New York City Taxi fare Prediction

Spoorthy Reddy Jarugu

Vellore Institute of Technology, Vellore
Deakin University, Australia

Introduction

This is a problem to predict the taxi fare in the city of New York. There are several parameters we need to consider to predict taxi fare. Parameters like distance (distance between pickup and dropoff loaction), hour (weather its a rush hour or normal hour or mid night hour of travel), Passenger (no of passengers tarvelling per trip or ride) and month (weather its snow fall month or normal month during travel).

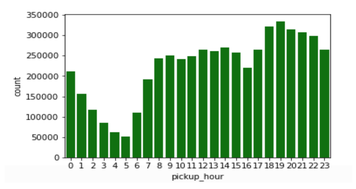
Dataset	Attribute
train.csv	key,fare_amount,pickup_datetime,pickup_longitude,pickup_latitude, dropoff_longitude,dropoff_latitude,passenger_count
test.csv	key,pickup_datetime,pickup_longitude,pickup_latitude, dropoff_longitude,dropoff_latitude,passenger_count

Data processing

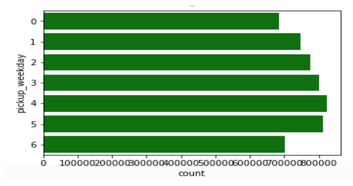
- Cleaning the dataset by removing missing and NAN values.
- Identify the boundary of the New York city and remove the outliers and duplicate data which is out of boundary.
- Print the maximum and minimum value in passenger and cleanign the data for maximum count of 6 passengers per ride.
- Extract Day, Month, Year from pickup_datetime column.

Extracting Data

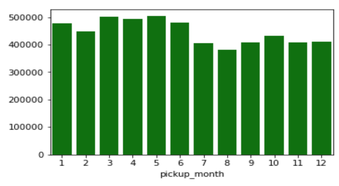
- To predict the taxi fare accurately we are extracting the
  - Hour is calcuted to find weather its mid_night_trip or rush_hour_trip is noted
  - Day on which the passenger is picked upon
  - Month of trip
  - Year of travel



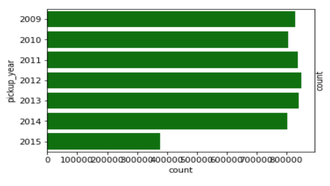
Hour of travel



Weekday of travel



Month of travel



Year of travel

- Trip_diatance is calculated from pickup_latitude, pickup_longitude, dropoff_latitude, dropoff_longitude and stored it in trip_distance.

```
In [35]: from geopy.distance import geodesic

         def distance_calculate(lat,long,drop_lat,drop_long):
             newport_ri = (lat,long)
             cleveland_oh = (drop_lat,drop_long)
             dist=geodesic(newport_ri, cleveland_oh).km
             return dist
```

```
In [36]: df['trip_distance']=list(map(distance_calculate,df['pickup_latitude'],df['pickup_longitude'],
                                    df['dropoff_latitude'],df['dropoff_longitude']))
         df.head()
```

Linear Regression

Using Linear Regression model to predict the taxi fare in the New York city.

```
In [42]: X=df.drop(columns=['key','fare_amount'])
         y=df['fare_amount']
```

Linear Regression

```
In [43]: from sklearn.model_selection import train_test_split

In [44]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=101)

In [45]: from sklearn.linear_model import LinearRegression

In [46]: lm = LinearRegression()

In [47]: lm.fit(X_train,y_train)
Out[47]: LinearRegression()
```

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable (fare_amount). The variable you are using to predict the other variable's value is called the independent variable (trip_distance).

Result

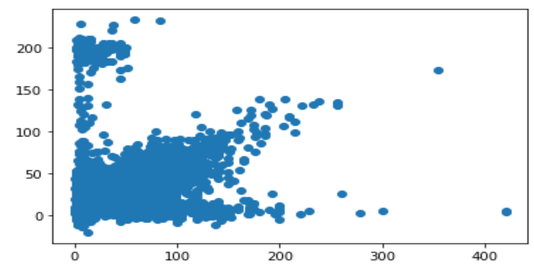
Technically, RMSE is the Root of the Mean of the Square of Errors and MAE is the Mean of Absolute value of Errors. Here, errors are the differences between the predicted values (values predicted by our regression model) and the actual values of a variable. RMSE score, MAE score and MSE score are calculated below.

```
In [58]: from sklearn import metrics

         print('MAE:', metrics.mean_absolute_error(y_test,prediction))
         print('MSE:', metrics.mean_squared_error(y_test,prediction))
         print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,prediction)))

MAE: 2.4184007631012028
MSE: 26.256078544064415
RMSE: 5.124068553802185
```

```
In [59]: lm.score(X_test,y_test)
Out[59]: 0.7109492969947017
```

Below given diagram is the visualization representation of predicted data.

```
In [51]: plt.scatter(y_test,prediction)
Out[51]: <matplotlib.collections.PathCollection at 0x1473954db80>
```



Conclusion

- Fare prediction using latitude and longitude information is showcased.
- Additionally mid_night_trip, Rush_hour_trip, show_season parameters are al considerd in fare calculation.
- The prediction model helps both passengers and dri diction compared to conventional predicti

Acknowledgement
- Kaggle Project and FLIP00 team