Machine Learning Engineer Nanodegree

Capstone Project - NYC Taxi Trip Duration Prediction

Kritika Rai

I. Definition

Project Overview

This project is for predicting NYC Taxi trip duration using the dataset released by the NYC Taxi and Limousine Commission, which includes pickup time, geo-coordinates, number of passengers, and several other variables.The dataset was obtained from Kaggle and can be found here https://www.kaggle.com/c/nyc-taxi-trip-duration/data.

Problem Statement

We are trying to predict trip duration of taxi rides in New York City. Our output variable (Trip duration) is continuous, so we will set up the model as a regression problem.

Metrics

1) Evaluation metric for the kaggle competition is RMSLE (Root Mean Squared Logarithmic Error), so we convert our dependant variable “Trip Duration” into “Log Trip Duration” and use RMSE as the evaluation Metric.

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

2) We also look at R squared as a secondary metric to understand the goodness of fit of the model.

$$R^2 = \frac{\text{Variance Explained By the model}}{\text{Total Variance}}$$

II. Analysis

Data Exploration and Visualization

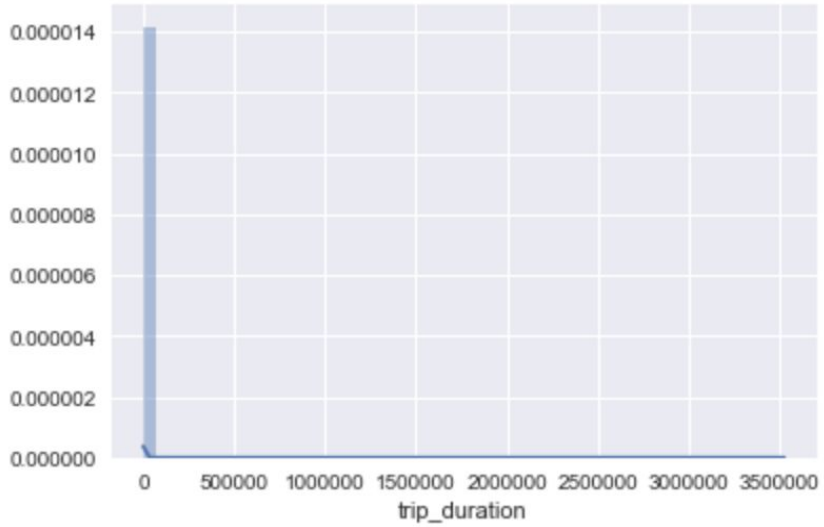
Attributes of the dataset:

- 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

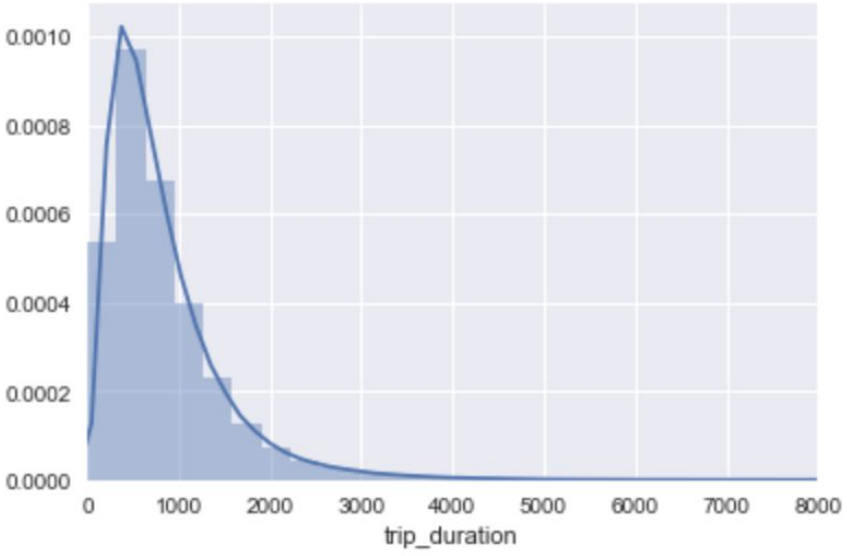
Removing Outliers

In the distribution below, we see that there are some outliers in the dataset (see X Axis is stretched to 3500K seconds). We remove all trip duration values greater than 3 standard deviation from the mean.

Distribution of Trip Duration(seconds) **Before** Removing Outliers



Distribution of Trip Duration(seconds) **After** Removing Outliers



Algorithms and Techniques

Gradient Boosting Regression

We apply gradient boosting regressor using sklearn to predict trip duration. Gradient boosting regression is an ensemble supervised learning method and builds an additive model of decision trees by selecting a tree that optimizes the objective function.

Cross Validation

We perform 4 fold cross validation with gradient boosting regression on our dataset to evaluate model performance. This helps ensure that the model is not overfitting and generalizes well to unseen data.

XGBoost
We also use XGBoost to try to improve model performance. XGBoost is heavily used in Kaggle competitions as it's efficient and scalable and provides robust solutions. A great introduction to xgb can be found here https://xgboost.readthedocs.io/en/latest/model.html#tree-boosting.

Benchmark

We use Kaggle's ranking mechanism as a benchmark for this project. The winning team had a RMSLE of 0.28976 and we will try to beat this benchmark.

III. Methodology

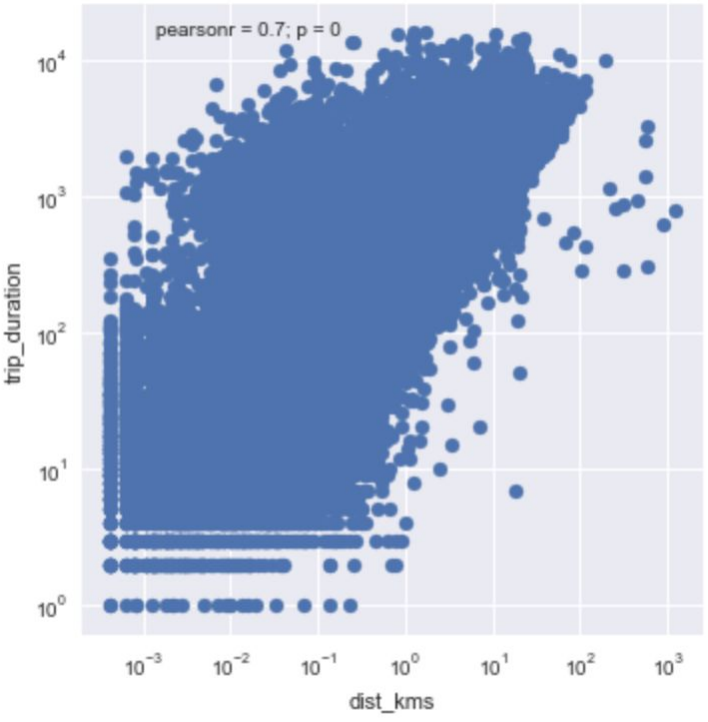
Data Preprocessing

We extract several features from the model and the descriptions are provided below.

Distance

We calculate trip duration using the haversine distance formula. Haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes. More information about haversine formula can be found here. We see that the correlation b/w distance and trip duration is really high (0.7), hence distance should be an important feature for prediction.

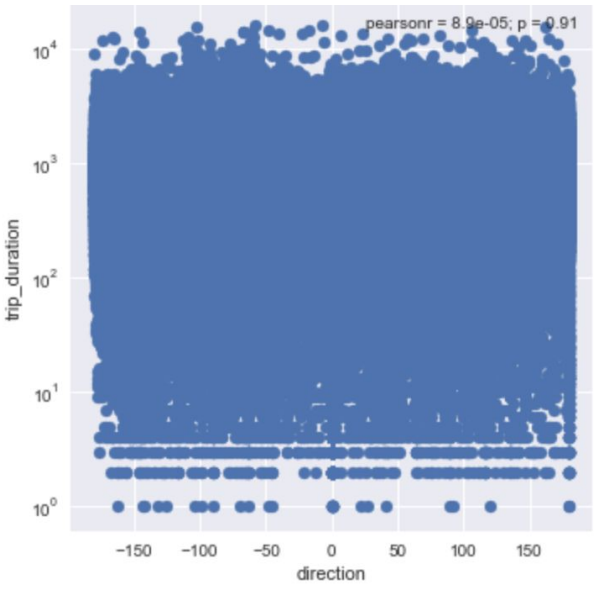
Corr Between Trip Duration and distance (kms)



Direction

We calculate direction of trip using the bearing distance formula. The bearing of a point is the number of degrees in the angle measured in a clockwise direction from the north line to the line joining the centre of the compass with the point. http://www.mathsteacher.com.au/year7/ch08_angles/07_bear/bearing.htm. We see that there is no correlation b/w duration and direction, but we'll still keep it in the model and the reason for that shall be seen later.

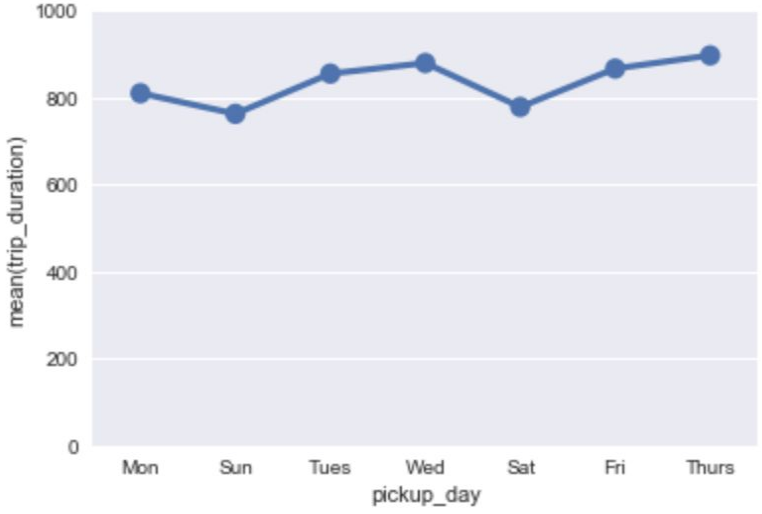
Corr Between Trip Duration and Direction



Day of the week

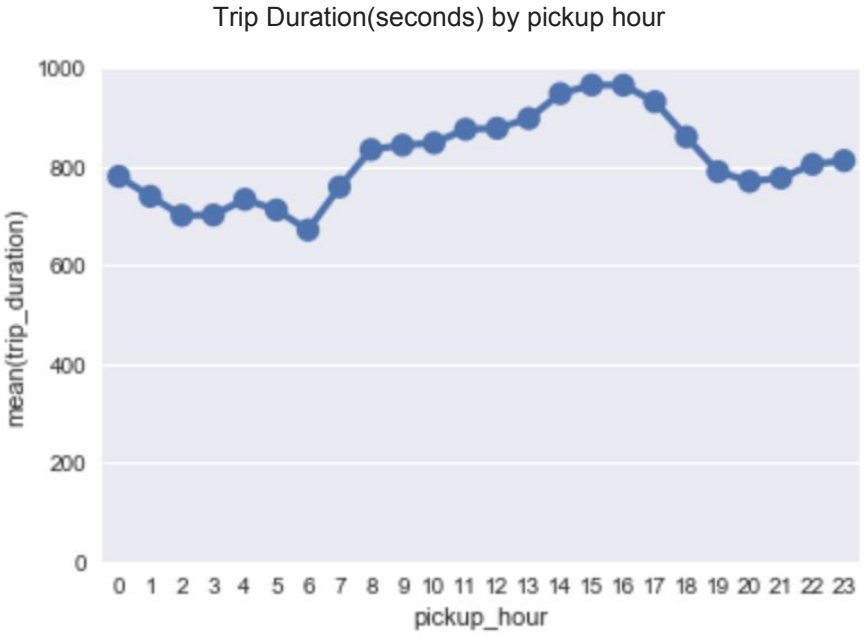
In our exploration, we see that trip duration on Sat and Sun is a lot shorter than trips on Mon-Fri. To capture this variability in the model, we use pickup_day as a feature.

Trip Duration(seconds) by pickup day



Time of day

We also see variability in trip duration by time of day, and use pickup_hour as a feature to capture this variability.



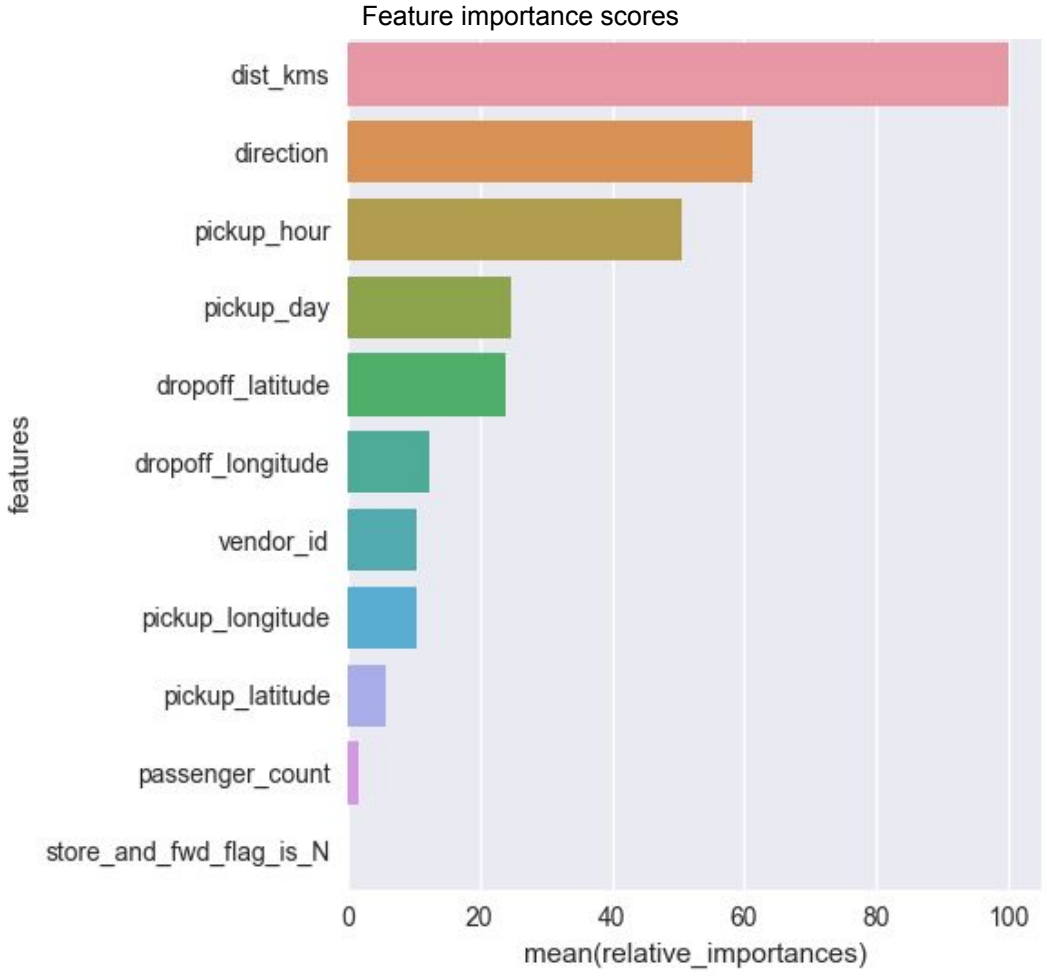
Features Selected For Model Development

Vendor_id - Given
Pickup_hour - Extracted from pickup_datetime
Pickup_day - Extracted from pickup_datetime
Distance - Extracted from pickup and dropoff lat and long
Direction - Extracted from pickup and dropoff lat and long
Passenger_count - Given
store_and_fwd_flag_is_N - Boolean extracted from store_and_fwd_flag
Pickup_longitude - Given
Pickup_latitude - Given
Dropoff_longitude - Given
Dropoff_latitude - Given

Implementation

Gradient Boosting Regression

A tree based approach gives us feature importance scores which are useful in understanding which features are actually useful to determine variability in the output variable. We see that distance and direction are the two most important predictors of trip duration. Even though direction wasn't correlated with trip duration, it is the second most important predictor of trip duration after distance. The model is capturing some non linear relationship between trip duration and direction.



With Gradient boosting regression, we get a RMSLE of 0.39 for the validation set and 0.43 for Kaggle's test dataset.

Gradient Boosting Regression Results

Mean R squared (Cross validated)	**Mean RMSLE (Cross Validated)**	**Kaggle Test Dataset RMSLE**
74.58%	0.3911	0.438

Refinement

We refine the subsample and learning rate for gradient boosting regression to account for overfitting. The results shown above for GBR were obtained after adjusting subsample and learning rate. We keep all the features in the model because running the model with the features given above is fairly quick. We try to use another model(xgboost) to see if that can improve model performance.

IV. Results

Model Evaluation and Validation

To try to improve our model performance, we use xgboost Regressor since one of the best results in the Kaggle competition were obtained using xgboost. Xgboost uses a more regularized model formalization than regular gradient boosting to control overfitting.
We get slight improvements in our model with XG Boost, but since the results are better than GBR, we settle with xgboost as the final model.

XG Boost Results

Training Dataset RMSLE	**Validation Dataset RMSLE**	**Kaggle Test Dataset RMSLE**
0.330	0.375	0.407

Justification

XGBoost's Test set RMSLE puts us at around rank ~550 out of ~ 1200 in the kaggle. Even though it is not very close to the winning solution, it provides a reasonable RMSLE score. There isn't a big difference between our Validation set RMSLE and Kaggle's Test set RMSLE, which means that the model generalizes well.

V. Conclusion

Reflection

This project involved data processing, exploration, outlier removal, heavy feature engineering, cross validation and model building. Most important part of this project was feature engineering. Largest increase in model RMSLE and R squared was seen by adding the feature "direction" to the model. Although, direction has very poor correlated with trip duration, as seen in feature importance scores, it is the second most important predictor of Trip Duration. This means that the model was able to capture nonlinear relationships in the dataset well.

Improvement

Several improvements could have been made to the model by:

- Getting more data about NYC from public data sources, we could use neighborhoods as a feature, or avg speed per neighbourhood etc. which could helped increase the model RMSLE and Rsquared.
- We could try to do a wider hyperparameter search on more parameters such as number of tree estimators used, max_depth etc.
- We could try to implement a neural network as well as they are well known to perform well when there are nonlinear relationships between inputs and output variable.

In general, more features, a more complex model and wider hyperparameter search could help improve the model performance.