## horizontal line



Cab Fare Prediction

29.10.2019

**─**

Sushant Thakur

# Overview

In this project we are going to predict the cab fare amount by training our model on a given dataset.As we have to predict the cab fare which is continuous it means the problem that we have is regression problem.

# Goals

1. Our goal is analyze and visualize the data which will help in improving the business.
2. We have to use the dataset and find some interesting patterns which allows us to build the model which will help us to predict the fare amount from given data

# DataSet

We have two csv files to work with:

1.train\_cab.csv : For training and validating the model

2.test.csv : For actual prediction

# Project Flow

In this project we will do following tasks

1.Reading both the files train and test data,outlier analysis and missing value analysis

2.Analyzing and visualizing the the dataset using graphs,barplots and some histograms.

3.Finding patterns and building model on that basis.

4.Validating the model.

5.Optimising the model and try to improve model performance.

6.Once model is freeze we will predict the fare amount.

# Reading files and some analysis.

1.In first step we have read the both files using pandas,First we will discuss about train\_cab.csv file.

2.Now we will grab the information about the dataset.

RangeIndex: 16067 entries, 0 to 16066

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

dtypes: float64(5), object(2)

As we can see we have 16067 entries in our dataset and contains 7 variables

3.Out of these 7 variables fare\_amount is to be predicted and hence it is target variable and rest are input variable

4.we will grab the information about the missing values so let us look at the number of missing values in our dataset

fare\_amount 24

pickup\_datetime 0

pickup\_longitude 0

pickup\_latitude 0

dropoff\_longitude 0

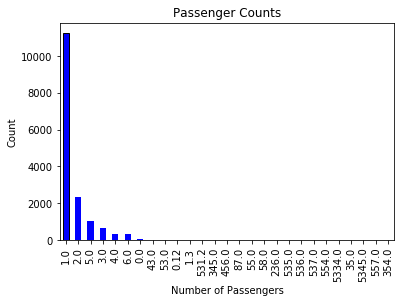
dropoff\_latitude 0

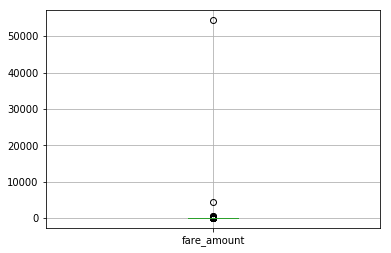
passenger\_count 55

As we can see we have 24 missing values in fare amount and 55 missing values in passenger count.

5. We can either remove the rows which are having missing values or we can impute using model like knn or we can us mean,median,mode to impute the data in place of missing values

6.As in below diagram we can see the most dominating value which we have for the passenger count is 1 so it is better idea to fill the empty value in passenger count as 1 rather than using something else.





7.So now we are left with 24 rows which has fare\_amount as NA.

As fare amount is to be predicted and also it contains many outliers can be seen in above image so its not a good idea to impute the value so we can remove the rows with NA in fare\_amount.

.**Finding useful information.**

We have latitude and longitude for the for both pickoff and drop so after some research,I have found that the distance can be found using this information which is called haversian distance.This can be really useful for predicting the fare\_amount.

Below is the link for finding the haversian distance

<https://stackoverflow.com/questions/27928/calculate-distance-between-two-latitude-longitude-points-haversine-formula>

**Performance metrics.**

This is one of the important things that we will have to freeze to evaluate our model performance.

1.As RMSE gives more importance to higher errors unlike mape,one high error is enough to give bad rmse and also we have removed most of the outliers,but still in reality there may be some free rides or rides with coupons or any rare weather condition so we will like to use rmse as it is very sensitive to outlier.it first squares the difference and then average it unlike mape which takes absolute difference than divide it with actual value and takes the average.

**Building Model.**

We will split our train data in to 80:20.we will train the model with 80 percent data and then validate the model with 20 % data and we care more about test rmse rather than train rmse because our model had not seen test data it is new for the model.

Features that we will use for building model

1.pickoff\_latitude.

2.pickoff\_longitude.

3.dropoff\_longitude

4.dropoff\_latitude.

5.passenger\_count.

We will go on using simpler to complex model like linear regression,decision trees,knn regression and then we will use ensemble model like random forest.

We will interpret the result in tabular form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Linear regression | Decision trees | Knn | Random Forest  (depth 7) |
| rmse\_train | 4.36 | 3.65 | 3.97 | 3.35 |
| rmse\_test | 4.12 | 4.99 | 4.25 | 3.87 |

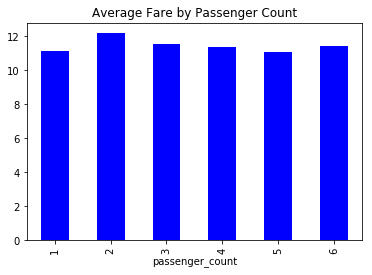
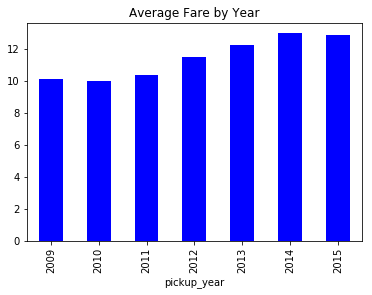
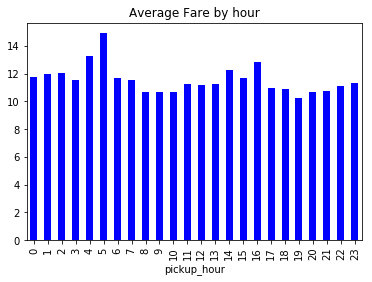
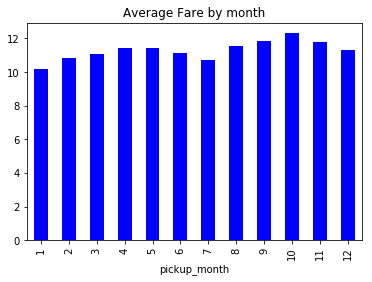
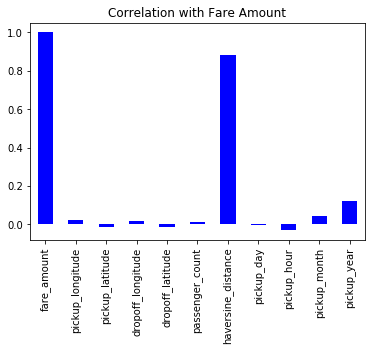
As we can see in above table linear regression has perform well on this model,

For decision tree train\_rmse is low as compare to test rmse so we can say that may be our decision tree model has overfit,knn has also perform well on this dataset but performance of random forest is dominating.

**Optimising Model.**

To improve the model we can try one thing such as using more number of features.

We also have pickoff datetime we can split this in year,month ,date and dat and try to find out the corelation with fare amount



As from the above visualization we can see that year,month,hour have some correlation with fare\_amount so trying those model and describing the data in tabular form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | linear\_regression | Decision trees | knn | Random forest |
| rmse\_train | 4.30 | 5.70 | 3.76 | 3.17 |
| rmse\_test | 3.81 | 4.84 | 3.74 | 3.40 |

As from the table we can see that adding more features has improved some model performance for linear regression,knn and random forest.But once again we can observed that the Random forest model has outperform the other models.

**Deployment.**

I have used random forest model to predict the fare\_amount of test.csv and write the predicted file in disc.

For deployment we can use pickle or database to save our model coefficients and we don’t have to run the model again and again we can just load the model from database or disc and throw value on that by doing some minor changes and Our Model is ready to perform.

**Instructions to run the file.**

**1.Python**

1.Make sure all the files are in same directory like py or ipynb,train\_cab.csv,test.csv else you will have to change the directory in code .

2.The file can be best viewed in jupyter notebook.I have made the project step by step including some great visualisations,missing value analysis, outlier analysis and model building.

3.If you have anaconda install you can just download the py from jupyter notebook and can run from the terminal only but it will be not as interactive as jupyter notebook.

4.use python3 <file\_name> to run the project.

**2.R**

1.First you will have to install R and R studio.The file can be best viewed in R studio,you can just uncomment all the install packages so that all the packages which are used in project is installed and select all the statement and run the project will run successfully.

2.Make sure to change the directory as per file(where train and test file is located else you will get error).

3.The code can be more effectively run on the r studio because it has some visualisations,graphs and histograms.

4.For running Rscript from terminal you can type Rscript <file\_name>