CAPABL PROJECT

Predict Taxi Trip Duration

Importing Packages

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

plt.style.use("ggplot")
warnings.filterwarnings('ignore')
```

Data Gathering

In [ ]:
```
df = pd.read_csv(r"C:\Users\HP\Downloads\train.csv")
test = pd.read_csv(r"C:\Users\HP\Downloads\test.csv")
df.head()
```

Out[ ]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.010040
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053

Data Pre-Processing

In [ ]:
```
print("Total number of samples in test dataset: ", df.shape[0])
print("Number of columns in test dataset: ", df.shape[1])
```

```
Total number of samples in test dataset:  1458644
Number of columns in test dataset:  11
```

Inference:

The train dataset has 1458644 rows ans 11 columns

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1458644 entries, 0 to 1458643
Data columns (total 11 columns):
 #   Column              Non-Null Count    Dtype  
---  ------              --------------    -----  
 0   id                  1458644 non-null  object 
 1   vendor_id           1458644 non-null  int64  
 2   pickup_datetime     1458644 non-null  object 
 3   dropoff_datetime    1458644 non-null  object 
 4   passenger_count     1458644 non-null  int64  
 5   pickup_longitude    1458644 non-null  float64
 6   pickup_latitude     1458644 non-null  float64
 7   dropoff_longitude   1458644 non-null  float64
 8   dropoff_latitude    1458644 non-null  float64
 9   store_and_fwd_flag  1458644 non-null  object 
 10  trip_duration       1458644 non-null  int64  
dtypes: float64(4), int64(3), object(4)
memory usage: 122.4+ MB
```

Inference:

It is observed that every variable has a correct datatype except pickup_datetime and dropoff_datetime which should be having datetime datatype

In [ ]:
```
df.describe()
```

Out[ ]:

	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	drop
count	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1
mean	1.534950e+00	1.664530e+00	-7.397349e+01	4.075092e+01	-7.397342e+01	4
std	4.987772e-01	1.314242e+00	7.090186e-02	3.288119e-02	7.064327e-02	3
min	1.000000e+00	0.000000e+00	-1.219333e+02	3.435970e+01	-1.219333e+02	3
25%	1.000000e+00	1.000000e+00	-7.399187e+01	4.073735e+01	-7.399133e+01	4
50%	2.000000e+00	1.000000e+00	-7.398174e+01	4.075410e+01	-7.397975e+01	4
75%	2.000000e+00	2.000000e+00	-7.396733e+01	4.076836e+01	-7.396301e+01	4
max	2.000000e+00	9.000000e+00	-6.133553e+01	5.188108e+01	-6.133553e+01	4

Inference:

The values of each variable have completely different ranges and have to be scaled

In [ ]:
```
new_df = df.copy(deep = True) #making a copy of the original dataset
test_df = df.copy(deep = True)
```

Missing value analysis

In [ ]:
```
new_df.isnull().sum()
```

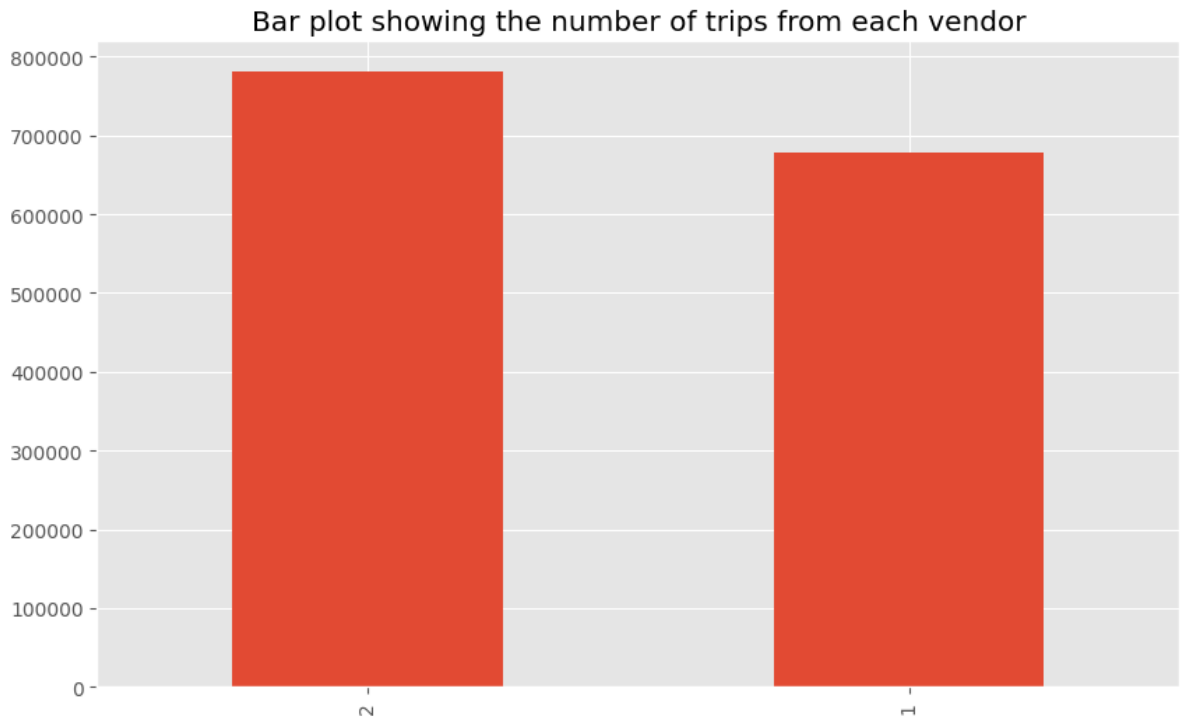
Out[ ]:
```
id                  0
vendor_id           0
pickup_datetime     0
dropoff_datetime    0
passenger_count     0
pickup_longitude    0
pickup_latitude     0
dropoff_longitude   0
dropoff_latitude    0
store_and_fwd_flag  0
trip_duration       0
dtype: int64
```

Inference:

Since there no missing values, there is no need for us to do the missing value analysis

Exploratory Data Analysis

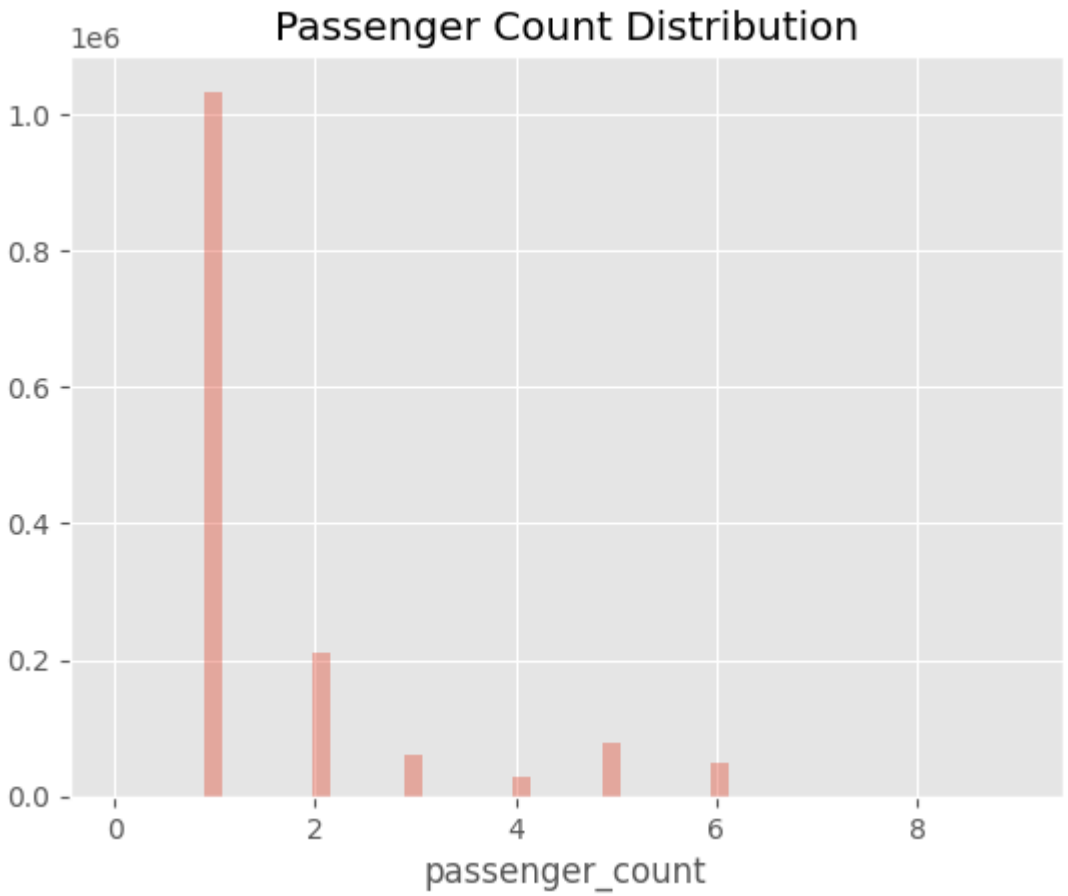
In [ ]:
```
plt.figure(figsize = (10, 6))
plt.title("Bar plot showing the number of trips from each vendor")
new_df["vendor_id"].value_counts().plot(kind = "bar")
plt.show()
```



Inference:

Both of the vendors have a significant amount of trips meaning that they would be useful in predicting the duration

In [ ]:
```
sns.distplot(new_df['passenger_count'],kde=False)
plt.title('Passenger Count Distribution')
plt.show()
```



Inference:

Here we see that the mostly 1 or 2 passengers available in the cab. The instance of large group of people travelling together are rare.

In [ ]:
```
new_df["pickup_datetime"] = pd.to_datetime(new_df["pickup_datetime"])
new_df["dropoff_datetime"] = pd.to_datetime(new_df["dropoff_datetime"])
```

In [ ]:
```
new_df['pickup_hour'] = new_df['pickup_datetime'].dt.hour
new_df['pickup_day'] = new_df['pickup_datetime'].dt.dayofweek
```

In [ ]:
```
new_df['dropoff_hour'] = new_df['dropoff_datetime'].dt.hour
new_df['dropoff_day'] = new_df['dropoff_datetime'].dt.dayofweek
```

In [ ]:
```
new_df
```

Out[ ]:

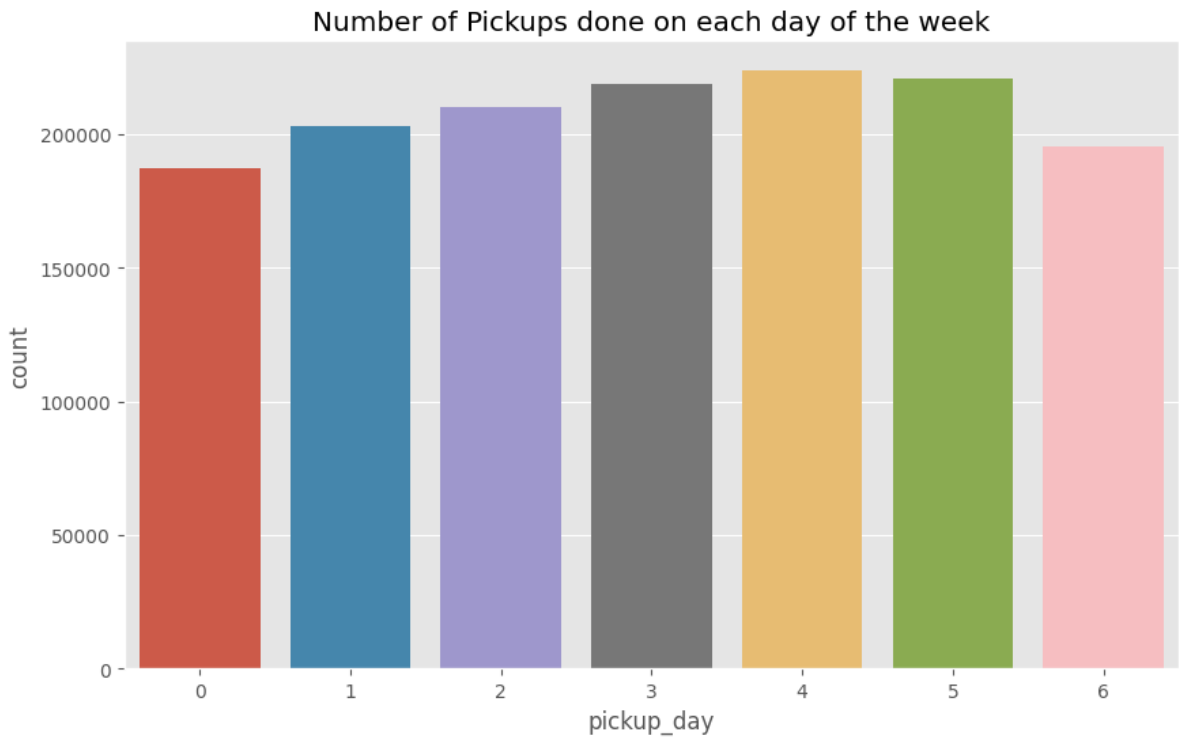
	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_long
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.98
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.98
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.97
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.01
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.97
...	...	...	...	...	...	
1458639	id2376096	2	2016-04-08 13:31:04	2016-04-08 13:44:02	4	-73.98
1458640	id1049543	1	2016-01-10 07:35:15	2016-01-10 07:46:10	1	-74.00
1458641	id2304944	2	2016-04-22 06:57:41	2016-04-22 07:10:25	1	-73.95
1458642	id2714485	1	2016-01-05 15:56:26	2016-01-05 16:02:39	1	-73.98
1458643	id1209952	1	2016-04-05 14:44:25	2016-04-05 14:47:43	1	-73.97

1458644 rows × 15 columns

In [ ]:
```
test_df["pickup_datetime"] = pd.to_datetime(test_df["pickup_datetime"])
test_df["dropoff_datetime"] = pd.to_datetime(test_df["dropoff_datetime"])
test_df['pickup_hour'] = test_df['pickup_datetime'].dt.hour
test_df['pickup_day'] = test_df['pickup_datetime'].dt.dayofweek
test_df['dropoff_hour'] = test_df['dropoff_datetime'].dt.hour
test_df['dropoff_day'] = test_df['dropoff_datetime'].dt.day_name()
```

In [ ]:
```
plt.figure(figsize = (10, 6))
sns.countplot(x='pickup_day',data = new_df)
plt.title('Number of Pickups done on each day of the week')
```

Out[ ]:
```
Text(0.5, 1.0, 'Number of Pickups done on each day of the week')
```

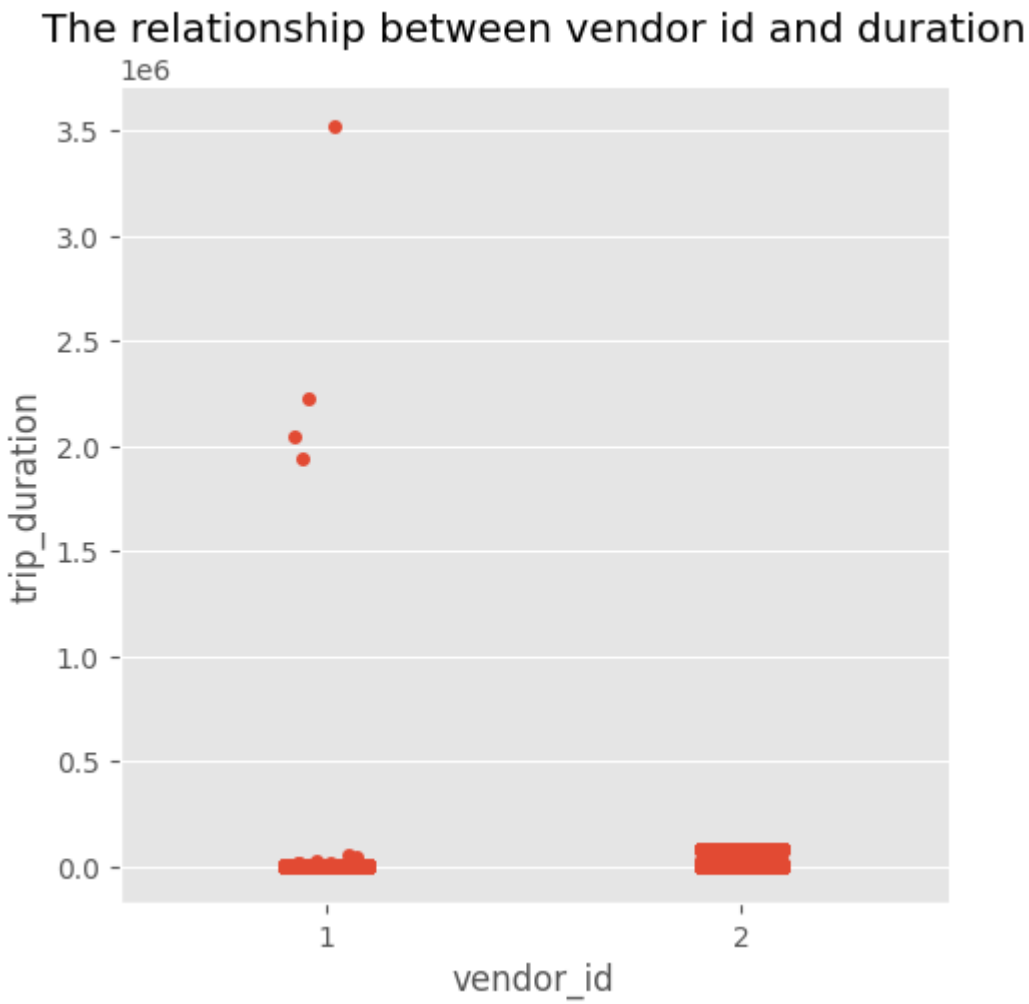


Inference

Thus we see most trips were taken on Friday and Monday being the least. The distribution of trip duration with the days of the week is something to look into as well.

In [ ]:
```
plt.figure(figsize = (15, 6))
sns.catplot(x="vendor_id", y="trip_duration",kind="strip",data=new_df)
plt.title("The relationship between vendor id and duration")
plt.show()
```

```
<Figure size 1500x600 with 0 Axes>
```



Inference

Here we see that vendor 1 mostly provides short trip duration cabs while vendor 2 provides cab for both short and long trips but there has been outliers in vendor 1's case

Feature Selection

In [ ]:
```
new_df["longitudinal_distance"] = abs(new_df["pickup_longitude"] - new_df["c
new_df["latitudinal_distance"] = abs(new_df["pickup_latitude"] - new_df["dro
```

In [ ]:
```
new_df.columns
```

Out[ ]:
```
Index(['id', 'vendor_id', 'pickup_datetime', 'dropoff_datetime',
       'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag',
       'trip_duration', 'pickup_hour', 'pickup_day', 'dropoff_hour',
       'dropoff_day', 'longitudinal_distance', 'latitudinal_distance'],
      dtype='object')
```

In [ ]:
```
X = new_df[["vendor_id", "passenger_count", "pickup_longitude", "pickup_lati
```

In [ ]:
```
X.head()
```

Out[ ]:

	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_lati
0	2	1	-73.982155	40.767937	-73.964630	40.765
1	1	1	-73.980415	40.738564	-73.999481	40.731
2	2	1	-73.979027	40.763939	-74.005333	40.710
3	2	1	-74.010040	40.719971	-74.012268	40.706
4	2	1	-73.973053	40.793209	-73.972923	40.782

In [ ]:
```
y = new_df.trip_duration
```

Training the model

In [ ]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_percentage_error
```

In [ ]:
```
x_train,x_test,y_train,y_test = train_test_split(X,y,test_size = 0.3,random_
```

In [ ]:
```
regression_model = LinearRegression()
regression_model.fit(x_train, y_train)
```

Out[ ]:
```
▼ LinearRegression
LinearRegression()
```

In [ ]:
```
regression_model.intercept_
```

Out[ ]:
```
149265.28284166197
```

In [ ]:
```
regression_model.coef_
```

Out[ ]:
```
array([ 1.97905863e+02,  9.46743403e+00, -3.17959649e+01, -1.09621649e+03,
        9.22815538e+02, -9.42473845e+02, -6.22246315e+00, -2.05902016e+02,
        1.99808399e+02,  9.61403407e+00,  7.73704143e+03,  6.91226594e+03])
```

In [ ]:
```
y_predict = regression_model.predict(x_test)
```

In [ ]:
```
y_predict
```

Out[ ]:
```
array([1157.38636882,  924.31649304,  769.96502678, ...,  674.18571211,
       1508.38661454,  789.25471437])
```

In [ ]:
```
predictions = pd.DataFrame({'Actual': y_test, 'Predicted': y_predict.flatter
```

In [ ]:
```
predictions
```

Out[ ]:

	Actual	Predicted
528024	1357	1157.386369
251854	268	924.316493
433742	116	769.965027
471693	1073	865.840080
1431726	485	869.308208
...	...	...
455353	491	638.950499
1132514	313	832.103683
990885	892	674.185712
177633	1651	1508.386615
385879	931	789.254714

437594 rows × 2 columns

Testing the model

In [ ]:
```
mean_absolute_percentage_error(y_test, y_predict)
```

Out[ ]:
```
1.0852261261993381
```

The Accuracy rate of the taxi_duration is 100% - 10.8% = 89.%(accuracy_score)

Testing with test dataset

In [ ]:
```
new_df["longitudinal_distance"] = abs(test_df["pickup_longitude"] - test_df[
new_df["latitudinal_distance"] = abs(test_df["pickup_latitude"] - test_df["d
X = new_df[["vendor_id", "passenger_count", "pickup_longitude", "pickup_lati
y = new_df.trip_duration
```

In [ ]:
```
regression_model.predict(X)
```

Out[ ]:
```
array([ 865.57126886,  652.91857821, 1275.99415462, ..., 1437.53517721,
        629.52744532,  566.48452541])
```

In [ ]:
```
predictions = pd.DataFrame({'Actual': y_test, 'Predicted': y_predict.flatter
predictions
```

Out[ ]:

	Actual	Predicted
528024	1357	1157.386369
251854	268	924.316493
433742	116	769.965027
471693	1073	865.840080
1431726	485	869.308208
...	...	...
455353	491	638.950499
1132514	313	832.103683
990885	892	674.185712
177633	1651	1508.386615
385879	931	789.254714

437594 rows × 2 columns

In [ ]:
```
mean_absolute_percentage_error(y_test, y_predict)
```

Out[ ]:
```
1.0852261261993381
```

The Accuracy rate of the taxi_duration is 100% - 10.8% = 89.%(accuracy_score)

Conclusion

We have successfully modelled a linear regression model to predict the time taken by vehicles and can be used in bussiness model to optimize transport