ASSIGNMENT - 1

In [1]:

```
#import required libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
#load the dataset
file_path="uber.csv"
uber_data=pd.read_csv(file_path)
```

In [3]:

```
#check the first few rows
print(uber_data.head())
```

```
   Unnamed: 0      key  fare_amount          pickup_datetime  \
0    24238194  52:06.0          7.5  2015-05-07 19:52:06 UTC   
1    27835199  04:56.0          7.7  2009-07-17 20:04:56 UTC   
2    44984355  45:00.0         12.9  2009-08-24 21:45:00 UTC   
3    25894730  22:21.0          5.3  2009-06-26 08:22:21 UTC   
4    17610152  47:00.0         16.0  2014-08-28 17:47:00 UTC   

   pickup_longitude  pickup_latitude  dropoff_longitude  dropoff_latitude  \
0        -73.999817        40.738354         -73.999512         40.723217   
1        -73.994355        40.728225         -73.994710         40.750325   
2        -74.005043        40.740770         -73.962565         40.772647   
3        -73.976124        40.790844         -73.965316         40.803349   
4        -73.925023        40.744085         -73.973082         40.761247   

   passenger_count  
0                1  
1                1  
2                1  
3                3  
4                5  
```

In [4]:

```
#check the shape of the dataset
print("Dimensions:",uber_data.shape)
```

```
Dimensions: (200000, 9)
```

In [5]:

```
#get column names and Types
print("Columns and Types:\n",uber_data.dtypes)
```

```
Columns and Types:
 Unnamed: 0             int64
key                   object
fare_amount          float64
pickup_datetime       object
pickup_longitude     float64
pickup_latitude      float64
dropoff_longitude    float64
dropoff_latitude     float64
passenger_count        int64
dtype: object
```

In [6]:

```
#check for missing values
uber_data.isnull().sum()
```

Out[6]:

```
Unnamed: 0           0
key                  0
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    1
dropoff_latitude     1
passenger_count      0
dtype: int64
```

In [7]:

```
#drop rows with missing values
uber_data.dropna(inplace=True)
```

In [8]:

```
#check missing values rows are drop or not?
uber_data.isnull().sum()
```

Out[8]:

```
Unnamed: 0           0
key                  0
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    0
dropoff_latitude     0
passenger_count      0
dtype: int64
```

In [9]:

```
#get statistical summary
print(uber_data.describe())
```

```
         Unnamed: 0    fare_amount  pickup_longitude  pickup_latitude  \
count  1.999990e+05  199999.000000     199999.000000    199999.000000   
mean   2.771248e+07      11.359892        -72.527631        39.935881   
std    1.601386e+07       9.901760         11.437815         7.720558   
min    1.000000e+00     -52.000000      -1340.648410       -74.015515   
25%    1.382534e+07       6.000000        -73.992065        40.734796   
50%    2.774524e+07       8.500000        -73.981823        40.752592   
75%    4.155535e+07      12.500000        -73.967154        40.767158   
max    5.542357e+07     499.000000         57.418457      1644.421482   

       dropoff_longitude  dropoff_latitude  passenger_count  
count      199999.000000     199999.000000    199999.000000  
mean          -72.525292         39.923890         1.684543  
std            13.117408          6.794829         1.385995  
min         -3356.666300       -881.985513         0.000000  
25%           -73.991407         40.733823         1.000000  
50%           -73.980093         40.753042         1.000000  
75%           -73.963659         40.768001         2.000000  
max          1153.572603        872.697628       208.000000  
```

In [10]:

```python
#Check and Convert Data Types
#Convert pickup_datetime to datetime type
uber_data['pickup_datetime']=pd.to_datetime(uber_data['pickup_datetime'])
```

In [11]:

```python
#Confirm data type
print(uber_data.dtypes)
```

```
Unnamed: 0                         int64
key                               object
fare_amount                      float64
pickup_datetime      datetime64[ns, UTC]
pickup_longitude                 float64
pickup_latitude                  float64
dropoff_longitude                float64
dropoff_latitude                 float64
passenger_count                    int64
dtype: object
```

In [12]:

```python
#Apply MinMaxScaler to numerical Columns
scaler=MinMaxScaler()
numerical_columns= ['fare_amount', 'pickup_longitude', 'pickup_latitude',
'dropoff_longitude', 'dropoff_latitude']
uber_data[numerical_columns] =scaler.fit_transform(uber_data[numerical_colum
```

In [13]:

```python
# Apply LabelEncoder to categorical columns
encoder = LabelEncoder()
uber_data['key'] = encoder.fit_transform(uber_data['key'])
```

In [ ]: