Capstone Report

New York City Taxi Trip Duration

Definition

Project Overview: This is a Kaggle competition. It is based on the taxi service in the New York city. The competition 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). The data was sampled and cleaned for the purposes of this playground competition. Based on individual trip attributes, participants should predict the duration of each trip in the test set. Prediction of trip duration is very important they can provide insights on traffic pattens, road blockage, With ride sharing apps gaining popularity, it is increasingly important for taxi companies to provide visibility to their estimated fare and ride duration, since the competing apps provide these metrics upfront. This is academic paper where machine learning was applied to this type of problem.

Ref: http://cs229.stanford.edu/proj2016/report/AntoniadesFadaviFobaAmonJuniorNewYorkCityCabPricing-report.pdf

Problem Statement: The competition is about regression building a model to predict the total trip duration based on given data about past taxi trip duration data. Explore the given data and predict which features are effecting the trip duration most so that they can be used in real cases every day. I treat this as a regression problem.

Datasets and inputs: The dataset for the competition can be found in kaggle platform. The features that are present in the data set are *id, vendor_id, pickup_datetime, dropoff_datetime, passenger_count, pickup_longitude, pickup_latitude,*

dropoff_longitude, dropoff_latitude, store_and_fwd_flag, trip_duration. Here *trip_duration* feature is the target variable. These are very important features one can guess for predicting the trip duration, but there are more important features that are absent like traffic flow density variation with time and route, weather data … which can be helpful in making accurate predictions. So more data can be collected or created and added to the given data if the aim is to generate the best model. There are 1458644 train examples with 11 features in the training data I want to split and use this training data for training and cross validation purposes. *Ref for data:* **https://www.kaggle.com/c/nyc-taxi-trip-duration/data**.

Evaluation Metric: As suggested by kaggle I want to use RMLSE as error metric it is a vey simple evaluation metric and works well even if there are very small and large values for target variable. This metric suits well for this competition

The RMSLE is calculated as :

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$$

Where:

ϵ is the RMSLE value (score)
n is the total number of observations in the (public/private) data set,
p_i is your prediction of trip duration, and
a_i is the actual trip duration for i.
$\log(x)$ is the natural logarithm of x

Analysis

Data Exploration: New York Taxi Duration train data contains nearly 1458644 rows and 11 data columns, Test data contains 625134 rows and 9

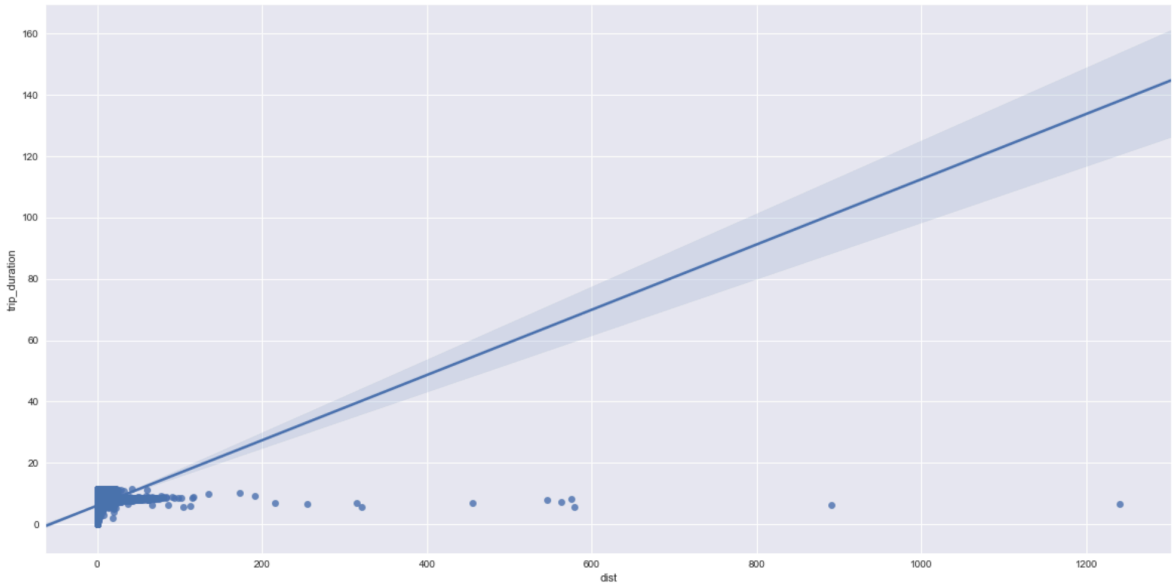
columns ‘dropoff_datetime’, ‘trip_duration’ features are absent in the test data, The column that are present in the data and their description.

Date fields**:**

- 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

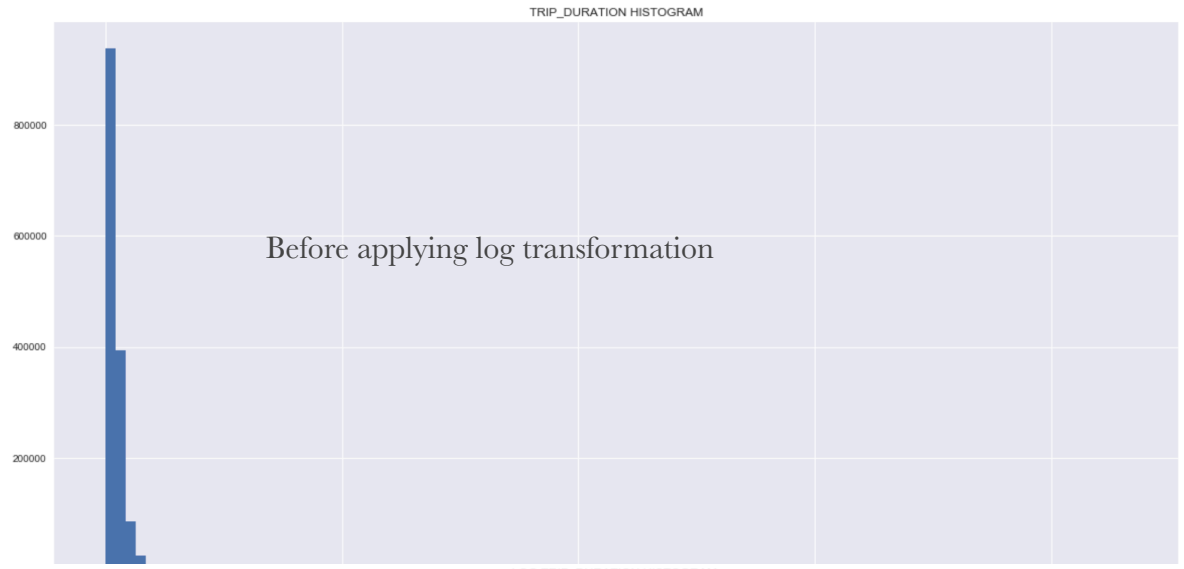
Exploratory Visualisation:

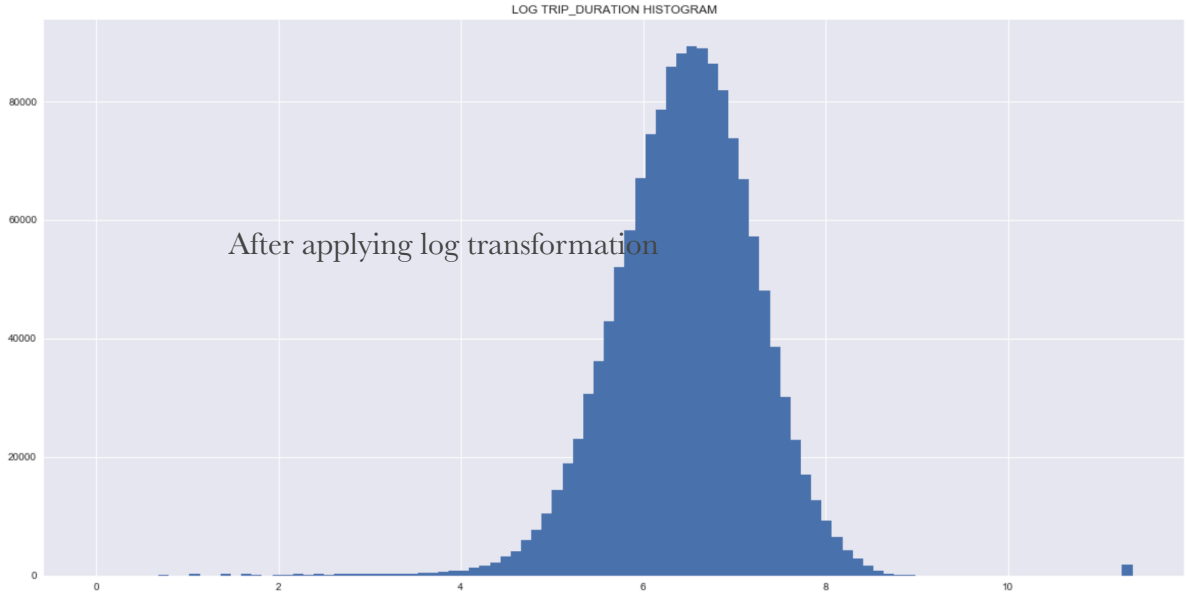
The data doesn’t have any missing values in any fields, All the floating type data fields are reduced to int32 type variables where ever possible in order to reduce the size of data and aid faster calculations. A new data field is creaeted to account for the distance between the pickup and drop-off locations. Many other time related variables are created to indicate weekday, monthday, hour, month these are created form given pickup date time data field, The scatter plot between Duration and Distance fields show that there are 4 data points which have abnormal Duration values those are removed, Distance data field is modified to represent more accurate distances now, After removing those four abnormal data points a better box plot is obtained for Trip Duration. To figure out if the trip duration have any correlation with other features several plot are drawn for Duration vs Distance, Passenger count, hour, day, weekday, of all the above features only Distance seemed to show some positive correlation



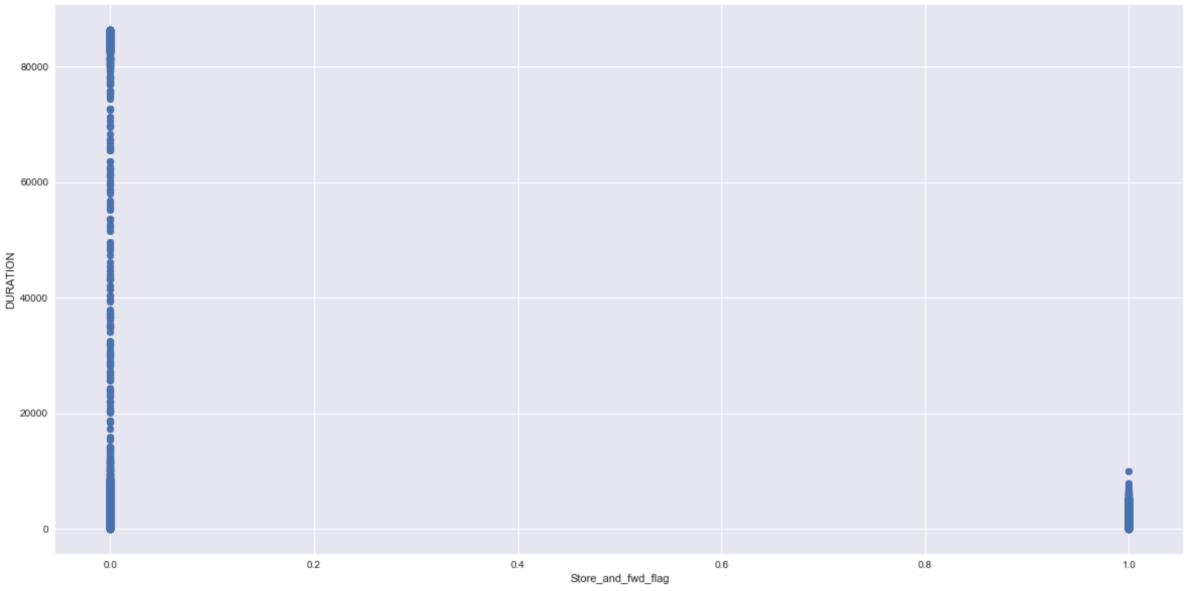
Trip Duration Vs Distance of travel

with Trip duration, It is found that the data field for trip duration was not normally distributed but it was log normally distributed so log transformation was applied to it in order to make it normally distributed. It is clearly evident from the below pictures that after applying log transformation data field was normally distributed.





It is also seen that store_and_fwd_flag data field is noted for very few data observations, plot below depicts this.



Algorithms and techniques:

Two algorithms I want to use for this project are Linear regression and XGBoost Regression algorithms, Since the trip duration is a continuous entity a regression model must be used, Linear regression model is easy to interpret and does not require much hyper parameter tuning, XGBoost Regression algorithm is used because it provides very good score and a chance to fine tune hyper parameters, It is a very sophisticated yet flexible algorithm.

Benchmark

For a baseline bench mark model linear regression model can be considered . For a best achievable model there are many great kernels in kaggle platform for this competition which can serve as for a best achievable model. I want to use this very good kaggle kernel by Beluga as evaluation or reference: https://www.kaggle.com/gaborfodor/from-eda-to-the-top-lb-0-367, The model presented is very sophisticated and is on top of the leader board. This model helps me in evaluating my performance to check what features need to be added to the data to improve the performance, the model in the kernel achieves a score of 0.37 which is very good I want to take help from few feature engineering techniques to see if my XGBoost model score better.

Methodology

Data Preprocessing:

A new data field was created to account for distance form pickup and drop-off locations using pickup and drop-off coordinates this feature is created because it is quite common to expect trip duration depending on distance of journey, All the floating type data fields are reduced to int32 type variables where ever possible in order to reduce the size of data and aid faster calculations, Many other time related variables are created to indicate weekday, monthday, hour, month these are created form given pickup date time data field, these time containing variables are created because a trip duration may be effected by hour of a day, or day of a week.

Implementation:

A scorer function is made for calculating RMLSE, The train data is further splitted in to train and test data, Linear regression model with default parameters is implemented on that train data using all the preprocessed and created features, when evaluated on test data this linear regression model yielded a RMLSE score of 0.6527. XGBoost regression model with sklearn API was implemented using GridSearchCV and 5 fold cross validation for tuning `max_depth, n_estimators, learning_rate` parameters, best values for these parameters as obtained by GridSearchCV are `max_depth = 4, n_estimators = 400, learning_rate = 1.` Now the XGBoost regression model is implemented using the above values for parameters and RMLSE error score obtained now is 0.4148 which is a very good improvement from base line bench mark linear regression model.

Refinement:

As I stated I used the kaggle kernel to see some way to do better feature engineering for that inspiration I added one more distance feature for manhattan_distance, direction of journey/travel is implemented by using a feature called bearing_array an additional time related feature called weak_hour is added, and Principal Component analysis is applied for pickup and drop-off coordinates features, pickup and drop-off coordinates are plotted before and after applying PCA to visualise the transformation. After doing these feature transformation techniques an XGBoost regression model using those features and the test data score was improved to 0.4121

Results

Model Evaluation and Validation:

All the models created in this project are evaluated by using RMLSE error metric, 5 fold cross validation was used along with GridSearchCV for tuning `max_depth, n_estimators, learning_rate` parameters for XGBoost Regression model, best values for these parameters as obtained by GridSearchCV are `max_depth = 4, n_estimators = 400, learning_rate = 1,` this cross validation technique will account for robustness of model's solution by reducing variance in evaluation. For the final model these same values are used for hyper parameters.

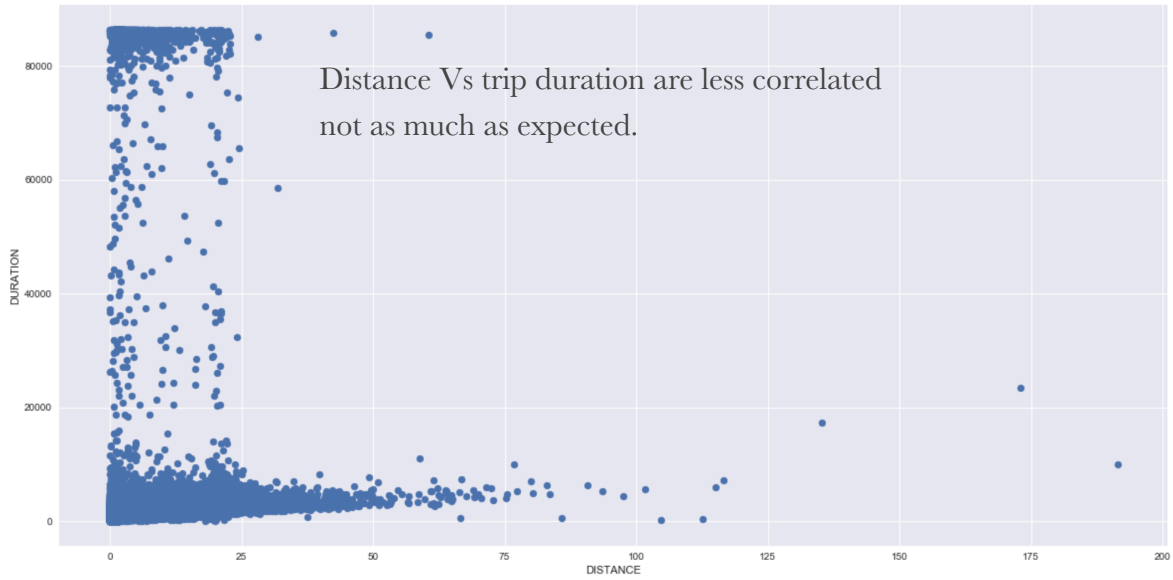
Justification:

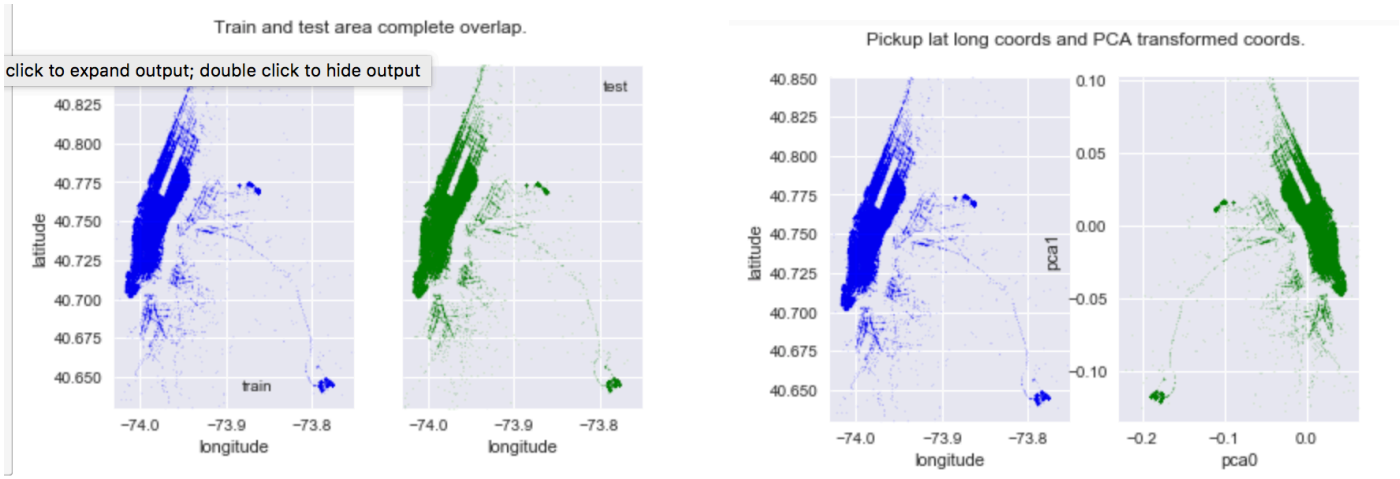
Final obtained result for RMLSE error is 0.4121, this is a very good improvement compared to base line benchmark linear regression which gave an RMLSE of 0.6527, but the final model RMLSE is slightly less than optimal bench mark model(Kaggle kernel model) RMLSE score of 0.37 but to obtain that score of 0.37 more external data need to be added which is huge in size and requires huge computational resources, but 0.4121 is very close enough to 0.37 , But with the available data I would say that the model solution with RMLSE of 0.4121 significant enough to have adequately solved the problem of predicting New York city taxi trip duration prediction.

Conclusion

Free-Form Visualisation:

It is interesting to observe how trip duration is very less correlated with the distance between pickup and drop off locations, because it is very fair to guess that trip_duration increases with distance of travel, may be because trip_duration is more influenced by traffic flow density and weather, It is also quite interesting for how PCA is exactly mirror image like transform the pickup coordinates





Its amazing how PCA is able to make such transformation of coordinates

Reflection:

The process used for this project can be summarised using the following steps:

1. An initial problem and relevant ,public datasets were found.
2. The data was downloaded and preprocessed and visualised.
3. Evaluation metric as created and benchmark model specified.
4. Relevant Feature Engineering was done.
5. Regression model was trained and hyper parameters were tuned

I found it interesting how XGBoost super out performs Linear regression or any model for that matter, Feature engineering was the difficult part in any machine learning project I also found it difficult.

Improvement:

The solution be improved by using external data related to weather, traffic flow intensity and trip duration. By tuning more hyper parameters and using more computational resources like AWS. It can observed form the Kaggle leader board for the competition where top performers used same techniques to improve the solution. By using above improvements we can better predict taxi trip duration than current prediction.