Capstone Project 1:

New York City Taxi Fare Prediction

Milestone Report

Mehrnaz Mireslami

1. Problem Statement

In big cities such as New York City, a huge number of taxi rides is taken per day. As the popularity of app-based vehicle hiring services grows, accurate prediction of taxi fare is essential for enhancing customers' satisfaction, since it is given as upfront data to the customers. There are many factors that should be considered such as the pickup time, pickup or dropoff locations, etc. in predicting taxi fare. Providing accurate taxi fare at a specific time enables both drivers and customers to decide whether to select the rides or not. The goal of this project is developing a Machine Learning (ML) based model to predict the fare amount for a taxi ride in New York City while some data such as the pickup and dropoff locations are given.

Predicting accurate taxi fares yields better results for taxi cab and ridesharing companies such as Uber, Lyft, etc. Also, this project can be used in traffic congestion prediction and autonomous vehicle research to develop accurate traffic models and choose the fastest and less congested routes.

2. Description of the Dataset

The data from a Kaggle competition is used for this project (https://www.kaggle.com/c/new-york-city-taxi-fare-prediction/overview).

The dataset for this project includes the features explained below:

- 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.

During the modeling phase of the project, these features can be extended.

Target:

- fare_amount : dollar amount of the cost of the taxi ride.

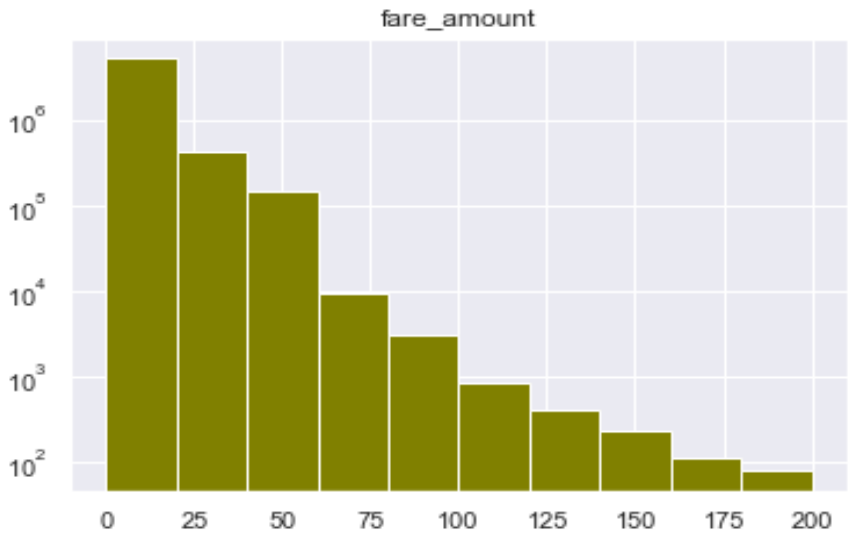
▢ Data Wrangling

This dataset obtained from Kaggle competition is almost clean. However, data analysis should dive deeper into the dataset to find potential data patterns, and extract new features to be able to model the problem.

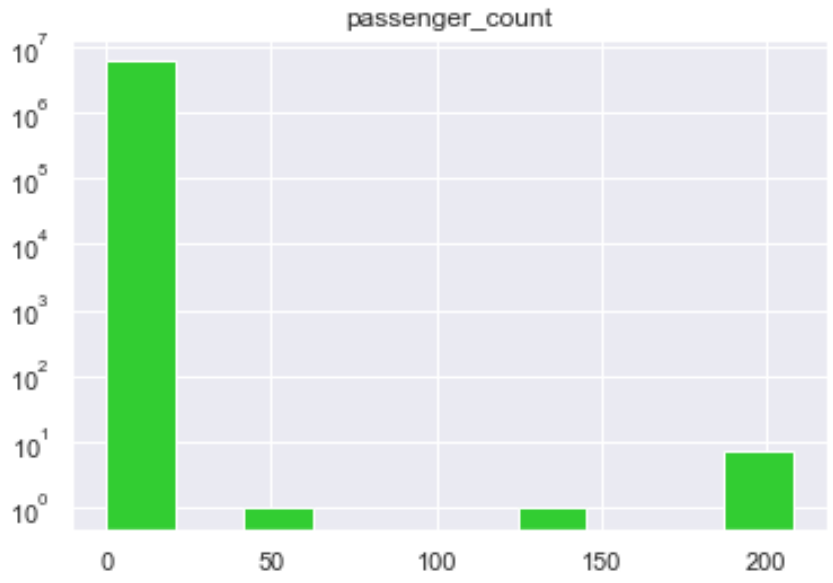
Considering that a lot of memory is required to read the dataset (which is 55M rows), first a limited number of rows is read (6,000,000). After finishing exploration code, the Jupyter Notebook will be rerun with the whole train dataset.

As the first step, rows with NaN values should be removed to make sure there are no incomplete data entries. In the next step, outliers should be found and removed from features. THen, new features are extracted by going through the target and the input set of features.

Fare Amount: Fare_amount with negative and zero values does not seem to be realistic and should be removed from the dataset. Based on the distribution of fare_amount which is shown in below, the fare_amount with greater than 200$ should be considered as outlier and dropped from the dataset.

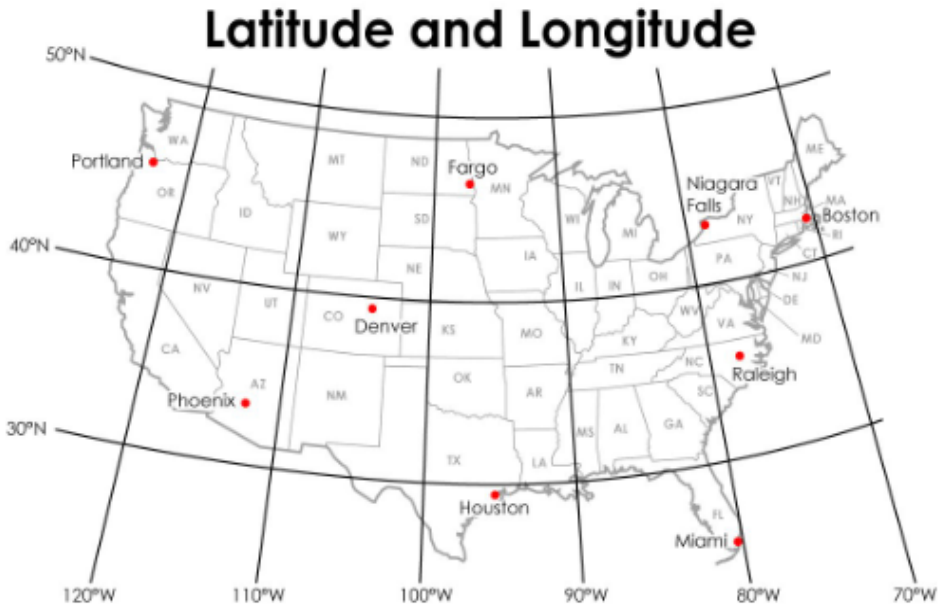


Passenger Count: The histogram of passenger_count shows passenger_count distribution. Maximum number of passengers is 208 that is not realistic for the number of seats on a taxi cab. The maximum allowed passengers for an SUV or a Van is 6. So, 6 is considered as an upper bound for the number of passengers in each ride.



Latitude and Longitude Features: In order to analyze latitude and longitude features, coordinates of NYC should be considered as boundaries.
The latitude and longitude range for NYC:

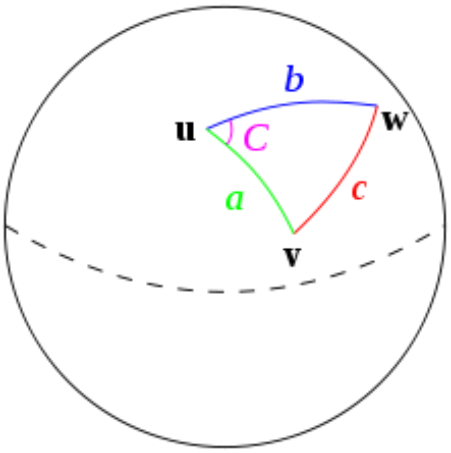
- The NYC's latitude is in the range of (40, 42)
- The NYC longitude is in the range of (-76, -71)



▢ Deriving New Features

In the next step, new features will be created based on the available data to see whether the new extracted features affect the fare_amount or not.

Distance Between Pickup and Dropoff Locations: Usually the distance between pickup and dropoff locations has great impact on fare_amount. Haversine formula is employed to calculate the distance between pickup and dropoff locations based on longitude and latitude.



The Haversine formula is (https://en.wikipedia.org/wiki/Haversine_formula):

Distance = 2 * r * arcsin(sqrt (sin((latitude2 - latitude1) / 2.0) ^ 2 + cos(latitude1) * cos(latitude2) * sin((longitude2 - longitude1) / 2.0) ^ 2))

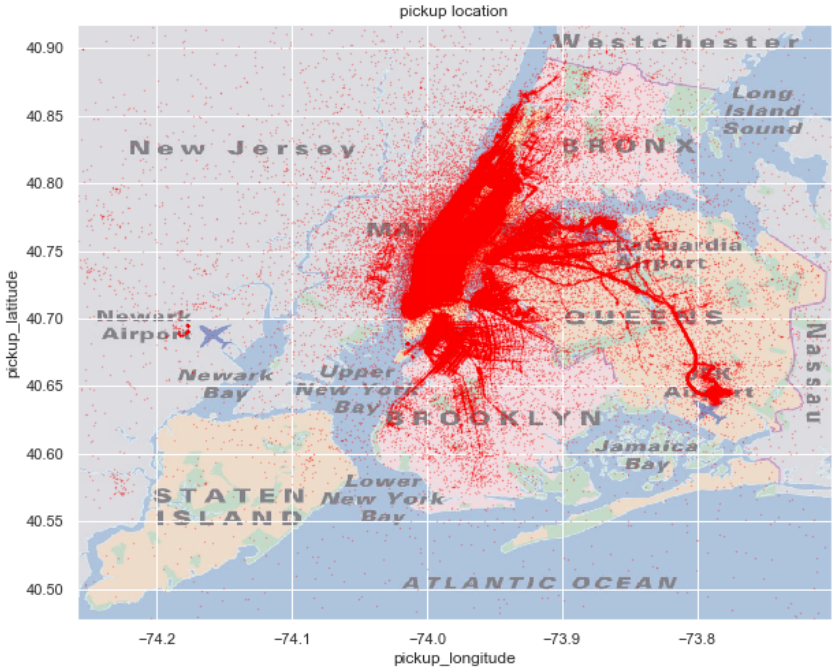
After creating the new feature that is named distance, it should be analyzed to remove any possible outliers. For example, distance zero does not make sense.

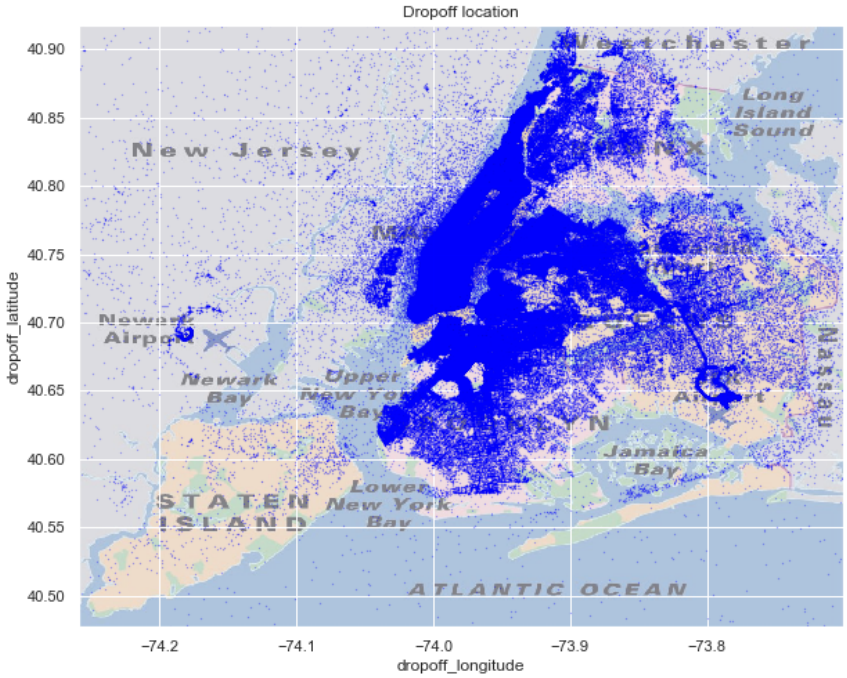
Time and Date Features: Days of the week can also affect the fare_amount. To explore it precisely, the type of the key column should be converted to datetime type to create new year, month, day, dayofweek, and hour columns. Since the 'key' and 'pickup_datetime' columns are the same, the 'key' column should be removed to get rid of duplicated data

3. EDA and Data Storytelling

Through exploratory data analysis, the correlation among data can be represented by visualizing data and performing inferential statistics that will be explained in detail in the following subsections.

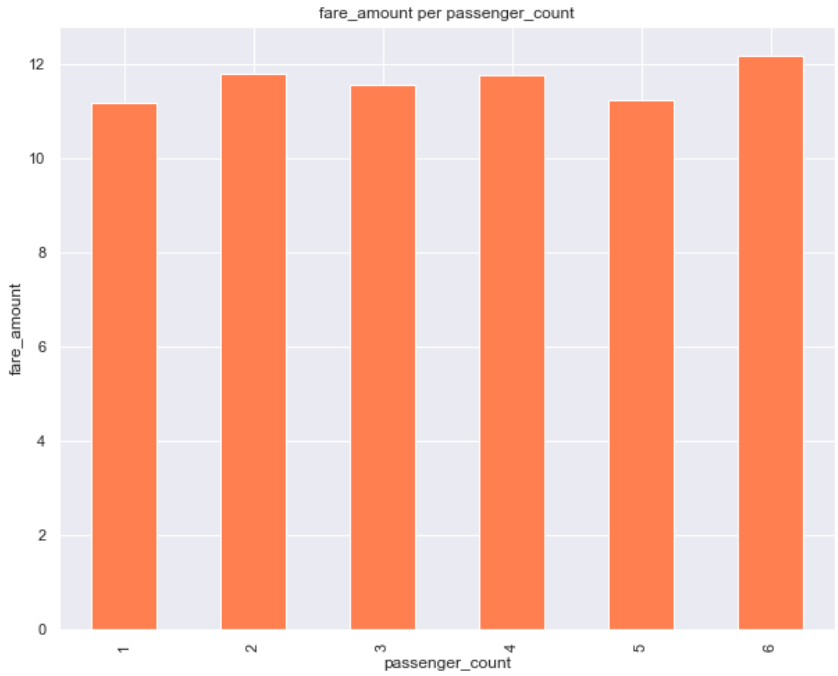
Pickup and dropoff visualization: In order to have a better view of location data, the coordinates are plotted on a NYC map for pickup and dropoff separately. As shown in these plots, the concentration of coordinates are related to Manhattan neighborhoods, since this area is considered as the crowded part of NYC.


pickup location
Westchester
Long Island Sound
New Jersey
BRONX
LaGuardia Airport
QUEENS
Newark Airport
Newark Bay
Upper New York Bay
BROOKLYN
Jamaica Bay
Nassau
Lower New York Bay
STATEN ISLAND
ATLANTIC OCEAN
40.90
40.85
40.80
40.75
40.70
40.65
40.60
40.55
40.50
pickup_latitude
−74.2
−74.1
−74.0
−73.9
−73.8
pickup_longitude

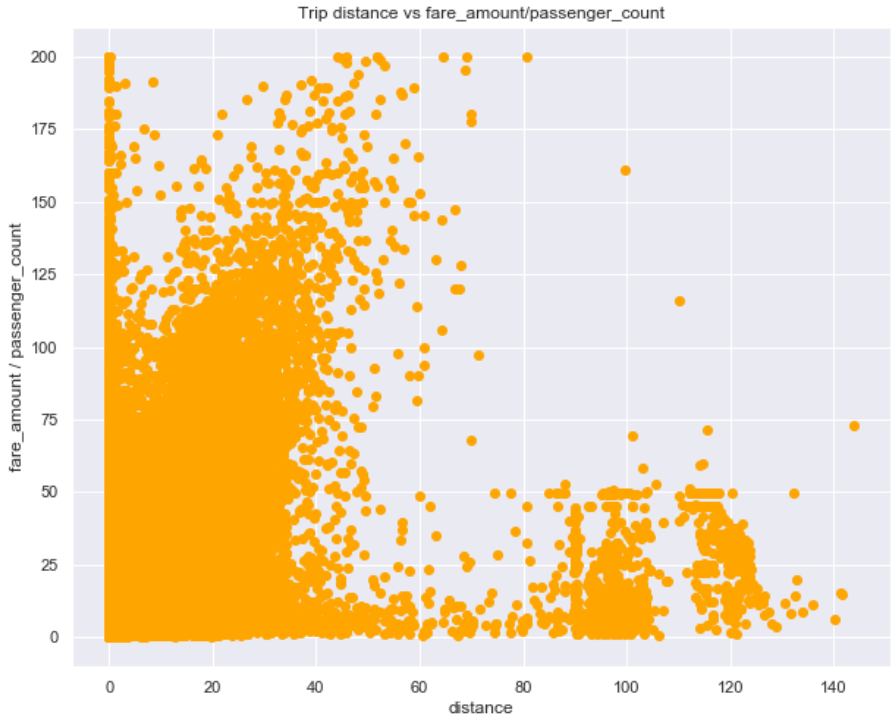

Dropoff location
Long Island Sound
New Jersey
Newark Bay
Jamaica Bay
Lower New York Bay
STATEN ISLAND
ATLANTIC OCEAN
40.90
40.85
40.80
40.75
40.70
40.65
40.60
40.55
40.50
dropoff_latitude
−74.2
−74.1
−74.0
−73.9
−73.8
dropoff_longitude

Fare_amount visualization: To be able to focus on the effect of one feature on fare_amount, new columns are added to obtain the normalized fare_amount values.

The below bar graph represents that the 'fare_amount' includes a base amount that increases slightly by the number of passengers in this case for single to five passengers. Then, the 'fare_amount' significantly increased for six passengers because another type of car is required.



The other feature is normalized fare_amount based on distance and passenger_count. The scatter plot denotes that for distance < 70, there is a positive correlation between distance and fare_amount per passenger.

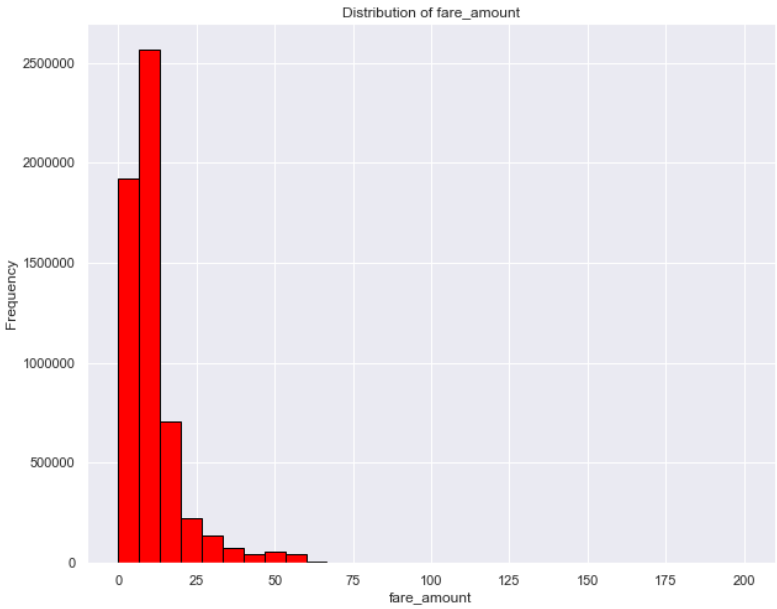


In order to show the trend of fare_amount changes over the years, a scatter plot has been plotted below for the fare_amount by ride distance. The size of the circles represent the passenger count for each ride. The pink color is used to identify the rides where the pickup or dropoff location is in the Manhattan area and the blue color is used for other NYC regions.

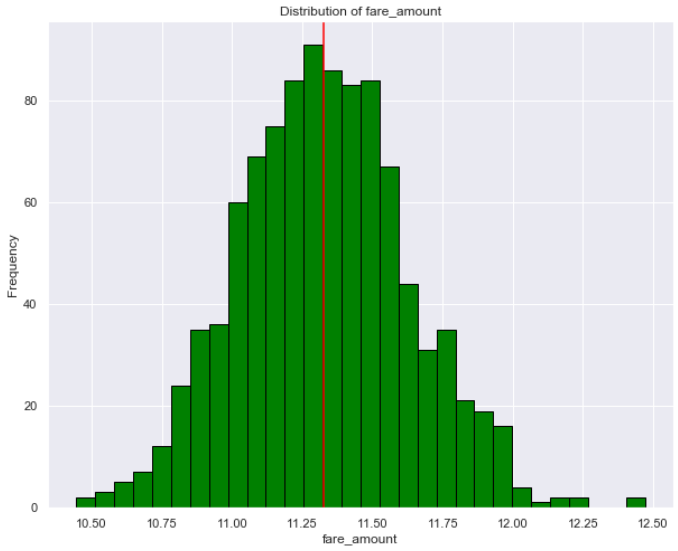
This scatter plot represents that the fare_amount in the Manhattan area is lower than the other regions. It matches the expectations since the rides in the Manhattan area are usually shorter than the ones in other regions. Also, usually the number of available taxi cabs is larger in Manhattan due to the higher demand. Moreover, passenger count for the rides in Manhattan is usually limited to one or two considering it is a business district. As it can be seen in the figure below, as the passenger count increases, the fare_amount increases proportionally as well.

Statistical Inference

The histogram of fare_amount is shown in below.



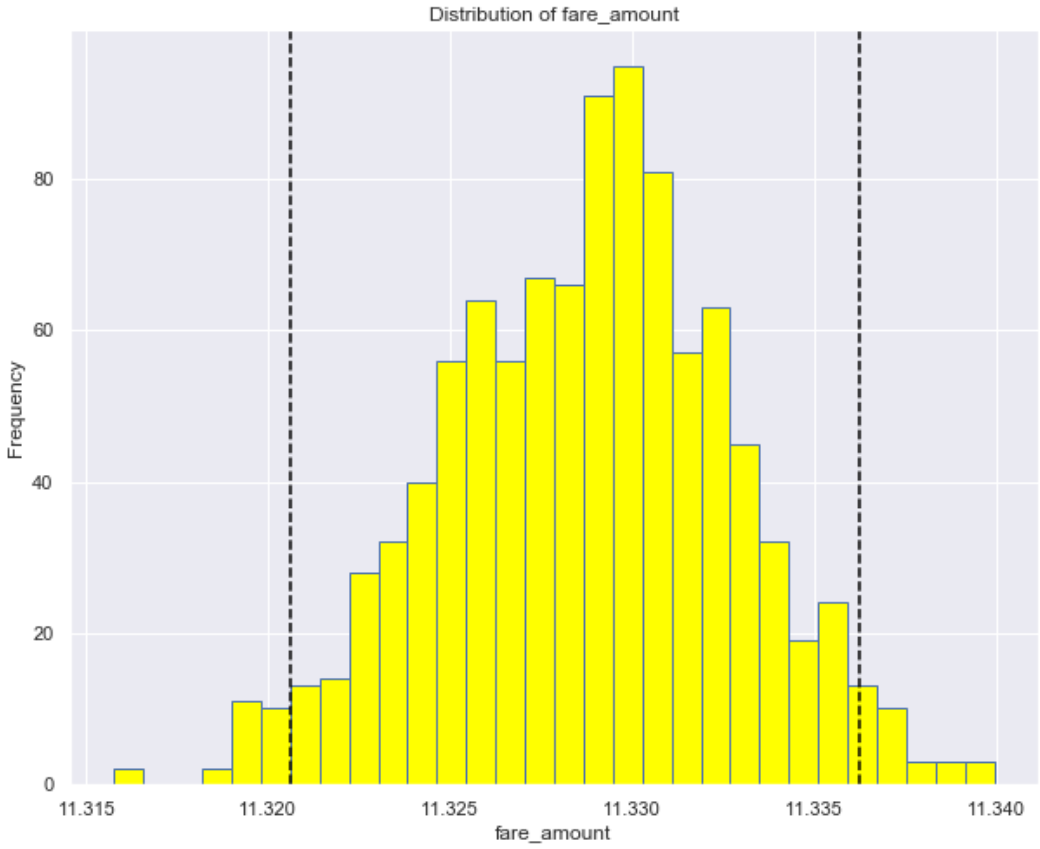
In order to show that the given sample is a good representation of the actual distribution, a sampling variability test should be performed. The histogram represents that the mean of the generated sample is really close to the mean of the actual data.



Generating Bootstrap Replicates

By performing bootstrapping, a sample is generated from the original sample data by resampling to perform statistical inference such as confidence interval. This way, inference can be made based on new datasets.

In this project the bootstrap replicates are generated for fare_amount and confidence interval is calculated to see if 95% of the observed values would lie within the 95% confidence interval.



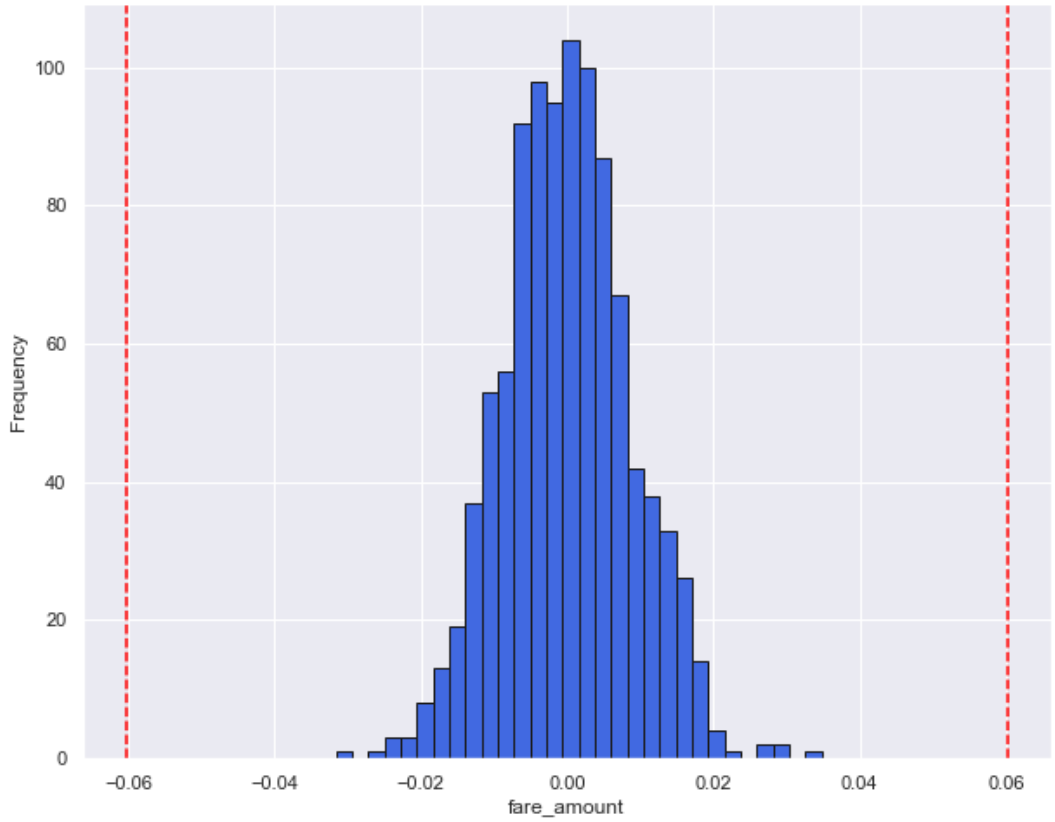
Hypothesis Testing (Null Hypothesis):

In this experiment, we go one step further and check if there is any relationship between two datasets.

Null Hypothesis1 :

fare_amount for weekdays vs. fare_amount for weekends: The null hypothesis is that the mean of fare_amount for weekdays and weekends are the same. First, bootstrap replicates should be drawn for the difference between the mean of weekend and weekdays. Then, p_value is calculated as the probability of obtaining test results that are as extreme as the observed results, by assuming the null hypothesis is true. Since the mean value of fare_amount for weekdays and weekends are not the same and in order to have fare comparison, sample arrays for

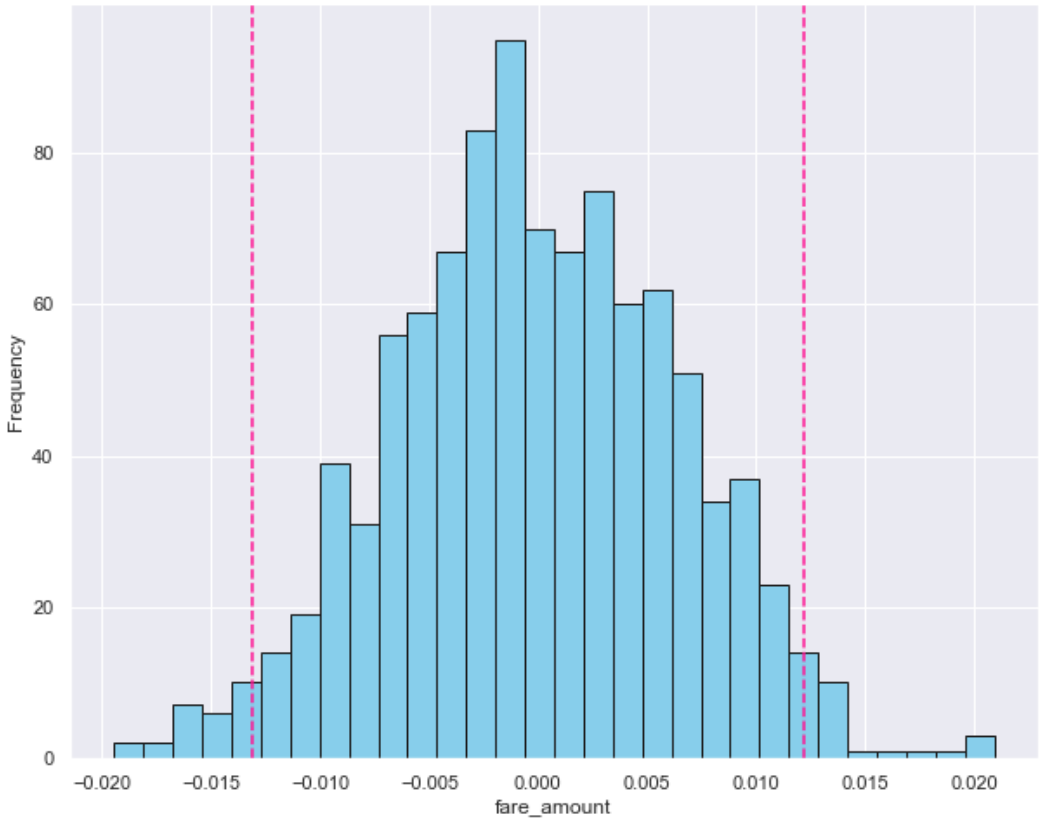
these datasets should be shifted. Then, the p_value and confidence interval are calculated to see if the null hypothesis should be accepted or not. The achieved p_value is zero, which means that the probability of the mean of the two datasets being the same is zero. Therefore, the null hypothesis should be rejected.



Null Hypothesis2 :

fare_amount of Manhattan vs. fare_amount of other locations: The null hypothesis is that the mean of fare_amount of Manhattan area is the same as other regions of NYC. In this experiment, aare_amount of Manhattan includes the fare_amount if pickup or dropoff is one (which means that pickup or dropoff is in the Manhattan area). After performing bootstrapping, p_value and confidence intervals are calculated to see the null hypothesis should be accepted or rejected. Since, the achieved p_value is zero the null hypothesis is rejected which means

that the mean of fare_amount of Manhattan is significantly different from the mean of fare_amount for other NYC locations.

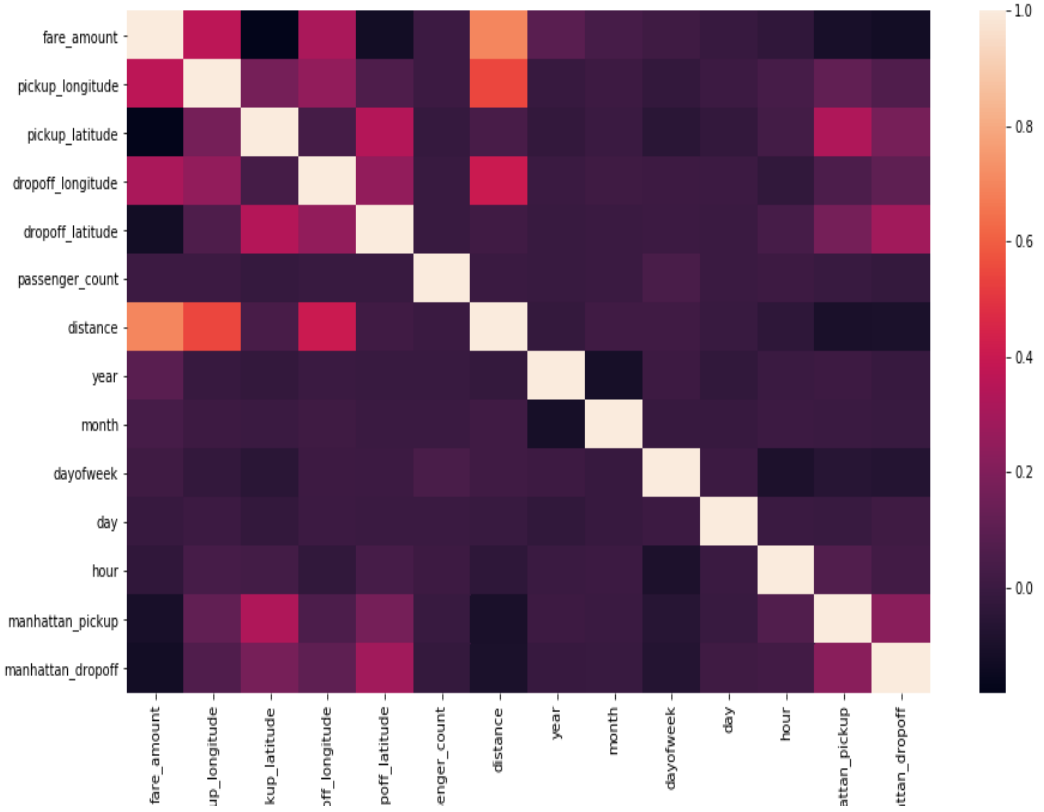


4. Results and In_Depth Analysis Using Machine Learning

By performing data wrangling and exploratory data analysis, required features for our model are obtained. In the machine learning step of this project, the relationship among these features and the target variable (which is taxi fare_amount) is determined. Considering that the model should be trained on one set of data and then its performance should be measured on unseen data sets, the data have been split into training (70%) and testing (30%) data sets.

Feature Correlation:

To identify the correlation among features in the data set, the heatmap correlation can be used. When a large number of features exist, heatmap represents correlation among features and their contributions in machine learning models. Correlated features should be avoided since they will cause inaccuracy to our model. Range of the heatmap can be between -1 and 1.



The correlation heatmap of the problem features shows that none of the features are highly correlated.

Naive approach:

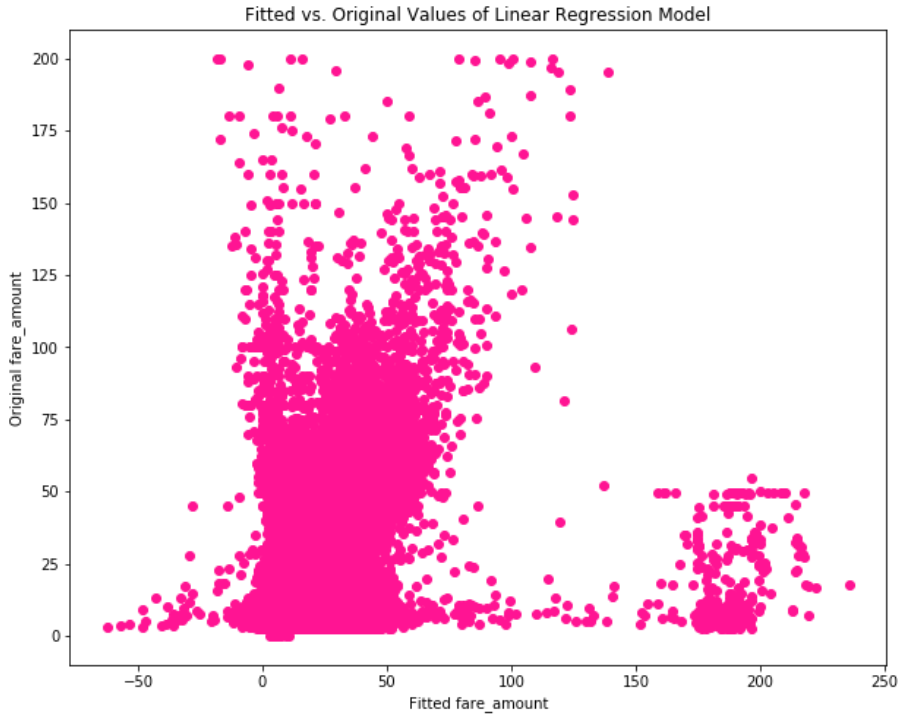
In order to show the effectiveness of using machine learning techniques, a naive approach is considered. Considering that the problem is regression, this naive

approach always returns the mean value of the target (fare_amount) in the training data. The correlation coefficient of naive approach is negative value. The correlation coefficient of machine learning technique will be compared to show the machine learning technique achieves better predictions.

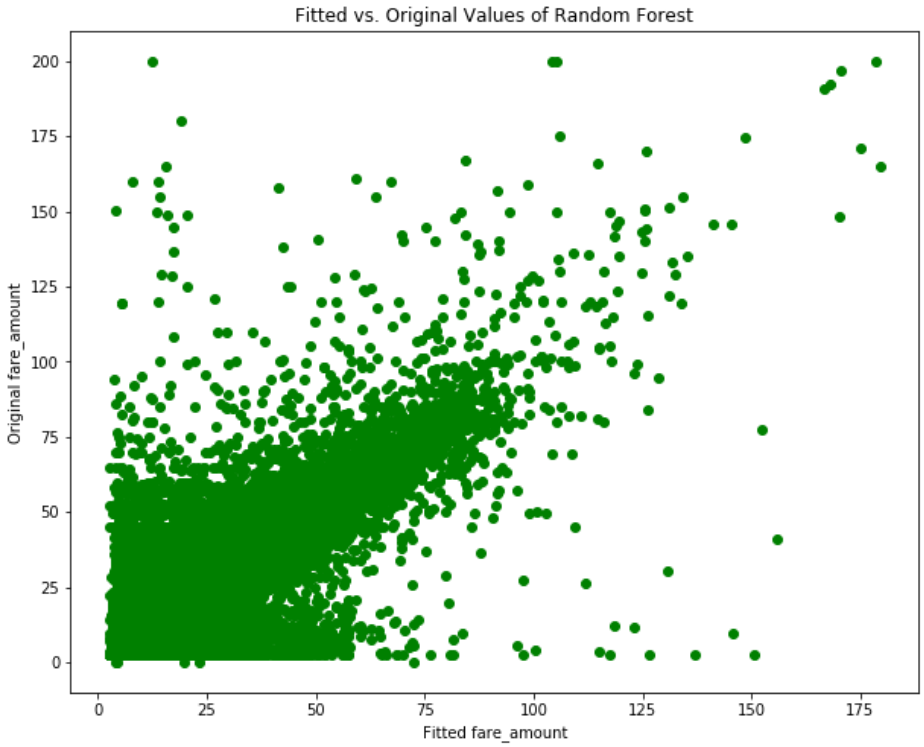
Supervised machine learning algorithms:

Since the target variable (taxi fare amount) is in the form of real variable, different regression models have been employed in this project. The algorithms that have been used in this project are Linear Regression, Random Forest Regression, Decision Tree Regression, Gradient Boosting Regressor, and Linear Support Vector Regression, which are all supervised learning algorithms.

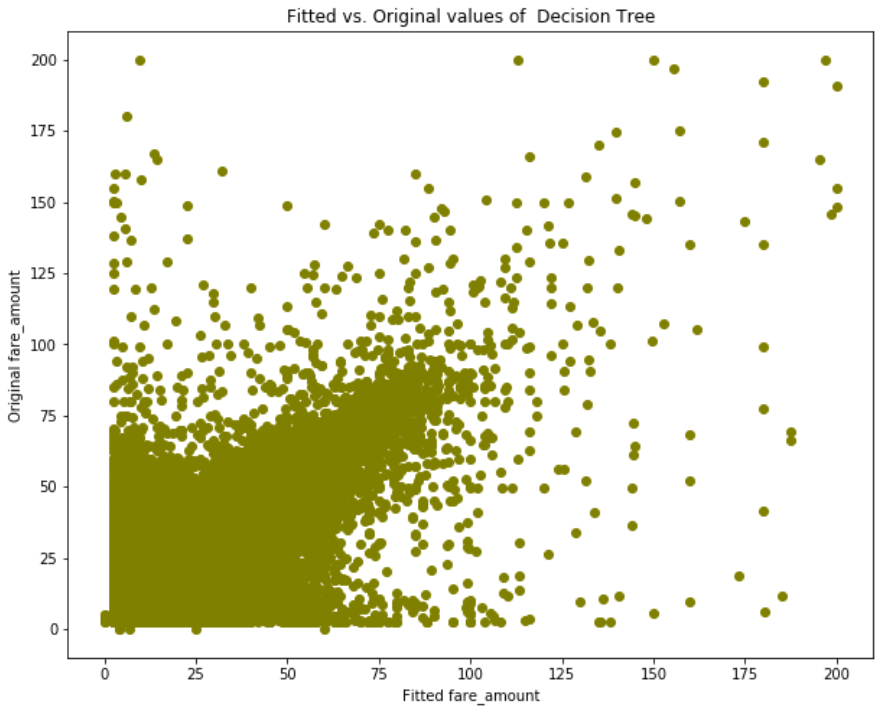
- **Ordinary Least Squares (OLS):** First the simple linear regression using statsmodels is performed to show the linear relationship among the features and target variable. The OLS results are summarised in a summary table.



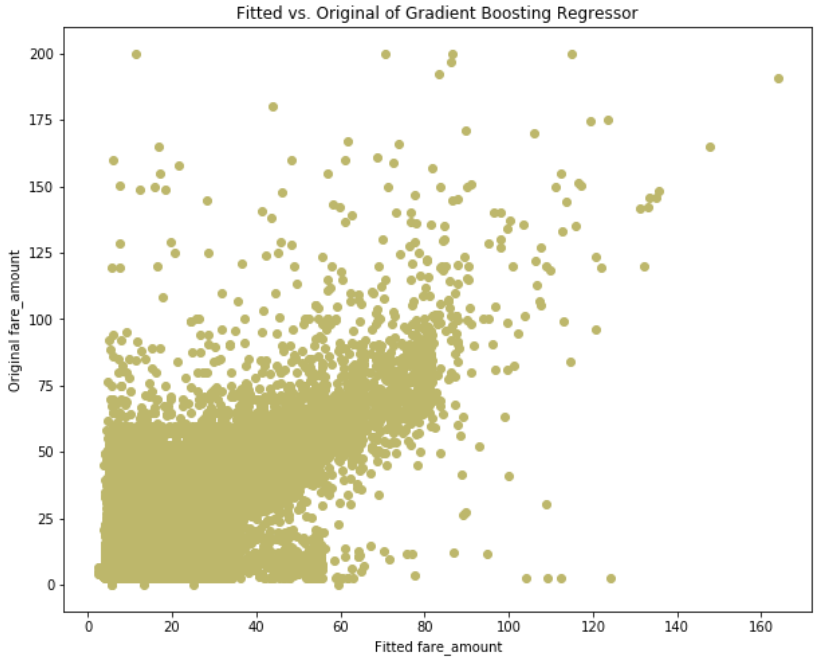
- **Random Forest Regression:** is considered as a first non-linear regression model in this project since it is one of the most accurate learning algorithms which runs efficiently on large datasets. The n_estimators is set to 10 trees in this project.



- **Decision Tree Regression:** is another regression model that has been employed for this project. Considering the Decision Tree does not require scaling and normalization, it is considered as the simplest and most powerful learning technique.



- **Gradient Boosting Regressor:** is a strong algorithm for predicting continuous values since simple models are added one at a time to obtain the correct target value. Gradient Boosting has been used in most of the competitions since it achieves more accurate results compared to other techniques such as Linear Regression.

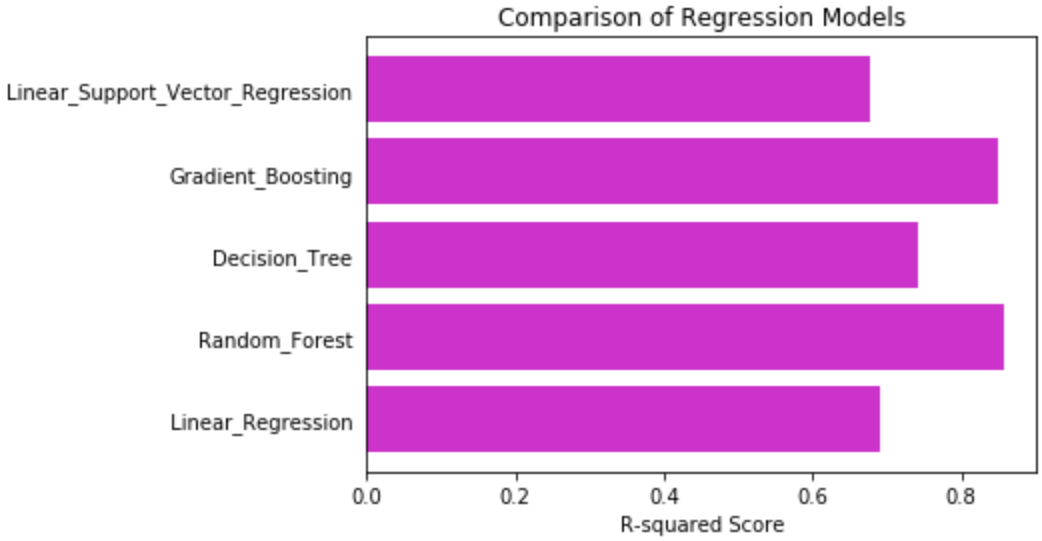


- **Linear Support Vector Regression (SVR):** SVR is a regression algorithm that tries to fit the error within a certain threshold. The computational complexity of SVR is O(n^3) where n is the number of training samples. Considering that the number of training samples is 1,400,000, it takes too long to run, Linear SVR is used which can produce a model with accurate results same as SVR. Since the kernel has not been used in linear SVR, it is a fast algorithm which is good for huge samples.

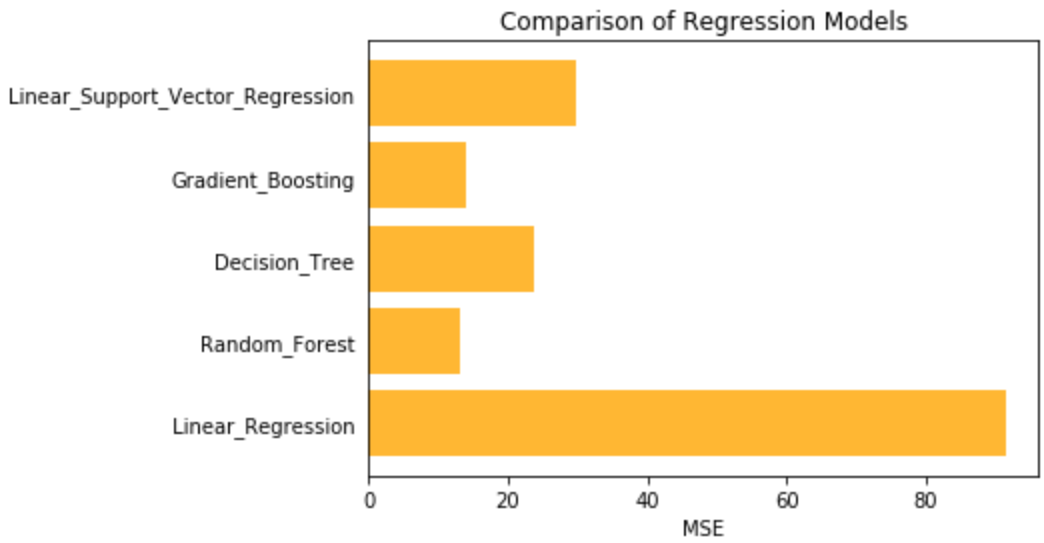
Evaluation:

In order to validate the performance of employed machine learning algorithms, some statistical measurements metrics have been considered that are described in detail in below:

- **R-squared Score:** The larger R-squared means the regression model fits the observations better. Both Random Forest and Gradient Boosting obtain high R-squared value.

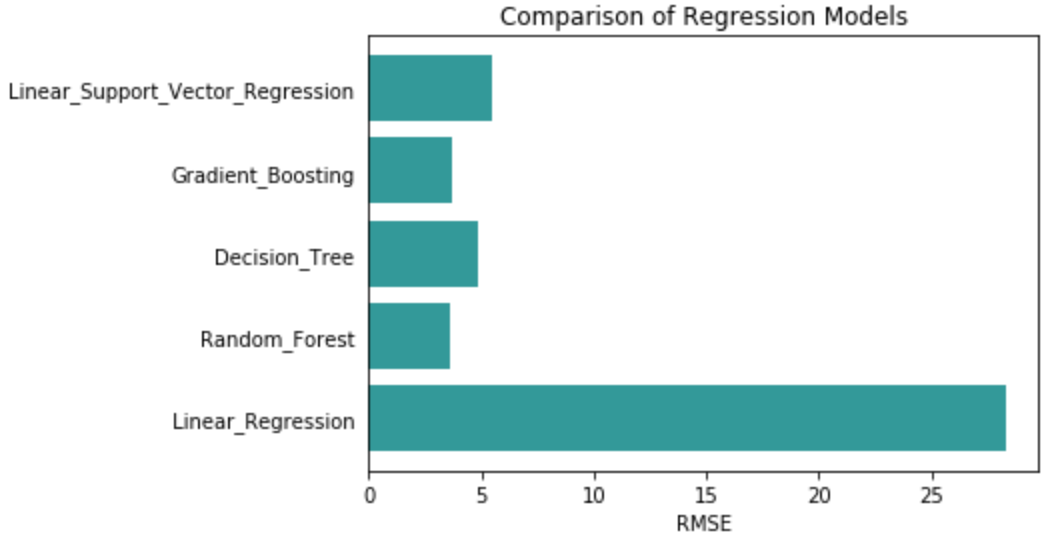


- **Mean square error (MSE) :** is the average of the square of the errors. The larger the number the larger the error.



As shown in this plot, both Random Forest and Gradient Boosting have lower MSE compared to other models.

- **Root Mean Square Error (RMSE):** is a method of measuring the difference between values predicted by a model and their actual values. In other words, RMSE represents how concentrated the data is around the regression line of the best fitted model. The non-negative values that are close to zero are better.



Both Random Forest and Gradient Boosting obtain low values.

- **Correlation Coefficient:** returns a value between -1 (full negative correlation) and 1 (full positive correlation) that indicates the strength of linear relationship among two variables.

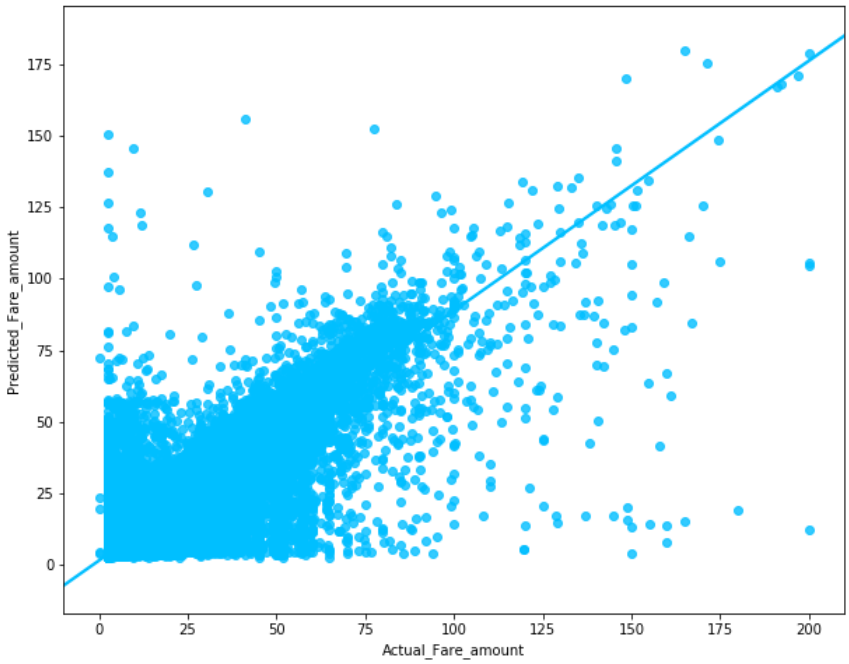
  Both Random Forest and Gradient Boosting indicate a notable correlation between target value and predicted values.

The linear regression improves its predictions compared to Naive approach. However, there is a limit to its performance due to nonlinear patterns in the dataset. Therefore, based on the evaluation results we can see that the non-linear algorithm such as the Random Forest model increases the performance. Based on the above evaluation both Random Forest and Gradient Boosting have pretty close results. Since Random Forest outperforms all other models, it is going to be used for the rest of the project parts.

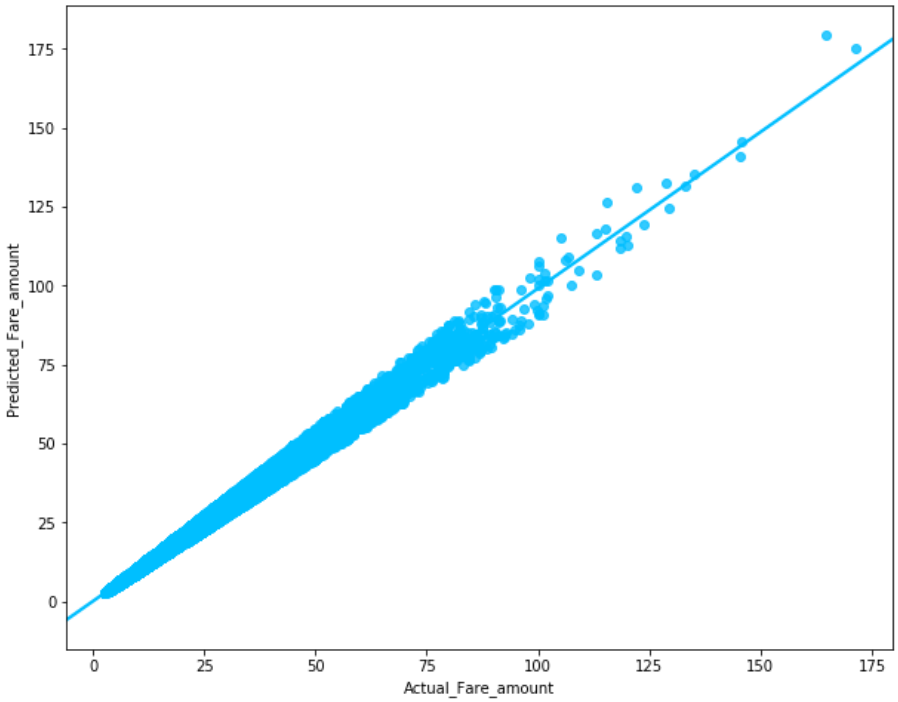
A Deeper Look at the Accuracy of the Model:

To see the error between actual fare_amount and predicted fare_amount, the Predicted_Fare_amount over the Actual_Fare_amount ratio is calculated. The ration value close to one shows that the prediction model is more accurate. The relationship between Actual_fare_amount and Predicted_Fare_amount has been

plotted a linear regression model fit.



In order to limit the visualization to more accurate prediction, the data with error lower than 10% is plotted below where we can see a more linear curve.

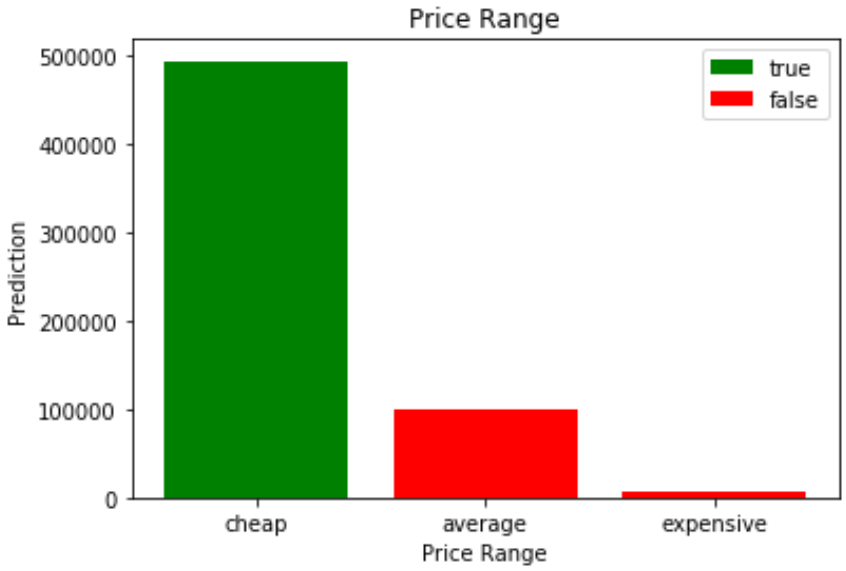


Calculated ratio shows 45% of the data has a good error (less than 10%)

Classification:

Classification is a type of supervised learning where the data is already labeled. In this project Naive Bayes Classifier is used which is a fast algorithm compared to other classification algorithms.

Naive Bayes Classifier: as explained before, New York city taxi fare prediction is a regression problem. In order to consider machine learning classification techniques, the problem should be changed to classification problem. In this matter, the continuous target is changed to labeled target by considering 3 different fare_amount categories. The data with fare_amount less than to 15$ are labeled as cheap, between 15 and 50 dollars are labeled as average, and greater than 50 are labeled as expensive.



Since most of the training dataset are in the cheap category, the prediction accuracy of this category is higher.

Additional Explorations

Feature Selections: In training machine learning models, the features have a huge impact on the performance and accuracy of the model. Feature selection can reduce overfitting and training time and increase accuracy.

To prepare the model using the above algorithms, all the features are selected. In order to see the effect of feature selection on the performance of prediction models, the OLS is used considering features that have the strongest relationship with the target variable. However, the results do not show a significant improvement. The reason can be because the number of original features is already not large.

Principal Component Analysis (PCA): is a dimensionality reduction algorithm that can be used for feature extraction. By performing PCA on the problem dataset features, the dimensions of the dataset are reduced from 13 to 3. Then, the OLS is performed using the new PCA features. Although usually the PCA increases the performance of the prediction model, the results show that it does not have significant influence on our OLS model since the number of features is not big.

Summary: For app-based hiring vehicles companies accurate prediction of taxi fare is important. In order to find the most appropriate model to predict the fare amount for a taxi ride in New York City, many factors are considered as problem features such as pickup or dropoff locations. Various regression and classification models are used in this project. The results show that Random Forest outperforms all other models and obtains an accurate prediction model.

As a future work, clustering techniques can be applied to grouped the pickup or dropoff locations. This way, this project can be used in traffic history prediction and autonomous vehicle research who might work on traffic modeling to choose the least congestion road. Moreover, more variables can be considered to improve the prediction accuracy.