## CAB FAIR PREDICTION

If you torture the data long enough, it will confess

**Objective of the Project:**

In this project i am given a training set and test dataset of size (16067, 8)

(9914, 6) respectively and requirement is to apply analytics and design a system

Which can predict the cab fare in the New York City.

### **Project Outline:**

### This course is divided into the below sections:

1. Intro to the problem
2. Exploratory Data Analysis and preprocessing
3. Feature engineering and
4. Model building

Table of Content:

1. Problem statement
2. Hypothesis generation
3. Loading and summarizing data
4. Univariate and bivariate EDA
5. Missing value and outlier treatment
6. Model Building and
7. Conclusion

Problem Statement:

I am given a dataset which contains below predictors

* Pickup\_datetime
* Pickup\_longitude
* Pickup\_lattitude
* Dropoff\_longitue
* Dropoff\_lattitude and
* Passengers Count and
* Target variable i.e. fair Amount

So this is a regression problem where I have to predict continuous target variable fare amount.

**Hypothesis generation:**

Below are some of the factors which I think can affect the target variable.

1. **Pickup and drop-off longitude and latitude:** since cab fare vary along with the pickup and drop-off locations. So these variable will play important role in fare prediction.
2. **Airport Locations:** Pic and drop from airport may affect the cab fare.
3. **Number of passenger:** Fair can be differ in case of the number of passengers.

# **Loading** **the data:**

### For this practice problem, we have been given two CSV files: train, test

* Train file will be used for training the model, i.e. our model will learn from this file. It contains all the independent variables and the target variable.
* Test file contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data.

# **Understanding the Data:**

In this section, we will look at the structure of the train and test datasets. Firstly, we will check the features present in our data and then we will look at their data types.

**Train Data:**

|  |
| --- |
| Train Data columns (total 7 columns): |
| fare\_amount 16043 non-null object |
| pickup\_datetime 16067 non-null object |
| pickup\_longitude 16067 non-null float64 |
| pickup\_latitude 16067 non-null float64 |
| dropoff\_longitude 16067 non-null float64 |
| dropoff\_latitude 16067 non-null float64 |
| passenger\_count 16012 non-null float64 |

|  |
| --- |
| Test Data columns (total 6 columns): |
| pickup\_datetime 9914 non-null object |
| pickup\_longitude 9914 non-null float64 |
| pickup\_latitude 9914 non-null float64 |
| dropoff\_longitude 9914 non-null float64 |
| dropoff\_latitude 9914 non-null float64 |
| passenger\_count 9914 non-null int64 |
| Dtypes: float64 (4), int64 (1), object (1) |
|  |

**Missing Values Treatment:**

As we can see below dataset contains some missing values we need to remove the same before processing

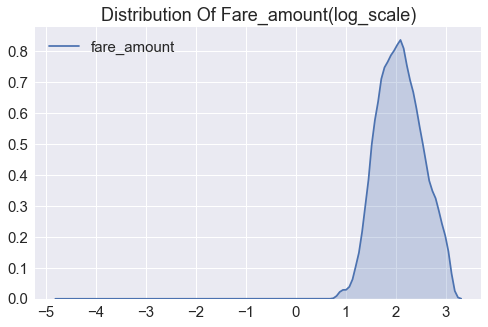
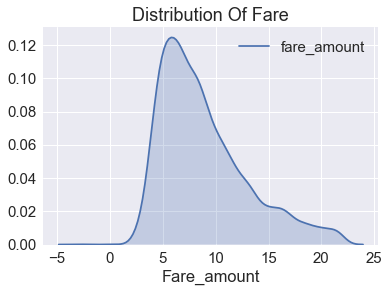
|  |
| --- |
| fare\_amount 24 |
| pickup\_datetime 0 |
| pickup\_longitude 0 |
| pickup\_latitude 0 |
| dropoff\_longitude 0 |
| dropoff\_latitude 0 |
| passenger\_count 55 |
| pickup year 0 |
| pickup\_day 0 |
| picup\_day 0 |
| pickup\_month 0 |
| pickup\_hour 0 |
| dtype: int64 |

**Exploratory data Analysis:**

In any analytics project, exploratory analysis and deriving new features is most crucial stage. In this project, we aim to clean the data, visualize the relationship between variables and also figure out new features that are better predictors of cab fare.

1. **Distribution of fare amount:**

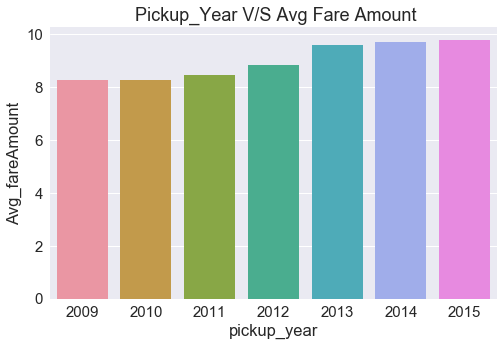
We first looked at the distribution of fare amount and found that there were many records where the fare was negative. Since, cost of a trip cannot be negative we removed such instances from the data. Also, fare amount follows long tail distribution. To understand the distribution of fare amount better we take a log transformation after removing the negative fares- this makes the distribution close to normal.



1. **Distribution of Pickup date time:**

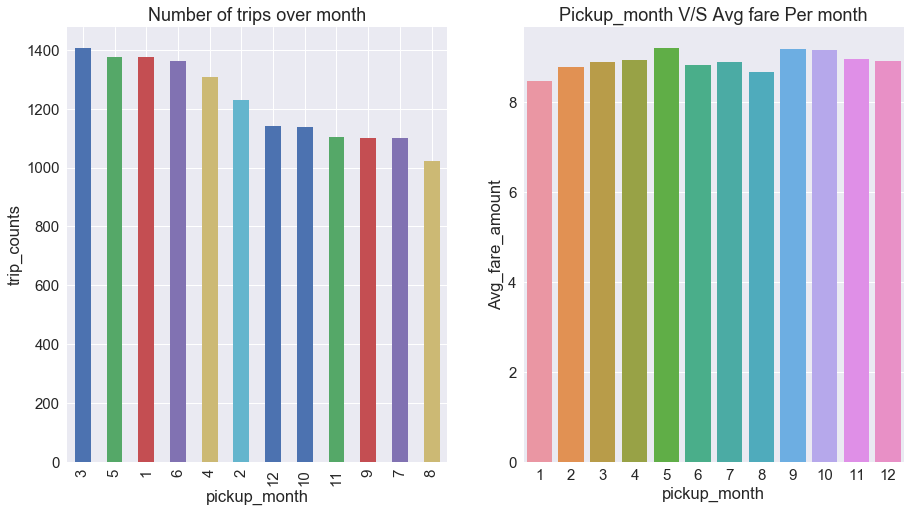
The first step to analyse how the fares have changed over time, is to create features like hour, day of the week, day, month, year from pickup date time. And will explore the year, Month, Days, Hours one by one

First let’s check how fair amount distributed with the year



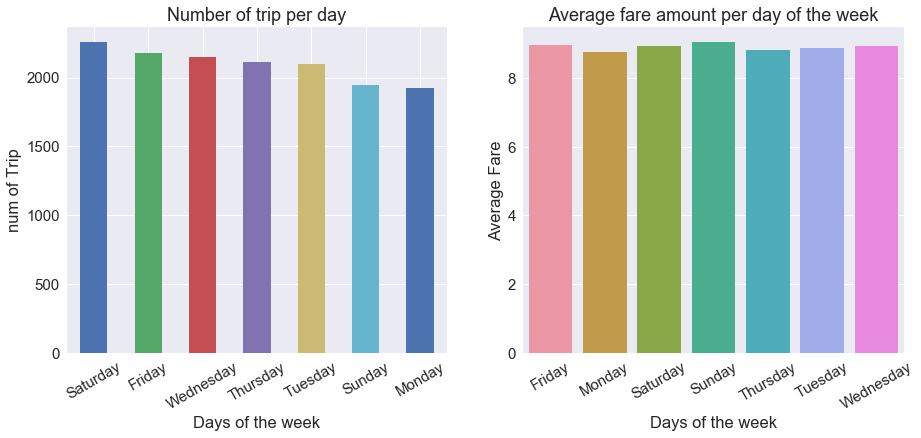
As expected, over years the average taxi fare has increased.

**Fare Amount Distribution over months:**

****

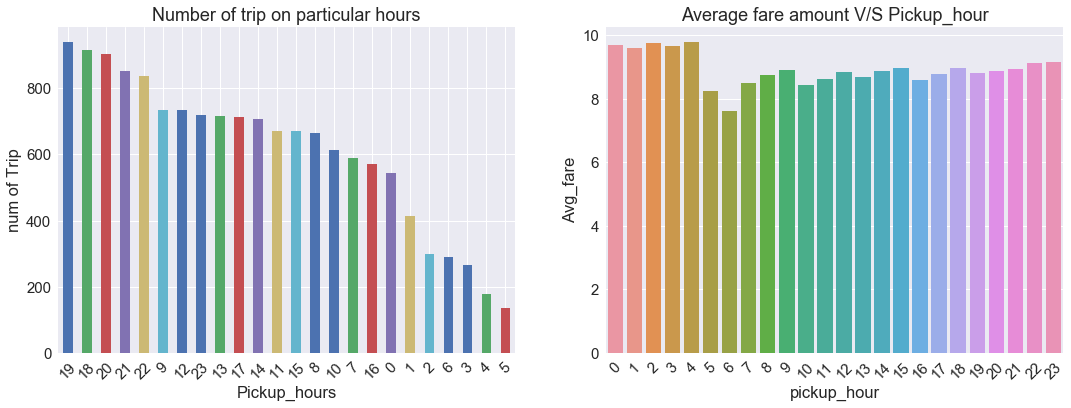
Over months, though there have been fewer pickups from July to December, the average fare is almost constant across months.

**Fare Amount Distribution over days of the week:**



We observed that though the number of pickups are higher on Saturday, the average fare amount is lower On Sunday though the number of trips are lower, avg fare amount is higher

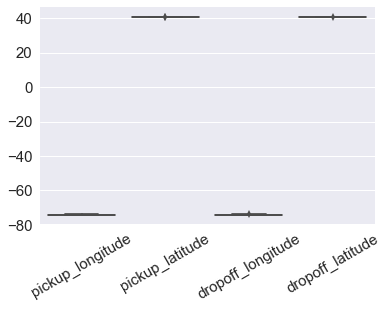
**Fare Amount Distribution over pickup hours:**



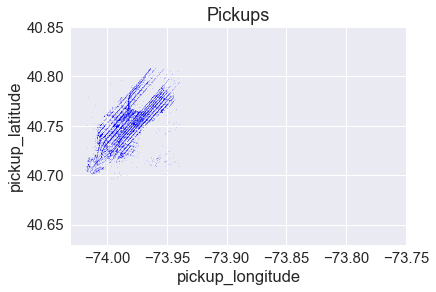
The average fare amount up to 5 am is the highest while the number of trips at 5 am are the least. The number of trips are highest after 18 and 19 hours

**Distribution of Geographical Features:** Given data is of New York City whose lat, long range is [((40,-74)] so train data having some outliers locations**.**

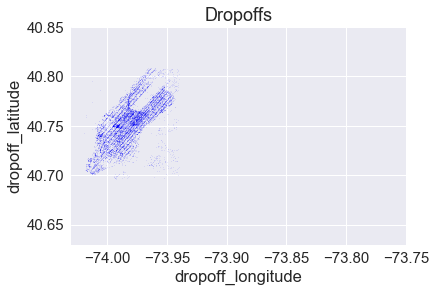
**Remove outlier locations using boxplot method**

****

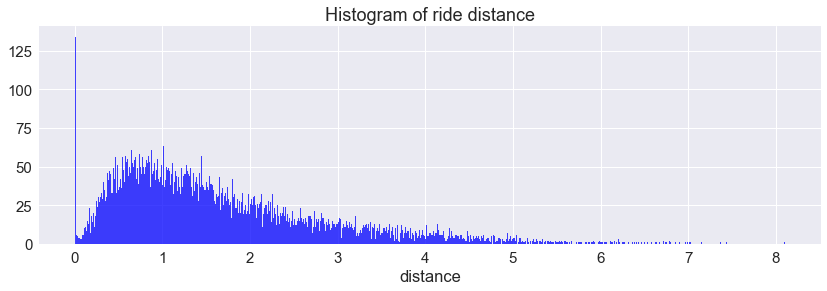
Let’s Check pick and drop off across New York:



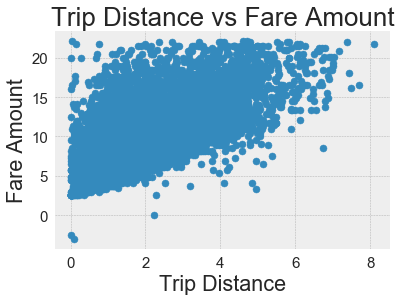
We can see that there is a high density of pickups near JFK



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

****

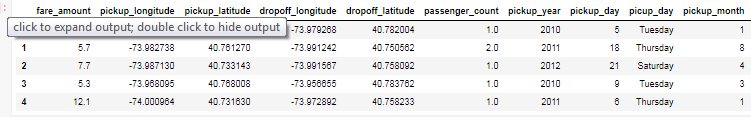
**Now let’s see distribution of distance Vs fare amount:**

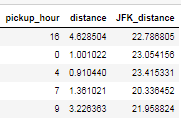
****

As we can see there is linear relationship between fare amount and Trip distance i.e. fare amount increasing with increase in distance.

**4 Feature Engineering:**

As we have seen earlier maximum pickup was from **JFK airport** and fare amount is highly depend upon **trip distance**, hence I will add these two new features into the train dataset.





1. **Modeling:**

After the Data Cleaning and Exploratory Analysis phase, we have finally arrived at the Model Building phase. The quality of results at the end of this phase depends on the data quality and features used for modelling.

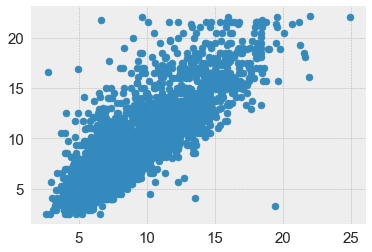
**Linear Regression:**It is used to find a linear relationship between the target and one or more predictors. The main idea is to identify a line that best fits the data. The best fit line is the one for which the prediction error is the least. This algorithm is not very flexible, and has a very high bias. Linear Regression is also highly susceptible to outliers as it tries to minimize the sum of squared errors.

Result:

**The test RMSE for Linear Regression model was 2.11, and the training RMSE was 2.19.** This model is an improvement on the baseline prediction. Still, the error rate is very high in this model, though the variance is very low (0.08). Let’s try a more complex model next.

|  |  |
| --- | --- |
| **Test RMSE for Linear Regression is** | **2.11** |
| **Train RMSE for Linear Regression is** | **2.19** |
| **Variance for Linear Regression is** | **0.08** |

**plot the predicted x\_test and y\_test values**

****

**Random Forest:**

Random Forest is far more flexible than a Linear Regression model. This means lower bias, and it can fit the data better. Complex models can often memorize the underlying data and hence will not generalize well. Parameter tuning is used to avoid this problem.

**Result: The Random Forest model gave an RMSE of 2.11 on validation data and train RMSE of 0.81.** There is a huge variation in the training and validation RMSE..

|  |  |
| --- | --- |
| Test RMSE for Random Forest is | 2.11 |
| Train RMSE for Random Forest is | 0.81 |
| Variance for Random Forest is | 1.29 |

## **Regression tree:**

 Regression tree building methodology allows input variables to be a mixture of continuous and categorical variables. A decision tree is generated when each decision node in the **tree** contains a test on some input variable's value.

**Result: Result:**

**The DT model gave an RMSE of 3.81 on validation data and train RMSE of 0.98.** There is a huge variation in the training and validation RMSE..

|  |  |
| --- | --- |
| Test RMSE for DT is | 3.81 |
| Train RMSE for DT is | 0.98 |
| Variance for DT is | 2.83 |

**Conclusion**

|  |  |  |  |
| --- | --- | --- | --- |
| Model Name | testRMSE | trainRMSE | Variance |
| Linear regression | 2.11 | 2.19 | 0.08 |
| Random Forest | 2.11 | 0.81 | 1.29 |
| Decision tree | 3.81 | 0.98 | 2.83 |

**As per above table I will go with the Linear regression model which having low RMSE and low variance between train and test data**