time_of_dayOvernight	-0.144*** (0.001)
Temperature	-0.0001*** (0.00002)
Precipitation	0.060*** (0.003)
Wind_Speed	0.001*** (0.00005)
hotel_nn3	57.186*** (0.227)
wifi_nn3	-56.223*** (0.234)
subway_nn5	
fdscrap_nn3	
store_nn4	
lagHour	-0.082*** (0.010)
lag2Hours	0.139*** (0.012)
lag3Hours	0.248*** (0.012)
lag4Hours	-0.092*** (0.010)
lag12Hours	-0.248*** (0.012)
Constant	-1,145,074.000*** (3,066.342)
Observations	37,497
Log Likelihood	-6,256,550.000
Akaike Inf. Crit.	12,513,148.000
Note:	*p<0.1; **p<0.05; ***p<0.01

Poisson Regression third Model

Poisson Regression Results - third model

	Dependent variable:
	trip_duration
store_and_fwd_flagY	0.127*** (0.003)
vendor_id	-0.003*** (0.0004)
passenger_count	0.012*** (0.0001)
hour	0.002*** (0.00003)
WeekDay.L	0.090*** (0.001)
WeekDay.Q	-0.132*** (0.001)
WeekDay.C	-0.028*** (0.001)
WeekDay4	0.038*** (0.001)
WeekDay5	-0.001** (0.0005)
WeekDay6	0.008*** (0.0005)
weekendWeekend	
time_of_dayMid-Day	0.004*** (0.001)
time_of_dayMorning Rush Hour	-0.063*** (0.001)
time_of_dayOvernight	-0.145*** (0.001)
hotel_nn3	50.526*** (0.245)
wifi_nn3	-49.578*** (0.252)
subway_nn5	
fdscrap_nn3	
pub_traveltime_60	0.00004*** (0.00000)
Total_Pop	-0.00001*** (0.00000)
Median_income	-0.00000*** (0.000)
Median_age	-0.0003*** (0.00004)
upper_rent	0.00001*** (0.00000)
Pct_publictrans_towork	-0.017*** (0.002)
pct_bachelor	0.103*** (0.003)
pct_white	-0.187*** (0.002)
pct_onemployment	0.364*** (0.009)
lag2Hours	0.134*** (0.012)
lag3Hours	0.234*** (0.012)
lag4Hours	-0.082*** (0.010)
Temperature	0.002*** (0.00001)
Precipitation	0.046*** (0.003)
Wind_Speed	0.001*** (0.00005)
Constant	-1,039,679.000*** (3,473.902)
Observations	37,497
Log Likelihood	-6,244,383.000
Akaike Inf. Crit.	12,488,828.000
Note:	*p<0.1; **p<0.05; ***p<0.01

Poisson Error Metrics - first model

poisson_MAE	poisson_MAPE
761.5915	0.9847421

Poisson Error Metrics - second model

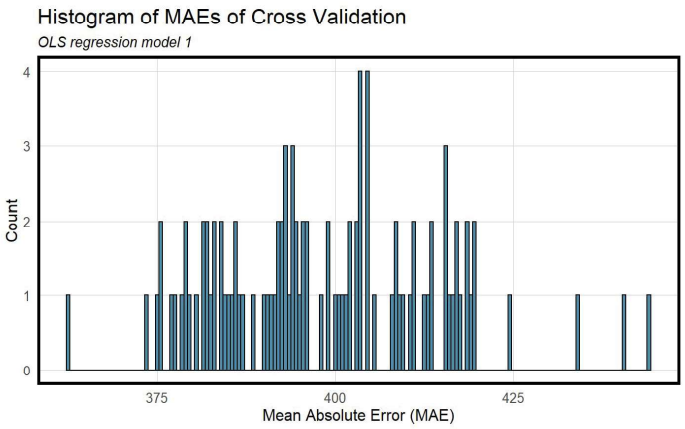
poisson_MAE2	poisson_MAPE2
761.5915	0.9847421

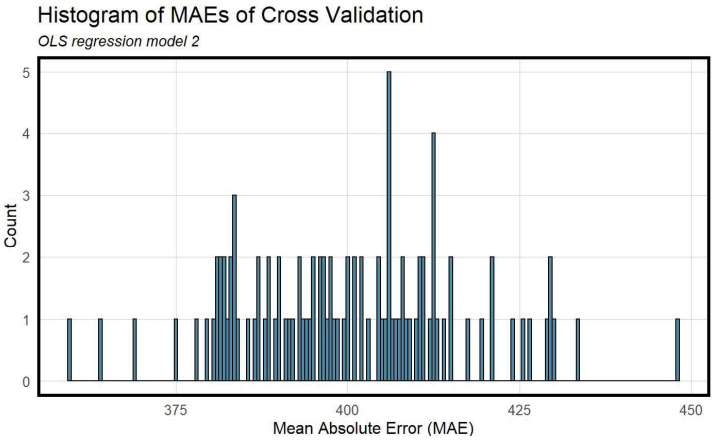
Poisson Error Metrics - third model

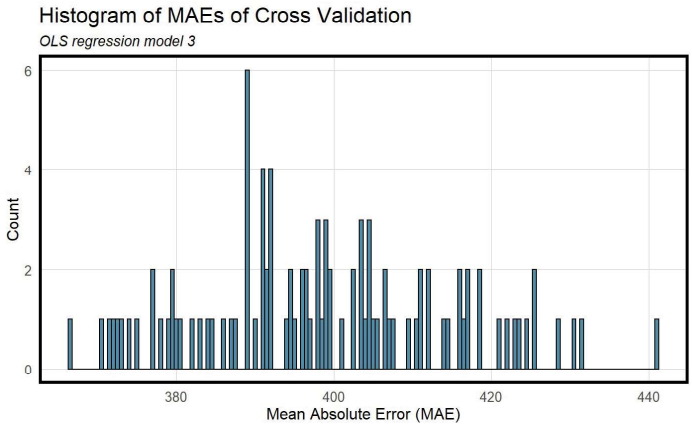
poisson_MAE3	poisson_MAPE3
761.5924	0.9847496

Cross Validation of Models.

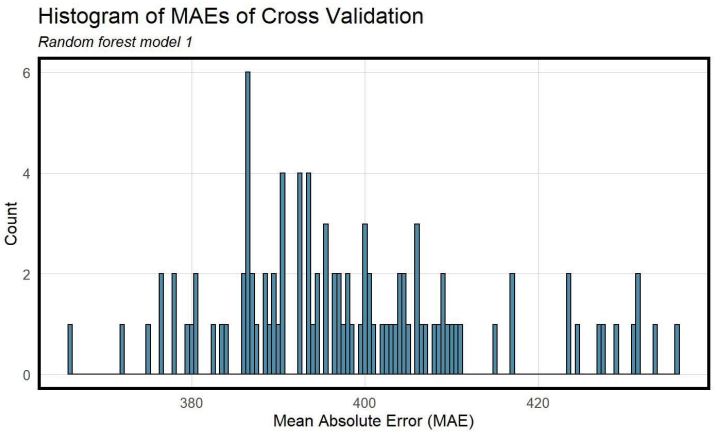
OLS Regression Model Cross Validation

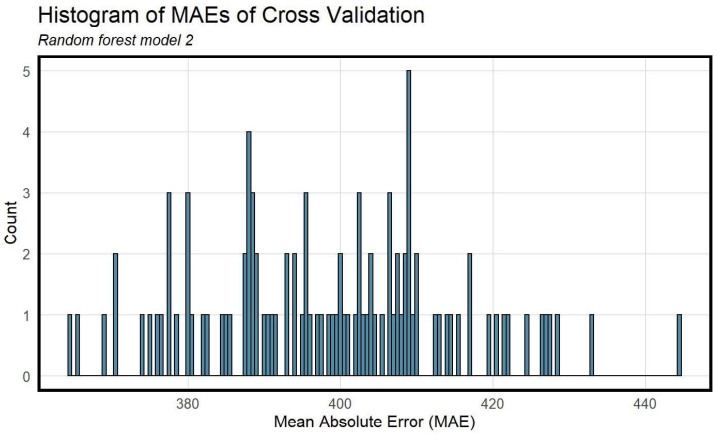


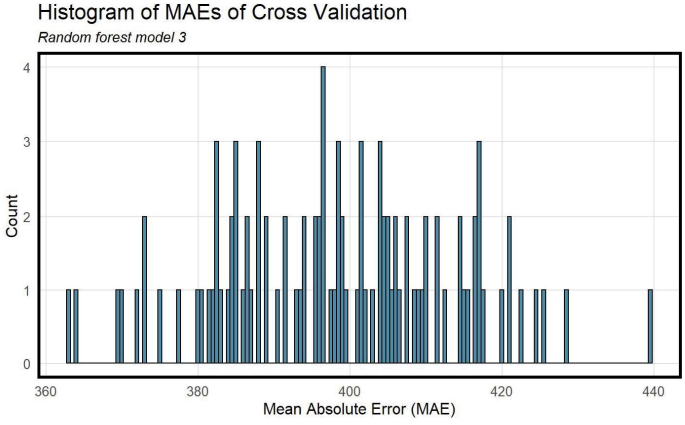




Random forest Regression Model Cross Validation







Discussion

When applying three set of models, the performance shows features selection across three models doesn't show much great difference. But the model with more complex features selection tends to show better performance. However, the MAE across three models are all nearly double the real trip duration, which means all the model doesn't perform well in predicting time consumption.

When conducting cross validation, it is shown that no matter how to suffle the sample dataset, the average value of error in predicting time consumption of each trip is about 400 seconds. The underlying reason might be firstly, in order to run successfully the model, I created subset of data with 100000 observations, although it's still a large dataset, the subset sample only takes 6% observasions totally, which actually tells little about the total data. For future analysis, it will be much helpful to conduct the same process with python, with the help of packages such as "datashader". Also, it is surprising that the poisson model acts best, since trip time consumption is not really a counted based feature, but the model shows least MAPE overall.

II. For Trip Demand

With the same feature selection set, I conducted OLS regression, Poisson regression and Random forest regression on the unit of census tract. Firslty I sum up the trip within each census tract and created a new column called "totalCount". After building the overall panel other features together, the final panel with regard to both origin tract and the trip totalcount contains overall 1261561 observasions. Also, in order to make the model run smoothly, I created subset sample of 100000 and to see the trip count prediction performance of each models.

OLS Regression First Model

```
Coefficients: (1 not defined because of singularities)
                              Estimate    Std. Error t value             Pr(>|t|)
(Intercept)                 4259.4703450   301.8667007  14.110 < 0.0000000000000002 ***
store_and_fwd_flagY         -516.5574399   276.0200962  -1.871              0.06129 .
hour                          16.0499699     3.5572811   4.512          0.00000644 ***
WeekDay.L                   -148.1512854    53.9269050  -2.747              0.00601 **
WeekDay.Q                   -129.0189848    54.8364737  -2.353              0.01864 *
WeekDay.C                     37.3434474    53.9885858   0.692              0.48913
WeekDay^4                   -134.7481932    54.3183126  -2.481              0.01311 *
WeekDay^5                    130.3775285    53.9383394   2.417              0.01564 *
WeekDay^6                      4.2943774    53.6713513   0.080              0.93623
week                          -0.8835094     4.6908986  -0.188              0.85061
passenger_count               -3.6302981    15.4856854  -0.234              0.81465
weekendWeekend                        NA           NA      NA                   NA
time_of_dayMid-Day           142.7570300    64.7979923   2.203              0.02759 *
time_of_dayMorning Rush Hour 181.5753815    79.7998446   2.275              0.02289 *
time_of_dayOvernight          62.3307650    55.7421045   1.118              0.26349
Temperature                    0.7322222     2.1418799   0.342              0.73246
Precipitation               -501.5387393   379.0737224  -1.323              0.18582
Wind_Speed                    -2.9737996     5.3247582  -0.558              0.57651
pub_traveltime_60             -2.1650595     0.1693671 -12.783 < 0.0000000000000002 ***
Total_Pop                     -0.1590937     0.0091378 -17.410 < 0.0000000000000002 ***
Median_income                  0.0132637     0.0007832  16.934 < 0.0000000000000002 ***
Median_age                   -90.2662117     3.8859941 -23.229 < 0.0000000000000002 ***
upper_rent                     3.2684688     0.0592098  55.201 < 0.0000000000000002 ***
Pct_publictrans_towork     -12775.9317656   178.4462371 -71.595 < 0.0000000000000002 ***
pct_bachelor                23882.8982112   287.1759061  83.165 < 0.0000000000000002 ***
pct_white                  -15428.8784672   225.0292011 -68.564 < 0.0000000000000002 ***
pct_onemployment            41764.0461679   949.5688475  43.982 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5550 on 74818 degrees of freedom
Multiple R-squared:  0.3711,    Adjusted R-squared:  0.3709
```

OLS Error Metrics

MAE_OLSreg1	MAPE_OLSreg1
3906.504	2.475058

```
OLS Regression Results First Model
==============================================================
                                        Dependent variable:
                                    ---------------------------
                                            totalCount
--------------------------------------------------------------
store_and_fwd_flagY                          -516.557*
                                             (276.020)
hour                                         16.050***
                                              (3.557)
WeekDay.L                                   -148.151***
                                             (53.927)
WeekDay.Q                                   -129.019**
                                             (54.836)
WeekDay.C                                     37.343
                                             (53.989)
WeekDay4                                    -134.748**
                                             (54.318)
WeekDay5                                    130.378**
                                             (53.938)
WeekDay6                                      4.294
                                             (53.671)
week                                          -0.884
                                              (4.691)
passenger_count                               -3.630
                                             (15.486)
weekendWeekend
time_of_dayMid-Day                          142.757**
                                             (64.798)
time_of_dayMorning Rush Hour                181.575**
                                             (79.800)
time_of_dayOvernight                          62.331
                                             (55.742)
Temperature                                    0.732
                                              (2.142)
Precipitation                               -501.539
                                             (379.074)
Wind_Speed                                    -2.974
                                              (5.325)
pub_traveltime_60                           -2.165***
                                              (0.169)
Total_Pop                                   -0.159***
                                              (0.009)
Median_income                                0.013***
                                              (0.001)
Median_age                                  -90.266***
                                              (3.886)
upper_rent                                   3.268***
                                              (0.059)
Pct_publictrans_towork                     -12,775.930***
                                             (178.446)
pct_bachelor                                23,882.900***
                                             (287.176)
pct_white                                  -15,428.880***
                                             (225.029)
pct_onemployment                            41,764.050***
                                             (949.569)
Constant                                    4,259.470***
                                             (301.867)
--------------------------------------------------------------
Observations                                    74,844
R2                                               0.371
Adjusted R2                                      0.371
Residual Std. Error                    5,550.316 (df = 74818)
F Statistic                          1,765.704*** (df = 25; 74818)
==============================================================
Note:                                *p<0.1; **p<0.05; ***p<0.01

Call:
lm(formula = totalCount ~ store_and_fwd_flag + hour + WeekDay +
    week + passenger_count + weekend + time_of_day + Temperature +
    Precipitation + Wind_Speed + pub_traveltime_60 + Total_Pop +
    Median_income + Median_age + upper_rent + Pct_publictrans_towork +
    pct_bachelor + pct_white + pct_onemployment, data = trip_train)

Residuals:
     Min       1Q   Median       3Q      Max
-16257.7  -3408.7   -171.3   2252.5  20373.7
```

OLS Regression Second Model

```
Coefficients: (6 not defined because of singularities)
                             Estimate Std. Error t value             Pr(>|t|)
(Intercept)                8958.4841   142.1219  63.034 < 0.0000000000000002 ***
store_and_fwd_flagY        -527.6401   347.3571  -1.519              0.12876
hour                         47.5232     4.4518  10.675 < 0.0000000000000002 ***
WeekDay.L                  -264.9681    67.8568  -3.905    0.000094380727168 ***
WeekDay.Q                  -511.7743    68.8180  -7.437    0.000000000000104 ***
WeekDay.C                   210.6626    67.9268   3.101              0.00193 **
WeekDay^4                  -289.6595    68.3393  -4.239    0.000022523511298 ***
WeekDay^5                   210.7778    67.8765   3.105              0.00190 **
WeekDay^6                    -8.9994    67.5364  -0.133              0.89399
week                          1.9018     5.9029   0.322              0.74732
passenger_count               7.8689    19.4864   0.404              0.68635
weekendWeekend                    NA         NA      NA                   NA
time_of_dayMid-Day          420.3617    81.4867   5.159    0.000000249365742 ***
time_of_dayMorning Rush Hour 30.6238   100.2981   0.305              0.76012
time_of_dayOvernight        122.0999    69.7608   1.750              0.08008 .
Temperature                  -0.8391     2.6953  -0.311              0.75554
Precipitation              -233.0156   477.0243  -0.488              0.62521
Wind_Speed                   -4.1722     6.7007  -0.623              0.53352
hotel_nn3                         NA         NA      NA                   NA
wifi_nn3                          NA         NA      NA                   NA
subway_nn5                        NA         NA      NA                   NA
fdscrap_nn3                       NA         NA      NA                   NA
store_nn4                         NA         NA      NA                   NA
lagHour                    -206.7426  1370.2680  -0.151              0.88007
lag2Hours                  1445.6378  1320.3837   1.095              0.27358
lag3Hours                 -1082.7637  1489.5916  -0.727              0.46730
lag4Hours                  2255.6212  1602.9559   1.407              0.15938
lag12Hours                 2368.4354  1489.6099   1.590              0.11185
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6985 on 74822 degrees of freedom
Multiple R-squared:  0.003922,  Adjusted R-squared:  0.003643
F-statistic: 14.03 on 21 and 74822 DF,  p-value: < 0.00000000000000022
```

OLS Error Metrics

MAE_OLSreg2	MAPE_OLSreg2
4473.355	4.446493

```
OLS Regression Results Second Model
==========================================================
                                        Dependent variable:
                                    ----------------------
                                            totalCount
----------------------------------------------------------
store_and_fwd_flagY                          -527.640
                                             (347.357)
hour                                        47.523***
                                             (4.452)
WeekDay.L                                   -264.968***
                                             (67.857)
WeekDay.Q                                   -511.774***
                                             (68.818)
WeekDay.C                                   210.663***
                                             (67.927)
WeekDay4                                    -289.660***
                                             (68.339)
WeekDay5                                    210.778***
                                             (67.876)
WeekDay6                                      -8.999
                                             (67.536)
week                                          1.902
                                              (5.903)
passenger_count                               7.869
                                             (19.486)
weekendWeekend
time_of_dayMid-Day                          420.362***
                                             (81.487)
time_of_dayMorning Rush Hour                 30.624
                                            (100.298)
time_of_dayOvernight                        122.100*
                                             (69.761)
Temperature                                   -0.839
                                              (2.695)
Precipitation                               -233.016
                                             (477.024)
Wind_Speed                                    -4.172
                                              (6.701)
hotel_nn3
wifi_nn3
subway_nn5
fdscrap_nn3
store_nn4
lagHour                                     -206.743
                                            (1,370.268)
lag2Hours                                   1,445.638
                                            (1,320.384)
lag3Hours                                   -1,082.764
                                            (1,489.592)
lag4Hours                                   2,255.621
                                            (1,602.956)
lag12Hours                                  2,368.435
                                            (1,489.610)
Constant                                   8,958.484***
                                             (142.122)
----------------------------------------------------------
Observations                                  74,844
R2                                             0.004
Adjusted R2                                    0.004
Residual Std. Error                 6,984.756 (df = 74822)
F Statistic                        14.030*** (df = 21; 74822)
==========================================================
Note:                          *p<0.1; **p<0.05; ***p<0.01

Call:
lm(formula = totalCount ~ store_and_fwd_flag + hour + WeekDay +
    week + passenger_count + weekend + time_of_day + Temperature +
    Precipitation + Wind_Speed + hotel_nn3 + wifi_nn3 + subway_nn +
    fdscrap_nn3 + store_nn4 + lagHour + lag2Hours + lag3Hours +
    lag4Hours + lag12Hours, data = trip_train)

Residuals:
   Min     1Q Median     3Q    Max
-10578  -3812  -1183   2092  29748
```

OLS Regression Third Model

```
Call:
lm(formula = totalCount ~ store_and_fwd_flag + vendor_id + passenger_count +
    hour + WeekDay + weekend + time_of_day + hotel_nn3 + wifi_nn3 +
    subway_nn5 + fdscrap_nn3 + pub_traveltime_60 + Total_Pop +
    Median_income + Median_age + upper_rent + Pct_publictrans_towork +
    pct_bachelor + pct_white + pct_onemployment + lag2Hours +
    lag3Hours + lag4Hours + Temperature + Precipitation + Wind_Speed,
    data = trip_train)

Residuals:
     Min       1Q   Median       3Q      Max
-16306.2  -3408.9   -164.2   2251.6  20374.0

Coefficients: (5 not defined because of singularities)
                              Estimate     Std. Error t value             Pr(>|t|)
(Intercept)                4416.7696236    307.1503761  14.380 < 0.0000000000000002 ***
store_and_fwd_flagY           -567.3081482    276.7858173  -2.050              0.04040 *
vendor_id                     -102.2618102     42.6345205  -2.399              0.01646 *
passenger_count                  7.5549661     16.1790338   0.467              0.64053
hour                            16.1135855      3.5374090   4.555          0.00000524 ***
WeekDay.L                     -147.2646428     53.9264504  -2.731              0.00632 **
WeekDay.Q                     -130.6711980     54.7542007  -2.387              0.01701 *
WeekDay.C                       37.6260854     53.9842476   0.697              0.48582
WeekDay^4                     -134.4846954     54.2950609  -2.477              0.01325 *
WeekDay^5                      130.3659276     53.9339687   2.417              0.01565 *
WeekDay^6                        3.3719339     53.6444036   0.063              0.94988
weekendWeekend                          NA            NA      NA                   NA
time_of_dayMid-Day            140.7576421     64.0791101   2.197              0.02805 *
time_of_dayMorning Rush Hour  179.8016883     78.9229627   2.278              0.02272 *
time_of_dayOvernight           61.8707413     55.6772335   1.111              0.26647
hotel_nn3                               NA            NA      NA                   NA
wifi_nn3                                NA            NA      NA                   NA
subway_nn5                              NA            NA      NA                   NA
fdscrap_nn3                             NA            NA      NA                   NA
pub_traveltime_60              -2.1632391      0.1693593 -12.773 < 0.0000000000000002 ***
Total_Pop                      -0.1589838      0.0091375 -17.399 < 0.0000000000000002 ***
Median_income                   0.0132475      0.0007832  16.914 < 0.0000000000000002 ***
Median_age                    -90.3221608      3.8858541 -23.244 < 0.0000000000000002 ***
upper_rent                      3.2680339      0.0592108  55.193 < 0.0000000000000002 ***
Pct_publictrans_towork       -12780.5223995    178.4432539 -71.622 < 0.0000000000000002 ***
pct_bachelor                  23881.6510882    287.1685990  83.162 < 0.0000000000000002 ***
pct_white                    -15432.6727909    225.0291862 -68.581 < 0.0000000000000002 ***
pct_onemployment             41734.4458412    949.5633874  43.951 < 0.0000000000000002 ***
lag2Hours                      477.4177907   1049.2822052   0.455              0.64911
lag3Hours                    -2323.6092074   1183.7138556  -1.963              0.04965 *
lag4Hours                     1039.3443770   1273.7621257   0.816              0.41452
Temperature                      0.3717748      1.3004821   0.286              0.77497
Precipitation                 -503.5339947    378.8633207  -1.329              0.18383
Wind_Speed                      -3.0352877      5.3231197  -0.570              0.56854
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5550 on 74815 degrees of freedom
Multiple R-squared:  0.3712,    Adjusted R-squared:  0.3709
F-statistic:  1577 on 28 and 74815 DF,  p-value: < 0.00000000000000022
```

OLS Error Metrics

MAE_OLSreg3	MAPE_OLSreg3
3906.499	2.474674

OLS Regression Results Third Model

	Dependent variable:
	totalCount
store_and_fwd_flagY	-567.308**
	(276.786)
vendor_id	-102.262**
	(42.635)
passenger_count	7.555
	(16.179)
hour	16.114***
	(3.537)
WeekDay.L	-147.265***
	(53.926)
WeekDay.Q	-130.671**
	(54.754)
WeekDay.C	37.626
	(53.984)
WeekDay4	-134.485**
	(54.295)
WeekDay5	130.366**
	(53.934)
WeekDay6	3.372
	(53.644)
weekendWeekend	
time_of_dayMid-Day	140.758**
	(64.079)
time_of_dayMorning Rush Hour	179.802**
	(78.923)
time_of_dayOvernight	61.871
	(55.677)
hotel_nn3	
wifi_nn3	
subway_nn5	
fdscrap_nn3	
pub_traveltime_60	-2.163***
	(0.169)
Total_Pop	-0.159***
	(0.009)
Median_income	0.013***
	(0.001)
Median_age	-90.322***
	(3.886)
upper_rent	3.268***
	(0.059)
Pct_publictrans_towork	-12,780.520***
	(178.443)
pct_bachelor	23,881.650***
	(287.169)
pct_white	-15,432.670***
	(225.029)
pct_onemployment	41,734.450***
	(949.563)
lag2Hours	477.418
	(1,049.282)
lag3Hours	-2,323.609**
	(1,183.714)
lag4Hours	1,039.344
	(1,273.762)
Temperature	0.372
	(1.300)
Precipitation	-503.534
	(378.863)
Wind_Speed	-3.035
	(5.323)
Constant	4,416.770***
	(307.150)
Observations	74,844
R2	0.371
Adjusted R2	0.371
Residual Std. Error	5,550.039 (df = 74815)
F Statistic	1,577.053*** (df = 28; 74815)
Note:	*p<0.1; **p<0.05; ***p<0.01

Discussion

Learning from the plots and summaries, the prediction doesn't show something great, but when I see through the prediction table and make comparison to the totalcount and predicted total count, it is obvious that most observations shows a great result, but there are bunch of outliers that make influence on the prediction. for example, most prediction output in test datset has MAPE of 0.5, but some observasions shows MAPE over 20! This super huge error put great impact on the final results!