Python: Cab Fare Prediction

Pre-requisites:

1. To run the python project the system shall have:
2. 64 bit python
3. IDLE-64 bit to run python application
4. Download the file and save it in your local in Documents folder. The file format should be in below format:

C:/Users/USER\_NAME/Documents/train.csv (in my case : C:/Users/Ananya Verma/Documents/train.csv)

AND

C:/Users/USER\_NAME/Documents/test.csv (in my case : C:/Users/Ananya Verma/Documents/test.csv)

Steps:

Run the application by following the below steps using Command Prompt:

1. Open command prompt by typing “cmd” in start.
2. Type “cd path-where –python-is-installed” Note that cd and path has a space between them.
3. Next, type the file name in that path(in .py format) and press enter.

Python 3.7.2 Shell will appear

The data types in train dataset are

fare\_amount object

pickup\_datetime object

pickup\_longitude float64

pickup\_latitude float64

dropoff\_longitude float64

dropoff\_latitude float64

passenger\_count float64

Next we check the min and max values in train dataset

* We see that the fare\_amount variable in train dataset has negative values which is not practicle, hence we delete the negative values from the dataset.
* Also after conversion of the fare\_amount variable to float(as it was an object) it tells us that we have a value 430- in the fare\_amount dataset. This looks to be noise data as the fare\_amount can not be 430- hence we remove that row too.

430- lies in index Int64Index([1123] hence we will drop this row.

* After dropping this row, we will make sure that the fare\_amount variable is float. Hence we will convert it to float as it makes sense to have them in float type.
* We see that the fare\_amount variable has negative values, hence we will remove all the negative values since the amount of fare can never be negative. After removing the negative values, the new count of rows is 16039.

Now we will check the missing values in the train dataset using isnull() function.

After using python code we find that only “passenger\_count” in train dataset variable had missing values, we have imputed them with the median of the passenger count variable.

The test dataset on the other hand had 0 missing values. So we’re good with the test dataset.

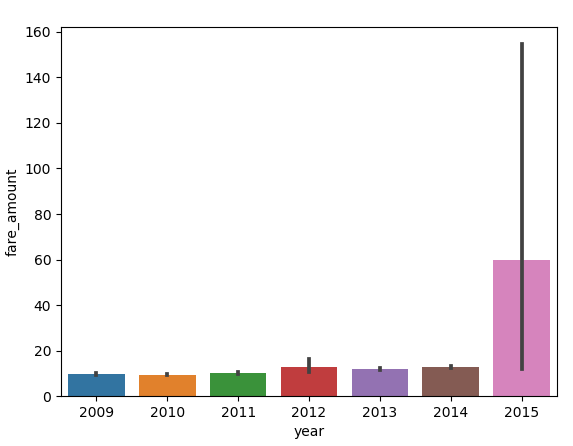
Using pd.DataFrame(train.isnull().sum()) we see that now our dataset has 0 missing values.

Next, we are dividing the datetime variable into year, month, weekday, day, hour, using python’s aligndatetime function and removing the old datetime variable from test and train dataset as it is of no use.

Now we check if the dataset has NA values since NA values are not beneficial for us, hence after using isnull().any() function we check if NA values are really in dataset or not. Later finding the NA values in train dataset we replace them with the mean of the variable.

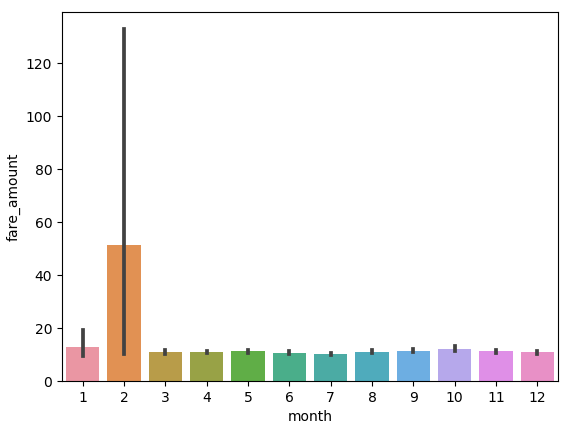
Next, We shall take the absolute values of the latitude and longitude variables in test and train dataset because we shall consider Manhattan distance, which calculates the vertical and Horizontal distances between two points.

Now we shall plot amount vs various parameters:

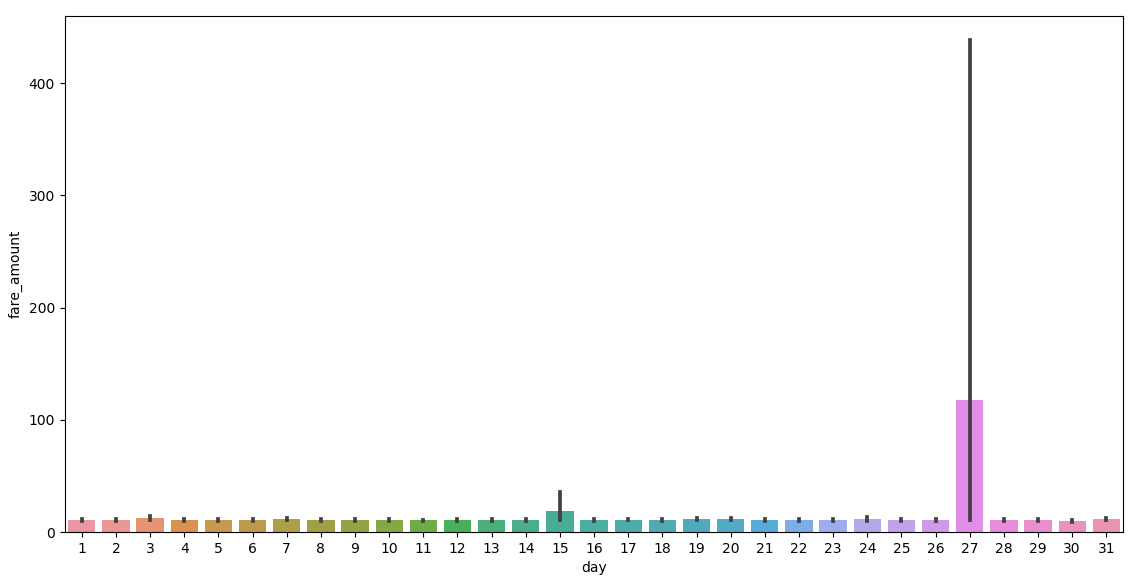


When we see the bar plot for fare\_amount vs year, we come to the conclusion that the fare amount was the highest in year 2015.

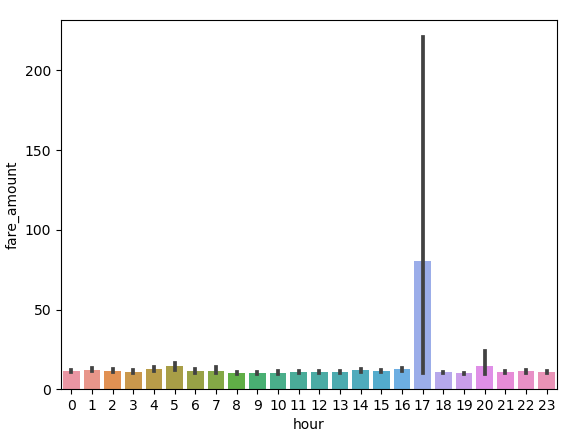
Now let’s check the other plots,



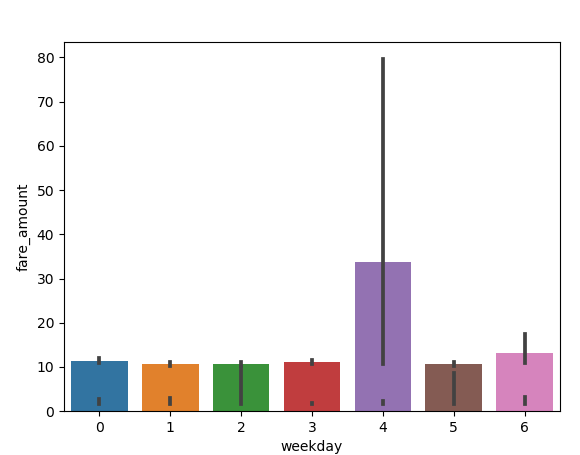
We observe that in the month of February, the fare amount came out to be the highest.



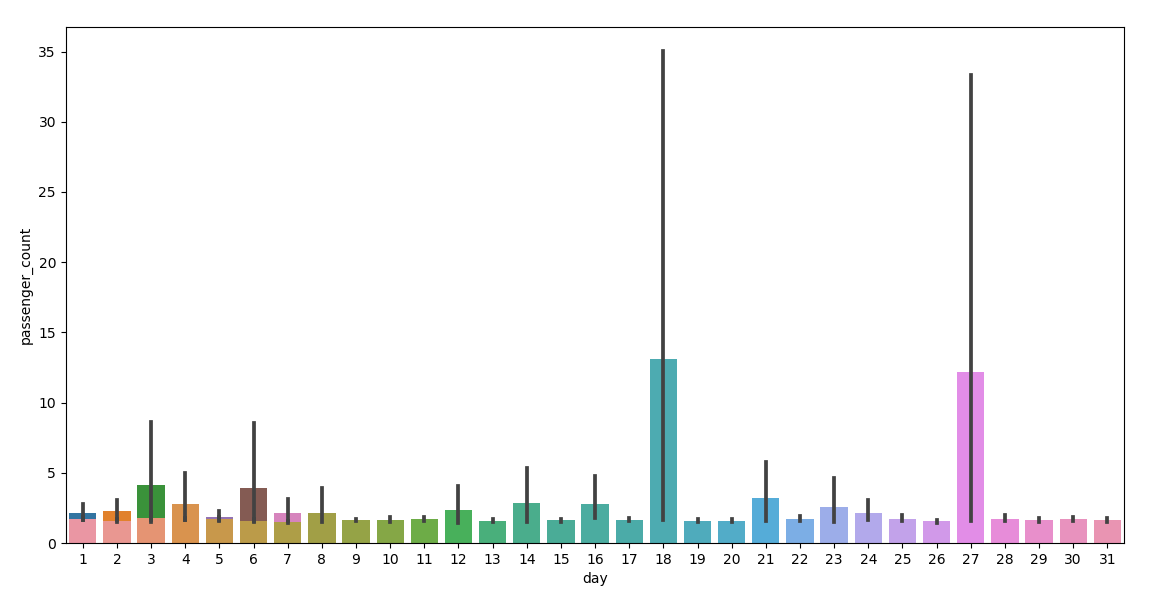
The fare amount turned out to be the highest on 27th of month.



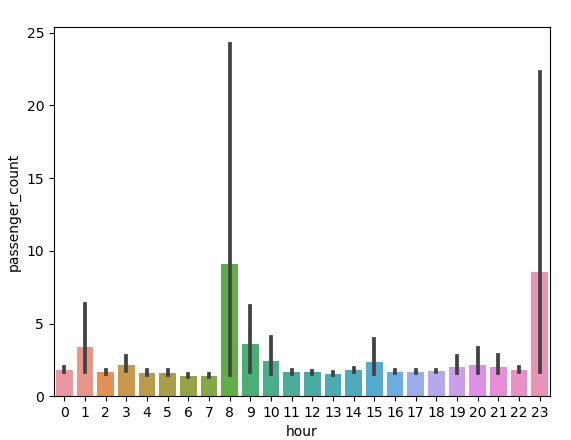
Amount was the highest at (17:00-12:00 hours) 5 in the evening, hence concluding that 5:00 PM had the maximum number of cab demands. It might be the case that the office going public used the cabs more at 5:00 PM.



Also, assuming 4th as Friday (0 being Monday), people took cabs on Friday the most. Assuming Friday as the day before weekends, mostly the day for public to hangout. Hence cabs had the most demand on Fridays.



Passenger count was the most on 18th of the day.

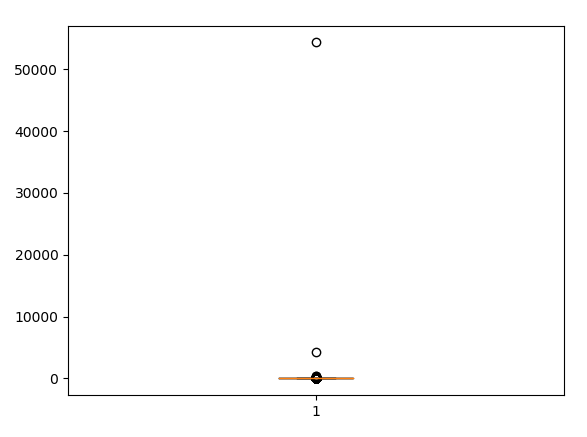


Passengers count was mostly found the highest at 8:00 AM in the morning keeping in mind it is the time when offices start.

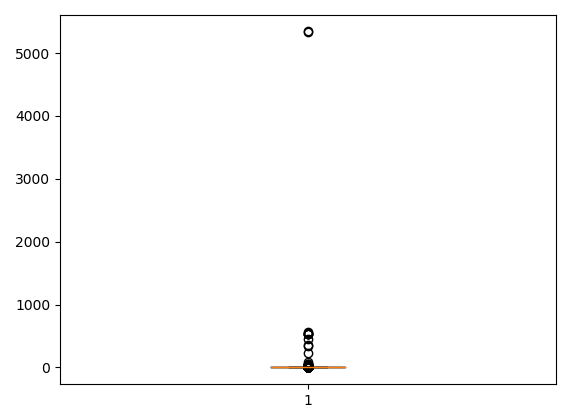
#### Finding Outliers

Outliers in some of the variables

In fare\_amount



In passenger count

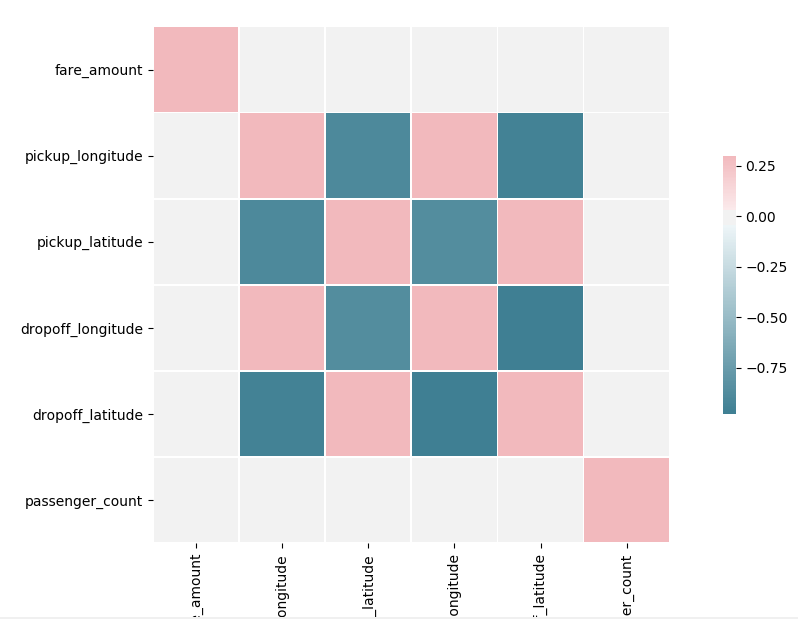


We have created function “outliers\_analysis” that drops all the outliers from test and train datasets.

We then split the dataset to dependent and independent variables, this will ease the process of prediction in Linear Regression model.

We use linear regression for the following below reasons:

* The prediction to be made is a regression model.
* Extremely simple method. Easy to use.



We shall check correlation between values,  
dropoff\_latitude is highly positively(Blue color) correlated with pickup\_longitude and dropoff\_latitude and negatively correlated with pickup\_latitude(red in color)

We now fine the error metrics to know the accuracy of this model.

While calculating MAPE we encountered a problem where MAPE gave infinite results. Hence we will use alternate method for calculating error metrics

We used RMSE and MAE which we will compare later for both the models used.

We now use decision tree model.

Decision tree is used for the below reasons:

* Decision trees require little effort from users for data preparation
* It’s okay even if the independent and dependent variables are non-linear in relationship.

All the predicted values are stored in predictions\_DT object.

|  |  |  |
| --- | --- | --- |
|  | LINEAR REGRESSION | DECISION TREE |
| MAE | 3.2667889809326742 | 5.375968 |
| RMSE | 4.1014448355035364 | 5.4297 |

Linear regression is better suited model for predictions in the case of Python since it gave lesser error,  
hence we will fix the predictions calculated by Linear regression model.