# Project Name: Cab Fare Prediction

# Contents

# 1.Introduction

**Problem Statement** –

You are a cab rental start-up company. You have successfully run the pilot project and now want to launch your cab service across the country. You have collected the historical data from your pilot project and now have a requirement to apply analytics for fare prediction. You need to design a system that predicts the fare amount for a cab ride in the city.

## Number of attributes:

· pickup_datetime - timestamp value indicating when the cab ride started.
· pickup_longitude - float for longitude coordinate of where the cab ride started.
· pickup_latitude - float for latitude coordinate of where the cab ride started.
· dropoff_longitude - float for longitude coordinate of where the cab ride ended.
· dropoff_latitude - float for latitude coordinate of where the cab ride ended.
· passenger_count - an integer indicating the number of passengers in the cab ride.

**Missing Values:** Yes

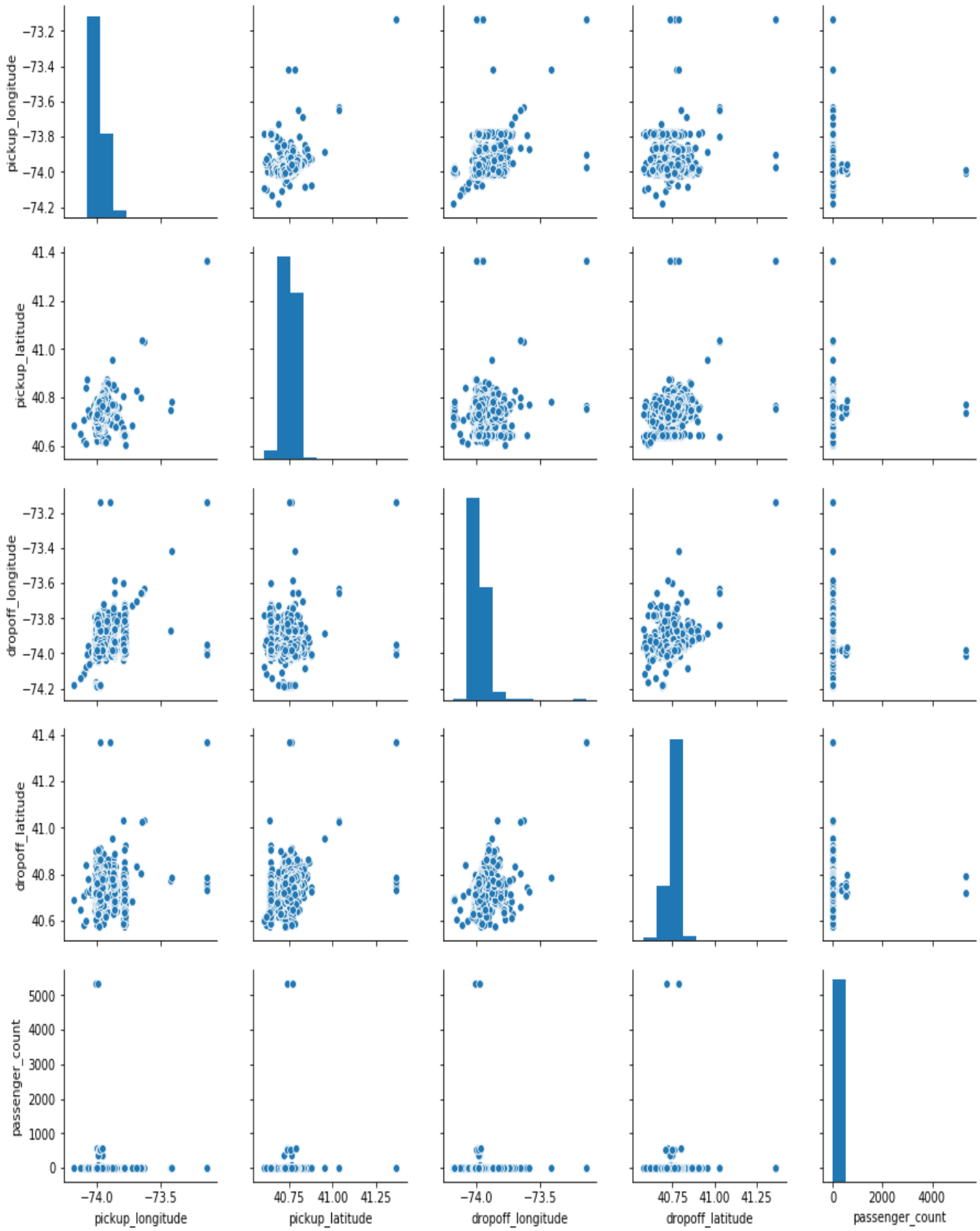
# 2.Methodology

Data pre-processing is the initial stage of any of project. In this stage we get the understanding of the data. We do this by looking at plots of independent variables vs target variables. If the data is messy, we try to improve it by sorting, deleting extra rows and columns. This stage is called as Exploratory Data Analysis. This stage generally involves data cleaning, merging, sorting, looking for outlier analysis, looking for missing values in the data, imputing missing values if found by various methods such as mean, median, mode, KNN imputation, etc.

Further we will look into what Pre-Processing steps do this project was involved in. Getting feel of data via visualization:

Some Scatterplots:

- They are used for Bivariate Analysis.
- Here we have plotted Scatter plot of all the variables to check and visualize the relationship between the variables.


pickup_longitude
pickup_latitude
dropoff_longitude
dropoff_latitude
passenger_count
−73.2
−73.4
−73.6
−73.8
−74.0
−74.2
41.4
41.2
41.0
40.8
40.6
5000
4000
3000
2000
1000
0
−74.0
−73.5
40.75
41.00
41.25
2000
4000

## 2.1 Imputing the data and removing values which are not within desired range(outlier) depending upon basic understanding of dataset.

In this step we will remove values in each variable which has NA values or fill with Imputation method like mean, median, mode, etc.
The rows of fare_amount variable are dropped where there are NA values because the fare_amount is dependent variable which is to be predicted and count of NA is 24, Overall it won't affect the model by dropping the rows.

```
fare_amount          24
pickup_datetime       0
pickup_longitude      0
pickup_latitude       0
dropoff_longitude     0
dropoff_latitude      0
passenger_count      55
dtype: int64
```

The NA values of passenger_count are filled with mode because this variable is of continuous type.

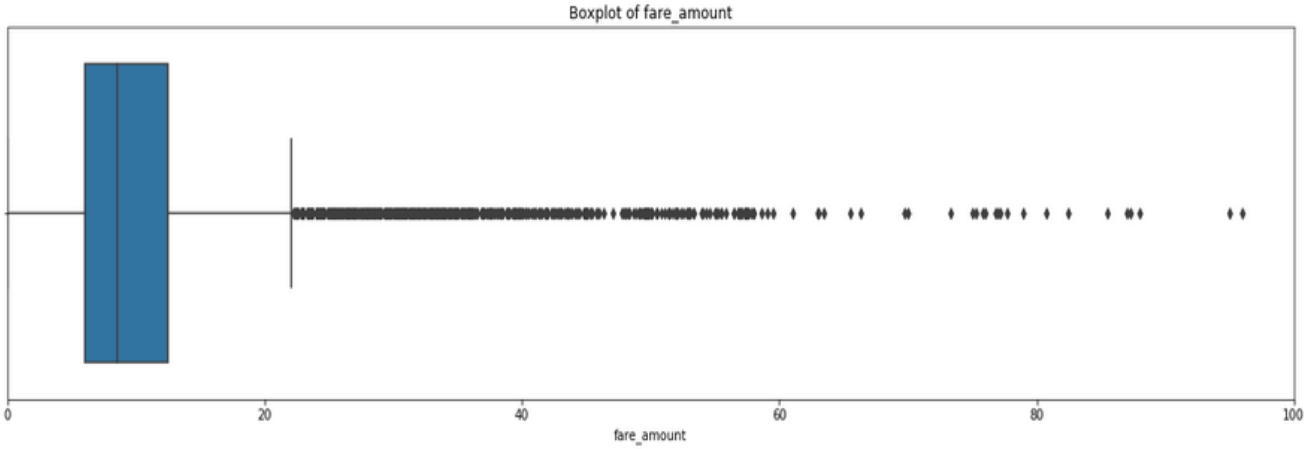
**Outlier Analysis :**

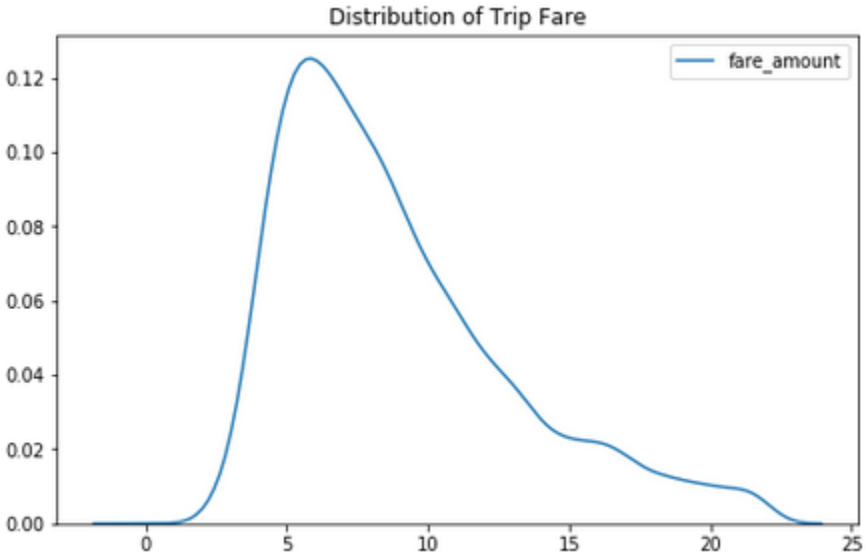
- The cab fare amount can't be in negative, so dropping the whole rows.

| | fare_amount | pickup_datetime | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_latitude | passenger_count |
|---|---|---|---|---|---|---|---|
| 2039 | -2.9 | 2010-03-09 23:37:10 UTC | -73.789450 | 40.643498 | -73.788665 | 40.641952 | 1.0 |
| 2486 | -2.5 | 2015-03-22 05:14:27 UTC | -74.000031 | 40.720631 | -73.999809 | 40.720539 | 1.0 |
| 13032 | -3.0 | 2013-08-30 08:57:10 UTC | -73.995062 | 40.740755 | -73.995885 | 40.741357 | 4.0 |

- By looking at the test dataset and calculating the range values of Pickup Latitude, Dropoff Latitude, Pickup Longitude, Dropoff Longitude we can say that there values which are out of range. So dropping such rows which are out of range.

- To remove outliers of fare_amount boxplot is plot. And using IQR(Inter quartile range) outliers are detected and removed.



# 2.2 Feature Engineering

Feature Engineering is used to drive new features from existing features.

- The 2 columns ‘latitude_diff’ and ‘longitude_diff’ which are latitude difference and longitude difference respectively are calculated using pickup_longitude, dropoff_longitude, pickup_latitude, dropoff_latitude.

- **For ‘pickup_datetime’ variable:**

We will use this timestamp variable to create new variables.
New features will be year, month, day_of_week, hour.
‘year’ will contain only years from pickup_datetime. For ex. 2009, 2010, 2011, etc. ‘month’ will contain only months from pickup_datetime. For ex. 1 for January, 2 for February, etc. ‘day_of_week’ will contain only week from pickup_datetime. For ex. 1 which is for Monday,2 for Tuesday,etc. ‘hour’ will contain only hours from pickup_datetime. For ex. 1, 2, 3, etc.

- By using the longitude and latitude values Manhattan distance and Euclidean distance is calculated.

# 3.Splitting train and Validation Dataset

a) We have used sklearn’s train_test_split() method to divide whole Dataset into train and validation dataset.

b) 20% is in validation dataset and 80% is in training data.

c) We will test the performance of model on validation dataset.

d) The model which performs best will be chosen to perform on test dataset provided along with original train dataset.

e) X_train y_train--are train subset.

f) X_test y_test--are validation subset.

# 4.Hyperparameter Optimization

a. To find the optimal hyperparameter we have used sklearn.model_selection.GridSearchCV. and sklearn.model_selection.RandomizedSearchCV

b. GridSearchCV tries all the parameters that we provide it and then returns the best suited parameter for data.

c. We gave parameter dictionary to GridSearchCV which contains keys which are parameter names and values are the values of parameters which we want to try for.

**Below are best hyperparameter we found for different models:**

**1. Multiple Linear Regression:**

Tuned Decision reg Parameters: {'copy_X': True, 'fit_intercept': False}
Best score is 0.38598545790826344

**2. Ridge Regression:**

Tuned Decision ridge Parameters: {'alpha': 0.0001, 'max_iter': 500, 'normalize': False}
Best score is  0.38576373297263056

**3. Lasso Regression:**

Tuned Decision lasso Parameters: {'alpha': 0, 'max_iter': 4500, 'normalize': True}
Best score is 0.3857343722187197

**4. Decision Tree Regression:**

Tuned Decision lasso Parameters: {'alpha': 0, 'max_iter': 4500, 'normalize': True}
Best score is 0.3857343722187197

**5. Random Forest Regression:**

Tuned Random Forest Parameters: {'n_estimators': 300, 'min_samples_split': 4, 'min_samples_leaf': 4, 'max_features': 'log2', 'max_depth': 16, 'bootstrap': True}
Best score is 0.7312742797500662

# 5. Model Development

Our problem statement wants us to predict the fare_amount. This is a Regression problem. So, we are going to build regression models on training data and predict it on test data. In this project I have built models using 5 Regression Algorithms:

I. Linear Regression
II. Ridge Regression
III. Lasso Regression
IV. Decision Tree
V. Random Forest

We will evaluate performance on validation dataset which was generated using Sampling. We will deal with specific error metrics like –
Regression metrics for our Models:
- r square
- MAPE(Mean Absolute Percentage Error)
- MSE(Mean square Error)
- RMSE(Root Mean Square Error)
- RMSLE( Root Mean Squared Log Error)

# 6.Model Performance

Here, we will evaluate the performance of different Regression models based on different Error Metrics.

**1. Multiple Linear Regression:**

R^2: 0.4686137747581447
Root Mean Squared Error: 3.0546730009060696
MAPE:  54.977680125669515
Mean Squared Error: 9.331027142464492
Mean Absolute Error: 2.325668627212664

**2. Ridge Regression:**

R^2: 0.4687912223924097
Root Mean Squared Error: 3.0541629295766226
MAPE:  54.525886659931686
Mean Squared Error: 9.327911200400058
Mean Absolute Error: 2.324710596189648

**3. Lasso Regression:**

R^2: 0.4934779566589604
Root Mean Squared Error: 2.9823510001334346
MAPE:  52.70949052952232
Mean Squared Error: 8.894417487996899
Mean Absolute Error: 2.259150982295565

**4. Decision Tree Regression:**

R^2: 0.7139731276057336
Root Mean Squared Error: 2.2411090989420903
MAPE:  36.80006737030525
Mean Squared Error: 5.022569993361029
Mean Absolute Error: 1.5787100309686268

**5. Random Forest Regression:**

R^2: 0.7444631247472726
Root Mean Squared Error: 2.118294653840564
MAPE:  37.26564340287496
Mean Squared Error: 4.487172240489516
Mean Absolute Error: 1.4948072045341703

**Conclusion: On the basis of RMSE metric, Random Forest Regression gave the best score as 2.118294653840564.**