Project Report
Cab Fare Prediction

Mohammed Shadir

20 SEPTEMBER 2019

Contents

Chapter 1
Introduction

1.1 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.

The objective of this project is to predict Cab Fare amount.

1.2 Data

There are 7 variables in our data in which 6 are independent variables and 1 (Fare Amount) is dependent variable. Since our target variable is continuous in nature, this is a regression problem.

Variables Information:

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.

• fare_amount - an integer indicating the amount incurred for the cab ride.

1.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics. In the given data set there are 7 variables and data types of all variables are either float64 or int64. There are 16067 observations and 7 columns in our training data set and 9914 observations and 6 columns excluding target variable in test data set. Missing value is also present in our data.

List of columns and their number of unique values -

```
fare_amount            467
pickup_datetime      16021
pickup_longitude     13789
pickup_latitude      14241
dropoff_longitude    13887
dropoff_latitude     14263
passenger_count         27
```

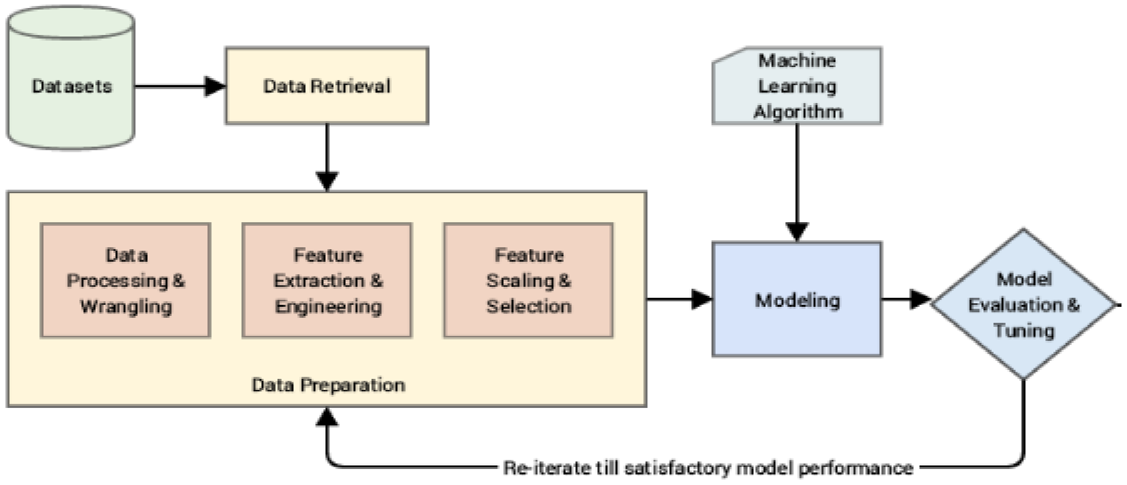
From EDA we have concluded that there are 5 continuous variable, 1 categorical variable and 1 Date Time Variable in nature.

Chapter 2

Methodology

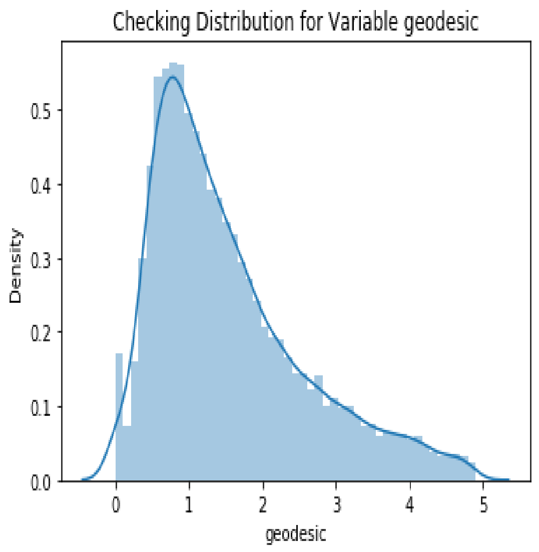
Before feeding the data to the model we need to clean the data and convert it to a proper format. It is the most crucial part of data science project we spend almost 80% of time in it.

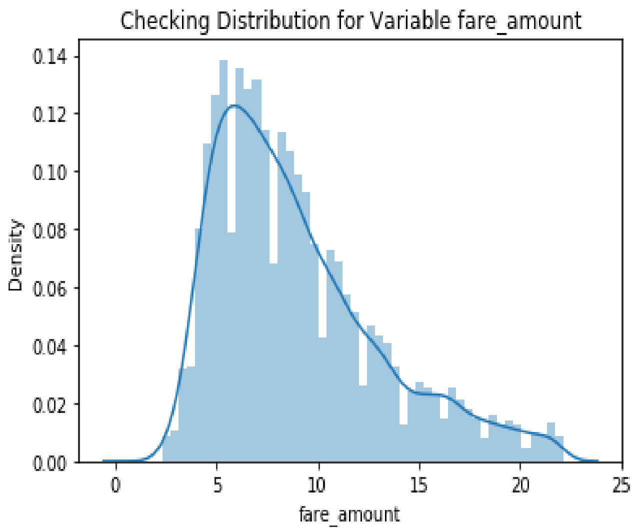
For implementation of this project, we're following the traditional CRISP-DM Process and has been divided into three major sections. The first section involves the data preparations. In the second section some exploratory data analysis is conducted and finally third section provides solution to a predictive model, and fits the model.

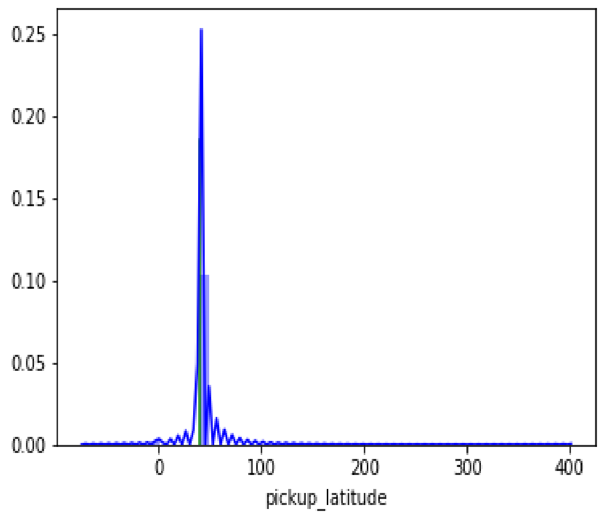

Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Re-iterate till satisfactory model performance

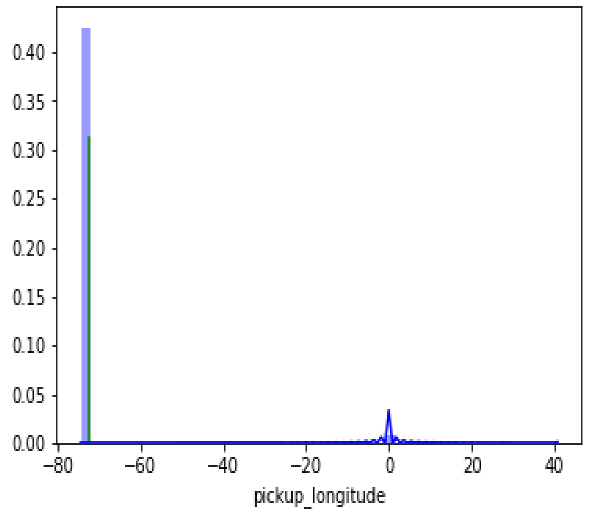
2.1 Pre Processing

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms looking at data refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis. To start this process we will first try and look at all the probability distributions of the variables. Most analysis like regression, require the data to be normally distributed. We can visualize that in a glance by looking at the probability distributions or probability density functions of the variable.









2.2.1 <u>Feature Engineering</u>

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

1. For 'pickup_datetime' variable:

We will use this timestamp variable to create new variables.

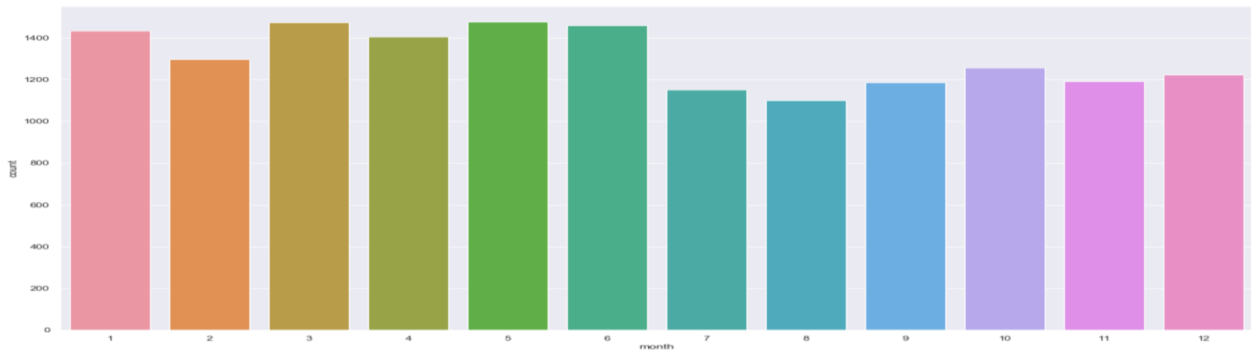
New features will be year, month, day_of_week, hour.

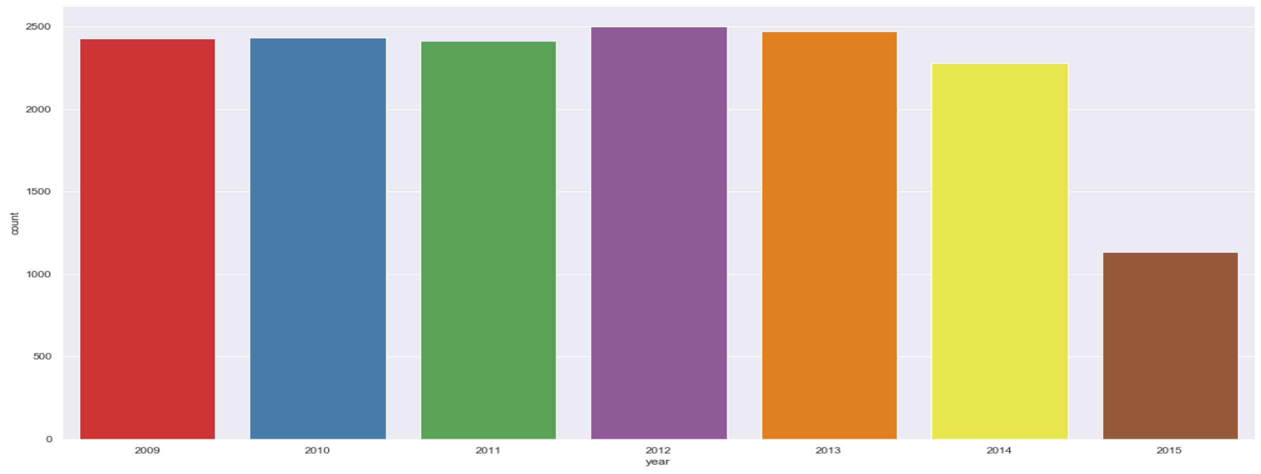
'year' will contain only years from pickup_datetime. For ex. 2009, 2010, 2011, etc.

'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





As we have now these new variables we will categorize them to new variables like Session from hour column, seasons from month column, week:weekday/weekend from day_of_week variable.

So, session variable which will contain categories—morning, afternoon, evening, night_PM, night_AM.

Seasons variable will contain categories—spring, summer, fall, winter.

Week will contain categories—weekday, weekend.

We will one-hot-encode session, seasons, week variable.

2. For 'Latitudes' and 'Longitudes' variables:

As we have latitude and longitude data for pickup and dropoff, we will find the distance the cab travelled from pickup and dropoff location.

We will use both haversine and vincenty methods to calculate distance. For haversine, variable name will be 'great_circle' and for vincenty, new variable name will be 'geodesic'.

Columns in training data after feature engineering:

```
Index(['fare_amount', 'passenger_count_1.0', 'passenger_count_2.0',
    'passenger_count_3.0', 'passenger_count_4.0', 'passenger_count_5.0',
    'year_2009', 'year_2010', 'year_2011', 'year_2012', 'year_2013',
    'year_2014', 'month_1', 'month_2', 'month_3', 'month_4', 'month_5',
    'month_6', 'month_7', 'month_8', 'month_9', 'month_10', 'month_11',
    'day_of_week_0', 'day_of_week_1', 'day_of_week_2', 'day_of_week_3',
    'day_of_week_4', 'day_of_week_5', 'hour_0', 'hour_1', 'hour_2',
    'hour_3', 'hour_4', 'hour_5', 'hour_6', 'hour_7', 'hour_8', 'hour_9',
    'hour_10', 'hour_11', 'hour_12', 'hour_13', 'hour_14', 'hour_15',
    'hour_16', 'hour_17', 'hour_18', 'hour_19', 'hour_20', 'hour_21',
    'hour_22', 'session_afternoon', 'session_evening', 'session_morning',
    'session_night_PM', 'seasons_fall', 'seasons_spring', 'seasons_summer',
    'week_weekend', 'great_circle', 'geodesic'],
   dtype='object')
```

Columns in testing data after feature engineering:

```
Index(['passenger_count_1', 'passenger_count_2',
       'passenger_count_3', 'passenger_count_4', 'passenger_count_5',
       'year_2009', 'year_2010', 'year_2011', 'year_2012', 'year_2013',
       'year_2014', 'month_1', 'month_2', 'month_3', 'month_4', 'month_5',
       'month_6', 'month_7', 'month_8', 'month_9', 'month_10', 'month_11',
       'day_of_week_0', 'day_of_week_1', 'day_of_week_2', 'day_of_week_3',
       'day_of_week_4', 'day_of_week_5', 'hour_0', 'hour_1', 'hour_2',
       'hour_3', 'hour_4', 'hour_5', 'hour_6', 'hour_7', 'hour_8', 'hour_9',
       'hour_10', 'hour_11', 'hour_12', 'hour_13', 'hour_14', 'hour_15',
       'hour_16', 'hour_17', 'hour_18', 'hour_19', 'hour_20', 'hour_21',
       'hour_22', 'session_afternoon', 'session_evening', 'session_morning',
       'session_night_PM', 'seasons_fall', 'seasons_spring', 'seasons_summer',
       'week_weekend', 'great_circle', 'geodesic'],
      dtype='object')
```

3. Passenge_count variable :

As passenger count variable contains unique values, we will convert it into a factor variable.

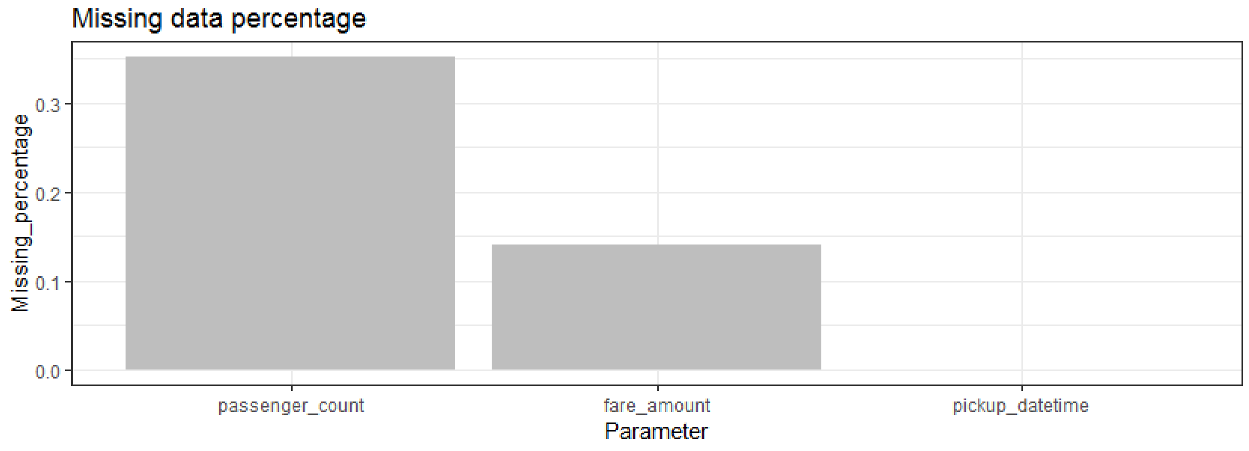
Also, Passenger count of any vehicle (i.e a Car can reach upto max 7, hence any number of passengers more than 7 is an outlier and can be removed)

After Feature Engineering, we lost 16067-15661=406 observations because of non-sensical values.

2.2.2 Missing Value Analysis

In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data. If a columns has more than 30% of data as missing value either we ignore the entire column or we ignore those observations. In the given data the maximum percentage of missing value is 0.35% for **passenger count** column. So we will compute missing value for all the columns.

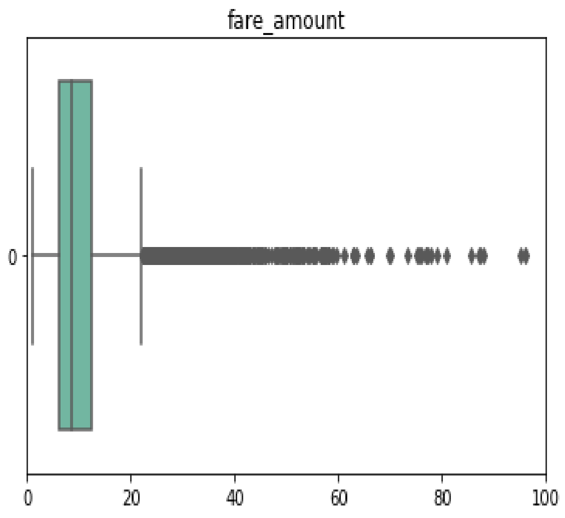
In this project we have used KNN imputation method to impute missing value.

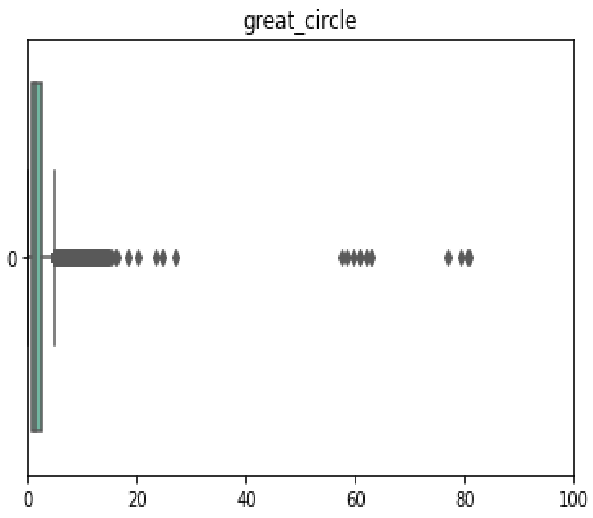


2.1.3 Outlier Analysis

We can clearly observe from these probability distributions that most of the variables are skewed. The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data. One of the other steps of pre-processing apart from checking for normality is the presence of outliers. In this case we use a classic approach of removing outliers. We visualize the outliers using boxplots.

In figure we have plotted the boxplots of the 11 predictor variables with respect to **Absenteeism time in hour**. A lot of useful inferences can be made from these plots. First as you can see, we have a lot of outliers and extreme values in each of the data set.

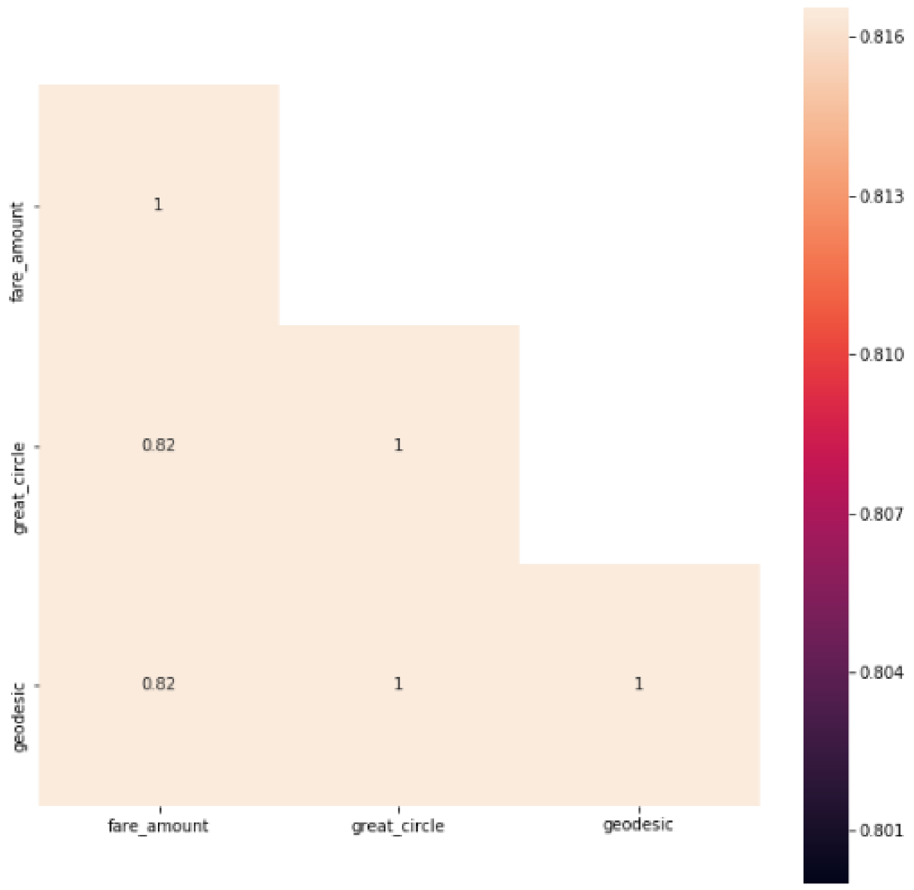




From the boxplot almost all the variables consists of outliers. We have converted the outliers (data beyond minimum and maximum values) as NA i.e. missing values and fill them by **KNN** imputation method.

2.1.4 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. Selecting subset of relevant columns for the model construction is known as Feature Selection. We cannot use all the features because some features may be carrying the same information or irrelevant information which can increase overhead. To reduce overhead we adopt feature selection technique to extract meaningful features out of data. This in turn helps us to avoid the problem of multi collinearity. In this project we have selected **Correlation Analysis** for numerical variable and **ANOVA** (Analysis of variance) for categorical variable.



From correlation analysis we have found that **great_circle** and **geodesic** has high correlation (>0.99), so we have excluded the **great_circle** column.

2.2.5 Feature Scaling

Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. Since our data is not uniformly distributed we will use **Normalization** as Feature Scaling Method

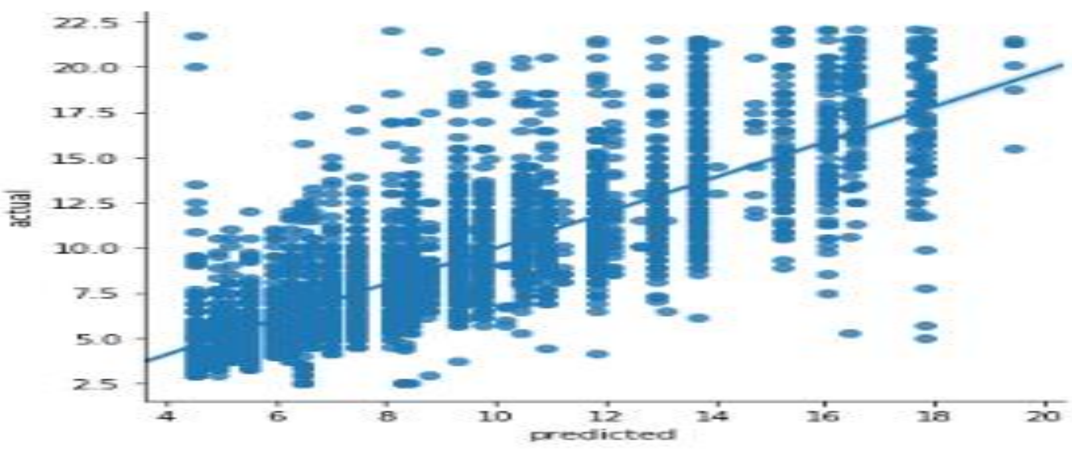
2.2 Modeling

After a thorough preprocessing we will be using some regression models on our processed data to predict the target variable. Following are the models which we have built –

2.2.1 Decision Tree

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Each branch connects nodes with "and" and multiple branches are connected by "or". It can be used for classification and regression. It is a supervised machine learning algorithm. Accept continuous and categorical variables as independent variables. Extremely easy to understand by the business users.

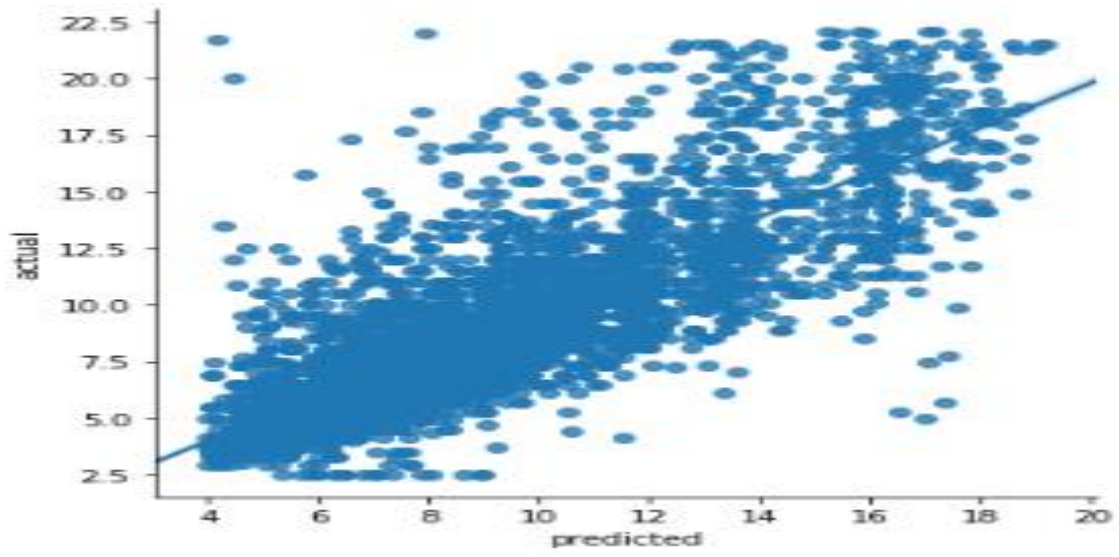
Error Metrics	R square	Adj R square	MAPE	MSE	RMSE	RMSLE
Train	0.6939	0.6923	19.2557	5.2269	2.2862	0.2154
Validation	0.6629	0.6575	20.1561	5.6740	2.3820	0.2249



2.2.2 <u>Random Forest</u>

Random Forest is an ensemble technique that consists of many decision trees. The idea behind Random Forest is to build n number of trees to have more accuracy in dataset. It is called random forest as we are building n no. of trees randomly. In other words, to build the decision trees it selects randomly n no of variables and n no of observations to build each decision tree. It means to build each decision tree on random forest we are not going to use the same data. The RMSE value and R^2 value for our project in R and Python are –

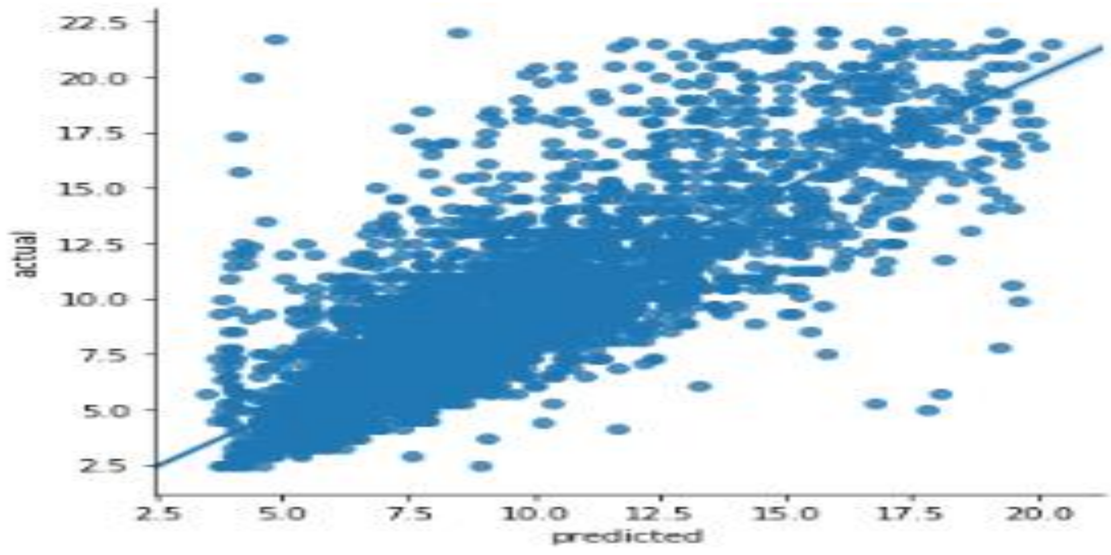
Error Metrics	R square	Adj R square	MAPE	MSE	RMSE	RMSLE
Train	0.7385	0.7371	18.0669	4.46611	2.1133	0.1996
Validation	0.6694	0.6642	20.1381	5.5641	2.3588	0.2243



2.2.3 <u>Multiple Linear Regression</u>

Linear Regression is one of the statistical methods of prediction. It is applicable only on continuous data. To build any model we have some assumptions to put on data and model. Here are the assumptions to the linear regression model.

Error Metrics	R square	Adj R square	MAPE	MSE	RMSE	RMSLE
Train	0.6679	0.6661	20.2960	5.672	2.3815	0.2264
Validation	0.6690	0.6637	20.2168	5.5715	2.3604	0.2244



Lasso Regression

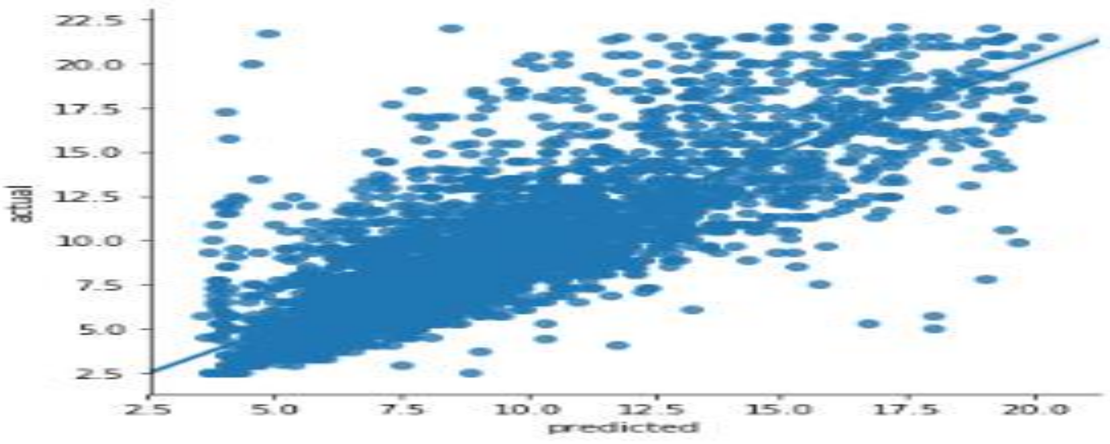
Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

The acronym "LASSO" stands for Least Absolute Shrinkage and Selection Operator.

L1 Regularization:

Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g. Ridge regression) doesn't result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge.

Error Metrics	R square	Adj R square	MAPE	MSE	RMSE	RMSLE
Train	0.6677	0.6660	20.2196	5.67469	2.3821	0.2263
Validation	0.6709	0.6657	20.0861	5.5395	2.3536	0.2236

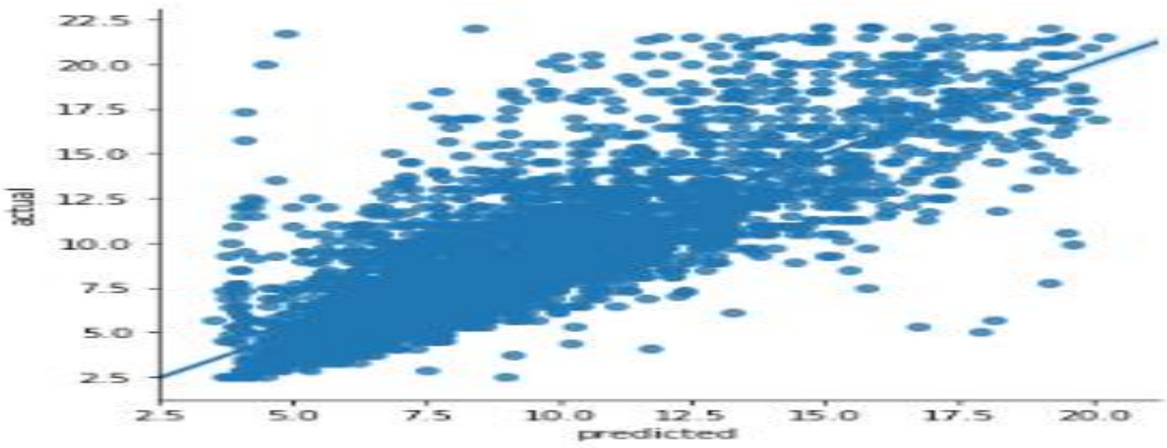


Ridge Regression

Ridge regression belongs a class of regression tools that use L2 regularization. The other type of regularization, L1 regularization, limits the size of the coefficients by adding an L1 penalty equal to the absolute value of the magnitude of coefficients. This sometimes results in the elimination of some coefficients altogether, which can yield sparse models. L2 regularization adds an L2 penalty, which equals the square of the magnitude of coefficients. All coefficients are shrunk by the same factor (so none are eliminated). Unlike L1 regularization, L2 will not result in sparse models.

A tuning parameter (λ) controls the strength of the penalty term. When λ = 0, ridge regression equals least squares regression. If λ = ∞, all coefficients are shrunk to zero. The ideal penalty is therefore somewhere in between 0 and ∞.

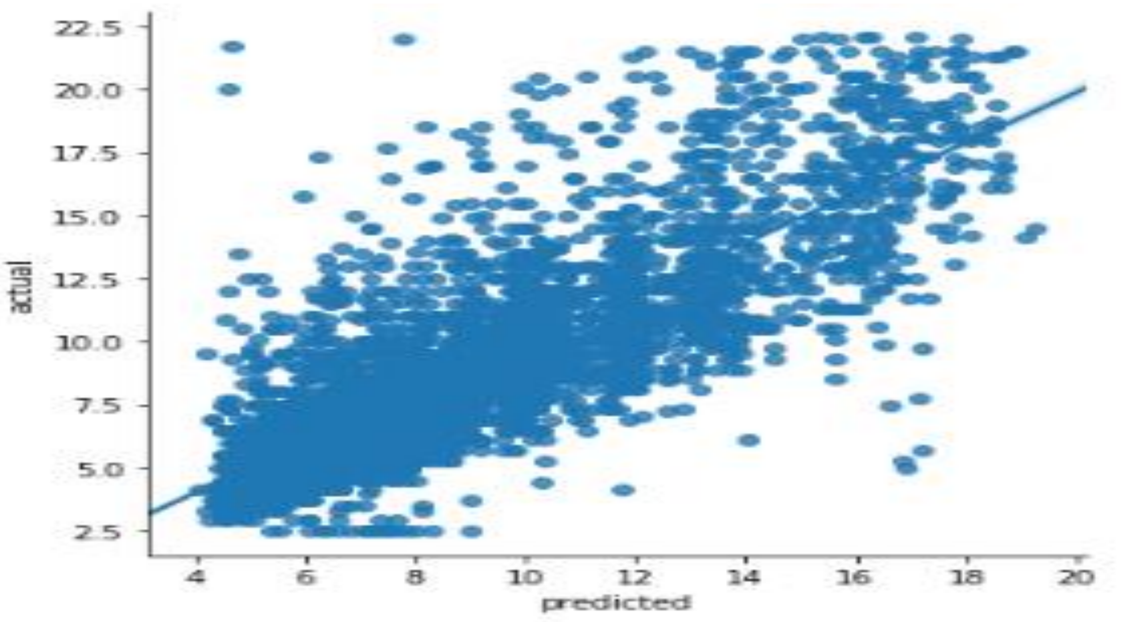
Error Metrics	R square	Adj R square	MAPE	MSE	RMSE	RMSLE
Train	0.6680	0.6662	20.2874	5.6706	2.3813	0.2264
Validation	0.6692	0.6639	20.2025	5.5681	2.3596	0.2243



2.2.4 Gradient boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Error Metrics	R square	Adj R square	MAPE	MSE	RMSE	RMSLE
Train	0.6899	0.6883	19.4158	5.2962	2.3013	0.2161
Validation	0.6719	0.6667	20.0783	5.5224	2.3499	0.2235



Chapter 3

Conclusion

In this chapter we are going to evaluate our models, select the best model for our dataset and try to get answers of the asked questions.

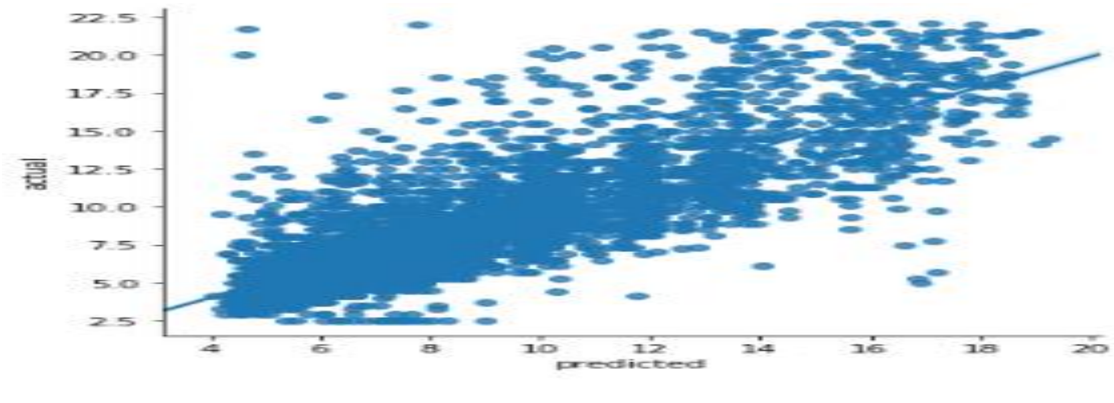
3.1 Model Evaluation

In the previous chapter we have seen the **Root Mean Square Error** (RMSE) and **R-Squared** Value of different models. **Root Mean Square Error** (RMSE) is the standard deviation of the residuals (prediction **errors**). Residuals are a measure of how far from the regression line data points are, RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Whereas **R-squared** is a relative measure of fit, **RMSE** is an absolute measure of fit. As the square root of a variance, **RMSE** can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of **RMSE** and higher value of **R-Squared Value** indicate better fit.

3.2 Model Selection

From the observation of all **RMSE Value** and **R-Squared** Value, we have concluded that We should train a XGBoost model on entire training dataset and use that model to predict on test data since it offers higher accuracy and less error. Also, we have saved model for later use.

Error Metrics	R square	Adj R square	MAPE	MSE	RMSE	RMSLE
Test	0.6870	0.6858	19.5191	5.3260	2.3078	0.2244



The RMSE value of Testing data and Training does not differs a lot this implies that it is not the case of overfitting.

Hence, XGBoost model has been implemented successfully with best accuracy scores and can now be implemented for predicting the cab fare prices for real time as well as future predictions.