ISSN No:-2456-2165

Route Agnostic Estimated Time of Arrival in Vehicle Trip Using Machine Learning

YMPHAIDIEN SUTONG
North-Eastern Hill University, Shillong

RAMNEET S. CHADHA
C-DAC, Noida

Abstract:- The transportation industry is an important industry sector in the economy that deals with the movement of people, goods and products. As vehicles become safer and more efficient, most individuals and companies adopt vehicles for midrange traveling, goods and product transportation and hence, opportunities to advance transportation abound. To make the most of them, we need to explore and develop different technology options. In this study, we explore the potential of Artificial Intelligence in predicting the Estimated Time of Arrival. Our method is by modelling historical-data based models. We find that several Nonlinear Machine Learning Regression Algorithms like Gradient Boosting Regressor, Random Forest Regressor, Light Gradient Boosted Machine, etc are suitable for this problem and are producing promising results in terms of RMSE and R2. Out of which the LightGBM model performs best.

Keywords:- *Estimated Time of Arrival, Regression, Gradient Boosting, Random Forest, Artificial Intelligence, Transportation, Light Gradient Boosted Machine.*

I. INTRODUCTION

Estimated Time of Arrival is one of the main attributes that could solve vast numbers of problems related to transportation and optimal traffic planning studies concluded that machine learning models outperform other approaches in terms of prediction accuracy. Most used methods for that purpose appear to be ANN, Support Vector Regressor and Tree based Regression Algorithm. Currently, some researchers focus on a path-based approach in which the hypothesis is that the total travel time of a vehicle traffic is highly dependent on the path.

In this study we approach the problem using Tree Based regression Algorithms like Gradient Boosting Regressor, Random Forest Regressor and Light Gradient Boosted Machine with Hyper Parameter tuning and extensive data pre-processing.

A well-designed prediction model for the ETA of a vehicle trip is an essential ingredient to make decisions that can optimize traffic planning and customer satisfaction. The main aim of this study is to show how efficient AI and Machine Learning are in the Fields of Estimated Time of Arrival.

II. METHODOLOGY AND EXPERIMENTS

A. Data Overview

The dataset was obtained from TLC Trip Record Data [4]. These taxi trip records include fields capturing pick-up and drop-off dates/times, pick-up and drop-off location coordinates, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts. The dataset contains *1458644* entries with *9* attributes.

B. Data Cleaning

The data obtained from data sources were channeled through pre-processing pipelines with the ultimate aim to optimize the model performance and accuracy. This process includes removal of duplicate, missing values and treating outliers accordingly.

C. Removal of outliers

Outliers were observed in *trip_duration*, *passsenger_count*, pickup and drop-off locations after a statistical analysis was done on the dataset. The *trip_duration* which is greater than 5900 seconds was removed. Trips where *passenger_count* = 0 were also removed. After visualizing the pickup and drop-off locations, positional outliers were observed in the *pickup_longitude* and *pickup_latitude* of the dataset in which *pickup_longitude* < -100 and *pickup_longitude* > 50 were removed.

D. Feature Engineering

The *trip_duration* data was visualized where the distribution was seen right-skewed, LogTransformation on the skewed column was done to approximately conform the data to normality.

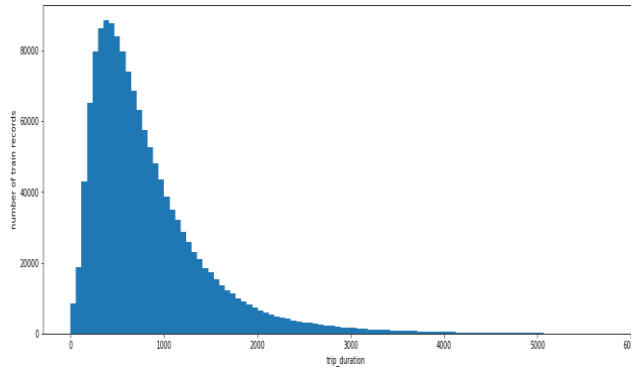

Fig 1: *Right skewed trip_duration distribution*

One-hot encoding was done on binary and categorical features to prepare it for the model to get a better prediction. We convert the datetime which are on YYYY-MM-DD HH:MM:SS format into individual features of months, week, weekday, hour and the minute of the day. Using the the values from the feature pickup and drop-off latitude and longitude coordinates, the distance feature was created using the *haversine* formula, for a given distance *D*, it is given by

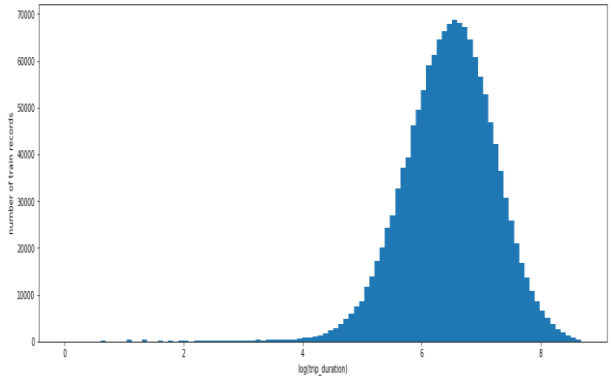

Fig 2: *Normalised trip_duration distribution*

$$D = 2r \times \arcsin\left(\sqrt{A + B}\right) \quad (1)$$

where, $A = \sin^2\left(\frac{\phi_1 - \phi_2}{2}\right)$

$$B = \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\alpha_1 - \alpha_2}{2}\right)$$

and ϕ_1, ϕ_2 are the latitude of pickup location and latitude of drop-off location in radians and α_1, α_2 are the longitude of pickup location and longitude of drop-off location in radians.

A direction feature was created using the formula

$$direction = arctan2(y, x) \quad (2)$$

where $x = \arctan(\sin(\alpha_1 - \alpha_2)) \cdot \cos(\alpha_2)$ and $y = cos(\alpha_1) \cdot sin(\alpha_2) - sin(\alpha_1) cos(\alpha_2 \cdot \cos(\alpha_1 - \alpha_2))$ and ϕ_1, ϕ_2 are the latitude of pickup location and latitude of drop-off location in radians and α_1, α_2 are the longitude of pickup location and longitude of drop-off location in radians.

The speed was created using the simple well-known formula given by

$$speed = \frac{distance}{time} \quad (3)$$

We finally split the data into two parts, features and target for training purposes, where the target is the *trip_duration* variable.

E. Model Selection

For this specific problem, we'll measure the error using the Root Mean Square Error (*RMSE*), Coefficient of Determination (R^2) and *Time* for evaluation and comparison.

Root Mean Square Error (*RMSE*): It is the standard deviation of the prediction errors i.e residuals. It gives you a relatively high weight to the large errors. A smaller value means it is a better model. It can be expressed as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y(i) - y'(i))^2}{N}} \quad (4)$$

where,
N is the number of data points, $y(i)$ is the i^{th} measurement and $y'(i)$ is its corresponding prediction.

The Coefficient of Determination (R^2): It is the proportion of variance in the dependent variable that can be told by the independent variable. It is used to measure the goodness of fit. A larger value means that it is a better model. It can be expressed as

$$R^2 = 1 - \frac{RSS}{TSS} \quad (5)$$

where,
RSS = sum of squares of residuals and
TSS = total sum of squares.

The *Time* is the time in which the algorithm takes to train a model.

F. Models

The entire code and implementation are written in the Python Programming Language. Python libraries like Pandas, Numpy load the dataset and perform the mathematical calculations on the dataset. Sklearn[3] is used to implement the three different machine learning algorithms. Matplotlib is needed to visualize the data in an interactive way. As introduced earlier, we choose three tree-based regression algorithms, which were implemented in Sklearn, namely, *Gradient Boosting Regressor*, *Random Forest Regressor* and *Light Gradient Boosting Machine*.

G. Cross Validation on the LightGBM model

Cross-validation is one of the most widely used data re-sampling methods to assess the generalization ability of a predictive model and to prevent overfitting [2]. To further test whether our LightGBM Model was stable, a *k*-Fold Cross Validation was done on our LightGBM Model with $k = 5$ to estimate the true prediction error of the model.

H. Hyperparameters Tuning

The process of hyperparameter tuning [1] (also called hyperparameter optimization) means finding the combination of hyperparameter values for a machine learning model that performs the best - as measured on a validation dataset - for a problem. In this study we perform *Randomized Search* on hyper parameters to find the best parameters. Typically, random search algorithms sacrifice a guarantee of optimality for finding a good solution quickly with convergence results in probability.

III. RESULTS

Out of the three models, LightGBM gave the highest R^2 score and lowest *RMSE* with the shortest amount of time to train the model. Upon Cross Validation we found that our LightGBM model is stable and hence a clear winner. The Table 1 presents an evaluation of our approach through Root

ISSN No:-2456-2165

Mean Square Error (*RMSE*), Coefficient of Determination (R^2) and Time comparisons.

Models	***RMSE***	***R^2***	***Time***
Gradient Boosting	0.333	0.186	*9mins 11s*
Random Forest	0.103	0.983	*25mins 48s*
LightGBM	0.056	0.994	*2min 53s*

Table 1: *Baseline results for ETA prediction on **TLC Trip Record Data** across three different models.*

The LightGBM model shows in Figure 3 that some features are more important than the others. It is seen that the time of the day and the pickup location is few of the most important features in predicting ETA using LightGBM models.

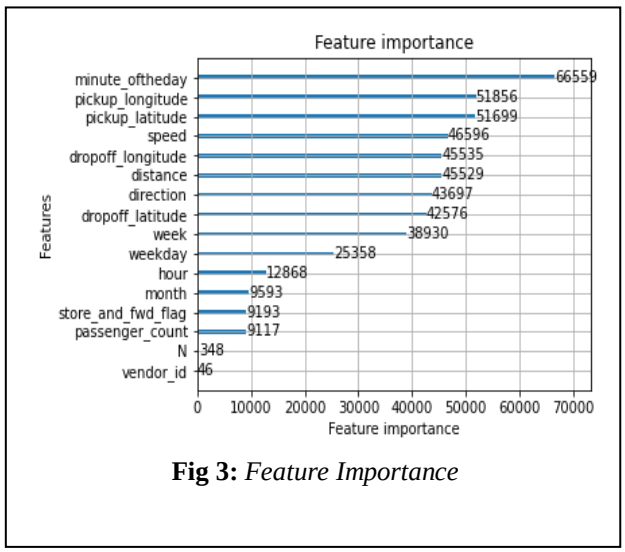

Fig 3: *Feature Importance*

IV. CONCLUSIONS

Machine learning has applications related to the prediction of stock price, customer sentiment analysis, etc. In order to predict the ETA of a vehicle trip, we need the historical data of trips. This paper analyses the Data Set with 9 attributes and makes a prediction using different regressors to get the ETA. It can be seen that the highest accuracy is obtained using the LightGBM model with the R^2 error being 0.99 and the *RMSE* being 0.05. We can summarize by saying that the LightGBM gave the best results out of all the models used for the ETA prediction.

REFERENCES

[1]. James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.
[2]. Daniel Berrar. Cross-validation., 2019.
[3]. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel,Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
[4]. TLC Trip Record Data, https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page