SURAJ MOHAN RAUT

15/NOV/2019

cab fare prediction project

DATA SCIENCE

**INDEX**

**Chapter 1 Introduction**

* 1. Problem Statement
  2. Data
  3. Programming languages

**Chapter 2 Methodology**

* 1. Pre-Processing
  2. Modelling
  3. Model Selection

**Chapter 3 Pre-processing**

* 1. Understanding the data
  2. Missing value analysis
  3. Outliers analysis
  4. (Feature Engineering) Creating some new variables from the given variables
     1. Dependent and Independent Variables
     2. Uniqueness of Variables
     3. Dividing the variables categories

**Chapter 4 Modelling**

* 1. Linear Regression
  2. Decision Tree
  3. Random Forest

**Chapter 5 Conclusion**

* 1. Model Evaluation
  2. Model Selection
  3. Some Visualization facts

5.3.1 Number of passengers and fare

5.3.2 Week day and fare

5.3.3 Impact of day on the number of cab rides

**Chapter 6 Instructions to deploy and run code**

**Appendix**

**References**

**Chapter 1**

# Introduction

Cab rental services are increasing day by day. There are many things /processes such as drivers, maintenances, software used for hiring, which make this industry easy to get inside. So, one of the major process of the cab rental industry is predicting the fare amount of the journey of the rented cab. One of the applications of software used is to predict the fare of the hired cab. This makes the cab rental services easy to use for both service providers: - drivers as well as consumers: - passengers. So, this is our introductory project to get glimpse of how data science is used in industry of the cab fare prediction

## 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.**

## Data

It is often said Data is new oil, this because the information derived from data is useful in vast areas. In our project data provided to us are train\_cab.csv as training data for our model/machine learning algorithm and test.csv is the data on which our built model will be tested upon. Train data contains all columns which are fare\_amount, pickup\_datetime, pickup\_longitude, pickup\_latitude, dropoff\_longitude, dropoff\_latitude and passenger\_count. But the test data contains all columns except fare\_amount which we have to predict using final model.

We have been provided with two data sets

1. train\_cab.csv

2. test.csv

|  |  |
| --- | --- |
| **Variables** | **Description** |
| **fare\_amount** | Fare amount |
| **pickup\_datetime** | Cab pickup date with time |
| **pickup\_longitude** | Pickup location longitude |
| **pickup\_latitude** | Pickup location latitude |
| **dropoff\_longitude** | Drop location longitude |
| **dropoff\_latitude** | Drop location latitude |
| **passenger\_count** | Number of passengers sitting in the cab |

* 1. **Programming Languages**
     1. R
     2. Python

# Chapter 2

# Methodology

* 1. **Pre-Processing**

Data pre-processing is the first step of any type of project. In this stage we get the feel 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, we also check the data type of each variable and convert if required that is known as coercion.

After cleaning the data (doing above things) the following processes are executed

* looking for missing values in the data,
* imputing missing values if found by various methods such as mean, median, mode, KNN imputation, etc.
* also performing outlier analysis,
* Outlier Analysis
* Feature engineering (extracting new features from existing features)
* Feature Selection
* Features Scaling o Skewness and Log transformation
* Visualization
  1. **Modelling**

Once all the Pre-Processing steps are done on our data set, next step is modelling. Modelling plays an important role to find out the good inferences from the data. **Choice of models depends upon the problem statement and data set.** As per our problem statement and dataset, we are predicting a continuous variable fare\_amount so we have to use regression modelling. we have following types of machine learning algorithms for regression type problem: -

* Linear regression
* Decision Tree
* Random forest
  1. **Model Selection**

The final step of our methodology will be the selection of the model based on the different output and results shown by different models. We have multiple parameters out of we will use **MAPE** which is suitable for our problem statement

**Chapter 3**

**Pre-processing**

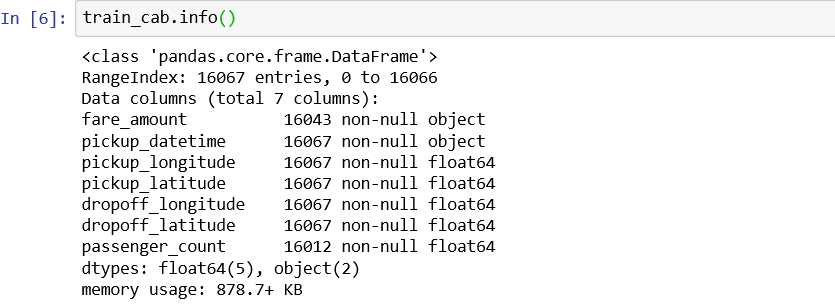
* 1. **Understanding the data**

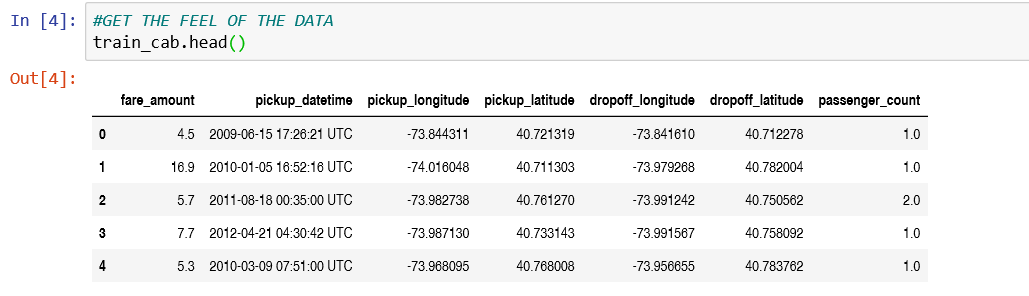
We have been provided with two data sets

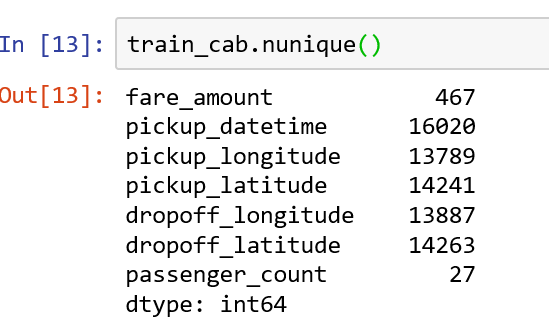
1. train\_cab.csv

2. test.csv

**Analysing train\_cab.csv**







From the above images from python

No. of variables =7

No. of independent variables are 6

No. of dependent variables is 1

Data type of the variables are shown in image

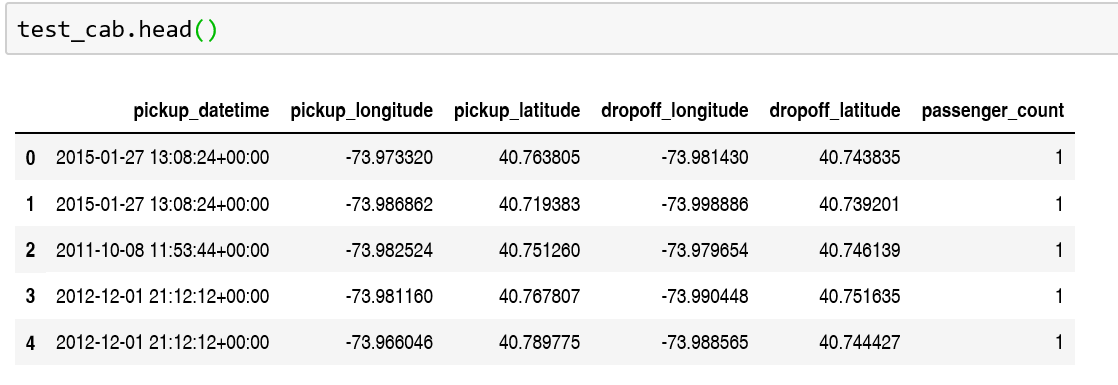
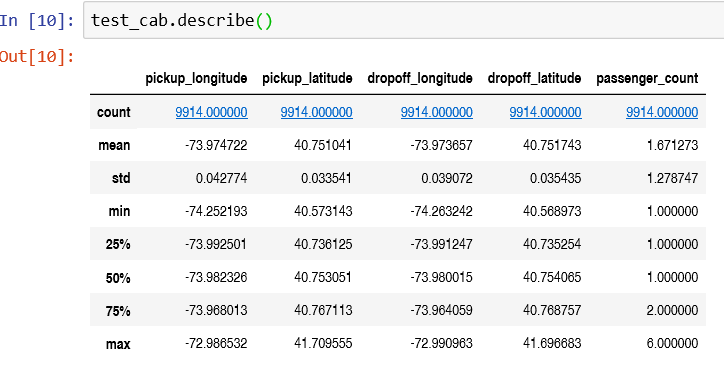
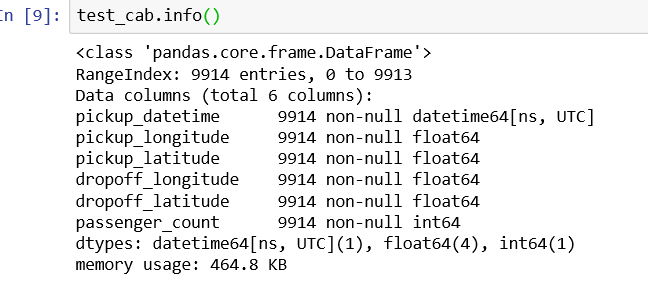
Data type of fare\_amount should be converted to numeric type

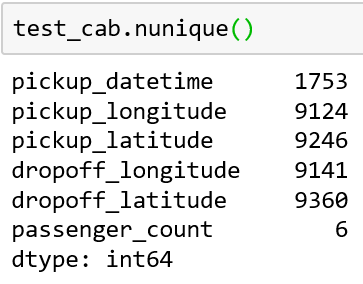
Data type of pickup\_datetime should be converted to datetime type

No. of unique data present in each variable is also presented in the image

Missing values can also be seen in fare\_amount, pickup\_datetime, passenger\_count variables which we will see in next section

**Analysing test.csv**

9914 observations with 6 variables no dependent variable****



From the above images from python

No. of variables =6

No. of independent variables are 6

No. of dependent variables is 0

Data type of the variables are shown in image

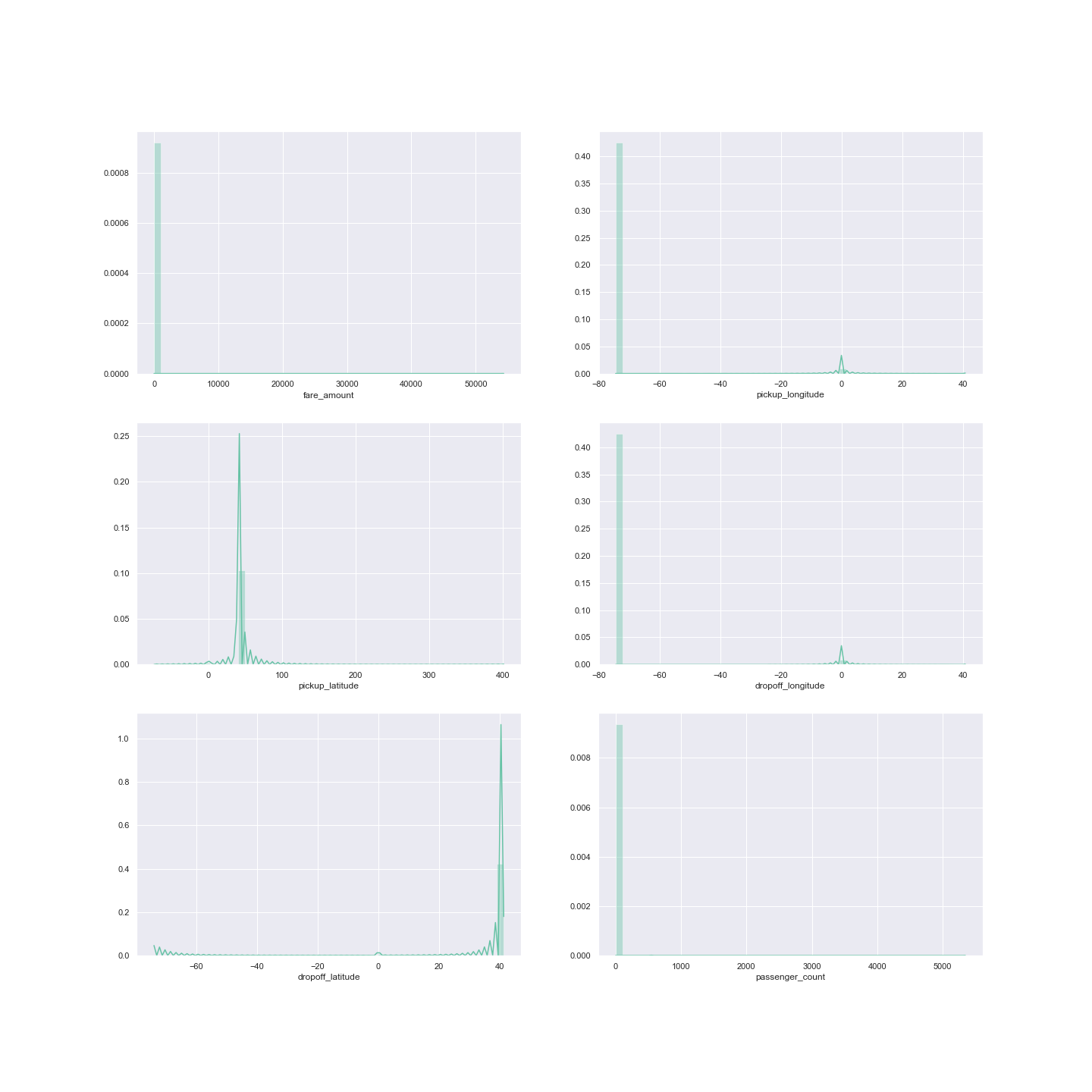
Data type of pickup\_datetime should be converted to datetime type

No. of unique data present in each variable is also presented in the image

There are no missing values in this test.csv

Lets do some graphical analysis

Using histogram

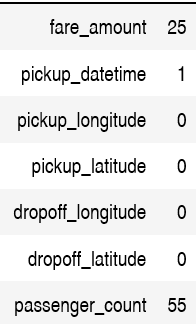


* 1. **Missing value analysis**

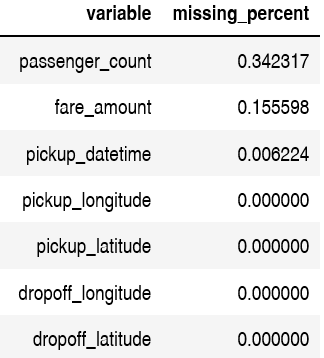
there are no missing values in test.csv

so missing value analysis will be done only on train\_cab.csv

the no. of missing values in each column are as follows

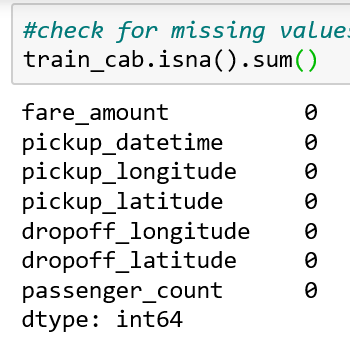
****

the percentage of missing value in each column, columns arranged in descending order are as follows

****

For removing the missing value, we will perform following steps

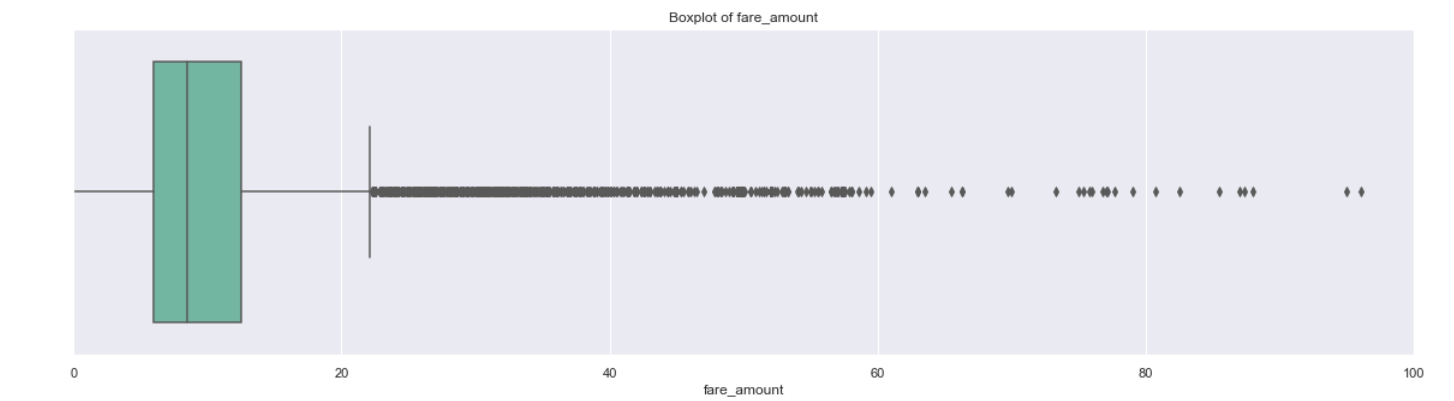
* take a random sample from each column and replace it with na value
* then impute the column with each method i.e. mean, median, and KNN imputation method ,
* then we will compare the actual value with the imputed value
* whichever value lies closet to the actual value we will freeze that method for imputation of that column
* we used knn imputation for the fare\_amout column as it was closest to the actual value
* for pickup\_datetime we used forward fill na method because there was only one na values
* for next four variable of longitude and latitude we do not have any missing value
* for passenger\_count we used median imputation as its value was closest to the actual value
* missing values after analysis are as shown in image



* 1. **Outliers analysis**

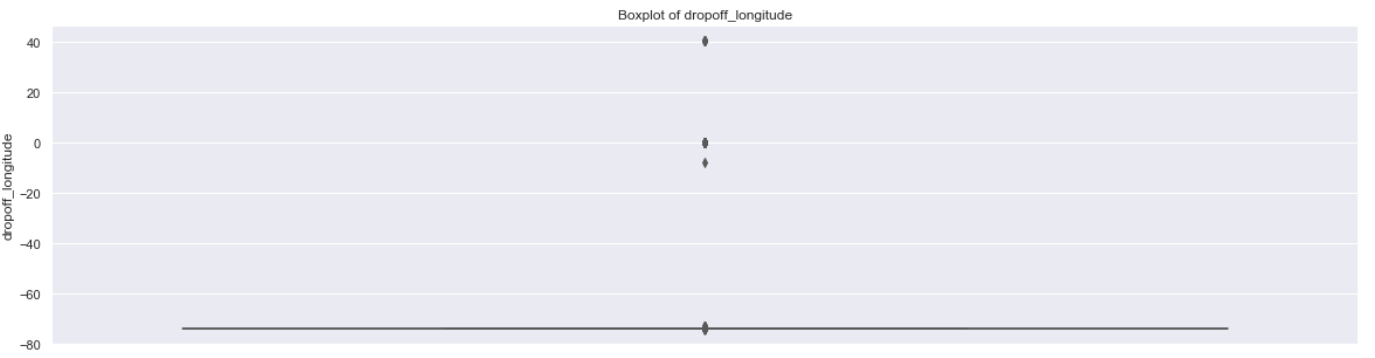
we have to outlier analysis for both the data set simultaneously

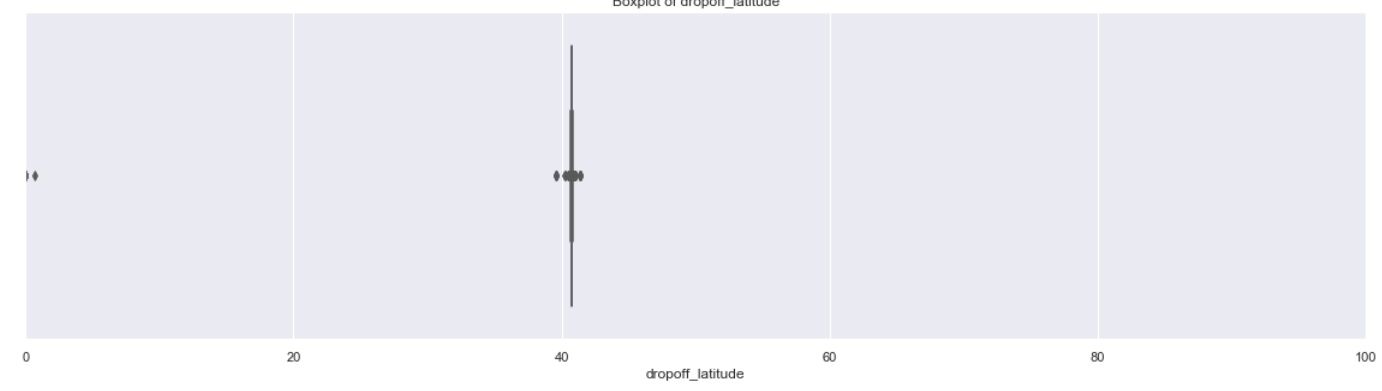
using boxplot lets first see the outliers in each column of train\_cab.csv

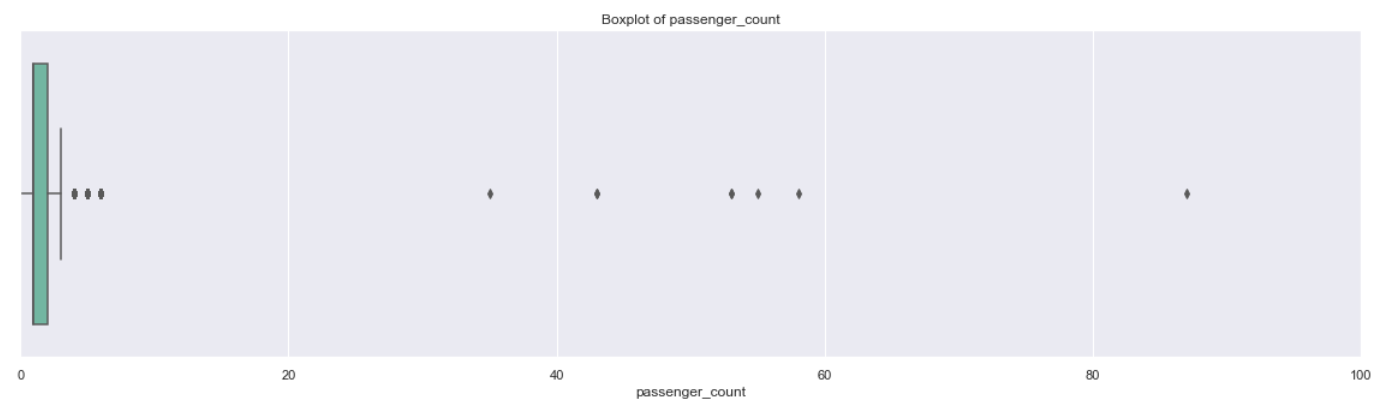


****

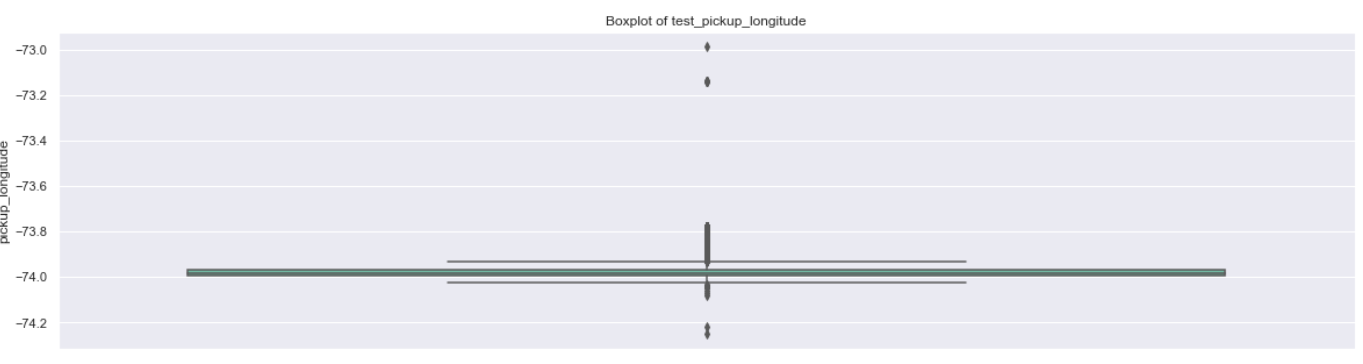
****

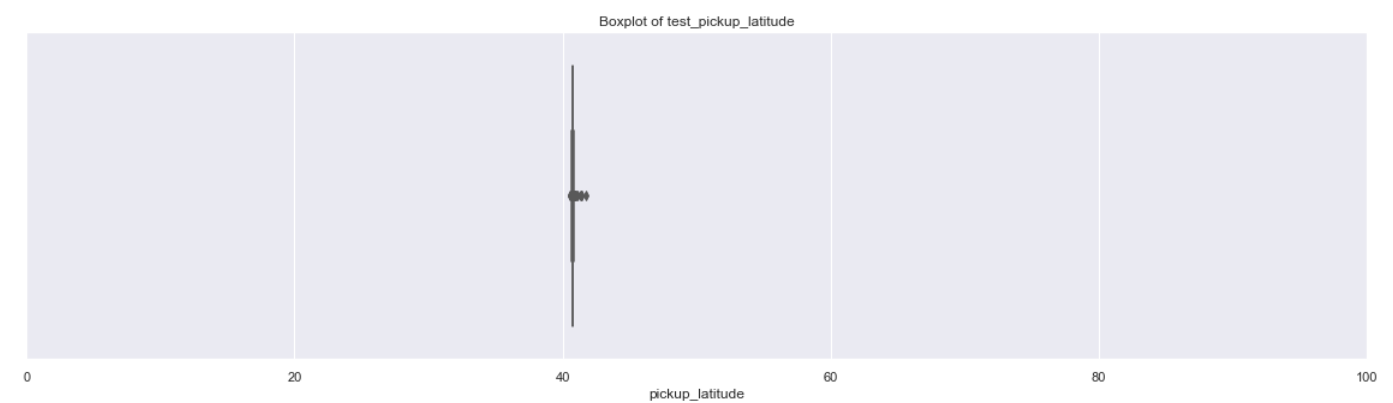
****

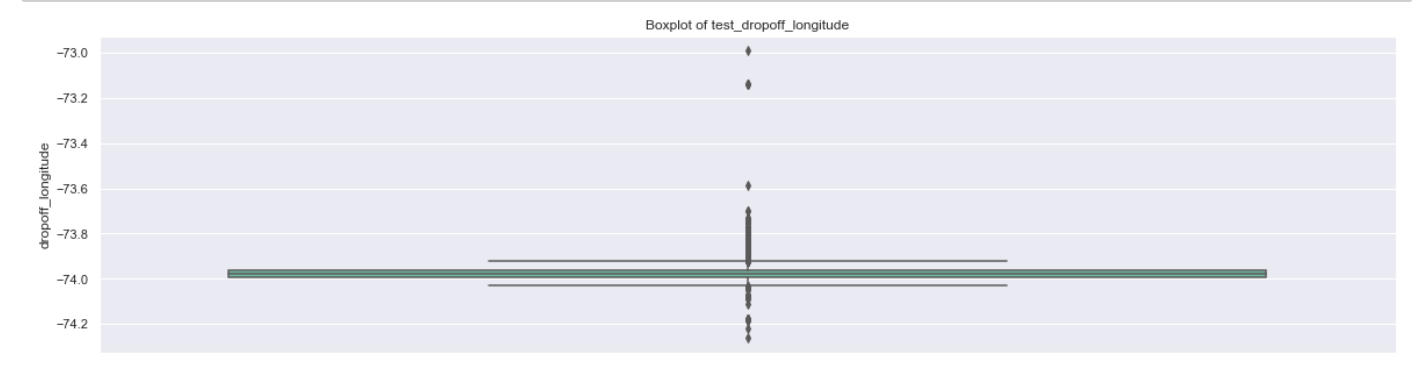
****

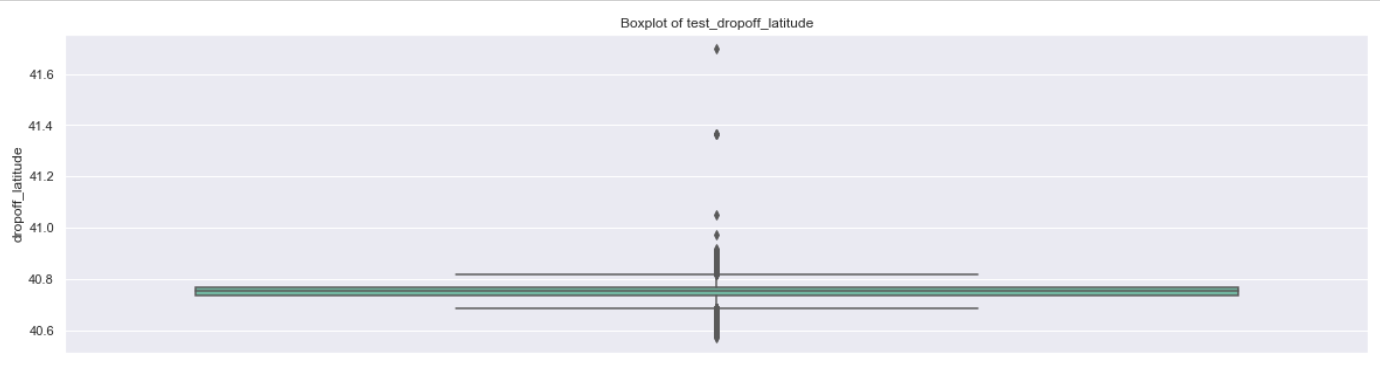


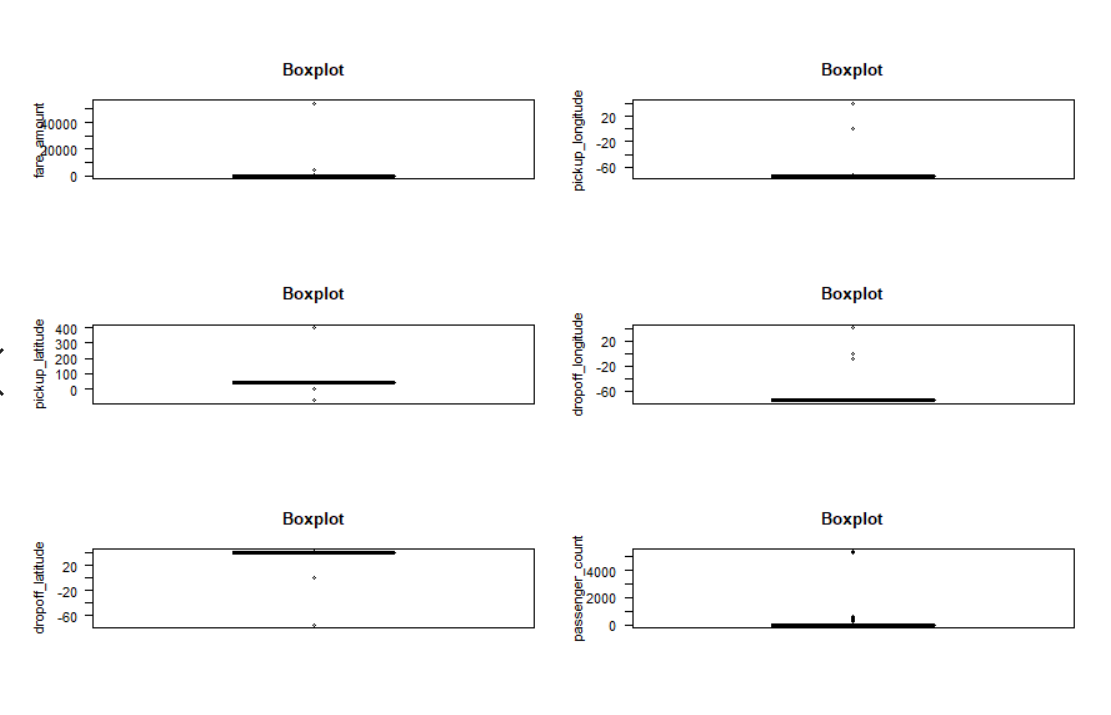
Boxplot for test.csv data columns

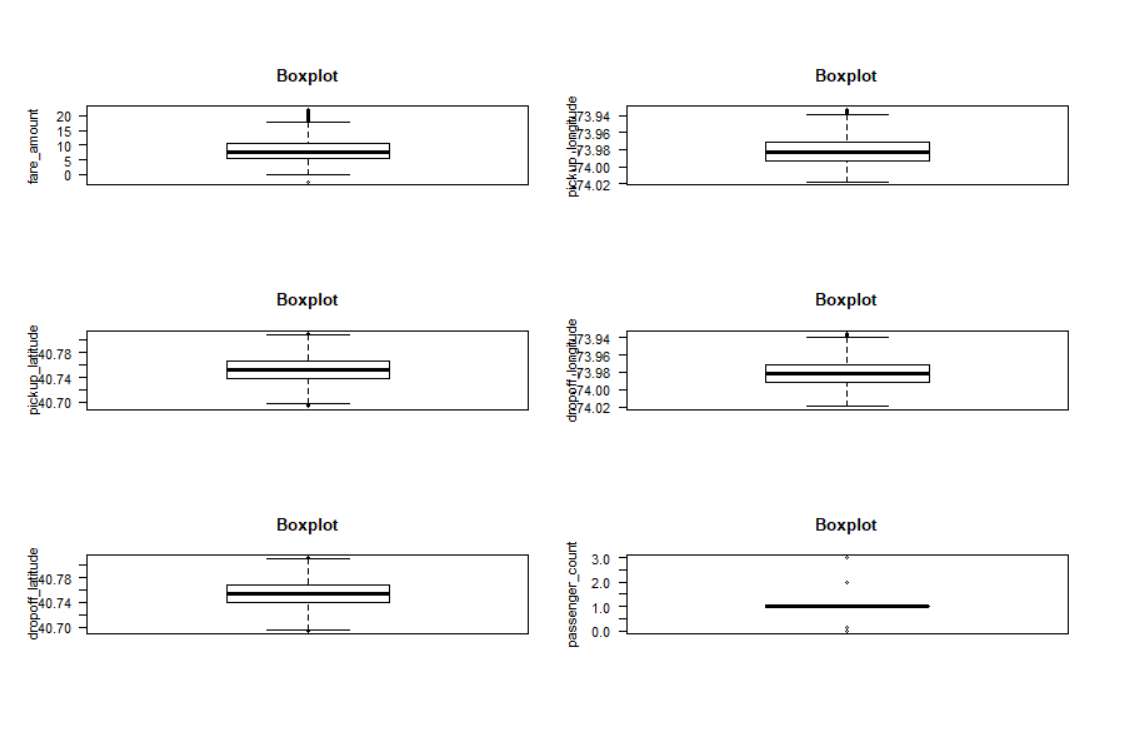












Steps followed in outlier analysis are as follows:

* for each variable find q75,q25 which are upper quartile and lower quartile respectively
* then calculate iqr which is interquartile range (iqr=q75-q25)
* then calculate upper fence and lower fence and name them max and min

(max=q75+(iqr\*1.5))

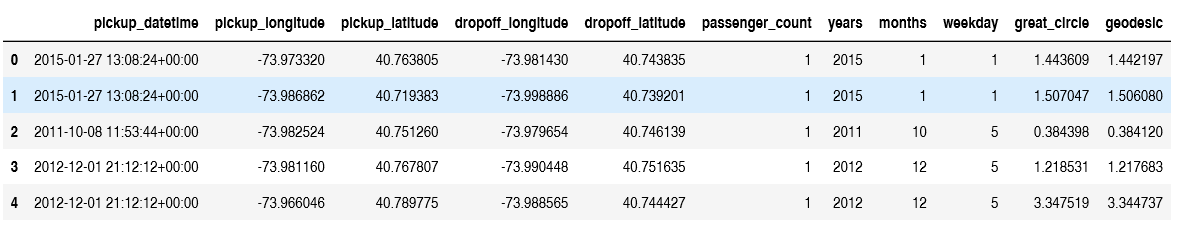
(min=q25-(iqr\*1.5))

* then we have to either remove or replace the values by na which are beyond this min and max fence
* I have replaced them with np.nan because if I removed it I felt we were loosing more than 30 percent of data in all
* And after replacing again performed the missing value analysis using KNN imputation method for all variables

**\*\*\*\*\*\*\*Now our data sets are free from missing values and outliers**

* 1. **(Feature engineering) Creating some new variables from the given variables.**

Here in our data set our variable name pickup\_datetime contains date and time for pickup. So we tried to extract some important variables from pickup\_datetime: years, months and week days



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’. As Vincenty is more accurate than haversine. Also, vincenty is prefered for short distances.

Varibles extaracted are geart\_circle and geodesic

* 1. **Dependent & Independent variables:**

Our Indpendent variable is: Passenger\_count, years, months, weekday, geodesic

Our Dependent variable is: fare\_amount

**3.5.1 Uniqueness in Variable: -**

We need to look at the unique number in the variables which help us to decide whether the variable is categorical or numeric. So, by using python script ‘nunique’ we tried to find out the unique values in each variable. We have also added the table below:

|  |  |
| --- | --- |
| **Variable Name** | **Unique Counts** |
| fare\_amount | 450 |
| passenger\_count | 7 |
| year | 7 |
| Month | 12 |
| Date | 31 |
| Day of Week | 7 |
| geodesic | 15424 |

**3.5.2 Dividing the variables into two categories basis their data types:**

Continuous variables - 'fare\_amount', 'distance'.

Categorical Variables - 'year', 'Month', 'Day of Week', 'passenger\_count'

**Chapter 4**

# Modelling

After a thorough pre-processing, we will use some regression models on our processed data to predict the target variable. Following are the models which we have built –

Linear Regression

Decision Tree

Random Forest

**Before running any model, we will split our data into two parts which is train and test data. Here in our case we have taken 80% of the data as our train data. Below is the snipped image of the split of train test.**

**4.1 Linear Regression**

[Multiple linear regression i](http://www.statisticssolutions.com/academic-solutions/membership-resources/member-profile/data-analysis-plan-templates/data-analysis-plan-multiple-linear-regression/)s the most common form of linear regression analysis. Multiple regression is an extension of simple linear regression. It is used as a predictive analysis, when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable).

Assumption of linear regression mode:

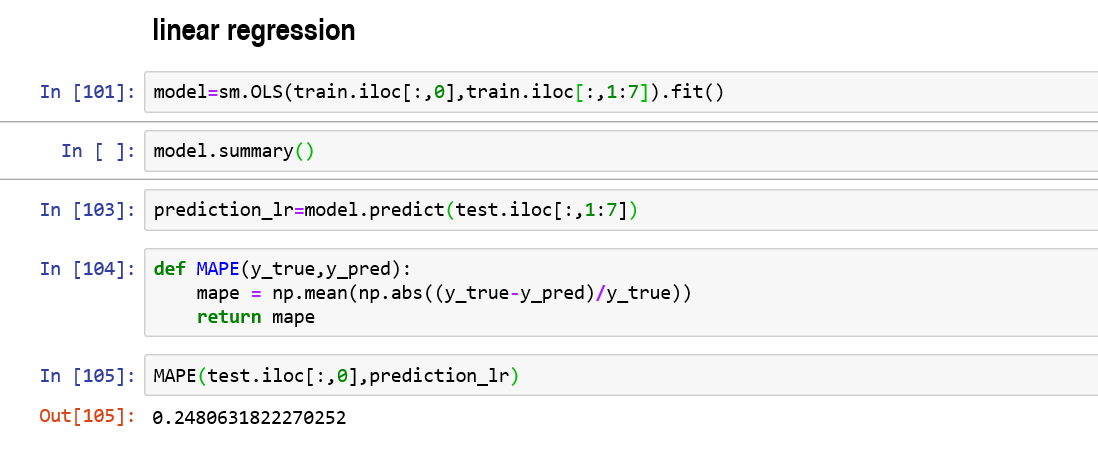
* There must be a linear relationship between the outcome variable and the independent variables.
* Scatterplots can show whether there is a linear or curvilinear relationship, correlation matrix also shows it.
* Multivariate Normality–Multiple regression assumes that the residuals are normally distributed.
* No Multicollinearity—Multiple regression assumes that the independent variables are not highly correlated with each other.  This assumption is tested using Variance Inflation Factor (VIF) values.
* Homoscedasticity–This assumption states that the variance of error terms are similar across the values of the independent variables.  A plot of standardized residuals versus predicted values can show whether points are equally distributed across all values of the independent variables.

Shortcomings of the linear regression model:

* Sensitive to outliers
* Sensitive to missing values

Below is a screenshot of the model we build and its output:

Python code:



R code:



## 4.2 Decision Tree

Decision Tree algorithm belongs to the family of [supervised learning algorithms](https://dataaspirant.com/2014/09/19/supervised-and-unsupervised-learning/). Unlike other supervised learning algorithms, decision tree algorithm can be used for solving [**regression and classification**](https://dataaspirant.com/2014/09/27/classification-and-prediction/)**problems** too.

# Assumptions while creating Decision Tree.

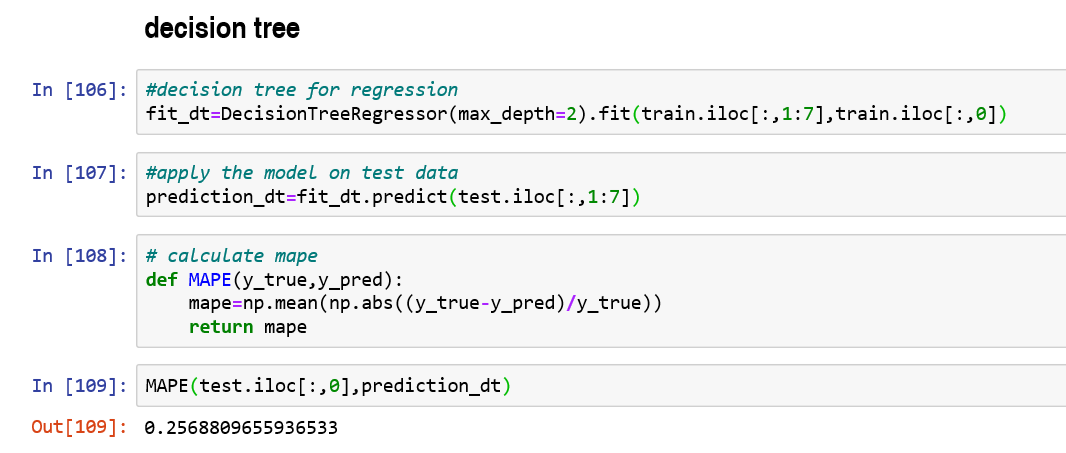
* At the beginning, the whole training set is considered as the **root.**
* Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
* Records are **distributed recursively** on the basis of attribute values.
* Order to placing attributes as root or internal node of the tree is done by using some statistical approach which are below mention.

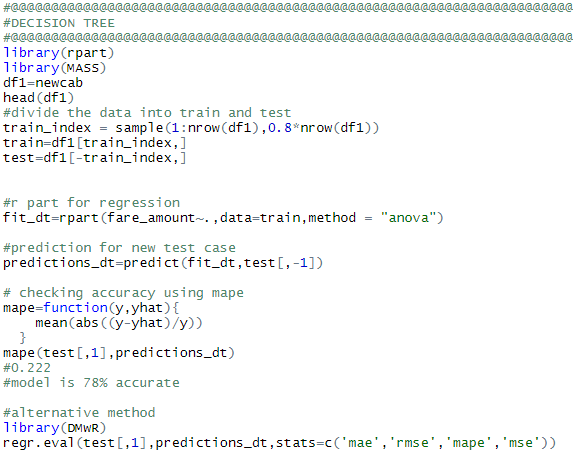
Advantages of Decision Tree.

* Decision Trees are easy to explain. It results in a set of rules.
* It follows the same approach as humans generally follow while making decisions
* Interpretation of a complex Decision Tree model can be simplified by its visualizations. Even a naive person can understand logic.
* The Number of hyper-parameters to be tuned is almost null.

Drawbacks of Decision Tree.

* There is a high probability of [**overfitting**](https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/)in Decision Tree.
* Generally, it gives low prediction accuracy for a dataset as compared to other machine learning algorithms.
* Information gain in a decision tree with categorical variables gives a biased response for attributes with greater no. of categories.
* Calculations can become complex when there are many **class labels**.





## 4.3 Random Forest

Random forests or random decision forests are an [ensemble learning m](https://en.wikipedia.org/wiki/Ensemble_learning)ethod for classification, regression and other task, that operate by constructing a multitude of [decision trees a](https://en.wikipedia.org/wiki/Decision_tree_learning)t training time and outputting the class that is the [mode o](https://en.wikipedia.org/wiki/Mode_(statistics))f the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of [overfitting t](https://en.wikipedia.org/wiki/Overfitting)o their [training set.](https://en.wikipedia.org/wiki/Test_set)

## ADVANTAGES

## One of the most accurate learning algorithms available

## It can handle many predictor variables

## Provides estimates of the importance of different predictor variables

## Maintains accuracy even when a large proportion of the data is missing

## LIMITATIONS

## Can overfit datasets that are particularly noisy

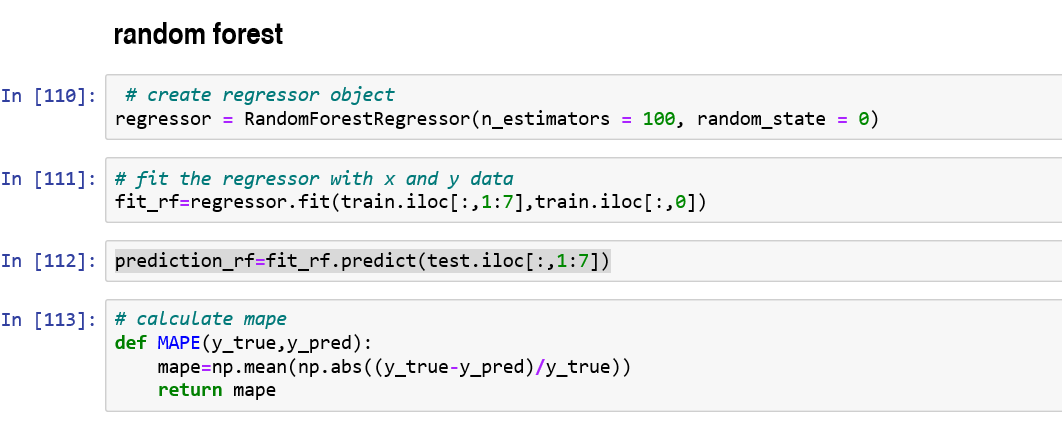
## For data including categorical predictor variables with different number of levels, random forests are biased in favor of those predictors with more levels. Therefore, the variable importance scores from random forest are not always reliable for this type of data

## ASSUMPTIONS

* No formal distributional assumptions, random forests are non-parametric and can thus handle skewed and multi-modal data as well as categorical data that are ordinal or non-ordinal.

**To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.**

Below is a screenshot of the model we build and its output:



In [114]: MAPE (test.iloc[:,0],prediction\_dt)

out [110]: 0.1682



**Chapter 5**

**Conclusion**

**Error matrix :**

## 5.1 Model Evaluation

## For python code

|  |  |  |
| --- | --- | --- |
| **MODEL** | **MAPE** | **ACCURACY** |
| Linear Regression | 0.25 | 75% |
| Decision Tree | 0.26 | 74% |
| Random Forest model | 0.16 | 84% |
|  |  |  |

## 

**For r code**

|  |  |  |
| --- | --- | --- |
| **MODEL** | **MAPE** | **ACCURACY** |
| Linear Regression | 0.20 | 80% |
| Decision Tree | 0.24 | 76% |
| Random Forest model | 0.14 | 86% |
|  |  |  |

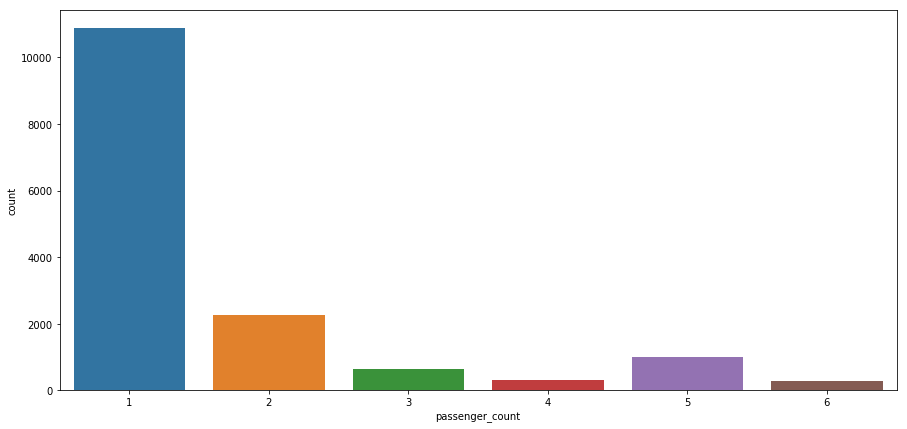
## 5.2 Model Selection

From above section results we get to know that the accuracy of random forest is highest among all so we will use random forest to predict target variable for the test\_cab.csv data shared in the problem statement result found are attached within my submitted program code files

## 5.3 Some more visualization facts:

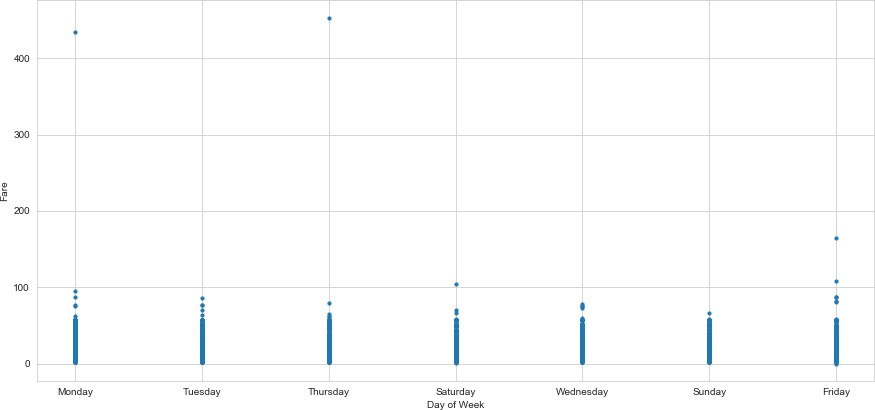
**5.3.1 Number of passengers and fare**

We can see in below graph that single passengers are the most frequent travelers, and the highest fare also seems to come from cabs which carry just 1 passenger.

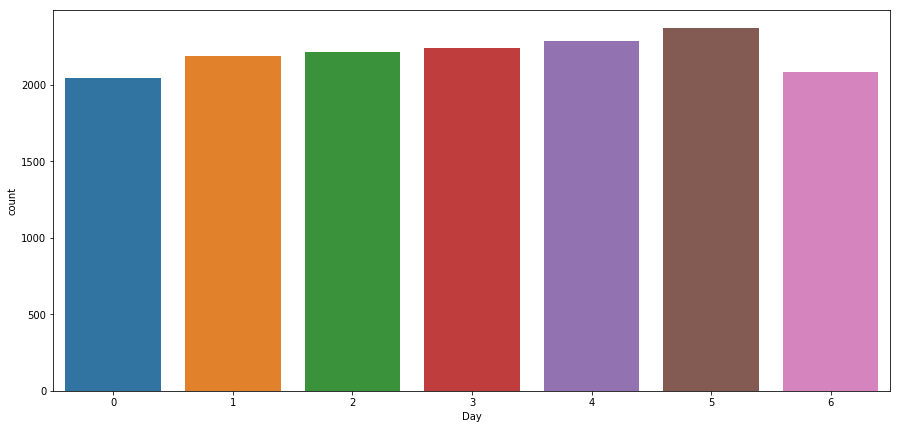


**5.3.2 Week Day and fare**

• Cab fare is high on Friday, Saturday and Monday, may be during weekend and first day of the working day they charge high fares because of high demands of cabs.

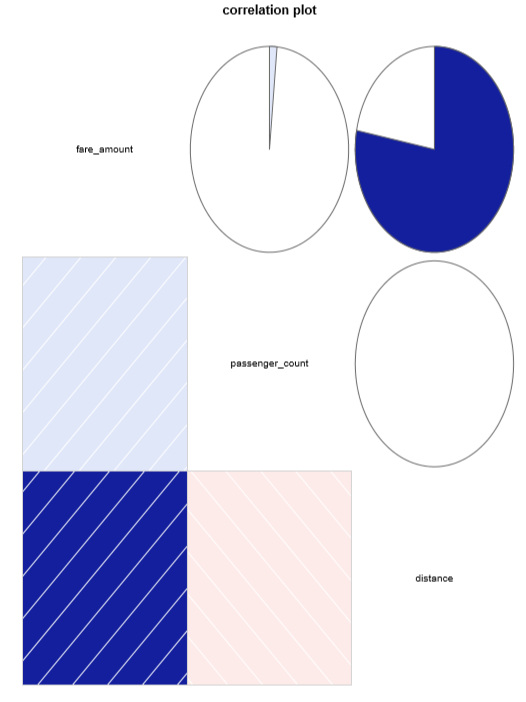


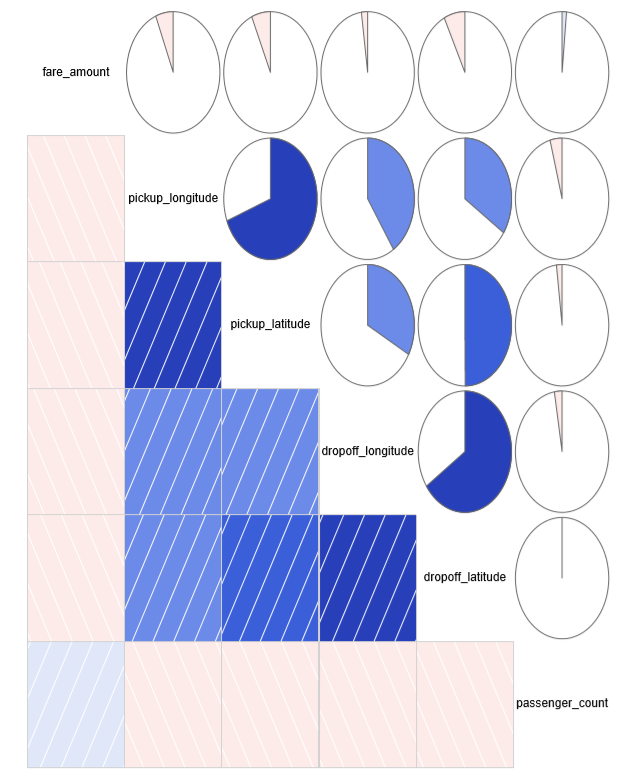
**5.3.3 Impact of Day on the Number of Cab rides :**



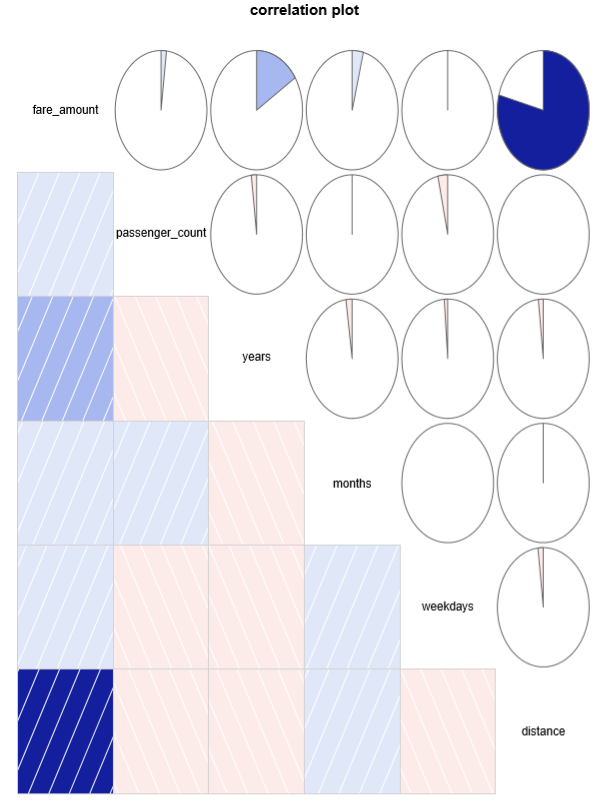
**Observation:** The day of the week does not seem to have much influence on the number of cabs ride

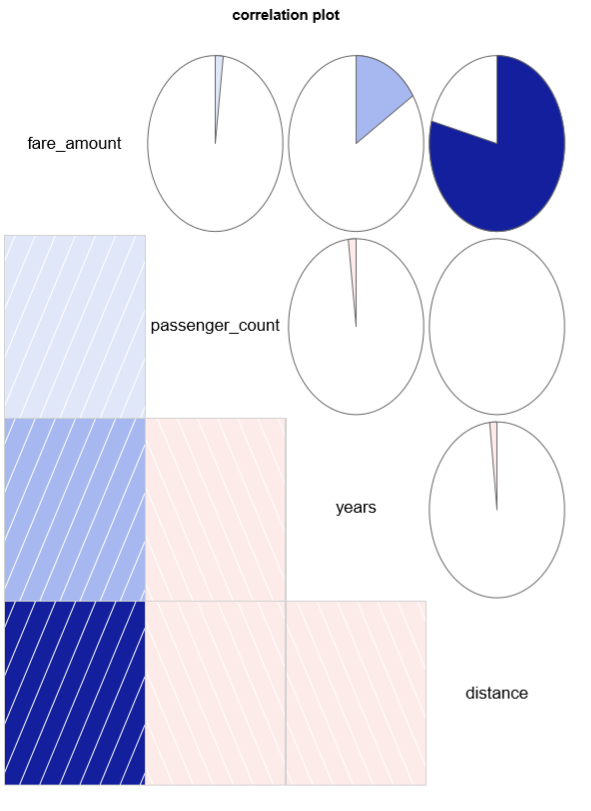
**5.3.4 correlation plot from r console**

****

****

Correlation plots after feature engineering





**Chapter 6**

**Instructions to deploy and run code.**

1. **Python instructions**
2. Open anaconda
3. Open jupyter notebook
4. Go to the file location and open finalcab.py file
5. Now open new python console
6. And copy the code from .py to console and
7. Import the data set in console using your system file location
8. Go to cell and click on run all cell at a time
9. The complete python file will be executed without inturuption
10. And result will be displayed of the test file
11. **R instructions**
12. Open r studio
13. Go to file loction and open mo.r file
14. First run first few lines to import the data set in r console using your system file loction
15. Then run complete file code lines one by one
16. The complete project will be executed without inturuption
17. The result will be displayed in console
18. Visualisations will be displayed in plots section

## References

1. For Data Cleaning and Model Development - <https://edwisor.com/career-data-scientist>
2. For other code related queries - [https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis- python/](https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/)
3. For Visualization – <https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/>
4. <https://towardsdatascience.com/>
5. <https://stackoverflow.com/>