Cairo University

Faculty of Engineering

Computer Engineering Department

New York Taxi Fare Prediction

Submitted to : Dr/ Ahmed Kaseb,
Eng/ Hussein Fadl

Submitted by: Abd-ElRhman Sobhy

Fatema Khalid

Feryal Hisham

Heba Ahmed

Brief problem description

Our task is to predict the fare amount for a taxi ride in New York city given the following information :

- 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.

Business Value

This project predicts the fare amount given the pickup datetime, passenger count, pickup location and the dropoff location. This serves the purpose of both the taxi drivers and the customers. For a driver they shall take rides from the pick up locations with high fare amount. As for customers, they would know the expected fare amount to decide on the appropriate means of transportation to take.

Project pipeline

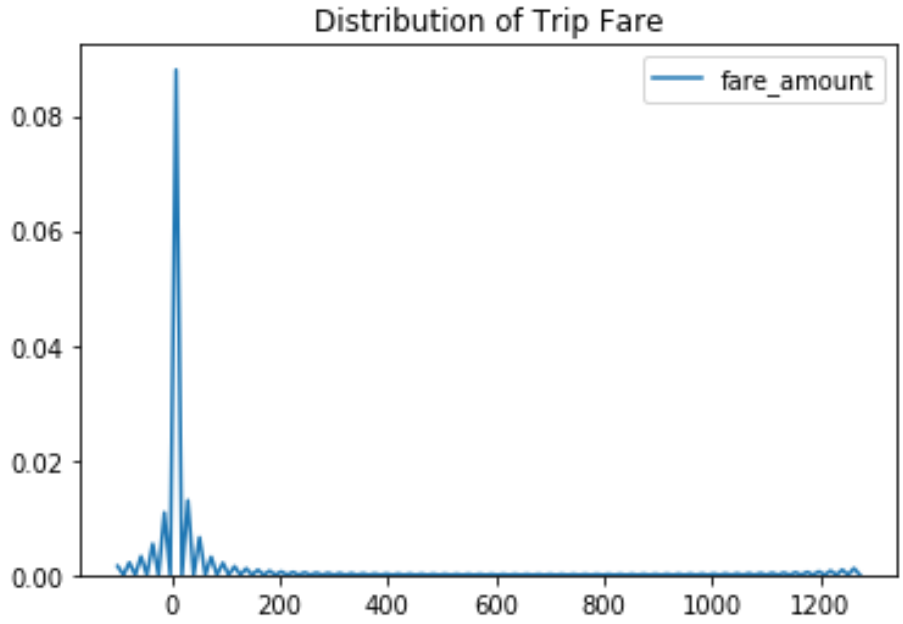
- Data Visualization.
- Data preprocessing and cleaning.
- Models Building and Training.
- Models Evaluation.
- Feature Engineering.
- Retraining the selected model with added features and Revaluation.
- Model Tuning.
- Final Evaluation.

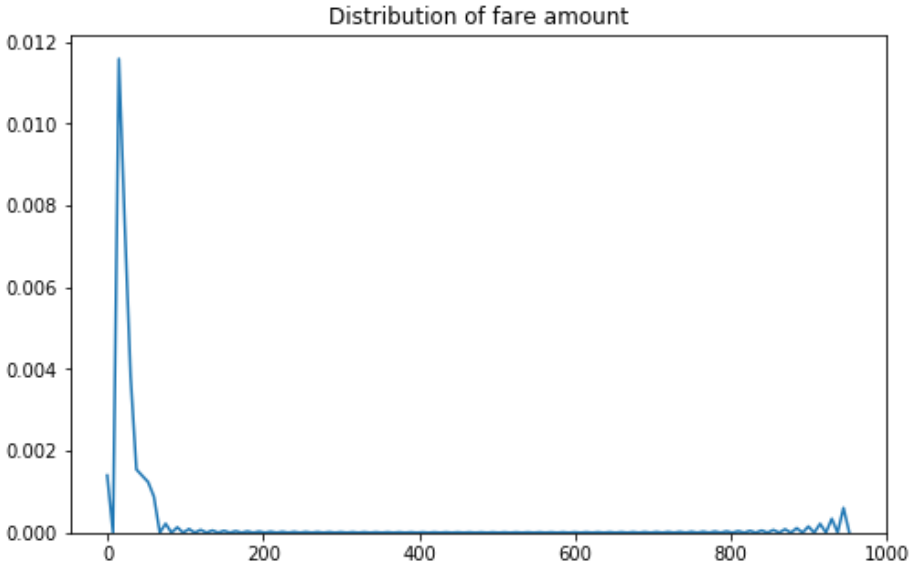
Data visualization

In this section, we will discuss various steps used to clean the data and understand the relationship between variables to use this understanding to create better features

1. Distribution of fare amount:

We first looked at the distribution of fare amount and found that there were 262 records where the fare was negative. Since the cost of a trip cannot be negative we removed such instances from the data.



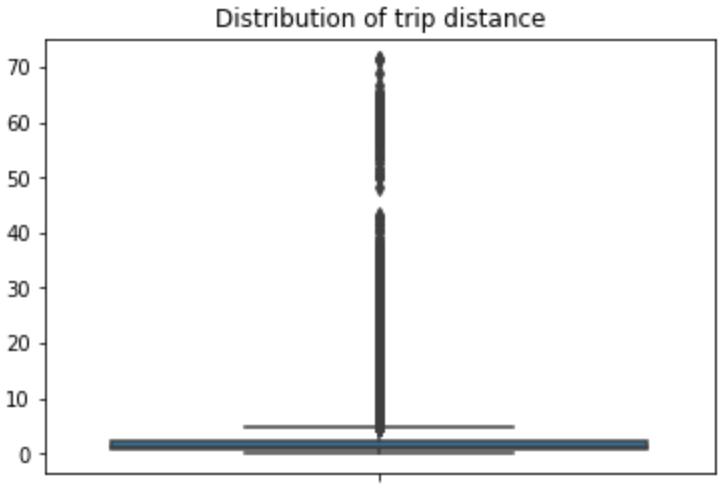


2. Distribution of Trip Distance:

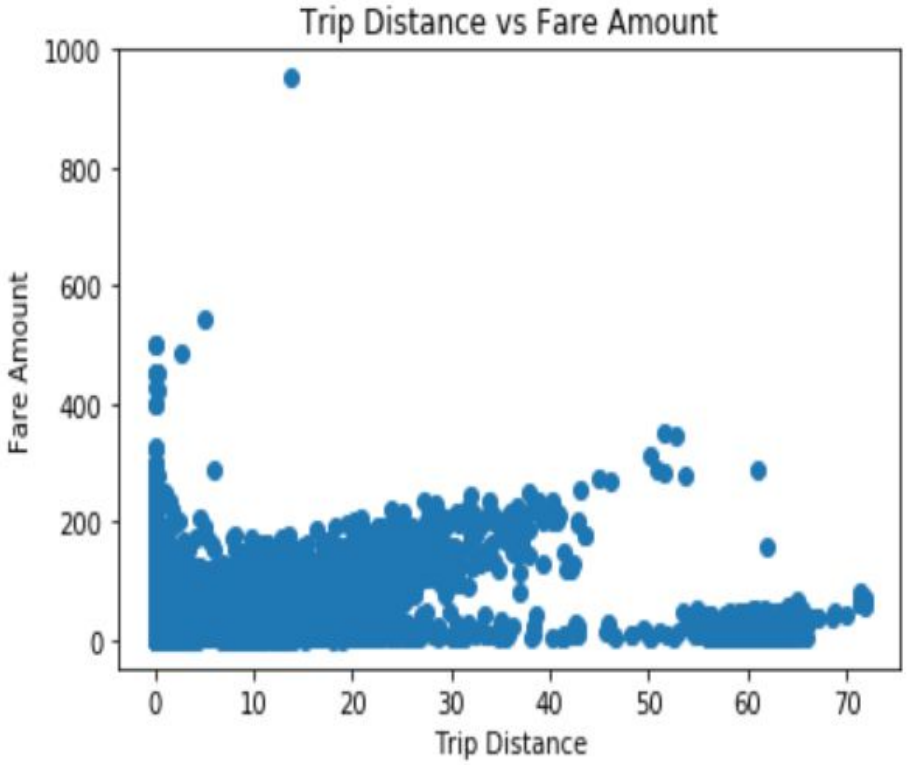
Using the pickup and drop-off coordinates we calculate the trip distance in miles based on Haversine Distance.

The mean of the trip distance is around 2 miles. There are few trips with large distance since about 75% of the trips are below ~2.4 miles.

mean	2.087566
std	2.337506
min	0.000044
25%	0.793985
50%	1.351586
75%	2.447495
max	71.828272

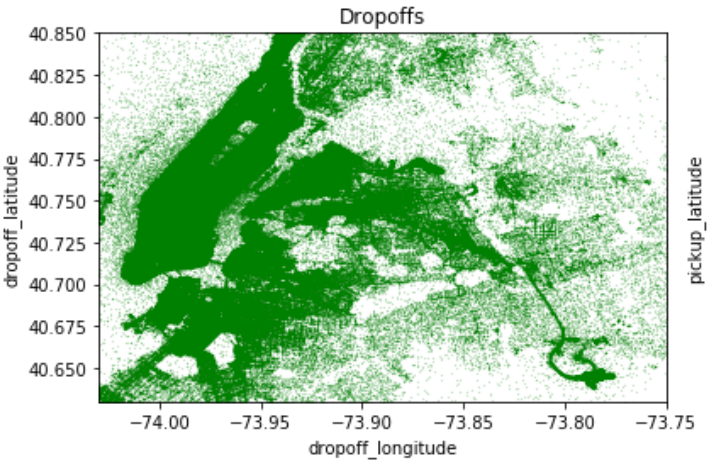


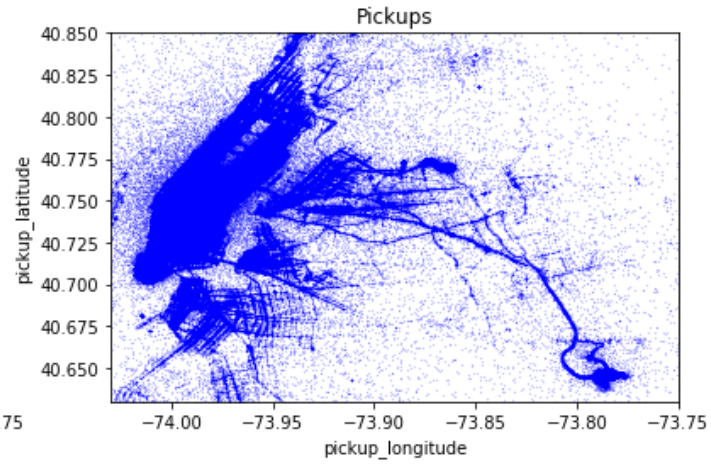
One of our hypothesis was that the fare amount should ideally increase with trip distance . A scatter plot between trip distance and fare amount showed that it is almost linear relationship. There were a lot of trips whose distance was greater than 50 miles, but the fare was relatively low This might indicate that the trip didn't pass congested areas.



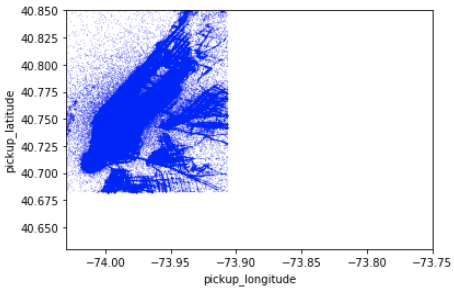
3. Distribution of Geographical Features:

The range of longitudes and latitudes are between -74.263242 to 40.573143 and -72.986532 to 41.709555 respectively. But in the training data set we observed latitudes and longitudes in range of (-3488.079513, 3344.459268) which is not possible, so we keep only rows with the specified range.

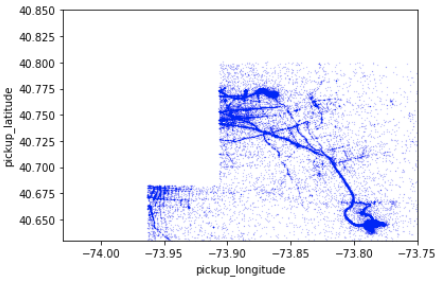




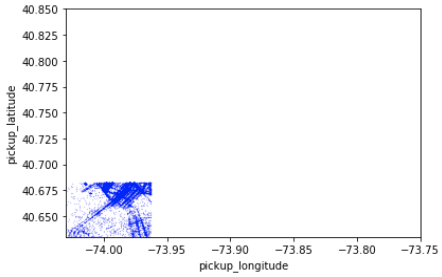
When we plot the number of pickups and dropoffs in all places we found that there are some places with high density than others like pickups near JFK and La Guardia airports and like Manhattan borough.



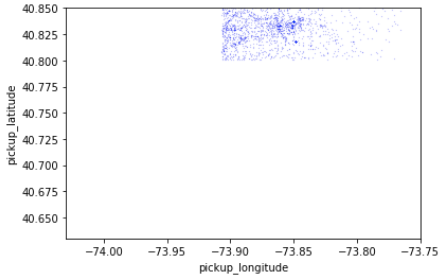
Manhattan



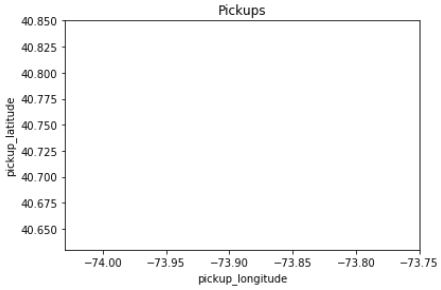
Queens



Brooklyn

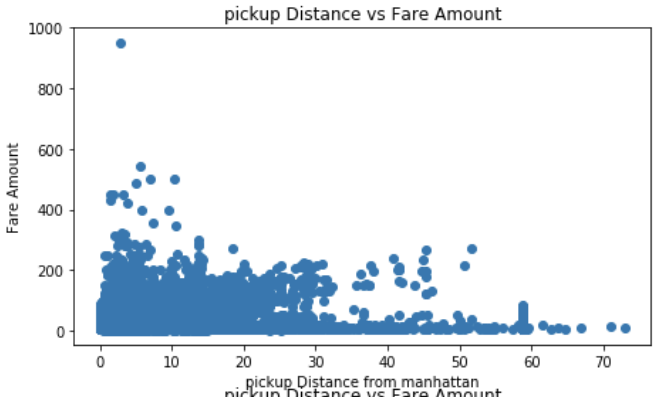


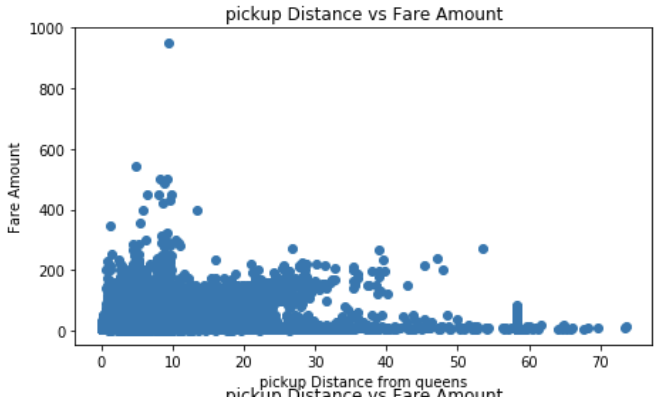
Bronx

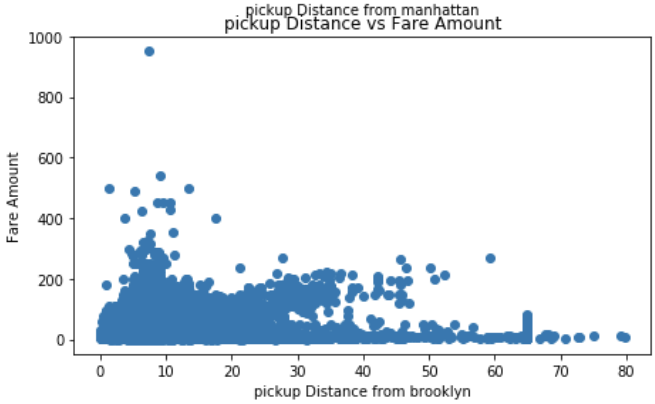


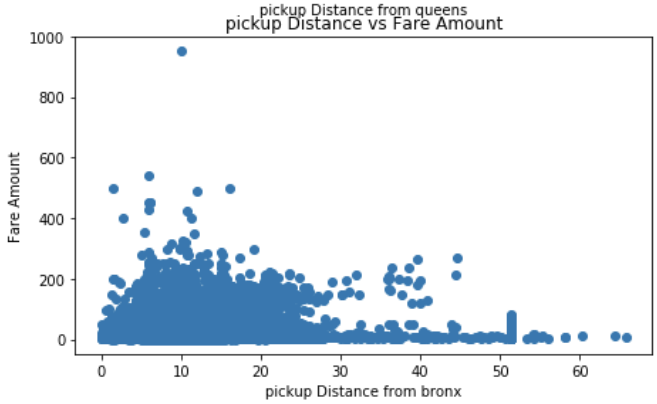
Staten_island

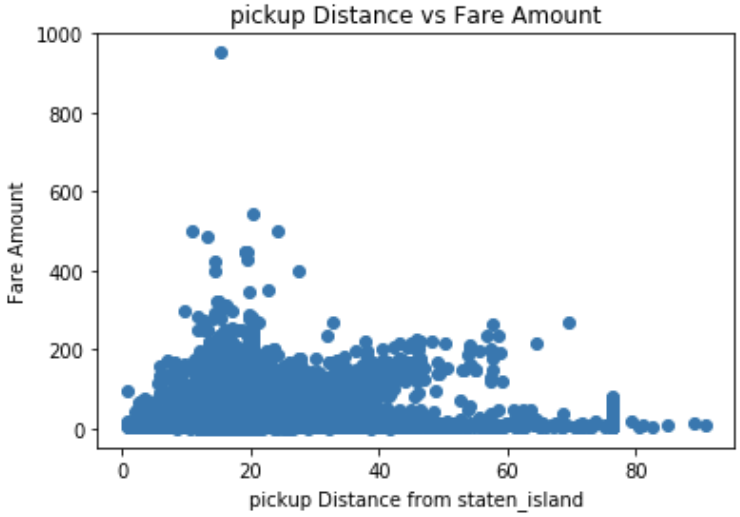
Our hypothesis is that if the trip passes through high density place this will increase the fare amount. This can be verified from observing the graphs below. As the distance from any borough center increases(more than 30 miles) the fare amount is relatively. So we will add extra features for the distance between the trip pickup/dropoff and each of the five boroughs or airports.



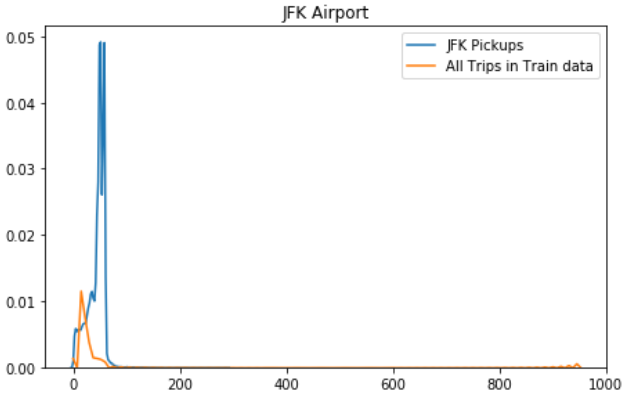


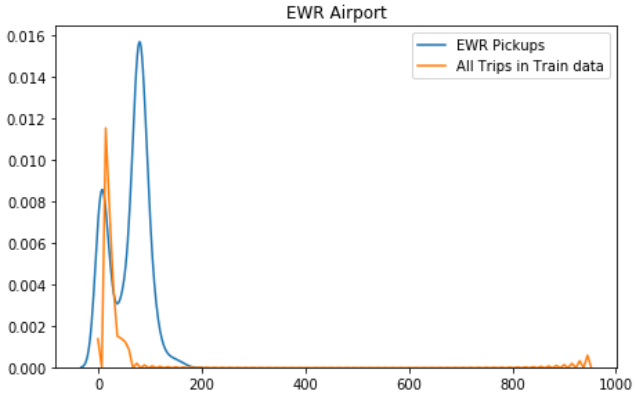


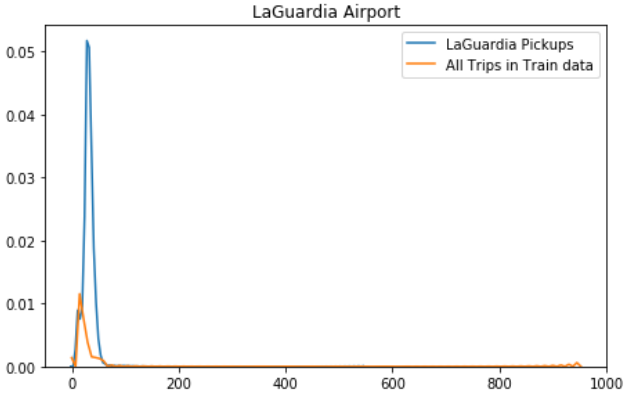




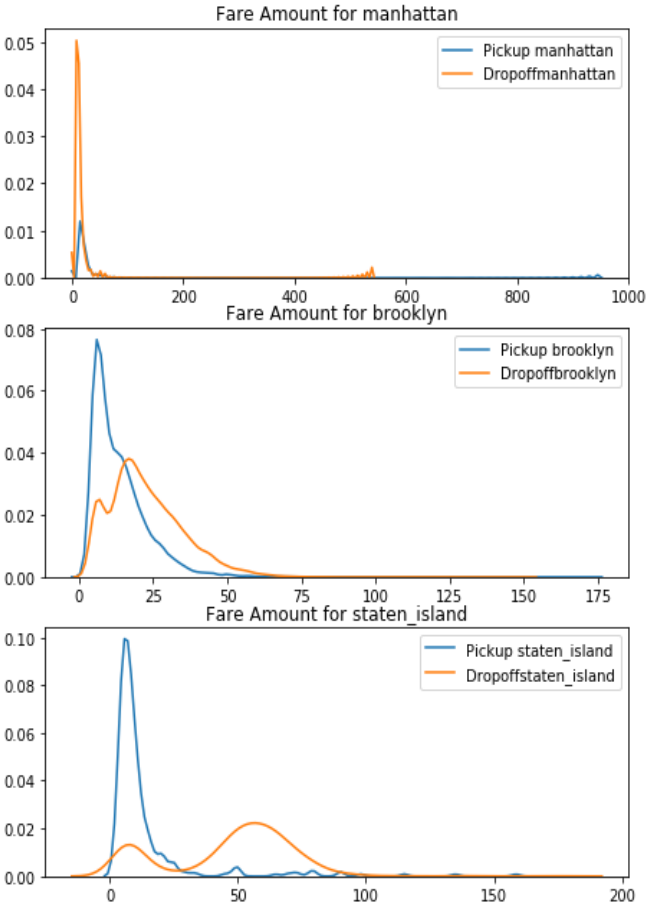
We then looked at what is the average fare amount for pickups and dropoffs to each of the airports - JFK,EWR and LaGuardia - compared to all trips in the train data and observed that the fare was higher for airport trips -the peak values of probability distribution for airports are at higher fare amount values-. Based on these observations we will create features to check whether a pickup or a drop-off is to any one of the three airports in New York .

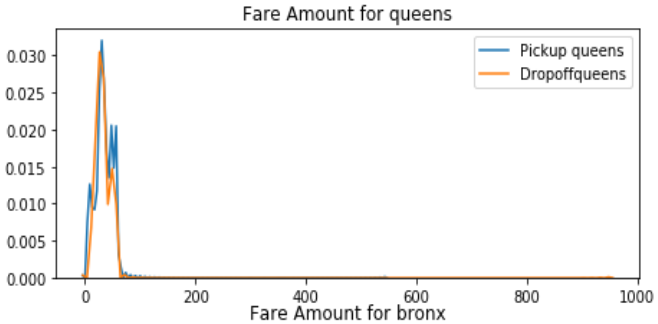


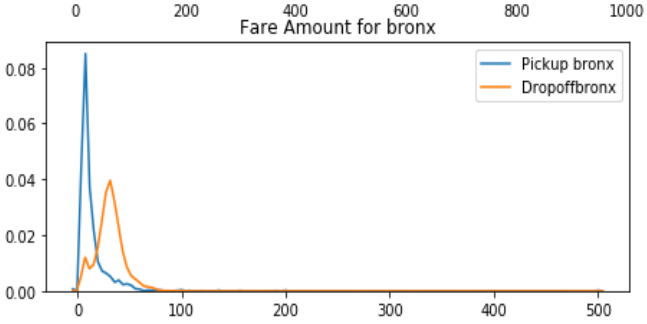




The next step was to check whether our hypothesis of the fare of trips from/to certain boroughs is higher than the rest, based on the 5 Boroughs New York city is divided -Manhattan, Queens, Brooklyn, Staten Island and Bronx- . After plotting we found that our hypothesis was right.







Also, Queens and Manhattan has a higher mean pickup fare compared to other Boroughs.

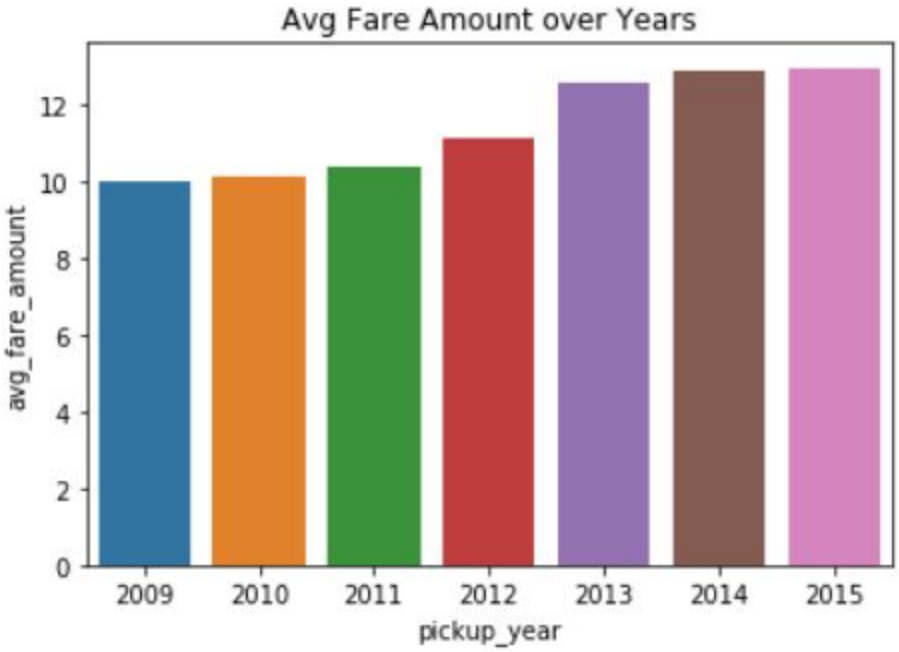
We also notice that Manhattan has higher values of fare amount for pickups than for dropoffs -1000 vs 500.

Hence we will add a feature to say the pickup/dropoff is from/to which borough.

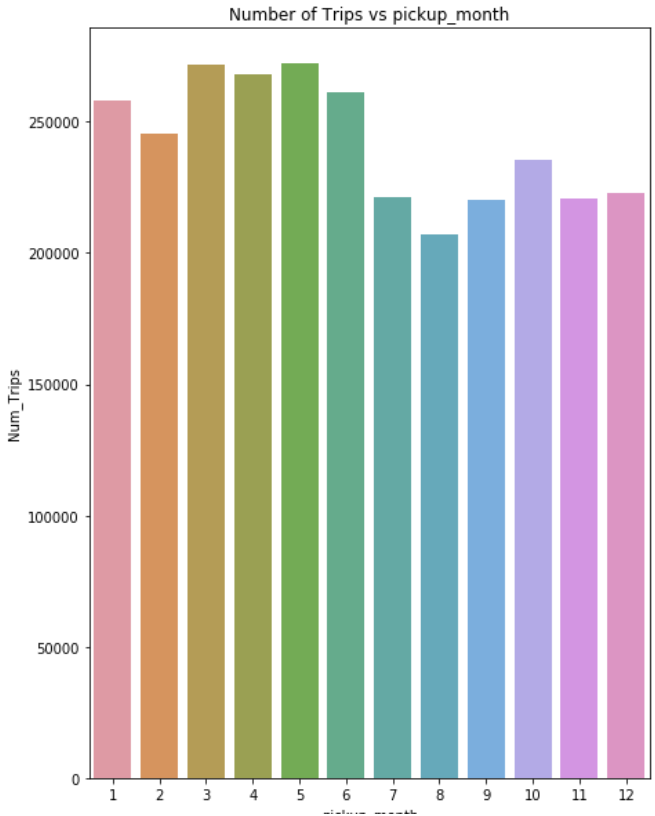
4. Distribution of Pickup date time:

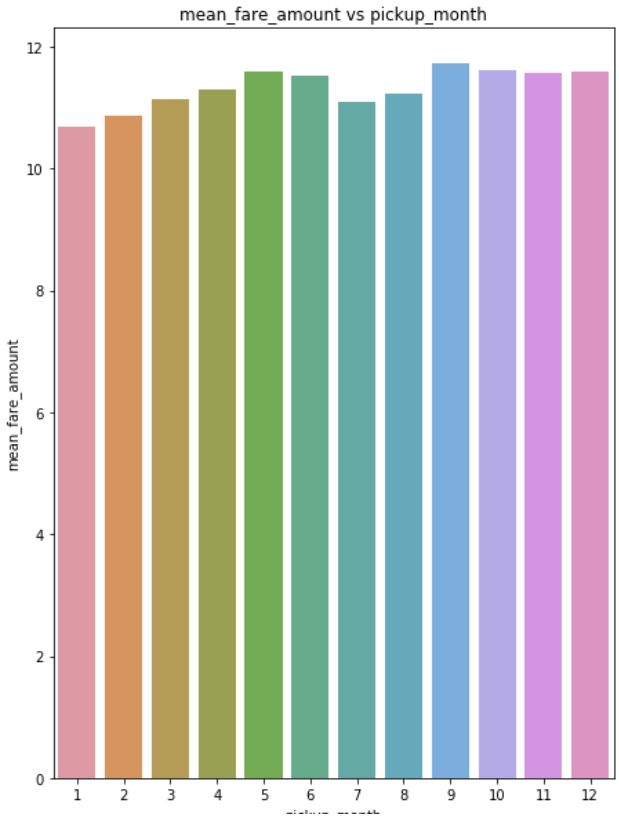
The first step to analyse how the fares have changed over time, is to create features like hour, day of the week, day, month, year from pickup datetime.

As expected, over years the average taxi fare has increased.

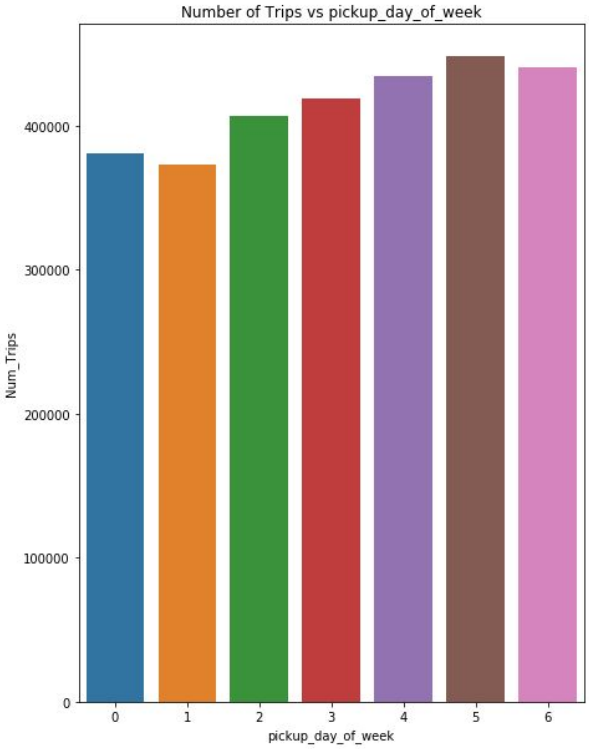


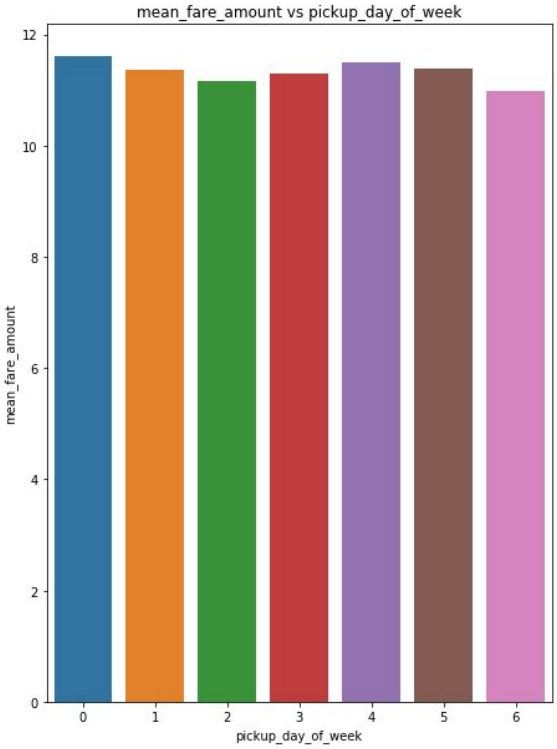
Over months, more trips are from January to June. While the average fare amount is almost the same for all months.





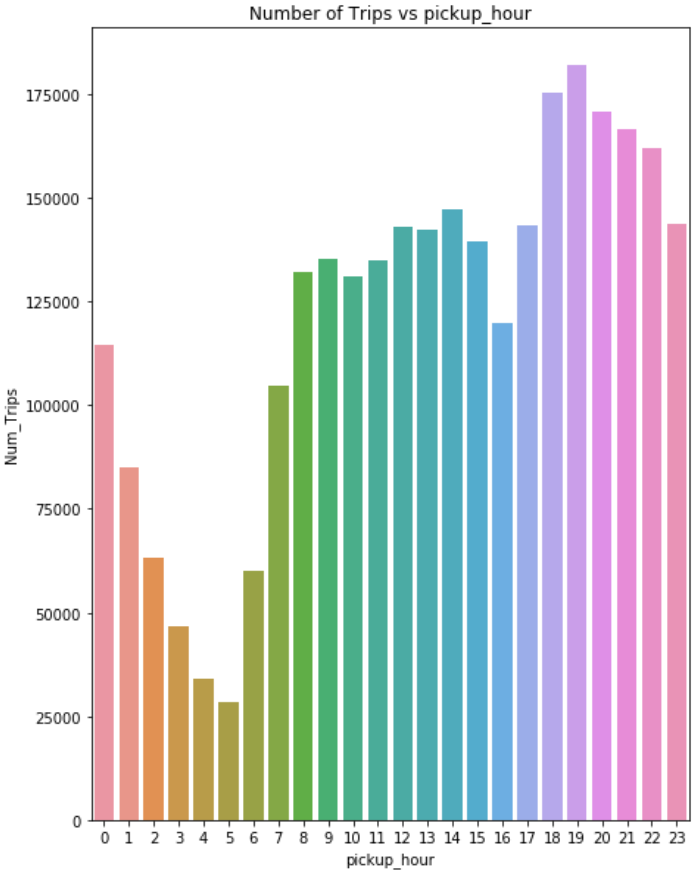
We observed that the number of pickups are higher on Friday(day 5) and Saturday (day 6).On Sunday(day 0) though the number of trips are lower, avg fare amount is higher. This is probably because Sunday is the most congested day since is the beginning of the working week.

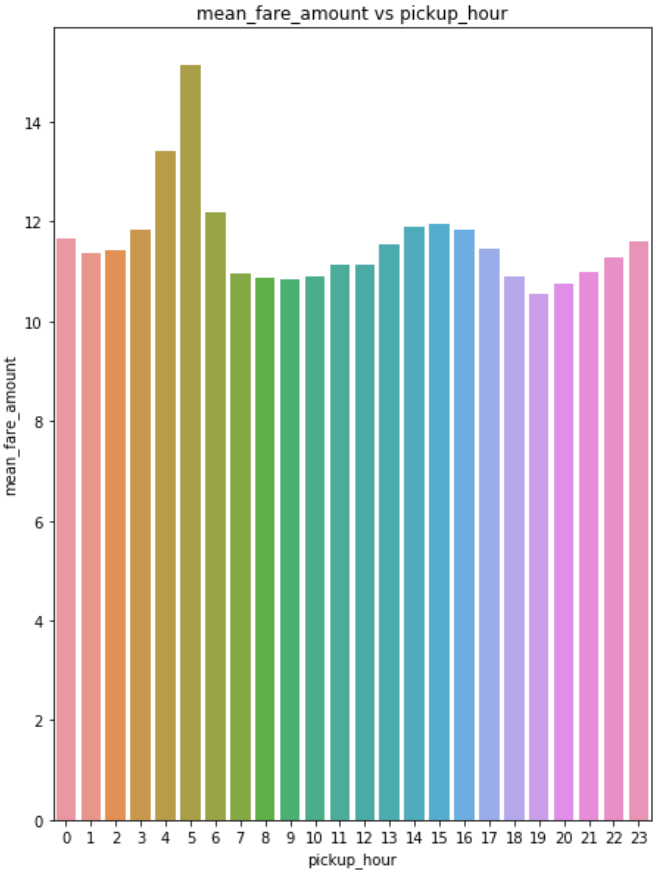




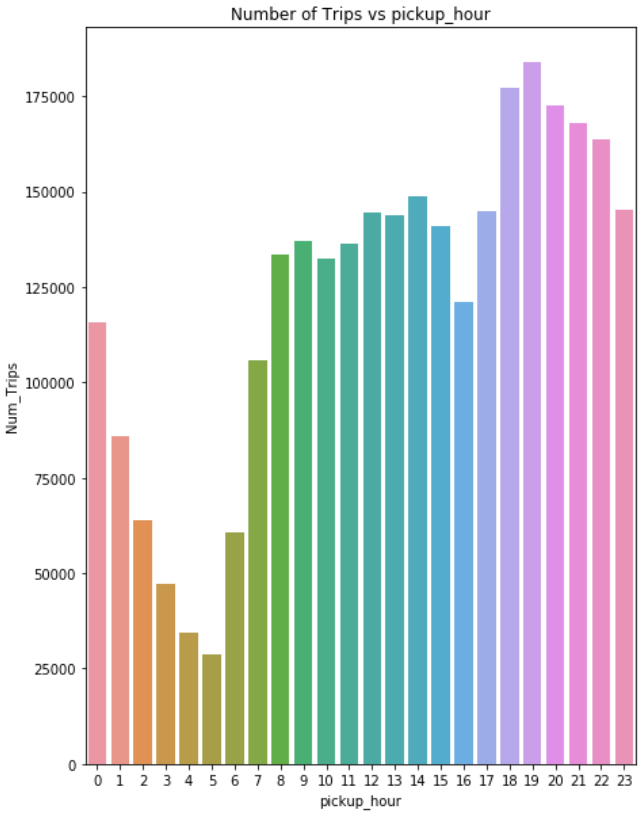
Number of trips increase in two periods of time indicating two rush hours on 7am and 6pm.
The average fare amount at 5 am is the highest while the number of trips at 5 am are the least.

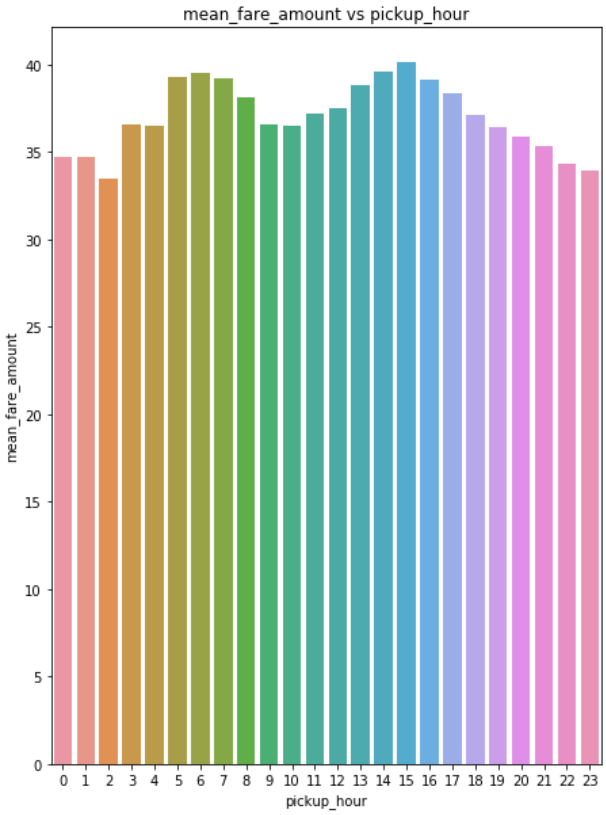
By observing the average fare amount we found that during all day the fare amount is approximately the same but for at 5am. We suspected that this is due to either the taxi services provided at such time is high or that most of the trips at that clock are airport trips.



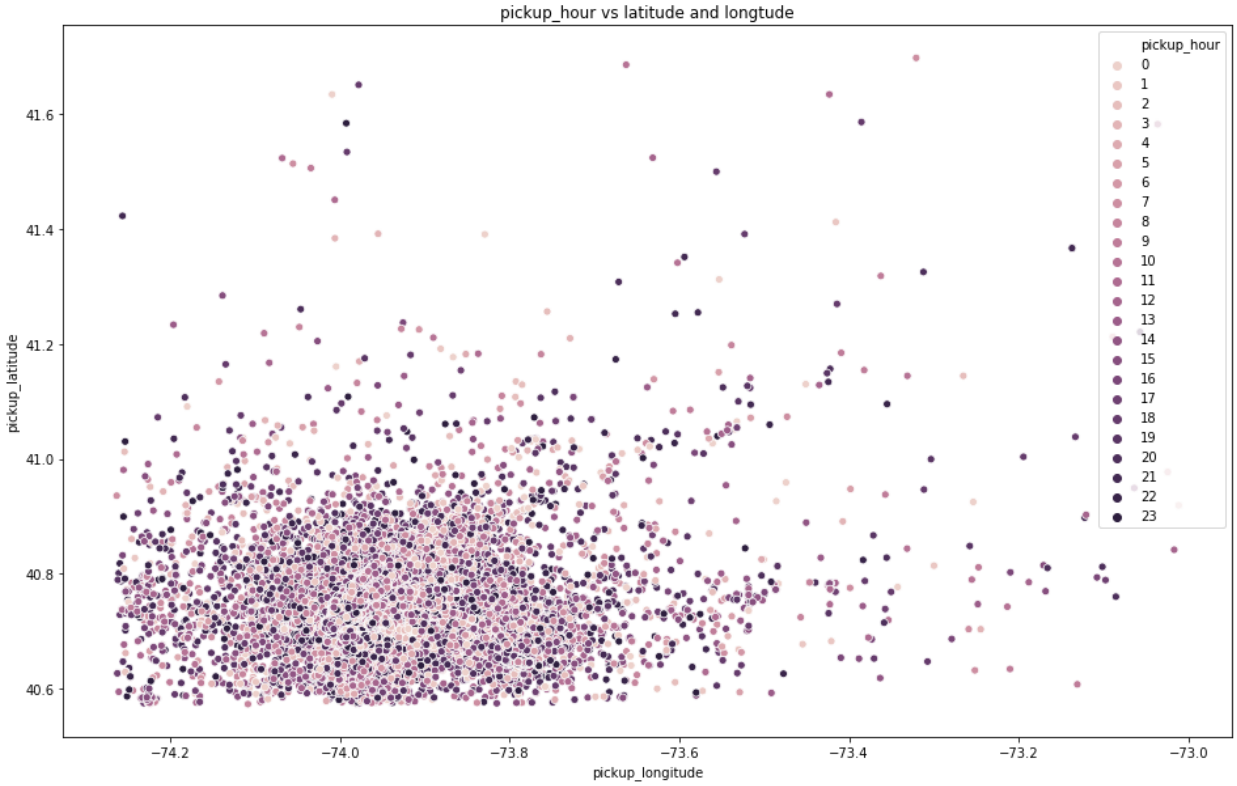


We want to see whether airport trips are at certain times are higher than others.
We can observe that most airport trips start from 7PM till midnight not at 5am as we speculated.



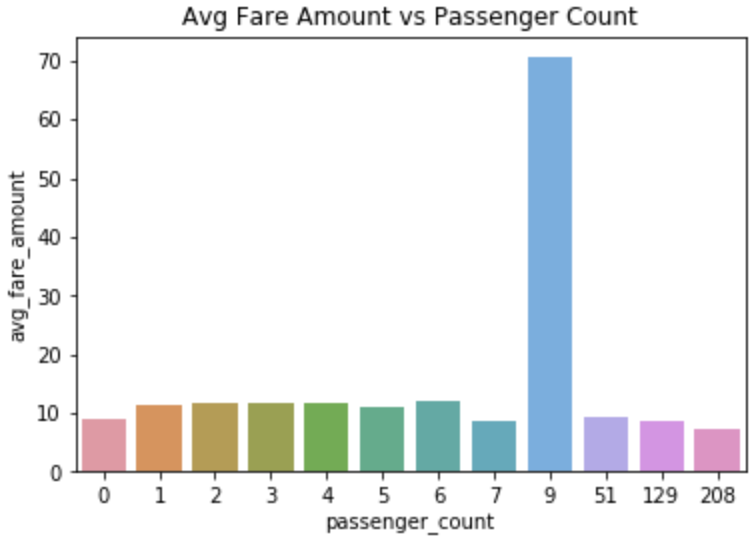


We want to see if there is a relation between pickup_longitude, pickup_latitude and pickup_time. In other words, if there are some places that are busy in certain hours. There is no pattern which means that there is no relation. Busy regions are busy almost in any hour.

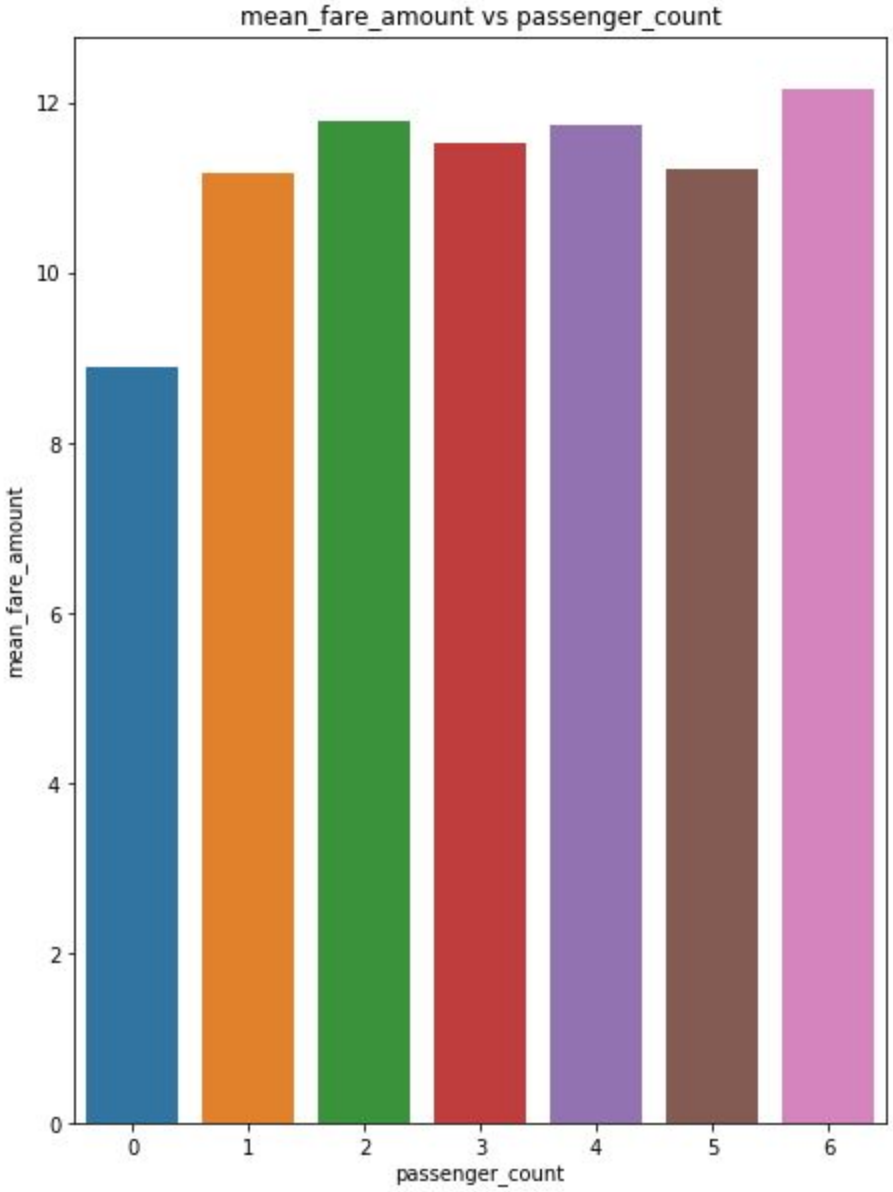


5. Distribution of Passenger Count:

When we plotted the passenger count against fare amount for the trips, we observed that average fair amount for 9 passengers is very high there are also trips with 51, 129 and 208 passengers.



This doesn’t makes sense. We counted the number of trips with such passenger count and found that these outliers (passenger count > 6) are only 12 rows from total 6 million rows, so they were removed.
Then we plotted the passenger count against fare amount for the trip. The trips with zero passenger probably means that customers ordered the taxi and then canceled the trip but a cancellation fee was paid.



Data preprocessing

- We removed records with null values.
- We removed unreasonable values observed from data visualization such as negative fare amounts, out of range longitudes and latitudes, and passenger counts greater than 6,etc.
- Extracted pickup year, month, day and hour from pickup date.

Models Building and Evaluation

1. Baseline Model

A baseline model is a solution to a problem without applying any machine learning techniques. The baseline we are using is calculating the average fare amount.
Any model we build should have an RMSE lower than **the baseline model's RMSE which is 9.71.**

2. Linear Regression

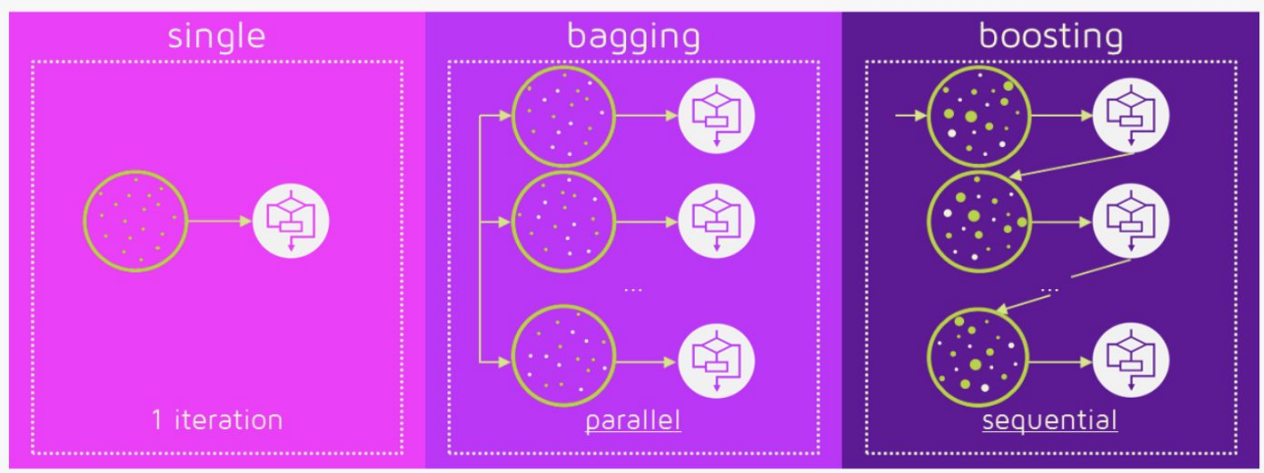
It identifies a line that best fits the data. The best fit line is the one for which the prediction error is the least. This algorithm is not very flexible, and has a very high bias. It is also highly susceptible to outliers to minimize the sum of squared errors.
The test RMSE for Linear Regression model was 8.14, and the training RMSE was 8.20. The error rate is very high in this model, though the variance is very low (0.069).

Brief description for Ensemble, Bagging and Boosting

When we try to predict the target variable using any machine learning technique, the main causes of difference in actual and predicted values are **noise, variance, and bias**. Ensemble helps to reduce these factors (except noise, which is irreducible error)
An ensemble is just a collection of predictors which come together (e.g. mean of all predictions) to give a final prediction. The reason we use ensembles is that many different predictors trying to predict same target variable will perform a better job than any single predictor alone. Ensembling techniques are further classified into Bagging and Boosting.

- **Bagging** is a simple ensembling technique in which we build many *independent* predictors/models/learners and combine them using some model averaging techniques. In Bagging, you take bootstrap samples (with replacement) of your data set and each sample trains a weak learner.

- **Boosting** is an ensemble technique in which the predictors are not made independently, but sequentially. Boosting uses all data to train each learner. But instances that were misclassified by the previous learners are given more weight, so that subsequent learners can give more focus to them during training.



3. Random Forest

It is an example of bagging ensemble. In Random Forest, multiple decision trees are created, and the output is the average prediction by each of these trees.
In general, a higher number of trees increases the performance and makes the predictions more stable, but it also slows down the computation.

The Random Forest model gave an RMSE of 3.83 on validation data and train RMSE of 3.27.

4. Light GBM

Light GBM is based on boosting method. It is a variation of the Gradient Boosting Decision Trees in which the decision trees are trained iteratively.
Light GBM is very fast as It is better to iterate quickly at high accuracy to try more different things, than waiting your neural network to finish after hours, it takes quite less RAM to run, and focuses on the accuracy of the result.

The Light GBM model gave an RMSE of 3.55 on validation data and train RMSE of 2.75.

Model name	Test RMSE	Train RMSE	Variance
Baseline model	9.61	9.65	0.042
Linear Regression	8.14	8.20	0.069
Random Forest	3.82	3.27	-0.54
LightGBM	3.79	3.07	-0.72

We find that the RMSE of Light GBM is smaller but with a slightly higher variance so we choose the Light GBM as it is also faster and takes less memory as mentioned before and we will tune the parameters to overcome the problem of overfitting - to reduce variance - .

Feature Engineering

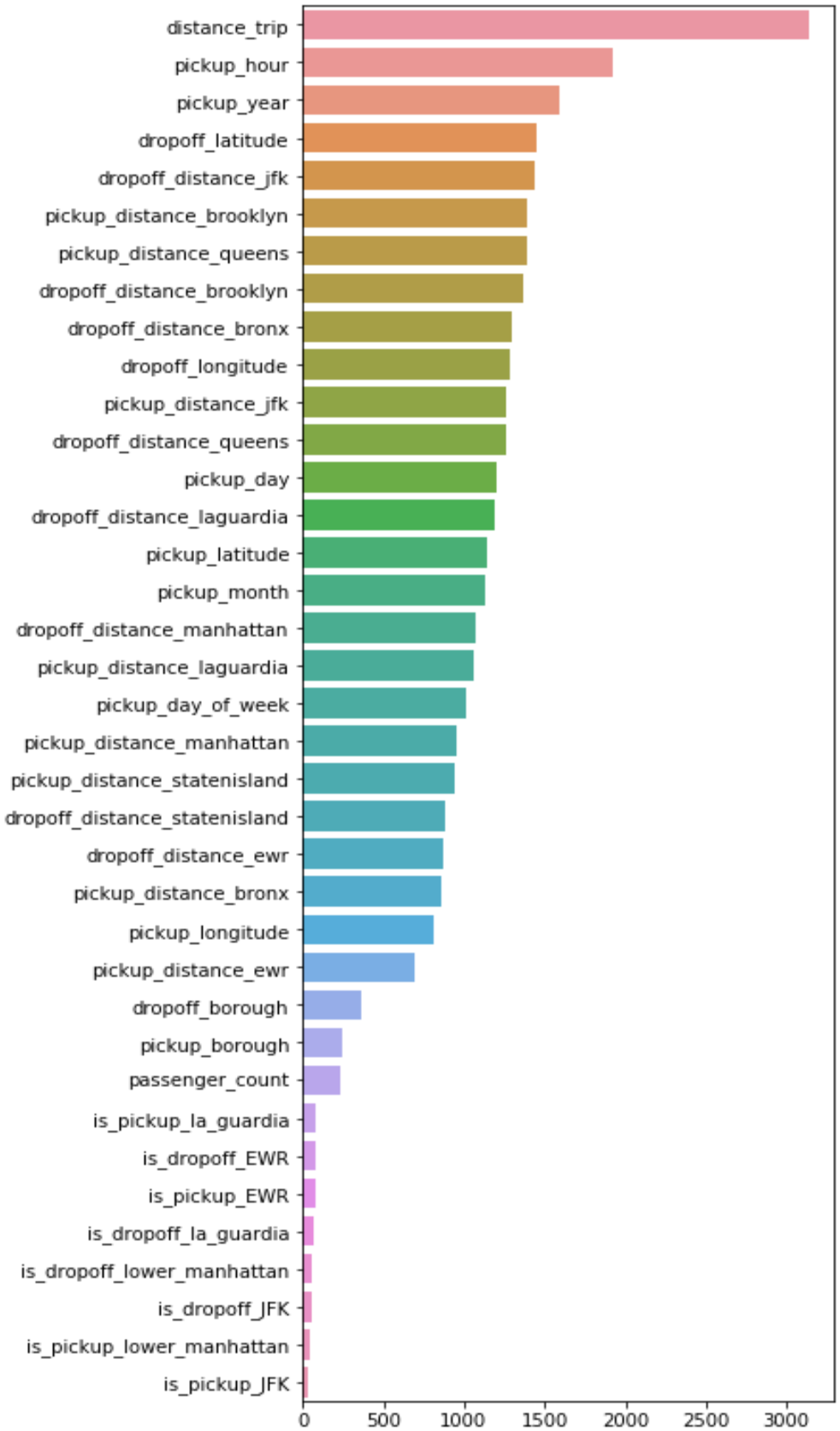
Feature Engineering is the process of transforming raw data into features that are input to the final models. The aim is to improve the accuracy of the models. Having good features means we can use simple models for producing better results**.**

We added these features:

- The distance of the trip.
- Whether the pickup or drop was from or to the airport.
- Which borough the pickup or drop was from.
- The pickup and drop distance from airports.
- The pickup and drop distance from each borough.
- Whether the pickup or drop was from or to lower_manhattan.

Applying the same Light GBM model discussed above on this feature engineering data resulted in an RMSE of 3.54 (drop from 3.79) and a variance of -0.83 (the variance increased so the next step is to tune the model to reduce it).

Here is a graph represents the importance of all features used in training:



Model Tuning and Final Results

In this phase we optimized some parameters in LightGBM to reduce overfitting like:

- **colsample_bytree**: This is the fraction of features that are to be used in each iteration of the tree building process.
- **subsample**: This indicates what fraction of the data is to be used in each iteration.
- **max_depth**: This indicates the maximum depth of the tree.
- **num_leaves:** number of leaves in full tree. Large number of leaves may cause overfitting.
- **n_estimators**: Number of boosted trees to fit.
- **learning_rate**: Boosting learning rate.

Where the first four parameters control overfitting while the last two are for better accuracy**.**

We used the hyperopt library in Python to tune the model. **Hyperopt** is a hyperparameter search package that implements various search algorithms to find the best set of hyperparameters within a search space**.**

To use Hyperopt, we must specify an objective/loss function to minimize, the search space and trials database.

- **The objective function** is to minimize the RMSE in a LightGBM Regressor, given a set of parameters.
- **The search space** defines the set of values a given parameter can take.
- After defining the objective function and the search space, we run **100 trials** and evaluate the result of the trials on our validation data to identify the best parameters.

The best parameters that we got for LightGBM with feature engineering were:

- `'colsample_bytree': 0.8288999698751736`
- `'subsample': 0.9920618722172806`
- `'subsample_freq': 100.0`
- `'max_depth': 9.0`
- `'num_leaves':  50`
- `'learning_rate': 0.06585147300487698`
- `'n_estimators': 1000`

Using these parameters, the LightGBM model resulted in an RMSE of 3.55 and variance of -0.25 (drop from -0.83).

Finally, This graph represents the probability density of the actual fare amount vs the predicted fare amount: