“New York City Taxi Trip Duration” Notebooks

2019. 10. 6. (Sat) Written by 이승진

1. Introduction
   1. Libraries & Functions

Documents

Pandas

Numpy

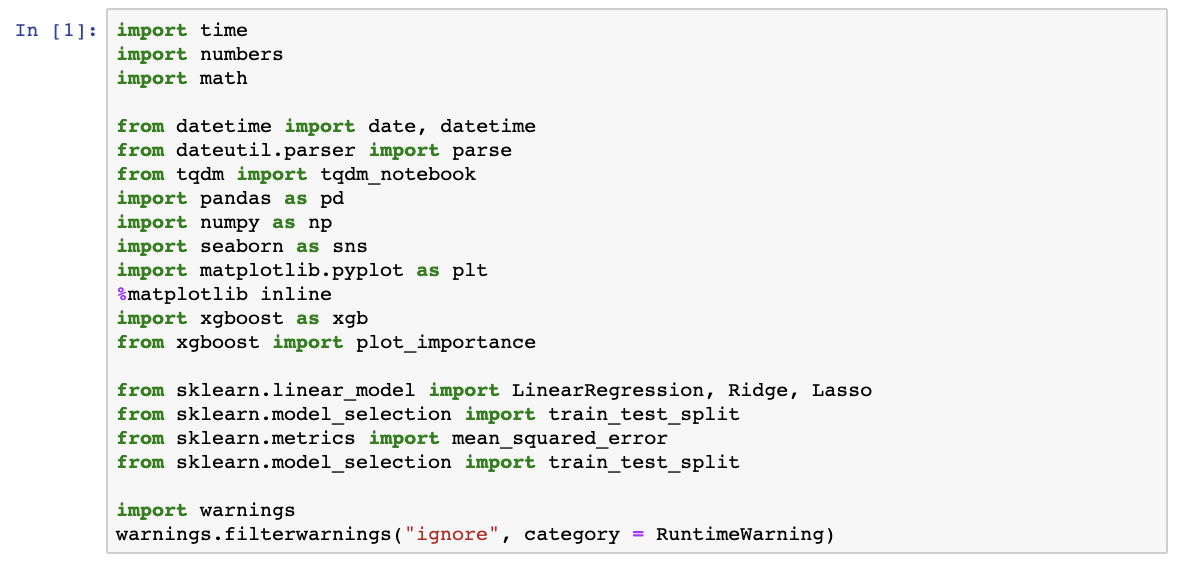
Seaborn

Matplotlib

Xgboost

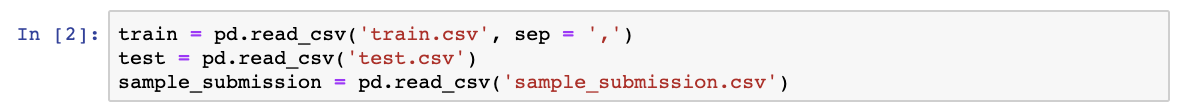
Scikit-learn (sklearn)

I used scikit-learn for machine learning operation like train-test-split, Lasso, Ridge, LinearRegression and etc. Numpy is the fundamental package for scientific computation. XGBoost is the classification (also Regression) algorithm used to make the final predictions. Seaborn and matplotlib is a nice tool for data visualization built on top of matplotlib. The import code is as follows :



* 1. Loading the Data

Load the train, test data using Pandas ‘read\_csv’ function.



1. Data Preparation

Let’s look about which column does dataset have. First, using pickup\_datetime, we can know when the traffic is heavy or not (extract hour of day, day of month.) For example, we can think about 8 am to 10 am or 5 pm to 7 pm, traffic will be heavy because of commuters, and weekdays vs weekend (including holiday of NYC) could have a major effect on results. If weekend is coming, many people can travel or go somewhere more comparing to weekdays then we’ll take a bit longer to reach our destination.

Second, using \_latitude and \_longitude, which is the most important feature I think, we can get some distance between coordinates. (pick-up and drop-off site)

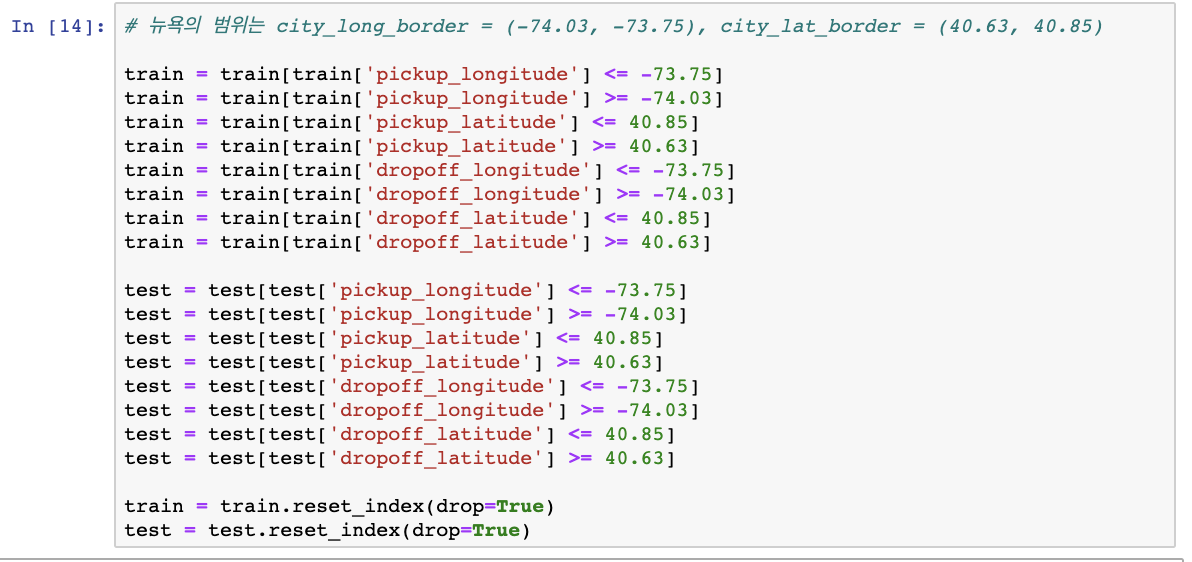
Using info, describe, shape function, see which columns do train, test set have.

* 1. Latitude and Longitude Cleaning

The border of NYC, in coordiantes comes out to be :

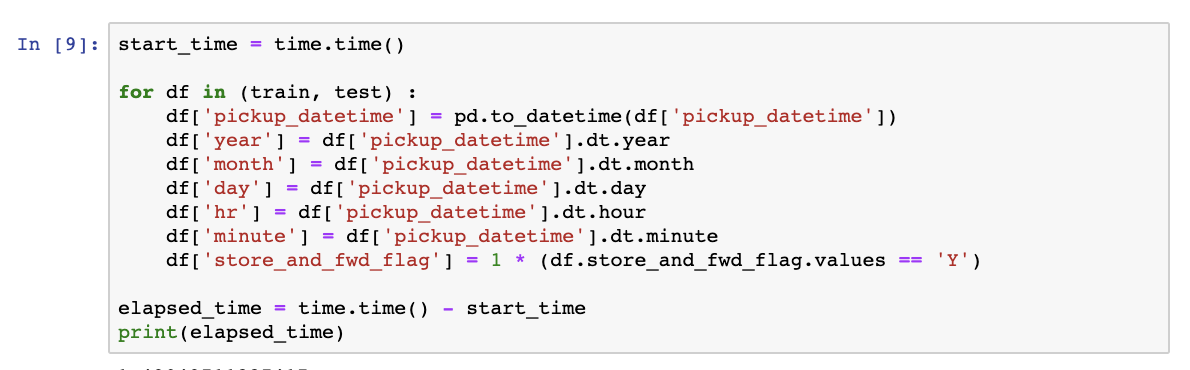
city\_long\_border = (-74.03, -73.75)  
city\_lat\_border = (40.63, 40.85)

So, comparing with ‘train.describe()’ result, we can see some odd number in every max value. So, I need to clean some incorrect value in some columns. The code I use is as follows :



* 1. Date Cleaning

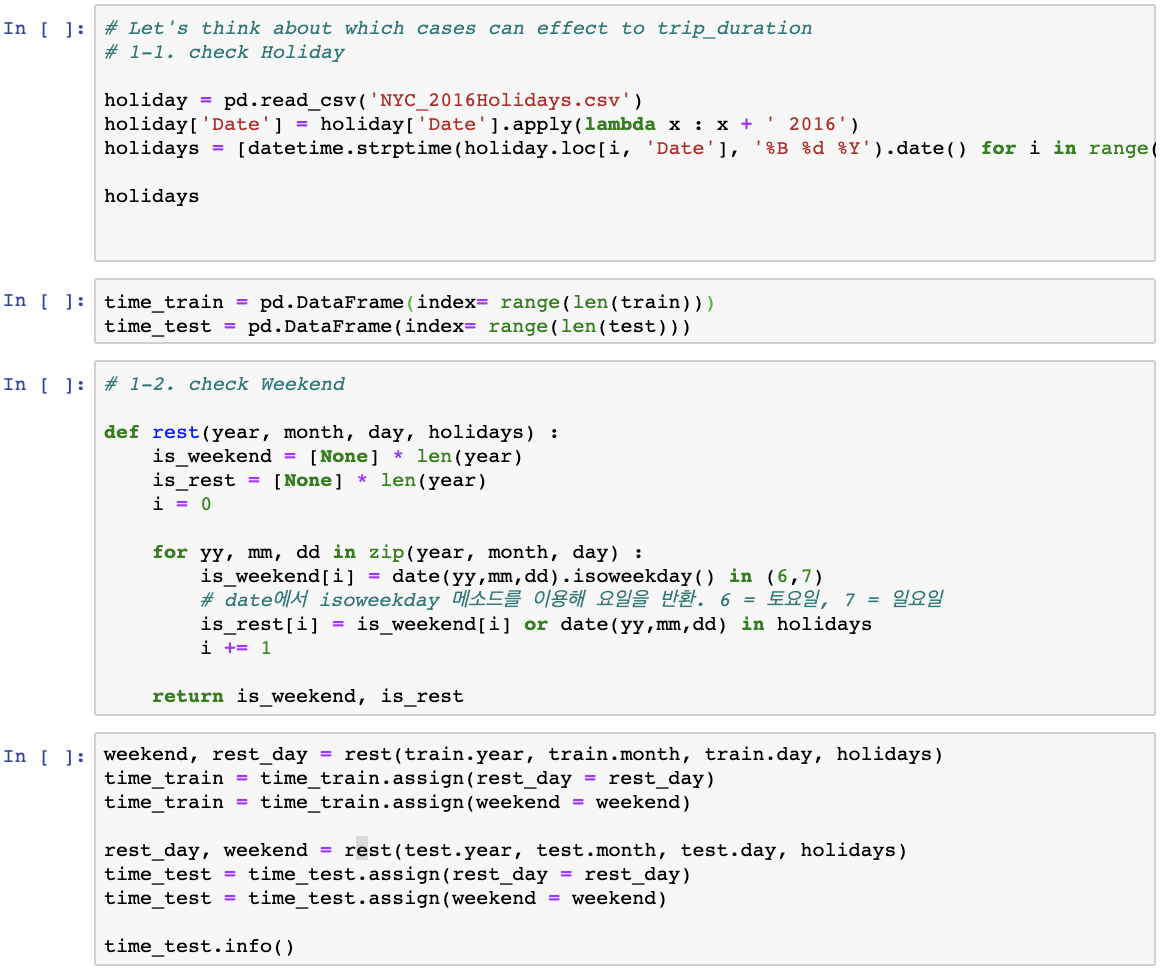
We need to change the formatting of ‘pickup\_datetime’ and ‘dropoff\_datetime’ to datetime type. This will be really important feature when we will get final results.



Let’s think about which will affect to our target data ( ‘trip\_duration’ ). We know that weekend has heavy traffic and cloudy, rainy, snowy day also has heavy traffic.

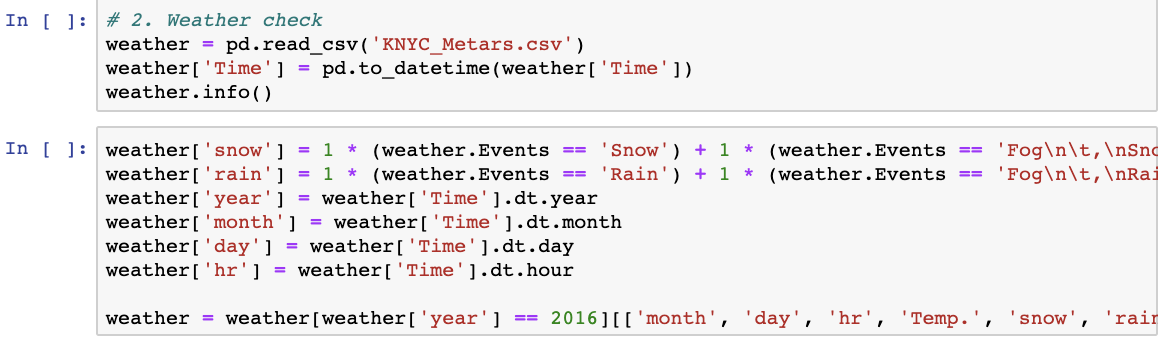
* 1. Add Holiday data

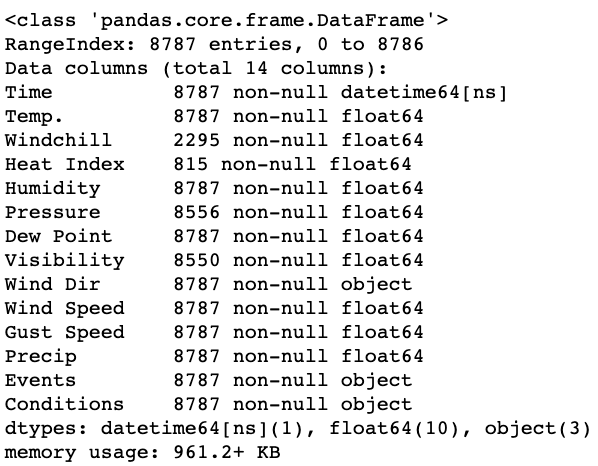
Thanks to Kaggle, I can get holiday data in 2016 NY City. With this data and datetime package, I can check weekend and holiday.



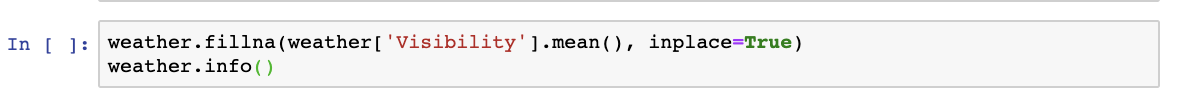
* 1. Add Weather data

Thanks to Kaggle, I can get weather data in 2016 NY City. With this data, I can check weather 2016.1.1 to 2016.6.30.





But when I check this data with info method, there are so many NAN number in some columns. So I choose some columns don’t have NAN number. (Time, Temp., Precip, VIsibility). I can get rain, snow data from Events column. There are some NAN number in ‘Visibility’ so I fill NAN number with mean() method like this.



* 1. Add Distance data

We have two method can get distance between two coordinates. First is Euclidean distance, second is Haversine distance. (Reference : <https://en.wikipedia.org/wiki/Haversine_formula>) But Euclidean distance is just for comparing two distance relatively. So, we need Haversine formula which is the great-circle distance between two points on a sphere given their longitudes and latitudes and give us exact distance between two points. With this formula, we can finally calculate the direction of distance traveled.





1. Data Visualization
   1. Initial Analysis

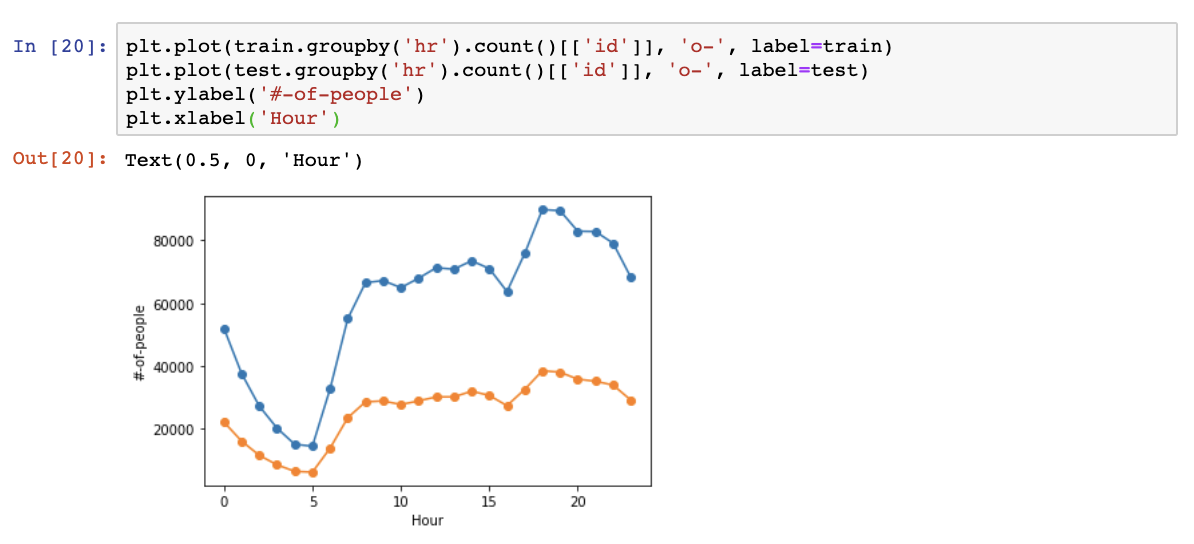
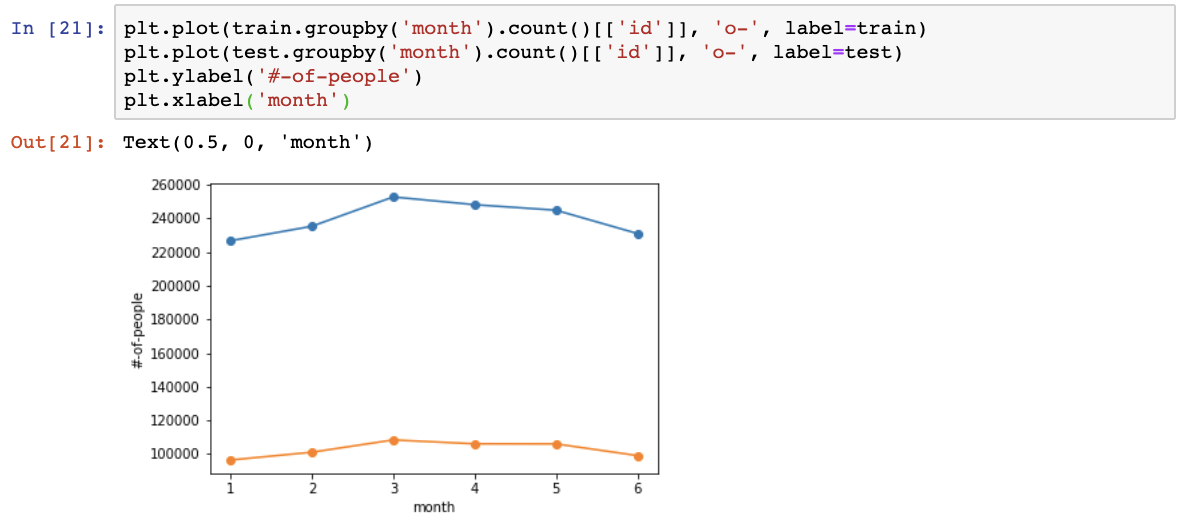
Let’s see our target value, ‘trip\_duration’ in histogram form. As you can see, shape of this histogram looks weird. Target values form is leaning toward to left. So I need to do ‘log transformation’ to make this histogram into normal distribution form. Like this. :



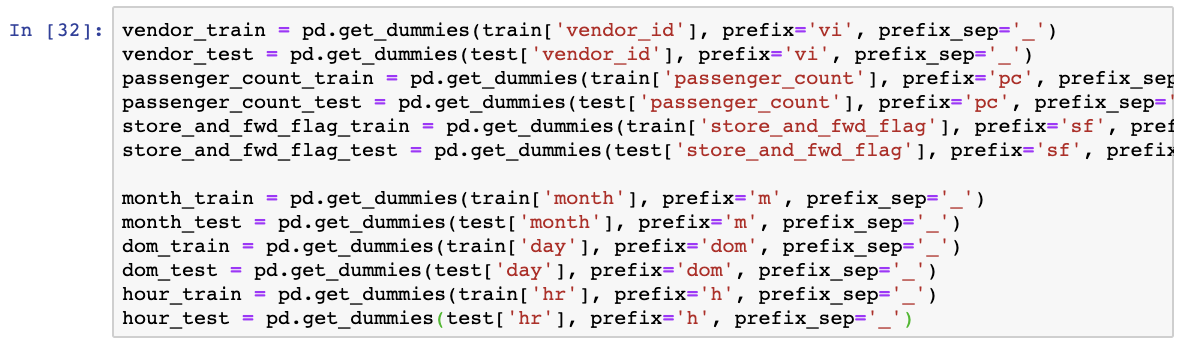


* 1. Pick-up date Analysis

I use group-by function to see how many people use taxi service in particular time and month. 12 am to 5 am, # of people are decreasing, in the morning and after 5, #-of-people are increasing steeply. I guess that this kind of situation is being held because of commuters. So, I think pick-up hour might be an important feature to get results. And there is no striking difference due to month.

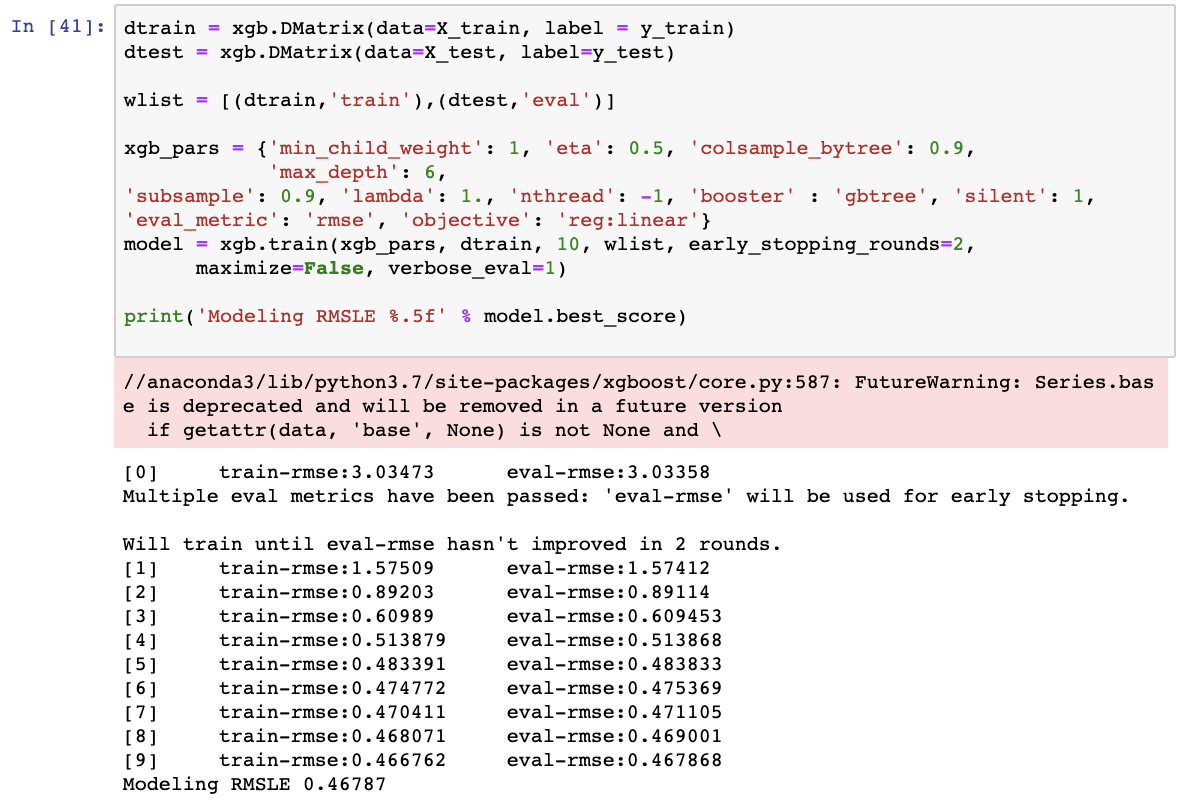


1. One-hot encoding



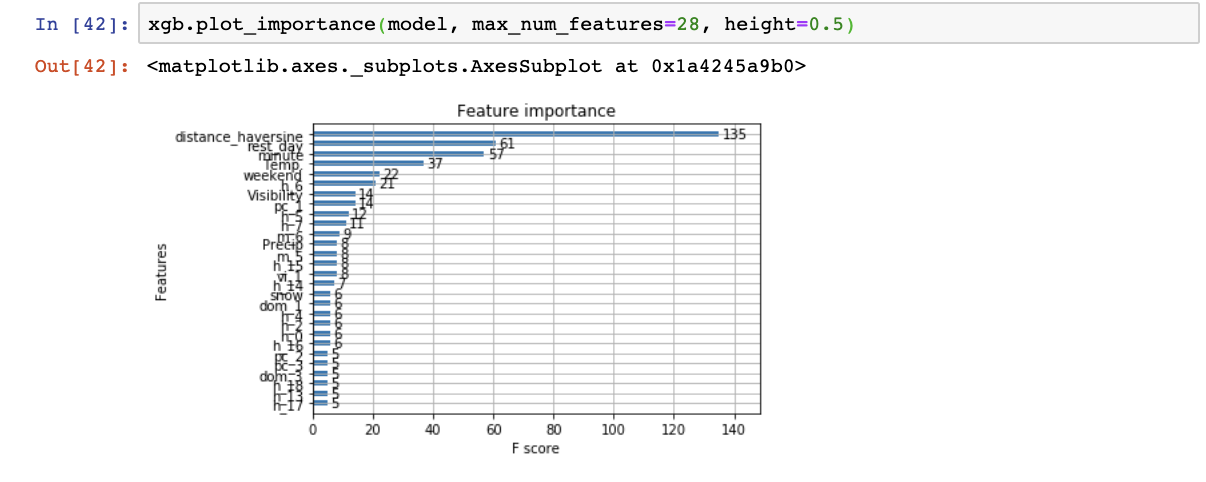
1. Training the Model and testing the accuracy

I notice that Lasso, Ridge, LinearRegression are not good at analyzing this kind of dataset even though engineering some features. Because, in my opinion, features are not in a linear relation. As you can see, there are no relation in day-of-hour, day-of-month, and weather. They have difference between several values in their features in particular time/season. So, I change the way to get some accuracy result by using XGBoost. And through GridSearchCV best parameter result, I use this kind of parameters. Like this :



The result came out 0.46787. The training was stopped in epoch [9]. (Due to early stopping.)

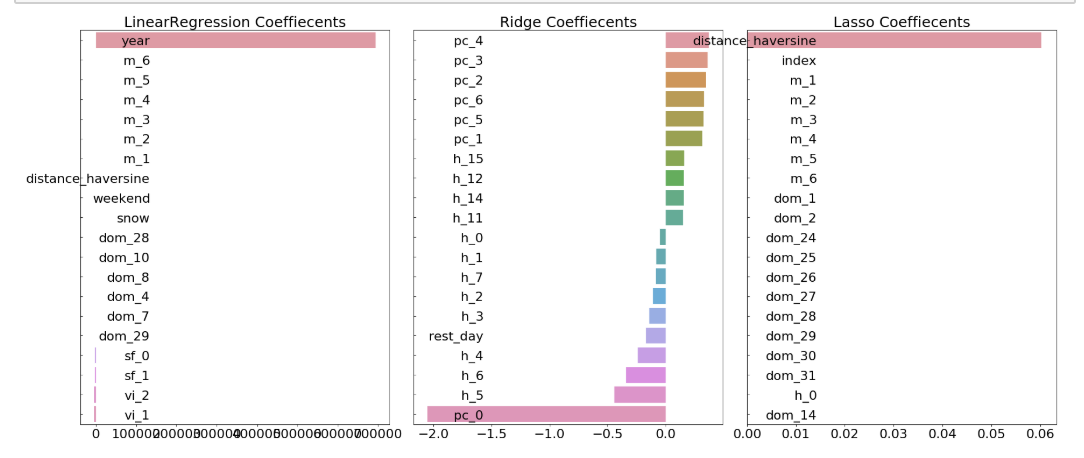
Out of interest, we can investigate the importance of each feature, to understand what affects the trip duration the most significantly. Here's how XGBoost allows us to do it:



From top to bottom we can see which features have the greatest effect on trip duration. It would make logical sense that distance has the greatest affect. The further you travel, the longer it'll take. Also, it’s a bit surprising result for me, temperature and minute is important compared with hour, month feature. The rest of the features follow a similar logic in why it's ranked the way it is.



As you can see, RMSE result are worse than XGBoost’s RMSE. I already mention that why this kind of results are came out.



3 kinds of classifier can’t reflect many kinds of features in this train data set. Lasso just use distance feature, and LinearRegression just use year feature (I don’t know why year feature is the most important to this classifier. All feature have the same value (2016)) to predict duration in this data.