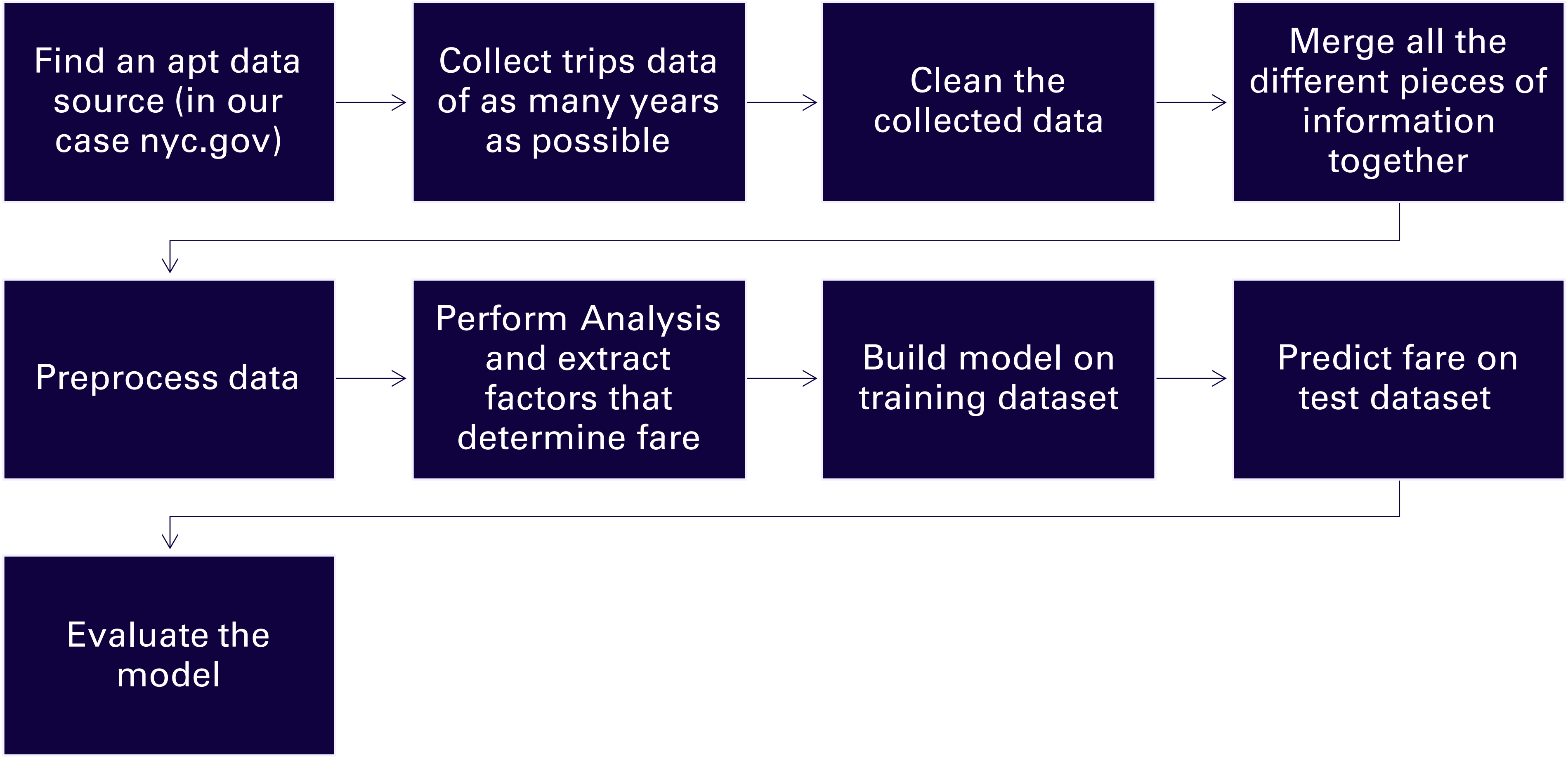
BIG DATA ANALYTICS

Taxi Fare Prediction

- Santhosh

Formulating the problem



Find an apt data source (in our case nyc.gov)

Collect trips data of as many years as possible

Clean the collected data

Merge all the different pieces of information together

Preprocess data

Perform Analysis and extract factors that determine fare

Build model on training dataset

Predict fare on test dataset

Evaluate the model

Data Sources and Mining Methods

- Data Source
  - Nyc yellow taxi data(~67 million data points collected)
  - Dataset link

- Mining methods
  - Data Miner Pro

Data Mining, Preprocessing and cleaning

- The data was available in the form of csv files spread across different webpages.
- Used DataMiner to collect all the links to the csv files mentioned above
- Since each month's data was in a separate csv file, the data from each of those csv files were loaded into different dataframes and merged
- Since there were a lot of columns and we needed only a few of them to predict fare, those redundant columns were removed
- The collected data was then cleaned to remove negative trip distances and negative fare amounts, negative passenger count
- The rows with zero passenger count were also removed because it could have been a logistic transportation ride and including that while training the model will lead to incorrect prediction

Data Mining, Preprocessing and cleaning

- The columns that we finally chose to build model on are Passenger Count, Pick up Location, Drop off Location, Trip Distance and Drop off time
- The rows that had Null values in any of the columns that we plan to use for our model were dropped
- Due to the pandemic in the current year, the trends in 2019 taxi data differs significantly with the 2020 data. So, 2 separate models were trained and tested on the datasets of respective years.
- Though both the models performed almost similarly, the facts and assumptions in this report is based primarily on the model trained with 2020 data. The final few slides shows the performance of the model built and tested on top of 2019 dataset

Data Preprocessing

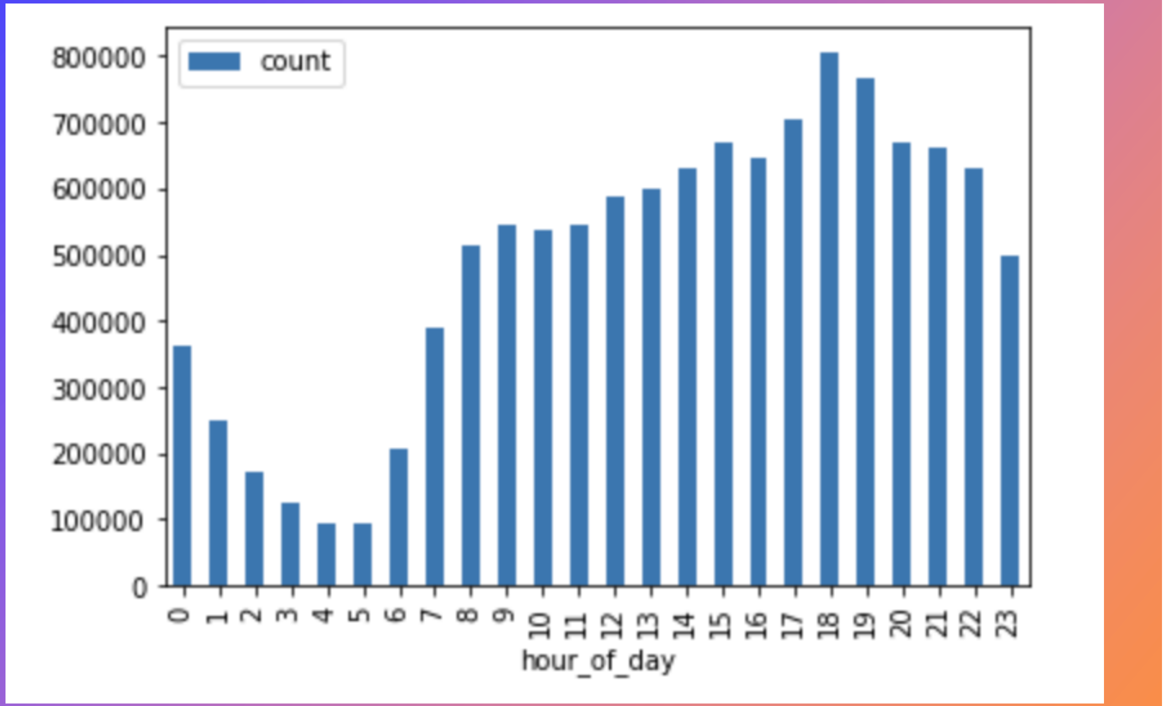
The column Pickup Time was specified as timestamp in the data

Since not much can be gleaned from a simple timestamp and after some primilinary analysis, the hour of day and the day of week on which the trip happened had an impact on the fare amount

The day of week and the hour of the trip were extracted out to two new column "day of week" and "hour of day"

Added a new column "Active day of week" and Out of the 7 days in a week, the most active 4 days (during which most trips took place) were figured out and all the trips that happened during these days were set to 1 and the remaining to 0

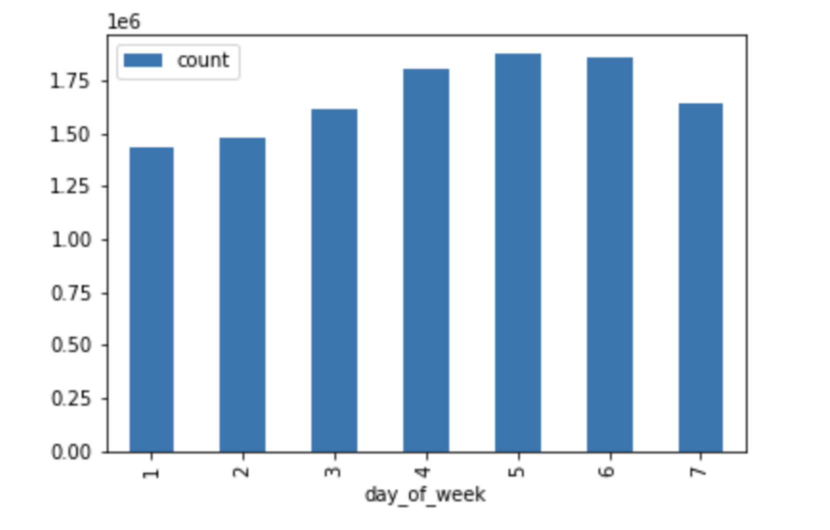
Added a new column "Active hour of day" and Out of the 24 hours in a day, the most active 12 hours (during which most trips took place) were figured out and all the trips that happened during these hours were set to 1 and the remaining to 0



Hour of day vs fare

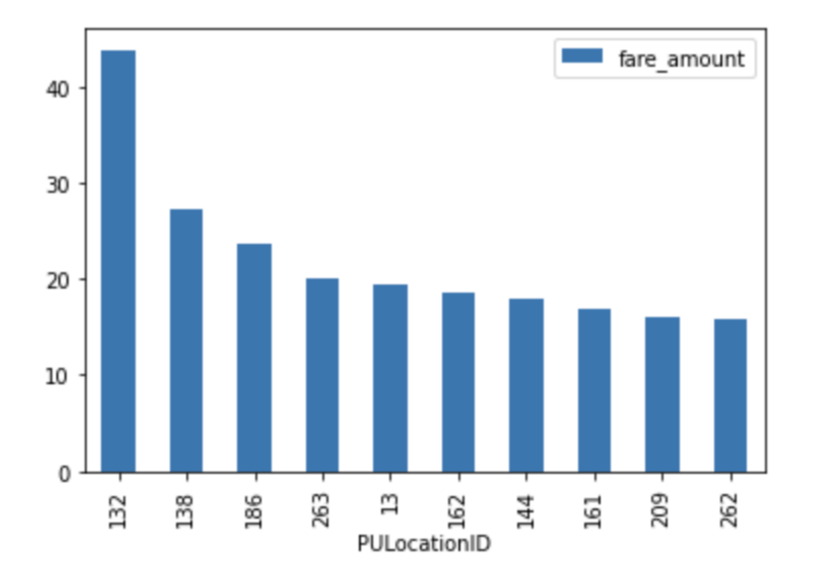
- As we can see from the graph, the number of trips that happened during the peak hours of the day clearly outnumber the trips that happen during the dull hours of the day
- Since in our case we are going to be considering only the active 12 hours of the day, the data points that have the following hour value will be set to 1 and the rest to 0
- Active hours: 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
- Busy hours correlate to higher demand and which correlates to higher fare

Day of week vs fare



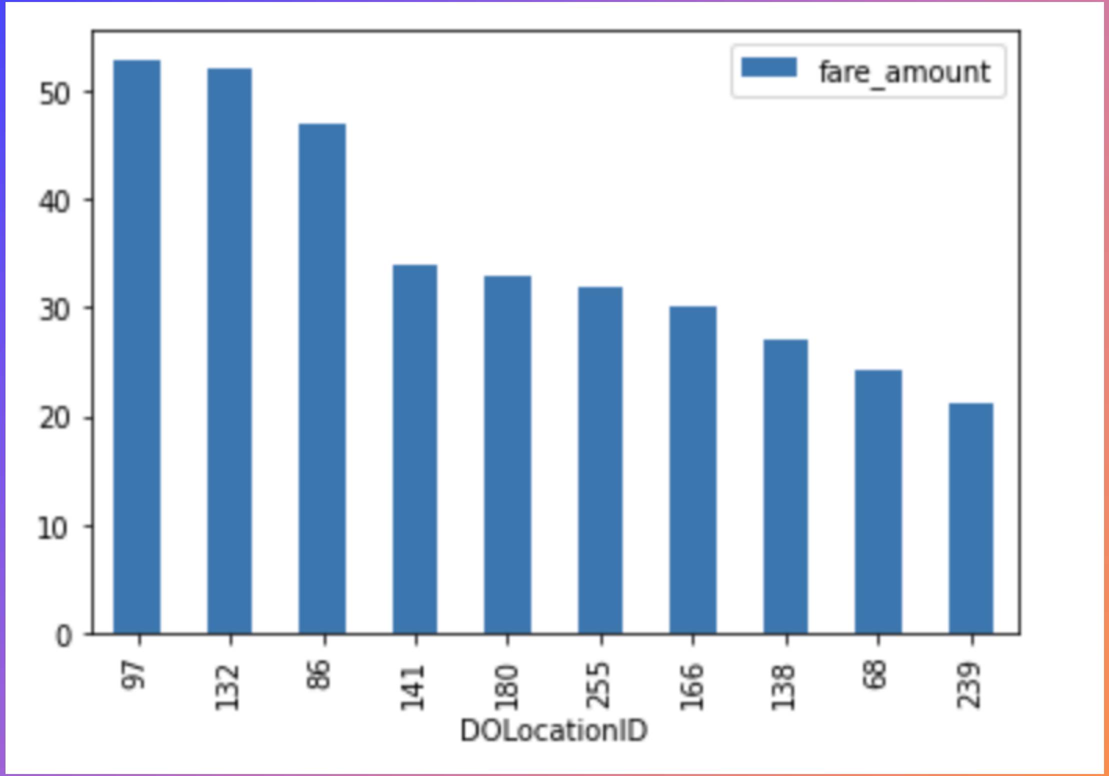
- As we can see from the graph, the number of trips that happened during Thursday, Friday, Saturday and Sunday clearly outnumber the trips that happened during the other days.
- Since in our case we are going to be considering only the active 4 days of the week, the data points that have the following day value will be set to 1 and the rest to 0
- Active days 4,5,6,7
- Busy days correlate to higher demand and which correlates to higher fare

Pick up location vs fare

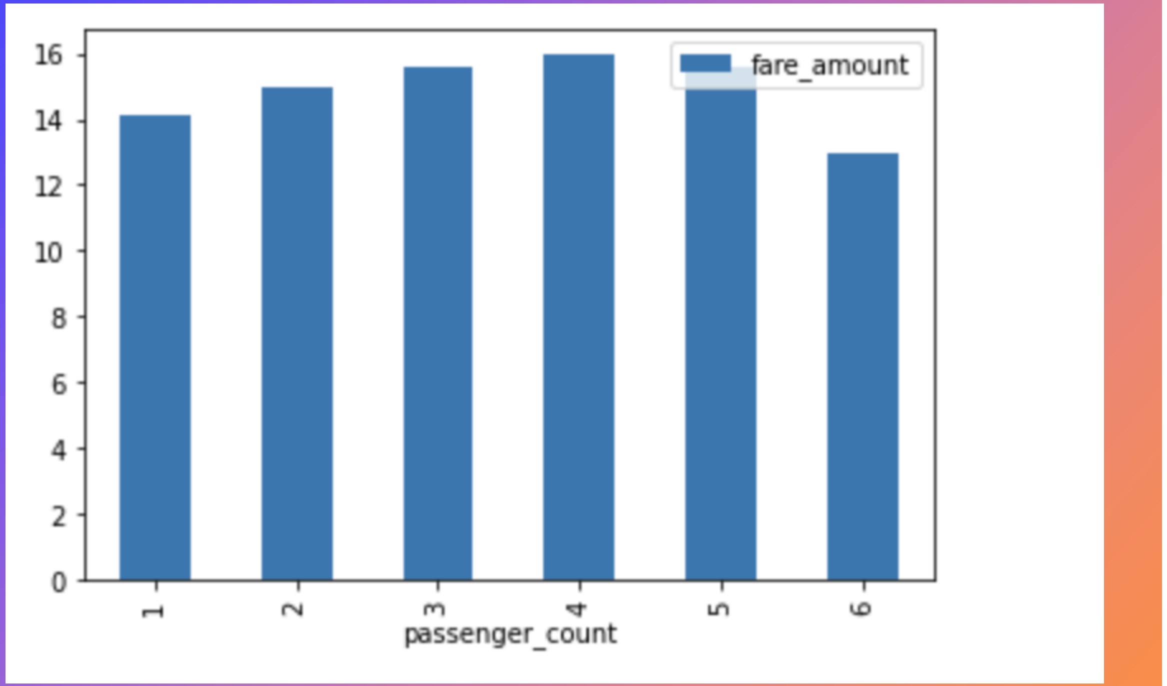


- The graph shows the fare amount for the trips originating from top 10 locations that accounted for a higher fare
- From the graph, we see that the trips origination at location ids 132, 138, 186, .. account for a higher fare than other locations
- These could be hot zones or busy place of the city like airport, Times square, etc. where the demand for taxi is always high
- Since the demand is always high, the fare charged by the taxi companies for rides originating from those locations is also high

Drop off location vs fare



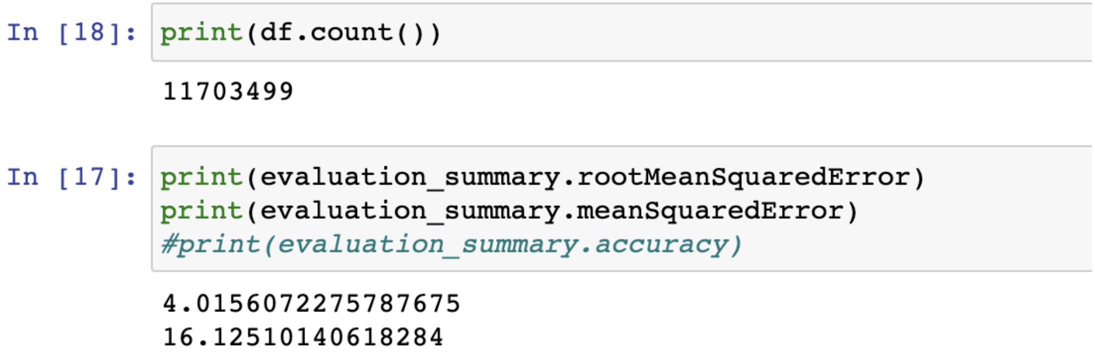
- The graph shows the fare amount for the trips ending at top 10 Drop off locations that accounted for a higher fare
- From the graph, we see that the trips ending at location ids 97, 132, 86, .. account for a higher fare than other locations
- These could be hot zones or busy place of the city like airport, Times square, etc. where the demand for taxi is always high
- Since the demand is always high, the fare charged by the taxi companies for rides culminating at those locations are also high for these locations



Passenger Count Vs fare

- The graph shows that as the passenger count increases the fare also increases
- One way to think about it could be multiple passengers can have multiple drop off locations and as a result the trip distance increases
- As the trip distance increases, the trip fare automatically increases and thereby we are seeing a relationship between passenger count and fare
- A drop in fare amount when the number of passengers is 6 is due an aberration in the dataset. A very few trips had 6 passengers and as these trips had lesser fare, we are seeing this drop in the fare for passenger count of 6

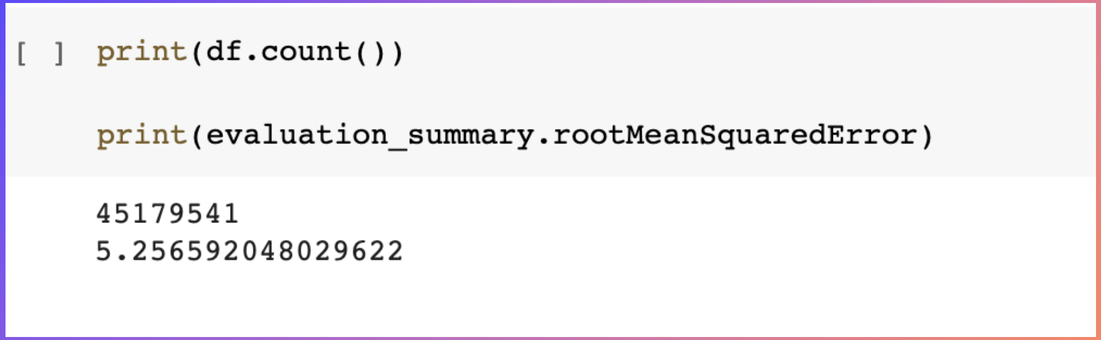
Model evaluation(2020 dataset)



```
In [18]: print(df.count())

         11703499

In [17]: print(evaluation_summary.rootMeanSquaredError)
         print(evaluation_summary.meanSquaredError)
         #print(evaluation_summary.accuracy)

         4.0156072275787675
         16.12510140618284
```

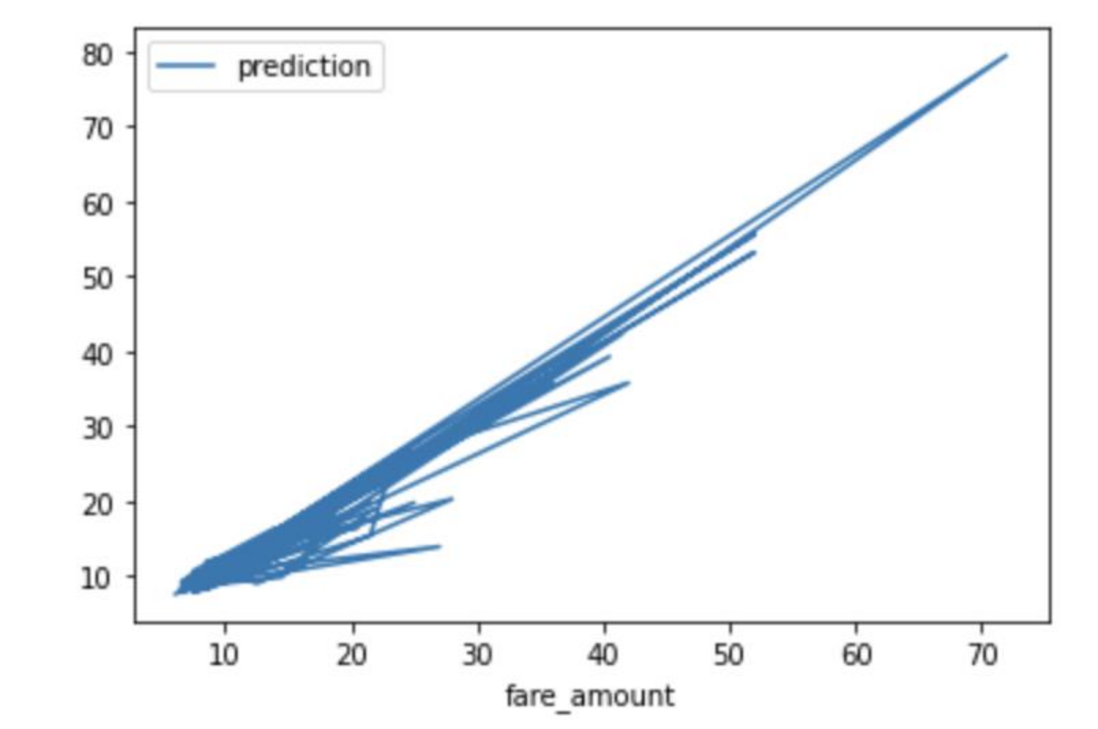
- Columns that were chosen to train the model are Passenger Count, Pick up Location, Drop off Location, Trip Distance, Active day of week and Active hour of day
- The model was trained on 70% of the data and tested on the remaining 30%
- The errors reported by the model after running it on the test data set are
- RootMeanSquaredError = 4.01
- MeanSquaredError = 16.13
- An error of 4 dollars while predicting the fare is not a considerable error.
- So, it can be concluded that the performance of the model is good

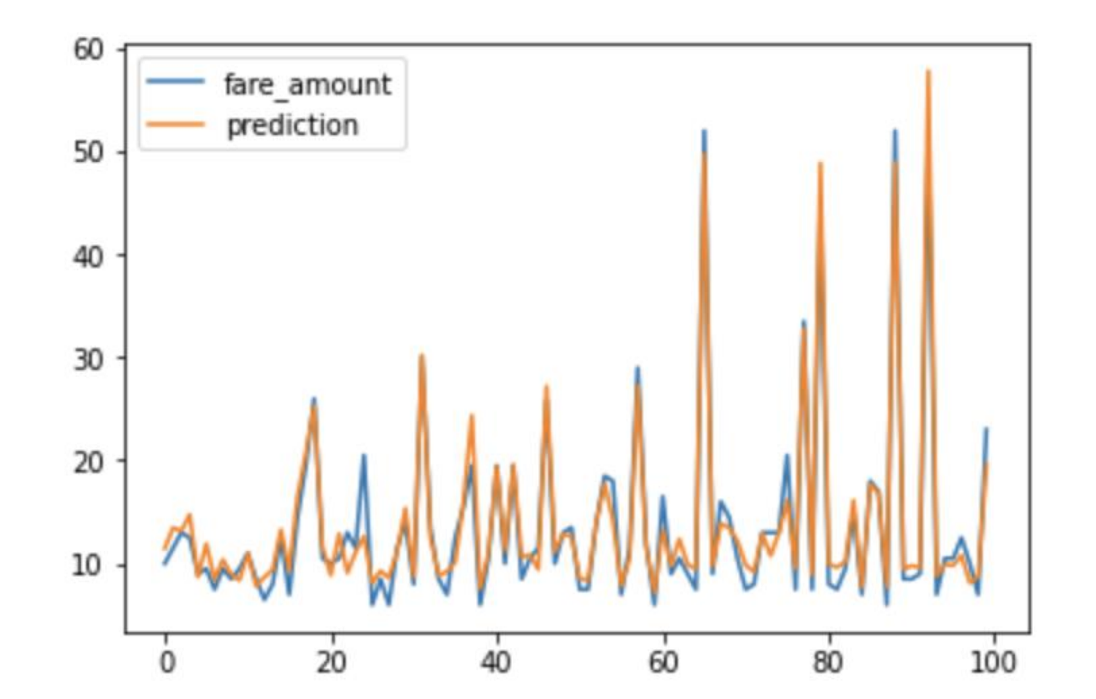
Model evaluation (2019 taxi records)



```
[ ] print(df.count())

    print(evaluation_summary.rootMeanSquaredError)

    45179541
    5.256592048029622
```

- Columns that were chosen to train the model are Passenger Count, Pick up Location, Drop off Location, Trip Distance, Active day of week and Active hour of day
- The model was trained on 85% of the data and tested on the remaining 15%
- The errors reported by the model after running it on the test data set are
- RootMeanSquaredError = 5.25
- MeanSquaredError = 27.6
- An error of 5 dollars while predicting the fare is not a considerable error.
- So, it can be concluded that the performance of the model is good





Actual vs Predicted fare curve

- Random 100 samples were chosen from the values predicted by the model and the predicted fare vs the actual fare are plotted
- Graph 1:
- We have an almost linear line at 45 degrees which shows that the model has performed better
- Graph 2:
- Blue curve represents the actual fare
- Orange curve represents the predicted fare
- Since the model performance is good there is no big difference between the two curves

Conclusion and Limitations

- Based on the linear regression model built using the above dataset, the fare predicted for rides, will now be more accurate
- One possible limitation could be, since the model was trained only on yellow taxi trips, any errors or mis judgements done by Yellow taxi fare predictor algorithm might have passed on to our model as well. This can be minimized in future by training the model with ride information of other taxi service providers like Uber
- Since the dataset did not have any information regarding any coupons or discounts applied by a customer for any of the rides, the model is unaware of these cases as well. This issue can be solved in future if yellow taxi makes this information available in future