Machine Learning Engineer Nanodegree

New York City Taxi Fare Prediction

Capstone Proposal

Dip Narayan Gupta

May 7, 2020

Domain Background

New York City taxi rides paint a vibrant picture of life in the city. The millions of rides taken each month can provide insight into traffic patterns, road blockage, or large-scale events that attract many New Yorkers. With ridesharing apps gaining popularity, it is increasingly important for taxi companies to provide visibility to their estimated fare , since the competing apps provide these metrics upfront. Predicting fare of a ride can help passengers decide to continue with ride based on predicted price, or help drivers decide which of two potential rides will be more profitable. In order to predict fare, only data which would be available at the beginning of a ride was used.Linear regression with model selection, random forest models and Gradient Boosting were used to predict fare amount.

Problem Statement

The New York City Taxi Fare prediction challenge is a supervised regression machine learning task. Given pickup and dropoff locations, the pickup timestamp, and the passenger count and other features the objective is to predict the fare of the taxi ride. Like most Kaggle competitions, this problem isn’t 100% reflective of those in industry, but it does present a realistic dataset and task on which we can apply our machine learning skills.

Datasets and Inputs

File descriptions

train.csv - Input features and target fare_amount values for the training set (about 55M rows) and **we are using 600000 instances.**

test.csv - Input features for the test set (about 10K rows). Goal is to predict fare_amount for each row.

Features

key – Unique string identifying each row in both the datasets.
pickup_datetime - timestamp value indicating when the taxi ride started.
pickup_longitude - float for longitude coordinate of where the taxi ride started.
pickup_latitude - float for latitude coordinate of where the taxi ride started.
dropoff_longitude - float for longitude coordinate of where the taxi ride ended.
dropoff_latitude - float for latitude coordinate of where the taxi ride ended.
passenger_count - integer indicating the number of passengers in the taxi ride.

Target

fare_amount - float dollar amount of the cost of the taxi ride. This value is only in the training set; this is what you are predicting in the test set.

Solution Statement

The problem is supervised Learning and training data is numeric. To start the first thing to do is data exploration to analyse data whether data has missing values or any outlier. Then data preprocessing technique is applied to remove erroneous data and important features are extracted. After this, modelling is performed to find best model with hyperparametres with GridSearchCV. Evaluating model with root mean square error metric and finally we will apply selected model to predict the test data. One possible solution to accomplish is to use scikit-learn built-in libraries, numpy, pandas, matplotlib.

Benchmark Model

As gradient Boosting model is the winner of this Kaggle problem with root mean square error of 1.38506. The good Benchmark Model will to get less root mean square error against the midmost competitors in leaderboard. Furthermore multple models will be evaluated as followes:

- Linear Regression
- RandomForestRegressor
- DecisionTree Regressor
- LightGBM
- Xgboost

Evaluation Metrics

The evaluation metrics will be simply root mean square error.

Project Design

To solve this problem, we'll follow a standard data science pipeline plan of attack:

1. Understand the problem and data: It describes the main objective to solve problem.

2. Data exploration / data cleaning: It is defined as the data visualisation and preprocessing data to remove erroneous data.

3. Feature engineering: Here we apply feature selection technique to find the best feature.

4. Model evaluation and selection: Here each model will be evaluated with cross validation using GridSearchCV and best model will be selected.

5. Interpretation of results and predictions: Finally we predict test data and interpret the result by root mean square error as metrics.