Capstone Project-2

NYC Taxi Trip Time Prediction (Supervised ML- Regression)

Individual project

Tanmaya Kumar Pattanaik

Introduction

- A typical taxi company faces a common problem of efficiently assigning the cabs to passengers so that the service is smooth and hassle free. One of main issue is determining the duration of the current trip so it can predict when the cab will be free for the next trip.

- So, Lets take a dataset and predict the trip duration and make taxi company life easier.

NYC Taxi Data

- The data set contains the data regarding several taxi trips and its duration in New York City. I will now try and apply different techniques of Data Analysis to get insights about the data and determine how different variables are dependent on the target variable **Trip Duration.**

- The dataset is based on the 2016 NYC Yellow Cab trip record data made available in Big Query on Google Cloud Platform. The data was originally published by the NYC Taxi and Limousine Commission (TLC).

Column Information

- **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
- **dropoff_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
- **dropoff_longitude** - the longitude where the meter was disengaged
- **dropoff_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

Problem Statement

Our task is to build a model that predicts the total ride duration of taxi trips in New York City.

Based on the individual trip attributes, we should predict the duration of each trip in the test set.

Required packages

- Numpy
- Pandas
- Seaborn
- Matplotlib
- Datetime
- XGBRegressor
- Haversine
- Sklearn

1. Linear Regression
2. Metrics
3. Train_test_split
4. GridsearchCV

Data exploration

- There are 1458644 rows and 11 columns in the dataset.
- There is no NaN/NULL record in the dataset, So we dont have to impute any record.
- The columns id and vendor_id are nominal.
- The columns pickup_datetime and dropoff_datetime are stored as object which must be converted to datetime for better analysis.
- Distance between pickup and dropoff coordinates using Haversine formula.
- Speed= Distance/Time
- The column store_and_fwd_flag is categorical
- The passenger count varies between 1 and 9 with most people number of people being 1 or 2
- The trip duration varying from 1s to 1939736s~538 hrs. There are definitely some outliers present which must be treated.

Univariate Analysis

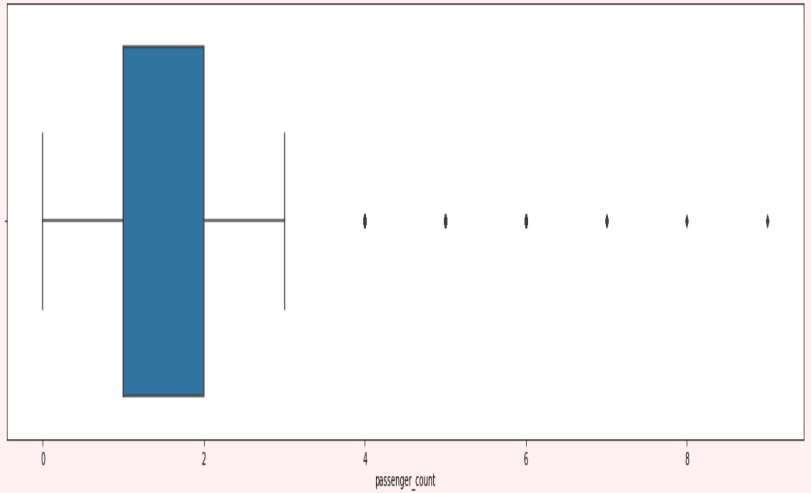
Univariate analysis is the analysis of one variable. Its major purpose is to describe patterns in the data consisting of single variable.

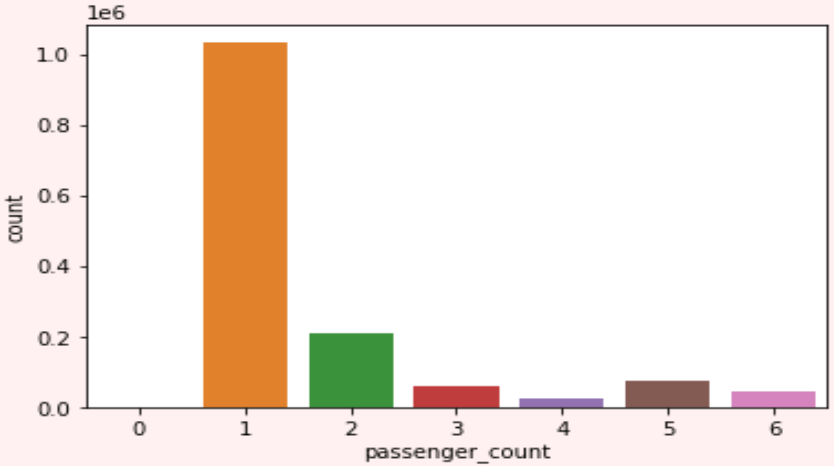
It doesn't deal with causes or relationships (unlike regression) and it's major purpose is to describe.

It takes data, summarizes that data and finds patterns in the data

Passengers

- There are some trips with 0 passenger count.
- Few trips consisted of even 7, 8 or 9 passengers. Clear outliers and pointers to data inconsistency
- Most of trip consist of passenger either 1 or 2.
- Passenger count is a driver entered value. Since the trip is not possible without passengers. It is evident that the driver forgot to enter the value for the trips with 0 passenger count.
- Mean median and mode are all approx equal to 1. So, we would replace the 0-passenger count with 1.
- Most of the trips was taken by single passenger and that is inline with our day-to-day observations

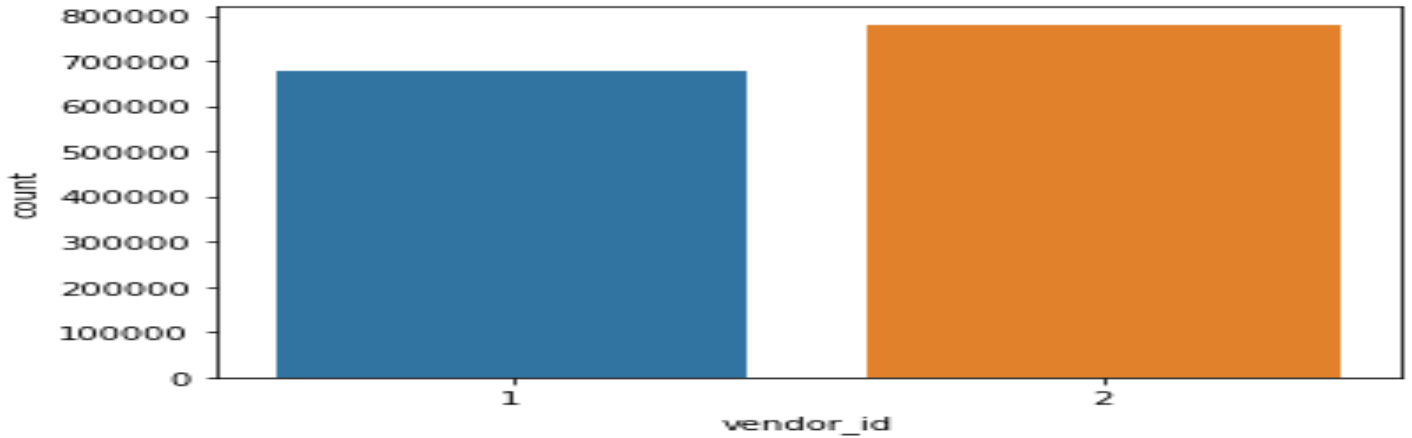




Vendors

Here we analyze taxi data only for the 2 vendors which are listed as 1 and 2 in the dataset.

Though both the vendors seems to have almost equal market share. But Vendor 2 is evidently more famous among the population as per the graph.

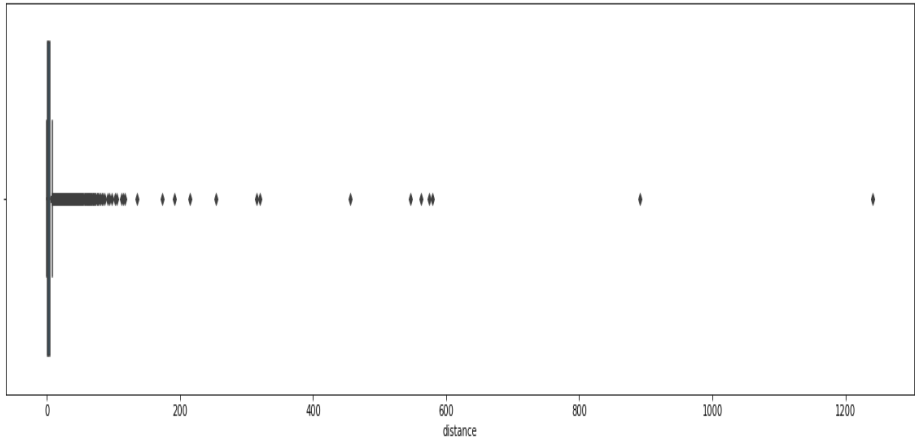


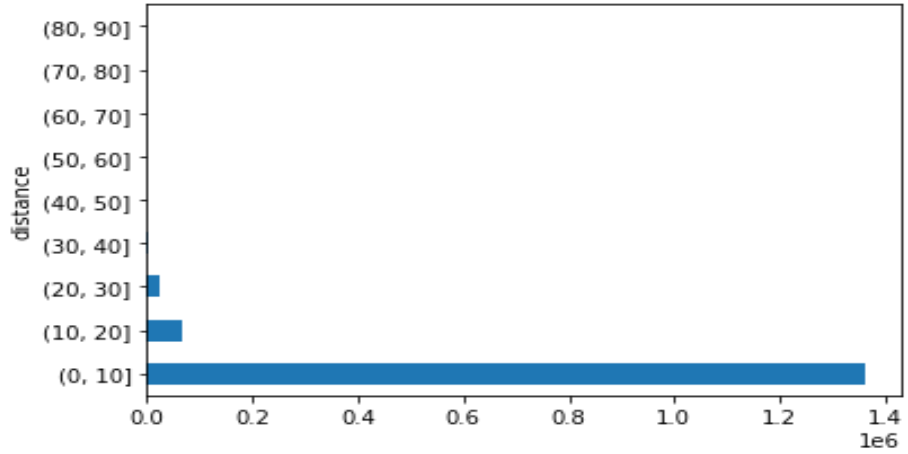
Distance

- There are some trips with over 100 km distance.
- Some of the trips distance value is 0 km.
- Mean distance travelled is approx 3.5 kms.
- Around 6K trip record with distance equal to 0. Below are some possible explanation for such records

1. Customer changed mind and cancelled the journey just after accepting it.
2. Software didn't record drop-off location properly due to which drop-off location is the same as the pickup location.
3. Issue with GPS tracker while the journey is being finished.
4. Driver cancelled the trip just after accepting it due to some reason. So, the trip couldn't start
5. Or some other issue with the software itself

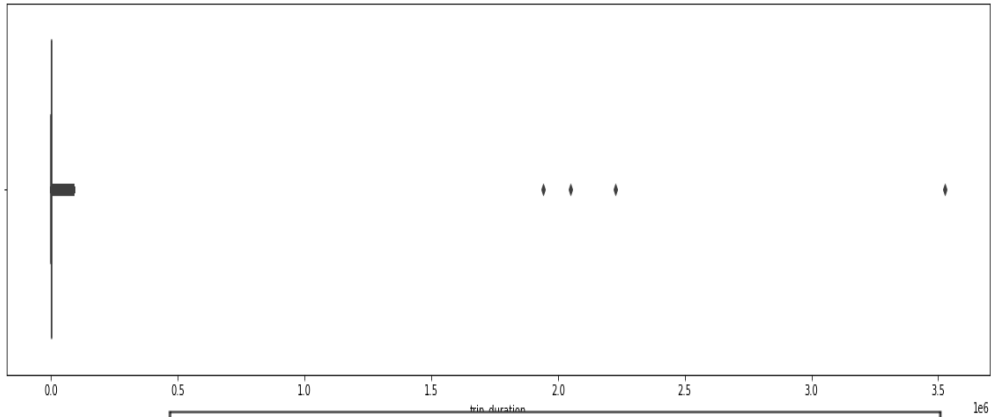
Most of the rides are completed between 1-10 Kms with some of the rides with distances between 10-30 kms.

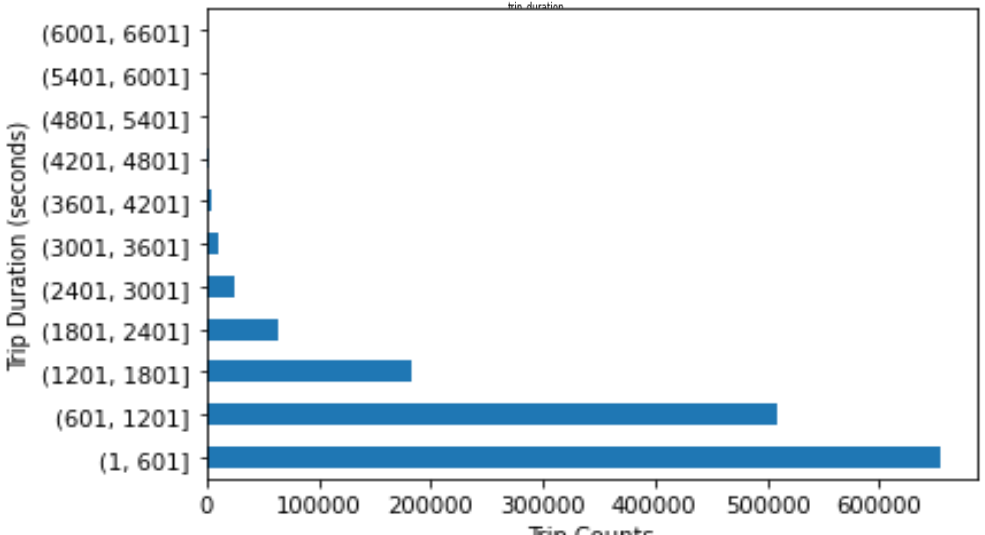




Trip duration

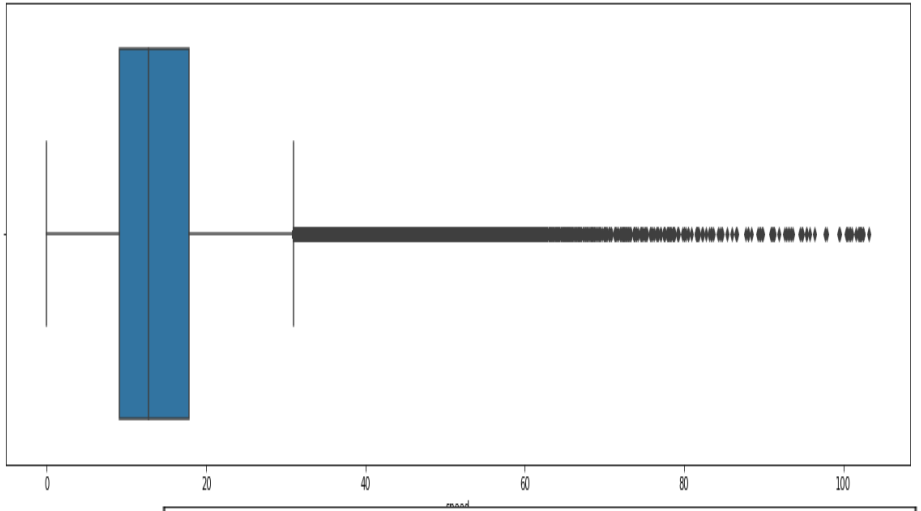
- Some trip durations are over 100000 seconds which are clear outliers and should be removed
- There are some durations with as low as 1 second. which points towards trips with 0 km distance.
- Major trip durations took between 10-20 mins to complete.
- Mean and mode are not same which shows that trip duration distribution is skewed towards right.
- Those trips with huge duration, these are outliers.
- These trips ran for more than 20 days, which seems unlikely by the distance travelled.
- All the trips are taken by vendor 1 which points us to the fact that this vendor might allows much longer trip for outstations.
- All these trips are either taken on Tuesdays in 1st month or Saturdays in 2nd month.
- Most of the trips took 0 - 30 mins to complete i.e. approx 1800 secs.

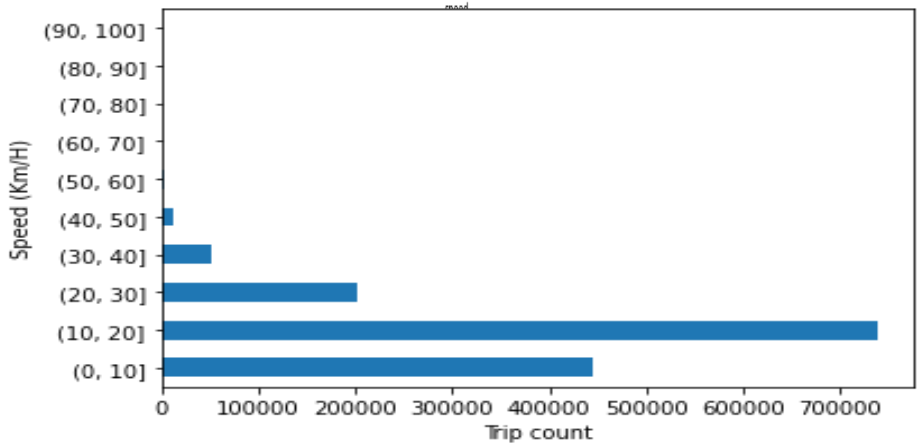




Speed

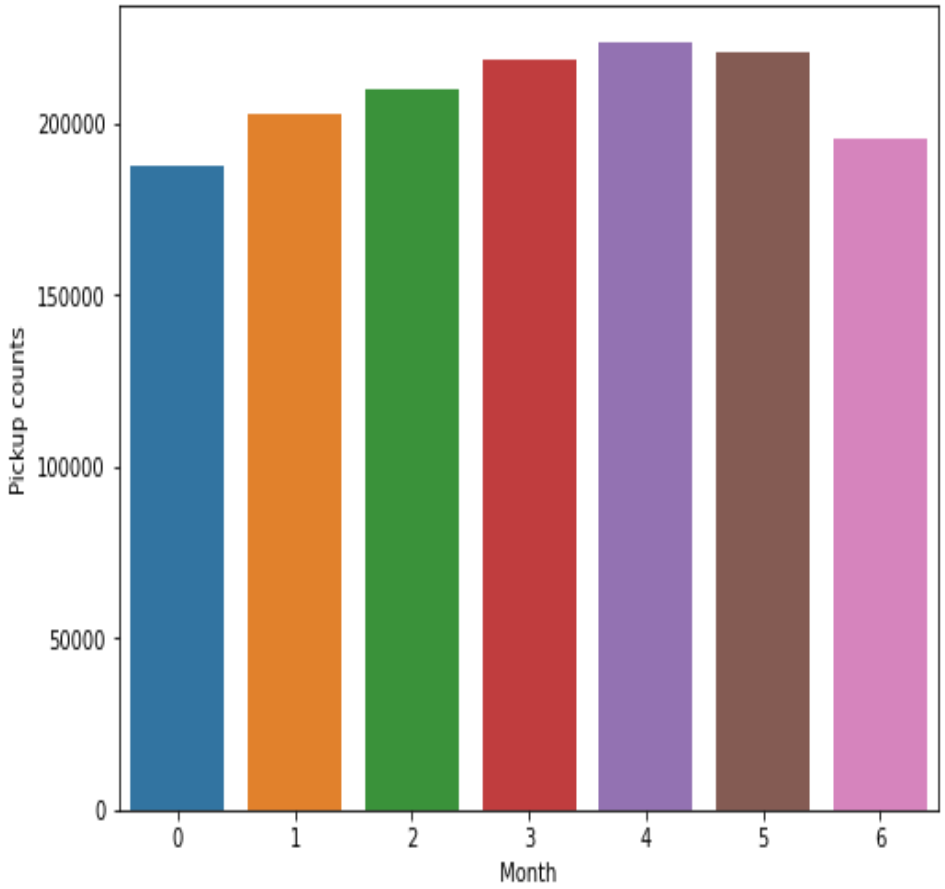
- Many trips were done at a speed of over 200 km/h.
- We focus on the trips which were done at less than 104 km/h as per the speed limits
- Trips over 30 km/h are being considered as outliers but we cannot ignore them.
- Mostly trips are done at a speed range of 10-20 km/h with an average speed of around 14 km/h.





Total trips per hour

- We can see an increasing trend of taxi pickups starting from Monday till Friday.
- The trend starts declining from Saturday till Monday which is normal where some office going people likes to stay at home for rest on the weekends.
- Taxi pickups increased in the late-night hours over the weekend possibly due to more outstation rides or for the late night leisure's nearby activities.
- Early morning pickups i.e before 5 AM have increased over the weekend in comparison to the office hours pickups i.e. after 7 AM which have decreased due to obvious reasons.
- Taxi pickups seems to be consistent across the week at 15 Hours i.e. at 3 PM.



Bivariate Analysis

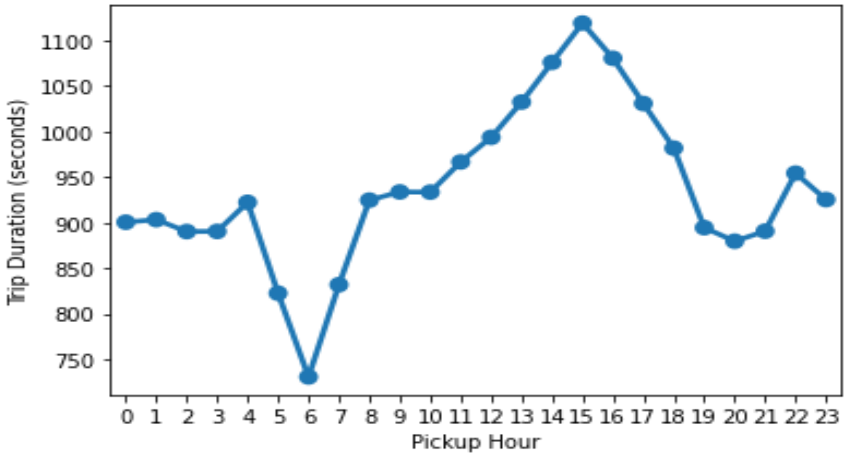
Bivariate analysis is used to find out if there is a relationship between two sets of values. It usually involves the variables X and Y.

Bivariate analysis is one of the simplest forms of quantitative analysis.

It is one of the simplest forms of statistical analysis, used to find out if there is a relationship between two sets of values.

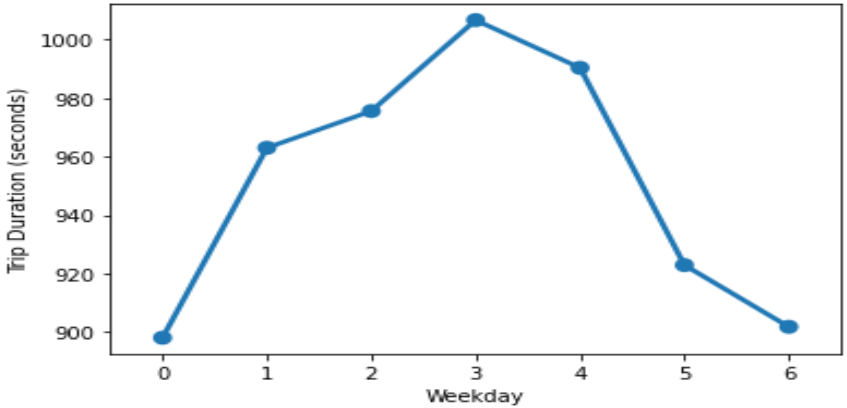
Trip Duration per hour

- Average trip duration is lowest at 6 AM when there is minimal traffic on the roads.
- Average trip duration is generally highest around 3 PM during the busy streets
- Trip duration on an average is similar during early morning hours i.e., before 6 AM & late evening hours i.e., after 6 PM.



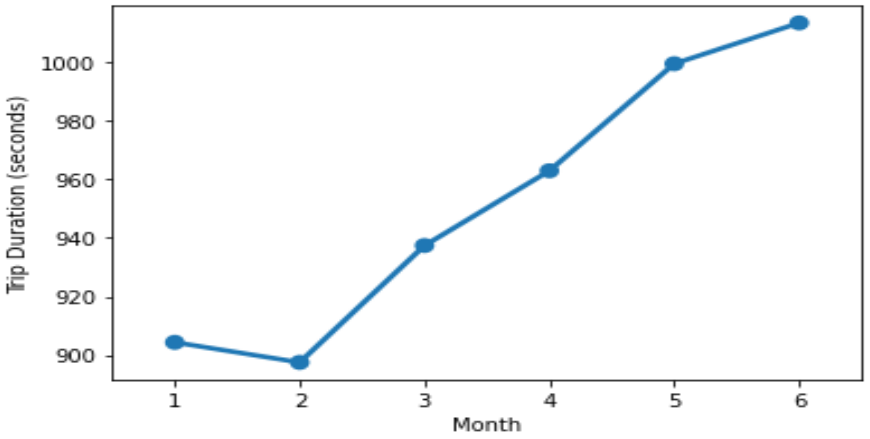
Trip duration per weekday

Trip duration is almost equally distributed across the week on a scale of 0-1000 minutes with minimal difference in the duration times. Also, it is observed that trip duration on Thursday is longest among all days.



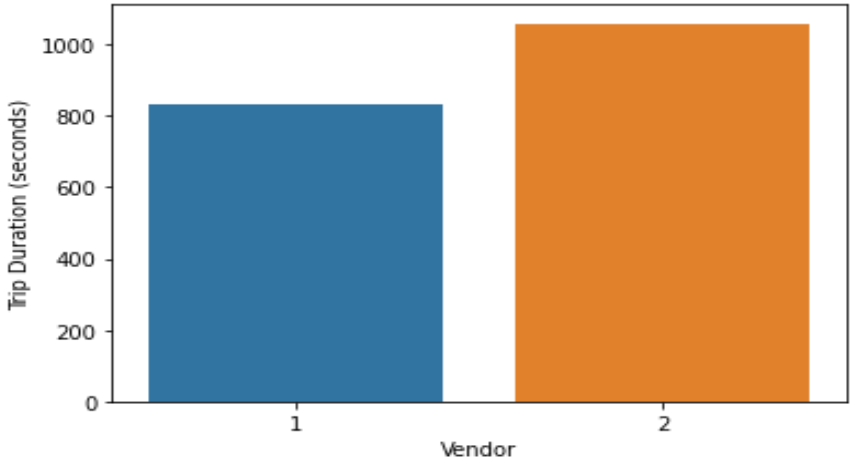
Trip duration per month

- We can see an increasing trend in the average trip duration along with each subsequent month.
- The duration difference between each month is not much. It has increased gradually over a period of 6 months.
- It is lowest during February when winters starts declining.



Trip duration per vendor

Average trip duration for vendor 2 is higher than vendor 1 by approx. 200 seconds i.e., at least 3 minutes per trip.

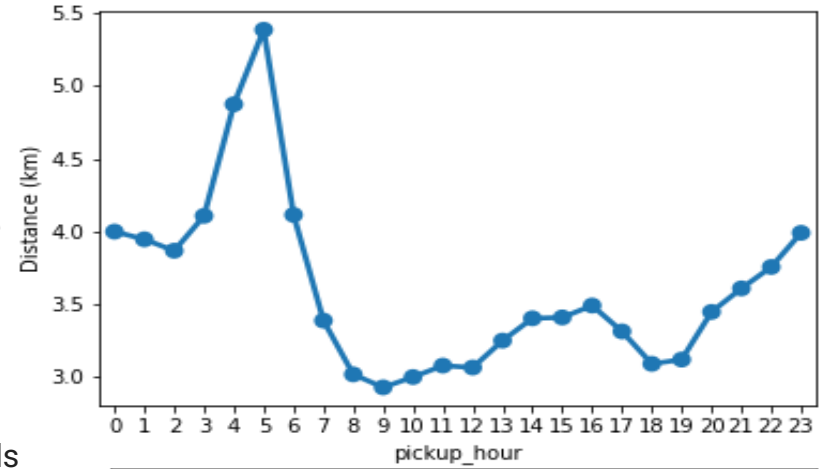


Distance per hour

- Trip distance is highest during early morning hours which can account for some things like:

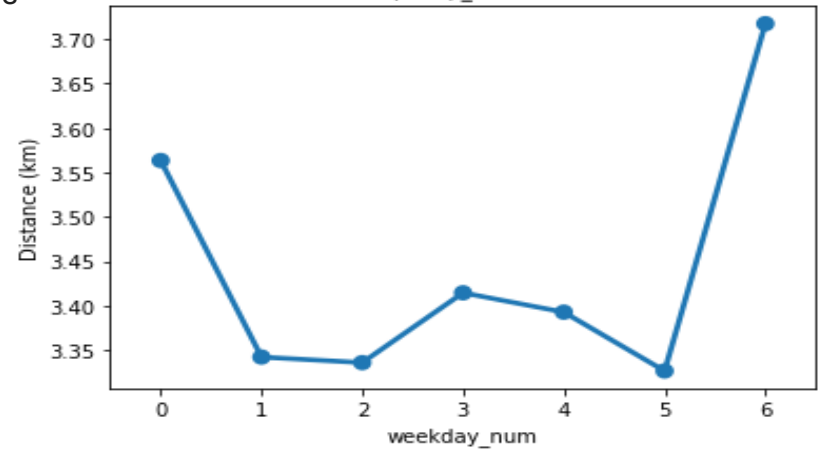
1. Outstation trips taken during the weekends.
2. Longer trips towards the city airport which is located in the outskirts of the city.

- Trip distance is fairly equal from morning till the evening varying around 3 - 3.5 kms.
- It starts increasing gradually towards the late-night hours starting from evening till 5 AM and decrease steeply towards morning.



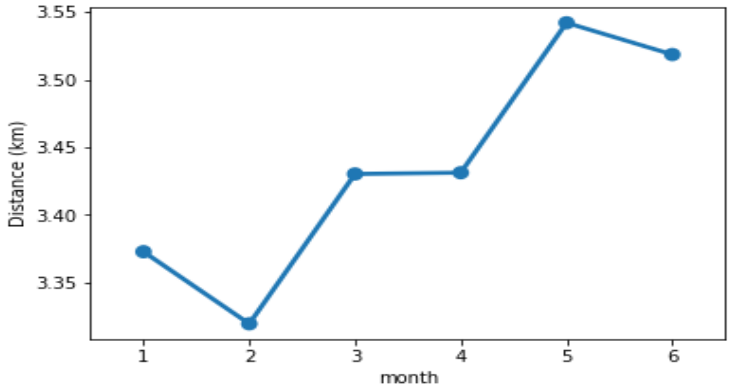
Distance per weekday

It's a fairly equal distribution with average distance metric varying around 3.5 km/h with Sunday being at the top may be due to outstation trips or night trips towards the airport.



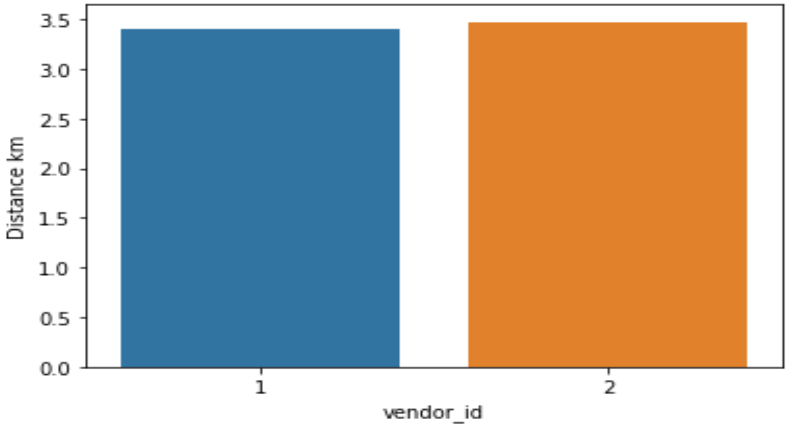
Distance per month

The distribution is almost equivalent, varying mostly around 3.5 km/h with 5th month being the highest in the average distance and 2nd month being the lowest.



Distance per vendor

This is more or less same picture with both the vendors.

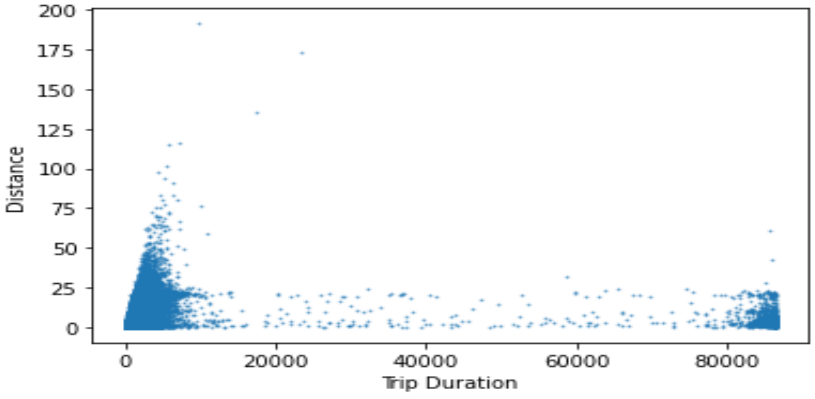


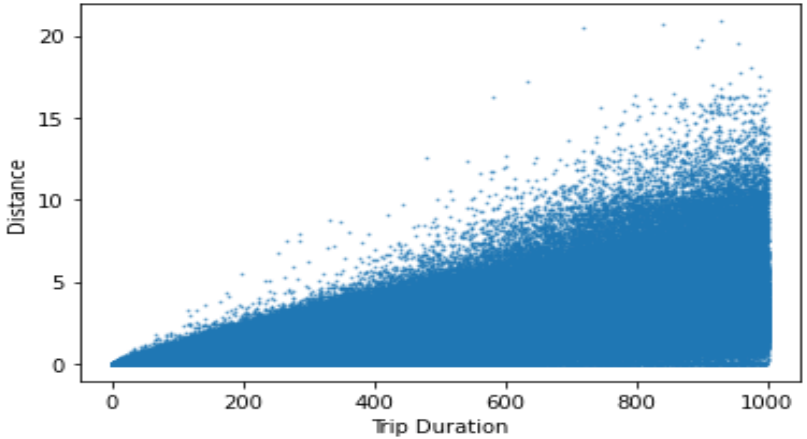
Distance v/s Trip duration

- There are lots of trips which covered negligible distance but clocked more than 20,000 seconds in terms of the Duration.
- Initially there is some proper correlation between the distance covered and the trip duration in the graph. but later on it all seems uncorrelated.
- There were few trips which covered huge distance of approx. 200 kms within very less time frame, which is unlikely and should be treated as outliers.

There should have been a linear relationship between the distance covered and trip duration on an average but we can see dense collection of the trips in the lower right corner which showcase many trips with the inconsistent readings.

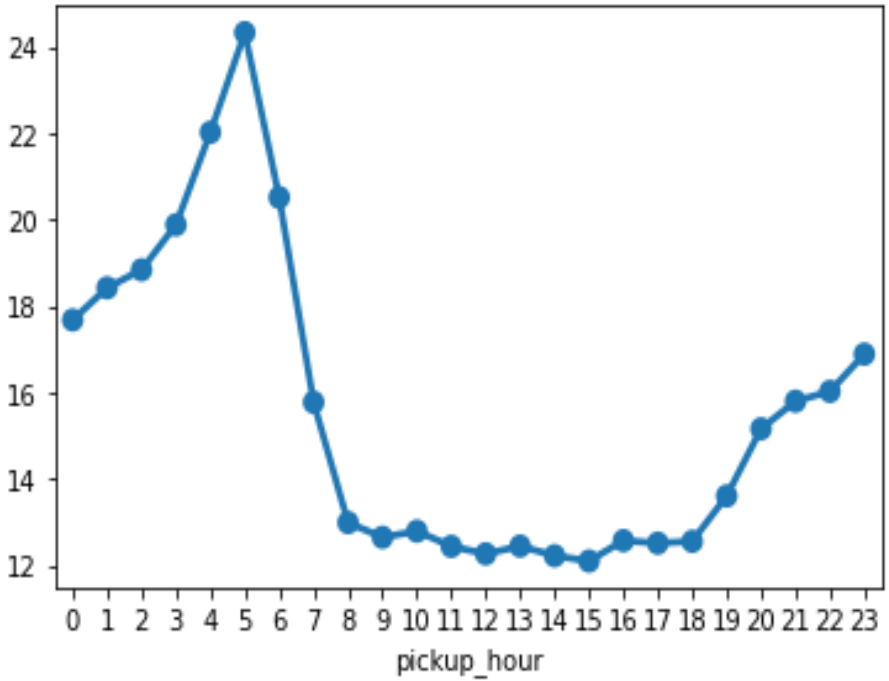
We removed those trips which covered 0 km distance but clocked more than 1 minute to make our data more consistent for predictive model. Because if the trip was cancelled after booking, then that should not have taken more than a minute time. This is our assumption.





Average speed per hour

- The average trend is totally inline with the normal circumstances.
- Average speed tend to increase after late evening and continues to increase gradually till the late early morning hours.
- Average taxi speed is highest at 5 AM in the morning, then it declines steeply as the office hours approaches
- Average taxi speed is more or less same during the office hours i.e. from 8 AM till 6PM in the evening.

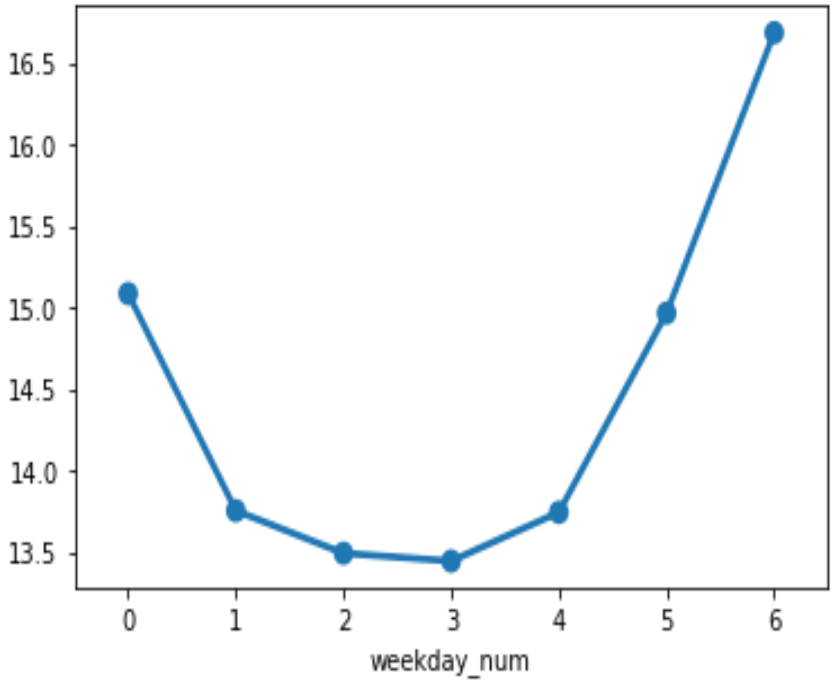


Average speed per weekday

- Average taxi speed is higher on weekend as compared to the weekdays which is obvious when there is mostly rush of office goers and business owners.

- Even on Monday the average taxi speed is shown higher which is quite surprising when it is one of the busiest day after the weekend. There can be several possibility for such behavior

1. Lot of customers who come back from outstation in early hours of Monday before 6 AM to attend office on time.
2. Early morning hours customers who come from the airports after vacation to attend office/business on time for the coming week.



Feature Engineering

Feature Selection

- We used backward elimination technique to select the best features to train our model
- It displays some statistical metrics with there significance value.
- Like, It shows the p values for each feature as per its significance in the whole dataset.
- It also shows the adjusted R squared values to identify whether removing or selecting the feature is beneficial or not.
- We only look at the P and adjusted R squared value to decide which features to keep and which needed to be removed.

Feature Selection(contd.)

- Duration variable assigned to Y because that is the dependent variable.
- Features such as id, timestamp and weekday were not assigned to X array because they are of type object. And we need an array of float data type.
- Fit stats model on the X array to figure out an optimal set of features by recursively checking for the highest p value and removing the feature of that index.
- Here we took the level of significance as 0.05 i.e., 5% which means that we will reject feature from the list of array and re-run the model till p value for all the features goes below .05 to find out the optimal combination for our model.

Split Data

Before training our model on the dataset, we need to split the dataset into training and testing datasets. This is required to train our model on the major part of our dataset and test the accuracy of the model on the minor part.

This will divide our dataset randomly with a ratio of 80/20 where training set consists of more than 1 million records and test dataset with more than .35 million records. Let's train our model on the training set now.

Feature Extraction

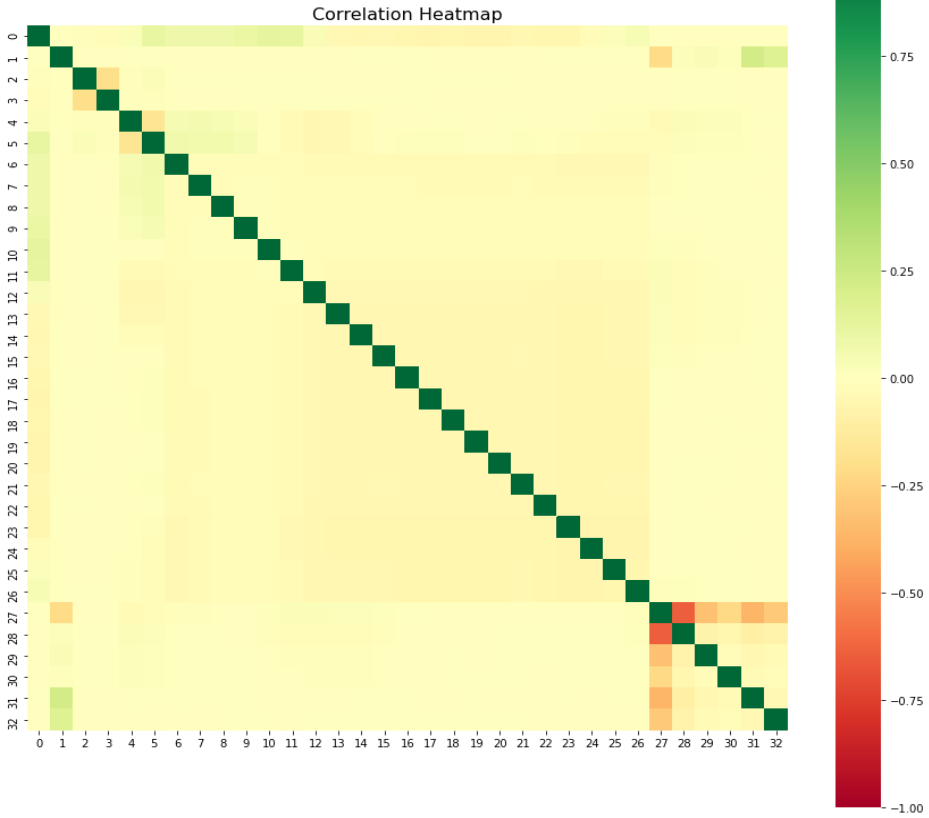
We used PCA for feature extraction i.e. Principal Component Analysis.

It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components

Here we see that almost 40 variables are needed for capturing atleast 99% of the variance in the training dataset. Hence, we will use the same set of variables.

Correlation Analysis

- Some combinations of features shows slight correlation but not above 0.5.
- Some features are infact negatively correlated.
- But most of the features shows no correlation. Which is a good thing.



Multiple Linear Regression

We first try with the default instantiation of the regressor object without using any generalization parameter. We will also not perform any scaling of the features because linear regression model takes care of that inherently. This is a plus point to use Linear regression model. It is quite fast to train even on very large datasets. So considering the size of our dataset this seems to be the correct approach as of now.

Let's see how it performs.

Multiple Linear Regression(Contd.)

RMSE Score for the Multiple linear regression for raw data, Feature selection data and PCA are same as 2739.0109

Variance score for the Multiple linear regression are same as 0.07

- Very poor Root mean squared value.
- And the low variance score which is also bad.
- Both the models i.e. from the feature selection and the feature extraction group resulted quite bad in prediction

XGBoost Regressor

- XGBoost (Extreme Gradient Boosting) is an optimized distributed gradient boosting library. It uses gradient boosting (GBM) framework at core. It belongs to a family of boosting algorithms that convert weak learners into strong learners. A weak learner is one which is slightly better than random guessing.

- 'Boosting' here is a sequential process; i.e., trees are grown using the information from a previously grown tree one after the other. This process slowly learns from data and tries to improve its prediction in the subsequent iterations.

XGBoost Regressor(Contd.)

Best hyperparameters used:

n_estimators = 300
Learning_rate = 0.08
Max_depth= 7
min_child_weight=4
n_jobs=-1

XGBoost Regressor(Contd.)

- RMSE score for the XGBoost regressor raw data- 297.967
  Variance score- 0.99
- RMSE score for the XGBoost regressor feature selection data with default params- 297.967
- Variance score- 0.99
- RMSE score for the XGBoost regressor feature selection data with tuned params-82.98
- Variance score- 1.00
- RMSE score for the XGBoost regressor PCA- 1719.324
- Variance score- 0.63

There is a significant improvement in the RMSE score for the tuned XGBoost regressor when trained on the feature selection group.

Also, the RMSE score on the raw data and feature selected data are same, which disproves the theory that it is always better to select the relevant features which are statistically important. As the data behaves differently in different models.

Conclusion

In this project we covered various aspects of the Machine learning development cycle. We observed that the data exploration and variable analysis is a very important aspect of the whole cycle and should be done for thorough understanding of the data. We also cleaned the data while exploring as there were some outliers which should be treated before feature engineering. Further we did feature engineering to filter and gather only the optimal features which are more significant and covered most of the variance in the dataset. Then finally we trained the models on the optimum featureset to get the results.

Thank You