Taxi Prediction:

Information on taxis:

Yellow Taxi: Yellow Medallion Taxicabs These are the famous NYC yellow taxis that provide transportation exclusively through street-hails. The number of taxicabs is limited by a finite number of medallions issued by the TLC. You access this mode of transportation by standing in the street and hailing an available taxi with your hand. The pickups are not pre-arranged.

For Hire Vehicles (FHVs) FHV transportation is accessed by a pre-arrangement with a dispatcher or limo company. These FHVs are not permitted to pick up passengers via street hails, as those rides are not considered pre-arranged.

Green Taxi: Street Hail Livery (SHL) The SHL program will allow livery vehicle owners to license and outfit their vehicles with green borough taxi branding, meters, credit card machines, and ultimately the right to accept street hails in addition to pre-arranged rides.

A maximum of four passengers may be carried in most cabs, although larger minivans may accommodate five passengers, and one child under seven can sit on an adult's lap in the back seat if the maximum has been reached. Drivers are required to pick up the first or closest passenger they see, and may not refuse a trip to a destination anywhere within the five boroughs

Business problem:

Given pickup and dropoff locations, the pickup timestamp, and the passenger count, the objective is to predict the fare of the taxi ride

Data:

Ge the data from : http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml The data used in the attached datasets were collected and provided to the NYC Taxi and Limousine Commission (TLC)

Objectives & Constraints:

Objectives:
Predict number of pick ups as accurately as possible within 10 minutes in a region.

Constraints:
Latency : A medium latency rate is required since the cab driver expect to find pick ups within 10 minutes.
Interpretability : As long as the output doesnt seem mean,the cab driver is not into the interpretation of the result.
Relative Error : Let say the predicted pickups for a particular location are 100, but in actual pickups are 102 the percentage error will be 2% and Absolute error is 2. The taxi driver into percentage error than the absolute error. Let say in some region the predicted pickups are 250, and if taxi driver knows that the relative error is 10% then predicted result will be considered in the range of 225 to 275 which is considerable

Machine Learning Problem :

As mentioned our goal is to predict the fare of taxi ride.Two important preprocessing tasks are involved:

1. Binning data into 10 mins interval(Since the average time taken to travel 1 mile is 10 minutes in any of the region)
2. Break NYC into clusters(regions)
   It is a Time-Series forecasting and regression problem.

Performance metrics :

1. Mean Absolute percentage error(MAPE).
2. Mean Squared error(MSE).

Features of the data:

Field Name	Description
VendorID	A code indicating the TPEP provider that provided the record. 1. Creative Mobile Technologies 2. VeriFone Inc.
tpep_pickup_datetime	The date and time when the meter was engaged.
tpep_dropoff_datetime	The date and time when the meter was disengaged.
Passenger_count	The number of passengers in the vehicle. This is a driver-entered value.
Trip_distance	The elapsed trip distance in miles reported by the taximeter.
Pickup_longitude	Longitude where the meter was engaged.
Pickup_latitude	Latitude where the meter was engaged.
RateCodeID	The final rate code in effect at the end of the trip. 1. Standard rate 2. JFK 3. Newark 4. Nassau or Westchester 5. Negotiated fare 6. Group ride
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka “store and forward,” because the vehicle did not have a connection to the server. Y= store and forward trip N= not a store and forward trip
Dropoff_longitude	Longitude where the meter was disengaged.
Dropoff_ latitude	Latitude where the meter was disengaged.
Payment_type	A numeric code signifying how the passenger paid for the trip. 1. Credit card 2. Cash 3. No charge 4. Dispute 5. Unknown 6. Voided trip
Fare_amount	The time-and-distance fare calculated by the meter.
Extra	Miscellaneous extras and surcharges. Currently, this only includes. the $0.50 and $1 rush hour and overnight charges.
MTA_tax	0.50 MTA tax that is automatically triggered based on the metered rate in use.
Improvement_surcharge	0.30 improvement surcharge assessed trips at the flag drop. the improvement surcharge began being levied in 2015.
Tip_amount	Tip amount – This field is automatically populated for credit card tips.Cash tips are not included.
Tolls_amount	Total amount of all tolls paid in trip.
Total_amount	The total amount charged to passengers. Does not include cash tips.

Exploratory Data Analysis:

We perform EDA on Jan 2015 data and try removing outliers if exist.

1. pickup_latitude & pickup_longitude:
It is inferred from the source https://www.flickr.com/places/info/2459115 that New York is bounded by the location cordinates(lat,long) - (40.5774, -74.15) & (40.9176,-73.7004).Hence any cordinates not within these cordinates are not considered by us as we are only concerned with pickups which originate within New York.All the pickup_latitude & pickup_longitude outside of the given specific location will be taken as an outlier and removed

2.dropoff_latitude & dropoff_longitude:
It is inferred from the source https://www.flickr.com/places/info/2459115 that New York is bounded by the location

It is inferred from the source https://www.flickr.com/places/info/2459115 that New York is bounded by the location cordinates(lat,long) - (40.5774, -74.15) & (40.9176,-73.7004) so hence any cordinates not within these cordinates are not considered by us as we are only concerned with dropoffs which are within New York.All the dropoff_latitude & dropoff_longitude outside of the given specific location will be taken as an outlier and removed

3.trip_duration(tpep_pickup_datetime & tpep_dropoff_datetime):
According to NYC Taxi & Limousine Commision Regulations the maximum allowed trip duration in a 24 hour interval is 12 hours.From tpep_pickup_datetime & tpep_dropoff_datetime we can calculate trip duration.All the records that give duration time greater than this will be considered as outliers and hence removed.

4.trip_speed:
The avg speed in Newyork speed is 12.45miles/hr, so a cab driver can travel 2 miles per 10min on avg.Trip speed can be calculated by trip_duration and trip_distance.All the records that give speed greater that this will be considered as outliers ad hence removed.

5.Total Fare:
For understanding outliers we use box plot and then plot percentiles to have clear picture of the higher rates.This percentile value could be considered as threshold and records above this will be considered as outlier and hence removed.

Conclusion: Applying all these we would be able to remove outliers.When I applied all these techniques,the fraction of data points that remain after removing outliers 0.9703576425607495

Clustering/Segmentation

One ML objective we have here is breaking down NYC data into regions/clusters.One important problem in accompkishing this objective is the features we would use to break NYC region into a certain number of clusters.Using K-Means clustering,the size of clusters would roughly be same.Here I have used K-Means clustering with objective function as minimising distance between clusters (around 0.5 miles) and found the optimal K would be 40.

Time Binning:

Since the average time taken to travel 1 mile is 10 minutes in any of the region,it is necessary that we bin data by time.Binning data into 10 mins interval,we devid whole months time into 10min intravels 24*31*60/10 =4464bins.

Smoothing:

It would lead to divide by zero problems in ratio feature if we have data containing zero pick ups. there are two ways to fill up these values

1. Fill the missing value with 0's
2. Fill the missing values with the avg values
   Case 1:(values missing at the start)

   ```
              Ex1: _ _ _ x =>ceil(x/4), ceil(x/4), ceil(x/4), ceil(x/4) <br/>
              Ex2: _ _ x => ceil(x/3), ceil(x/3), ceil(x/3)<br/>
   ```

   Case 2:(values missing in middle)

   ```
              Ex1: x _ _ y => ceil((x+y)/4), ceil((x+y)/4), ceil((x+y)/4), ceil((x+y)/4) <br/>
              Ex2: x _ _ _ y => ceil((x+y)/5), ceil((x+y)/5), ceil((x+y)/5), ceil((x+y)/5), ce
      il((x+y)/5)<br/>
   ```

   Case 3:(values missing at the end)

   ```
               Ex1: x _ _ _ => ceil(x/4), ceil(x/4), ceil(x/4), ceil(x/4) <br/>
               Ex2: x _ => ceil(x/2), ceil(x/2)<br/>
   ```

For example lets say:

At t we have 50 pick ups
At t+1 we have 0 pick ups
At t+2 we have 150 pick ups
What smoothing does is divide those 30 min pick ups to three equal pick ups leading each and every 10 min time bin to 50 pick ups.

We use 'Fill the missing values with the avg values ' method for 2015 data which is used as training data.
We use 'Fill the missing value with 0's' method for 2016 data which is used to test the model

Data Preparation:

Data Preparation for the months of Jan,Feb and March 2016

1. get the dataframe which inlcudes only required colums
2. adding trip times, speed, unix time stamp of pickup_time
3. remove the outliers based on trip_times, speed, trip_duration, total_amount
4. add pickup_cluster to each data point
5. add pickup_bin (index of 10min travel to which that trip belongs to)
6. group by data, based on 'pickup_cluster' and 'pickup_bin'
7. Fill zero pick ups using fill_misssing method.

Featurizations:

We could consider two important features:

1. Using Ratios of the 2016 data to the 2015 data i.e Rt=P2016t/P2015t
2. Using Previous known values of the 2016 data itself to predict the future values

Exponential Weighted Moving Averages: Using previous values we could predict the next value.But there is always an issue using previous values, it is necessary that we provide weightage to the recent values and less weights to the subsequent ones.But we still do not use which is the correct weighting scheme as there are infinitely many possibilities in which we can assign weights in a non-increasing order and tune the hyper-parameter window-size.To simplify this process we use Exponential Moving Averages which is a more logical way towards assigning weights and at the same time also using an optimal window-size.

$R't=\alpha*Rt-1+(1-\alpha)*R't-1$
$P't=\alpha*Pt-1+(1-\alpha)*P't-1$

In exponential moving averages we use a single hyper-parameter alpha (α) which is a value between 0 & 1 and based on the value of the hyper-parameter alpha the weights and the window sizes are configured.

We could use fourier transform components if we have time-series repeating data.Since the taxi data is reapeating for every 24 hours we could use its fourier components as feature.I have considered the top five amplitudes and corresponding frequencies as features in data.We have taken the components from the second peak since DC component is capturing the information of the previous part of the wave

Split data

we take 3 months of 2016 pickup data and split it such that for every region we have 70% data in train and 30% in test, ordered date-wise for every region

Features:

'freq1':frequency of 1st highest amplitude
'freq2':frequency of 2nd highest amplitude
'freq3':frequency of 3rd highest amplitude
'freq4':frequency of 4th highest amplitude
'freq5':frequency of 5th highest amplitude
'amp1':Amplitude of 1st highest fourier wave
'amp2':Amplitude of 2nd highest fourier wave
'amp3':Amplitude of 3rd highest fourier wave
'amp4':Amplitude of 4th highest fourier wave
'amp5':Amplitude of 5th highest fourier wave
'ft_5':Number of pick-ups at (t-5)th interval
'ft_4':Number of pick-ups at (t-4)th interval
'ft_3':Number of pick-ups at (t-3)rd interval
'ft_2':Number of pick-ups at (t-2)nd interval
'ft_1':Number of pick-ups at (t-1)st interval
lat:Latitude of cluster centre

lat:Latitude of cluster centre
long:Longitude of cluster centre
weekday:Pick up weekday
exp_avg:Exponential Weighted Moving Averages predicted values

Models

Using these features I have used Linear regression,Random forest and XGBoost models to train and test.Below is the summary of the output

Model	Train error	Test error
Linear Regression	0.113648123791	0.129361804204
Random Forest	0.0717619563182	0.124542461769
XGBoost	0.119387790351	0.115742475694