```
!pip install haversine
```

```
Collecting haversine
  Downloading haversine-2.9.0-py2.py3-none-any.whl.metadata (5.8 kB)
Downloading haversine-2.9.0-py2.py3-none-any.whl (7.7 kB)
Installing collected packages: haversine
Successfully installed haversine-2.9.0
```

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import haversine as hs
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error,root_mean_squared_error
```

```
df = pd.read_csv('uber.csv')
df
```

```
        Unnamed: 0                            key  fare_amount  \
0         24238194    2015-05-07 19:52:06.0000003          7.5   
1         27835199    2009-07-17 20:04:56.0000002          7.7   
2         44984355   2009-08-24 21:45:00.00000061         12.9   
3         25894730    2009-06-26 08:22:21.0000001          5.3   
4         17610152  2014-08-28 17:47:00.000000188         16.0   
...            ...                            ...          ...   
199995    42598914   2012-10-28 10:49:00.00000053          3.0   
199996    16382965    2014-03-14 01:09:00.0000008          7.5   
199997    27804658   2009-06-29 00:42:00.00000078         30.9   
199998    20259894    2015-05-20 14:56:25.0000004         14.5   
199999    11951496   2010-05-15 04:08:00.00000076         14.1   

                pickup_datetime  pickup_longitude  pickup_latitude  \
0       2015-05-07 19:52:06 UTC        -73.999817        40.738354   
1       2009-07-17 20:04:56 UTC        -73.994355        40.728225   
2       2009-08-24 21:45:00 UTC        -74.005043        40.740770   
3       2009-06-26 08:22:21 UTC        -73.976124        40.790844   
4       2014-08-28 17:47:00 UTC        -73.925023        40.744085   
...                         ...               ...              ...   
199995  2012-10-28 10:49:00 UTC        -73.987042        40.739367   
199996  2014-03-14 01:09:00 UTC        -73.984722        40.736837   
199997  2009-06-29 00:42:00 UTC        -73.986017        40.756487   
199998  2015-05-20 14:56:25 UTC        -73.997124        40.725452   
199999  2010-05-15 04:08:00 UTC        -73.984395        40.720077   

        dropoff_longitude  dropoff_latitude  passenger_count  
```

```
0               -73.999512        40.723217                1
1               -73.994710        40.750325                1
2               -73.962565        40.772647                1
3               -73.965316        40.803349                3
4               -73.973082        40.761247                5
...                    ...              ...              ...
199995          -73.986525        40.740297                1
199996          -74.006672        40.739620                1
199997          -73.858957        40.692588                2
199998          -73.983215        40.695415                1
199999          -73.985508        40.768793                1

[200000 rows x 9 columns]
```

```
df.head()
```

```
   Unnamed: 0                            key  fare_amount  \
0    24238194    2015-05-07 19:52:06.0000003          7.5
1    27835199    2009-07-17 20:04:56.0000002          7.7
2    44984355   2009-08-24 21:45:00.00000061         12.9
3    25894730    2009-06-26 08:22:21.0000001          5.3
4    17610152  2014-08-28 17:47:00.000000188         16.0

           pickup_datetime  pickup_longitude  pickup_latitude  \
0  2015-05-07 19:52:06 UTC        -73.999817        40.738354
1  2009-07-17 20:04:56 UTC        -73.994355        40.728225
2  2009-08-24 21:45:00 UTC        -74.005043        40.740770
3  2009-06-26 08:22:21 UTC        -73.976124        40.790844
4  2014-08-28 17:47:00 UTC        -73.925023        40.744085

   dropoff_longitude  dropoff_latitude  passenger_count
0         -73.999512         40.723217                1
1         -73.994710         40.750325                1
2         -73.962565         40.772647                1
3         -73.965316         40.803349                3
4         -73.973082         40.761247                5
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 9 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   Unnamed: 0         200000 non-null  int64
 1   key                200000 non-null  object
 2   fare_amount        200000 non-null  float64
 3   pickup_datetime    200000 non-null  object
 4   pickup_longitude   200000 non-null  float64
 5   pickup_latitude    200000 non-null  float64
 6   dropoff_longitude  199999 non-null  float64
 7   dropoff_latitude   199999 non-null  float64
 8   passenger_count    200000 non-null  int64  
dtypes: float64(5), int64(2), object(2)
memory usage: 13.7+ MB
```

```
df.columns
```

```
Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count'],
      dtype='object')
```

```
df = df.drop(['Unnamed: 0', 'key'], axis = 1)
```

```
df.shape
```

```
(200000, 7)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 7 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   fare_amount        200000 non-null  float64
 1   pickup_datetime    200000 non-null  object 
 2   pickup_longitude   200000 non-null  float64
 3   pickup_latitude    200000 non-null  float64
 4   dropoff_longitude  199999 non-null  float64
 5   dropoff_latitude   199999 non-null  float64
 6   passenger_count    200000 non-null  int64  
dtypes: float64(5), int64(1), object(1)
memory usage: 10.7+ MB
```

```
df.describe()
```

```
         fare_amount  pickup_longitude  pickup_latitude
dropoff_longitude  \
count  200000.000000     200000.000000    200000.000000
199999.000000
mean       11.359955        -72.527638        39.935885         -
72.525292
std         9.901776         11.437787         7.720539
13.117408
min       -52.000000      -1340.648410       -74.015515       -
3356.666300
25%         6.000000        -73.992065        40.734796         -
73.991407
50%         8.500000        -73.981823        40.752592         -
```

```
73.980093
75%         12.500000        -73.967154        40.767158          -
73.963658
max        499.000000         57.418457      1644.421482
1153.572603
```

```
       dropoff_latitude  passenger_count
count     199999.000000    200000.000000
mean          39.923890         1.684535
std            6.794829         1.385997
min         -881.985513         0.000000
25%           40.733823         1.000000
50%           40.753042         1.000000
75%           40.768001         2.000000
max          872.697628       208.000000
```

```
df.isnull().sum()
```

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

```
df.dtypes
```

```
fare_amount          float64
pickup_datetime       object
pickup_longitude     float64
pickup_latitude      float64
dropoff_longitude    float64
dropoff_latitude     float64
passenger_count        int64
dtype: object
```

```
df.pickup_datetime = pd.to_datetime(df.pickup_datetime,
                                    errors='coerce')
```

```
df.dtypes
```

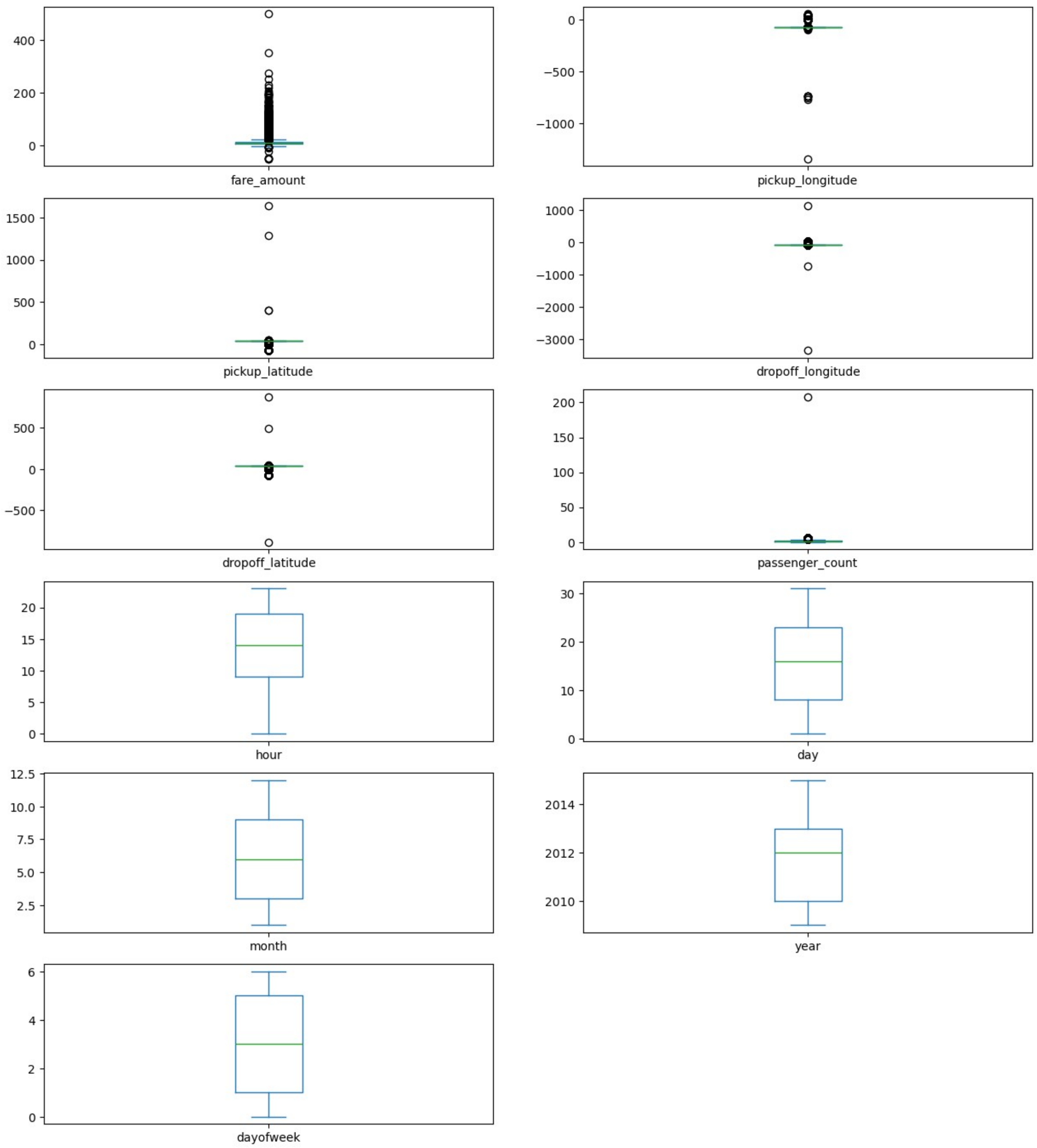
```
fare_amount                      float64
pickup_datetime      datetime64[ns, UTC]
pickup_longitude                 float64
pickup_latitude                  float64
dropoff_longitude                float64
dropoff_latitude                 float64
passenger_count                    int64
dtype: object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

```
df= df.assign(hour = df.pickup_datetime.dt.hour,
              day= df.pickup_datetime.dt.day,
              month = df.pickup_datetime.dt.month,
              year = df.pickup_datetime.dt.year,
              dayofweek = df.pickup_datetime.dt.dayofweek)

df.head()

   fare_amount           pickup_datetime  pickup_longitude
pickup_latitude  \
0          7.5 2015-05-07 19:52:06+00:00        -73.999817
40.738354
1          7.7 2009-07-17 20:04:56+00:00        -73.994355
40.728225
2         12.9 2009-08-24 21:45:00+00:00        -74.005043
40.740770
3          5.3 2009-06-26 08:22:21+00:00        -73.976124
40.790844
4         16.0 2014-08-28 17:47:00+00:00        -73.925023
40.744085

   dropoff_longitude  dropoff_latitude  passenger_count  hour  day
month  \
0         -73.999512         40.723217                1    19    7
5
1         -73.994710         40.750325                1    20   17
7
2         -73.962565         40.772647                1    21   24
8
3         -73.965316         40.803349                3     8   26
6
4         -73.973082         40.761247                5    17   28
8

   year  dayofweek
0  2015          3
1  2009          4
2  2009          0
3  2009          4
4  2014          3

df = df.drop('pickup_datetime',axis=1)
df.dtypes

fare_amount          float64
pickup_longitude     float64
pickup_latitude      float64
dropoff_longitude    float64
dropoff_latitude     float64
passenger_count        int64
```

```
hour                   int32
day                    int32
month                  int32
year                   int32
dayofweek              int32
dtype: object
```

```
df.plot(kind = "box",subplots = True,layout = (7,2),
        figsize=(15,20))
```

```
fare_amount                Axes(0.125,0.786098;0.352273x0.0939024)
pickup_longitude        Axes(0.547727,0.786098;0.352273x0.0939024)
pickup_latitude            Axes(0.125,0.673415;0.352273x0.0939024)
dropoff_longitude       Axes(0.547727,0.673415;0.352273x0.0939024)
dropoff_latitude           Axes(0.125,0.560732;0.352273x0.0939024)
passenger_count         Axes(0.547727,0.560732;0.352273x0.0939024)
hour                       Axes(0.125,0.448049;0.352273x0.0939024)
day                     Axes(0.547727,0.448049;0.352273x0.0939024)
month                      Axes(0.125,0.335366;0.352273x0.0939024)
year                    Axes(0.547727,0.335366;0.352273x0.0939024)
dayofweek                  Axes(0.125,0.222683;0.352273x0.0939024)
dtype: object
```



```python
def remove_outlier(df1 , col):
    Q1 = df1[col].quantile(0.25)
    Q3 = df1[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_whisker = Q1-1.5*IQR
    upper_whisker = Q3+1.5*IQR
    df[col] = np.clip(df1[col] , lower_whisker , upper_whisker)
    return df1
```

```
def treat_outliers_all(df1 , col_list):
    for c in col_list:
        df1 = remove_outlier(df , c)
    return df1

df = treat_outliers_all(df , df.iloc[: , 0::])

travel_dist = []
for pos in range(len(df['pickup_longitude'])):
    long1,lati1,long2,lati2 = [df['pickup_longitude'][pos],
                               df['pickup_latitude'][pos],
                               df['dropoff_longitude'][pos],
                               df['dropoff_latitude'][pos]]
    loc1=(lati1,long1)
    loc2=(lati2,long2)
    c = hs.haversine(loc1,loc2)
    travel_dist.append(c)
print(travel_dist)
df['dist_travel_km'] = travel_dist
df.head()
```

```
IOPub data rate exceeded.
The Jupyter server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--ServerApp.iopub_data_rate_limit`.

Current values:
ServerApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
ServerApp.rate_limit_window=3.0 (secs)


   fare_amount  pickup_longitude  pickup_latitude
dropoff_longitude  \
0          7.5        -73.999817        40.738354         -73.999512

1          7.7        -73.994355        40.728225         -73.994710

2         12.9        -74.005043        40.740770         -73.962565

3          5.3        -73.976124        40.790844         -73.965316

4         16.0        -73.929786        40.744085         -73.973082


   dropoff_latitude  passenger_count  hour  day  month  year
dayofweek  \
0         40.723217              1.0    19    7      5  2015
3
1         40.750325              1.0    20   17      7  2009
```

```
4
2         40.772647             1.0    21   24       8  2009
0
3         40.803349             3.0     8   26       6  2009
4
4         40.761247             3.5    17   28       8  2014
3
```

```
   dist_travel_km
0        1.683325
1        2.457593
2        5.036384
3        1.661686
4        4.116088
```

```
df= df.loc[(df.dist_travel_km >= 1) | (df.dist_travel_km <= 130)]
print('Observations left in the dataset:', df.shape)
```

```
Observations left in the dataset: (199999, 12)
```

```
incorrect_coordinates = df.loc[(df.pickup_latitude > 90) |
                               (df.pickup_latitude < -90) |
                               (df.dropoff_latitude > 90) |
                               (df.dropoff_latitude < -90) |
                               (df.pickup_longitude > 180) |
                               (df.pickup_longitude < -180) |
                               (df.dropoff_longitude > 90) |
                               (df.dropoff_longitude < -90)]

df.drop(incorrect_coordinates, inplace = True,
        errors = 'ignore')
```

```
C:\Users\VEDIKA\AppData\Local\Temp\ipykernel_22680\1102255182.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df.drop(incorrect_coordinates, inplace = True,
```

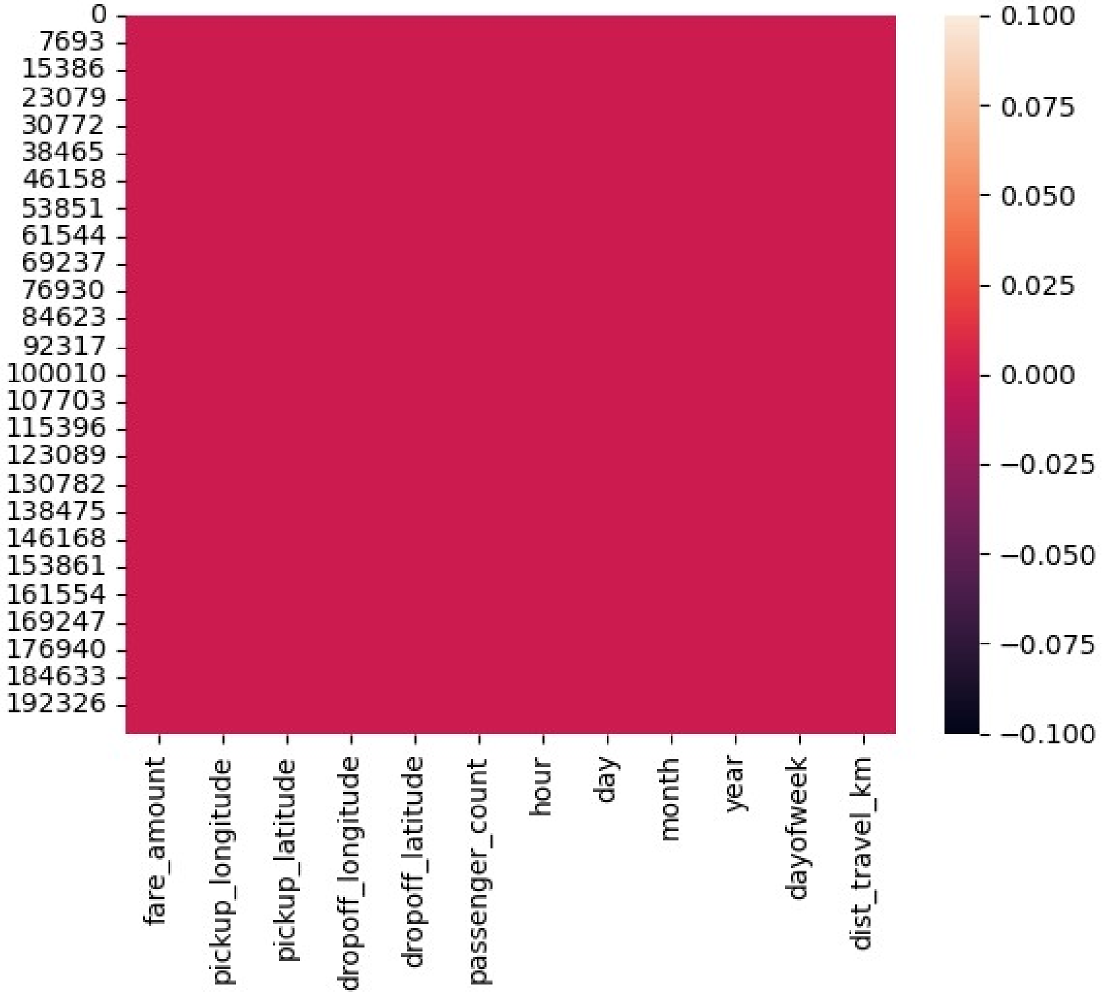
```
df.isnull().sum()
```

```
fare_amount          0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    0
dropoff_latitude     0
passenger_count      0
hour                 0
day                  0
```

```
month              0
year               0
dayofweek          0
dist_travel_km     0
dtype: int64
```

```
sns.heatmap(df.isnull())
```

```
<Axes: >
```



```
corr = df.corr()
corr
```

```
                   fare_amount  pickup_longitude  pickup_latitude  \
fare_amount           1.000000          0.154056        -0.110856
pickup_longitude      0.154056          1.000000         0.259492
pickup_latitude      -0.110856          0.259492         1.000000
dropoff_longitude     0.218681          0.425622         0.048889
dropoff_latitude     -0.125874          0.073309         0.515736
```

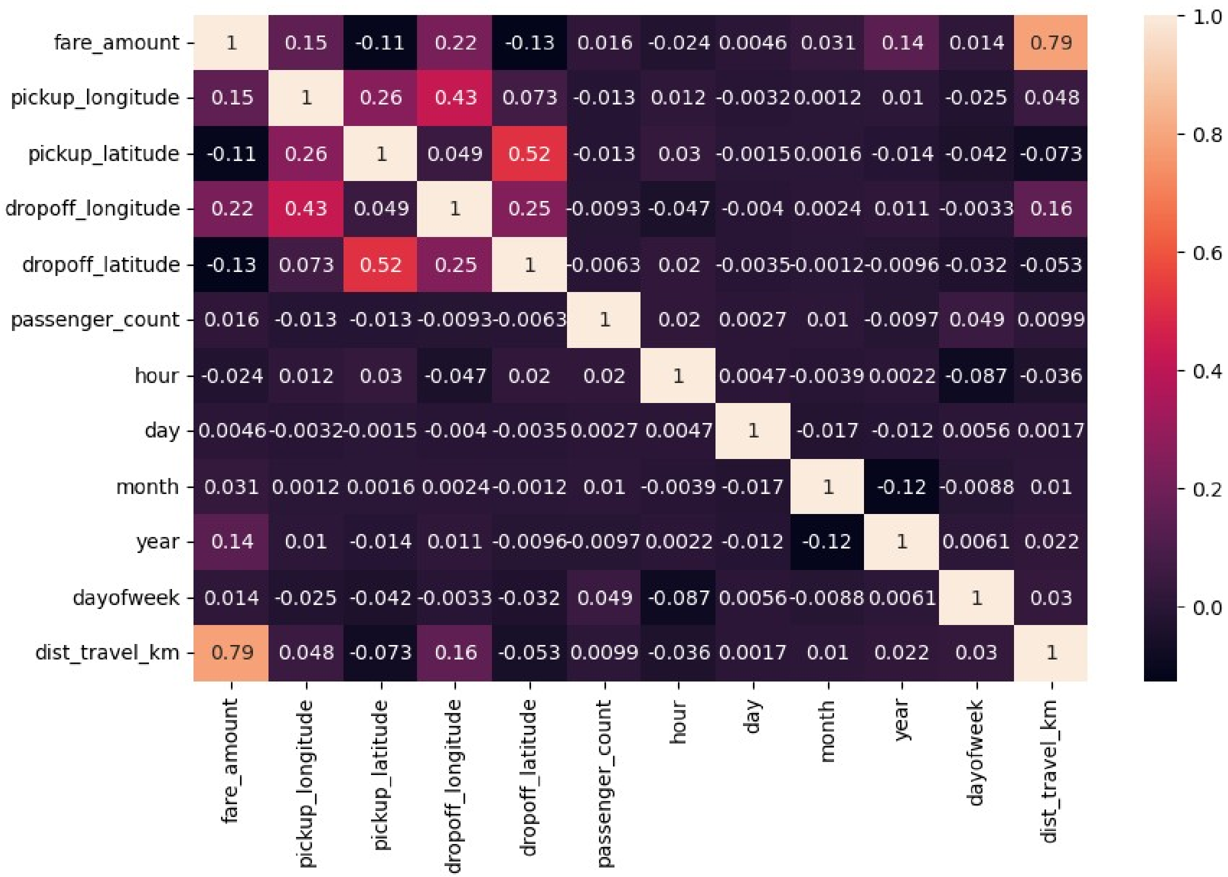
```
passenger_count       0.015798         -0.013202        -0.012879
hour                 -0.023605          0.011590         0.029691
day                   0.004552         -0.003194        -0.001544
month                 0.030815          0.001168         0.001561
year                  0.141271          0.010193        -0.014247
dayofweek             0.013664         -0.024645        -0.042304
dist_travel_km        0.786381          0.048423        -0.073385

                   dropoff_longitude  dropoff_latitude
passenger_count  \
fare_amount                 0.218681         -0.125874
0.015798
pickup_longitude            0.425622          0.073309         -
0.013202
pickup_latitude             0.048889          0.515736         -
0.012879
dropoff_longitude           1.000000          0.245670         -
0.009304
dropoff_latitude            0.245670          1.000000         -
0.006329
passenger_count            -0.009304         -0.006329
1.000000
hour                       -0.046560          0.019765
0.020260
day                        -0.004008         -0.003498
0.002699
month                       0.002392         -0.001191
0.010353
year                        0.011347         -0.009595         -
0.009743
dayofweek                  -0.003337         -0.031932
0.048542
dist_travel_km              0.155200         -0.052657
0.009916

                       hour       day     month      year
dayofweek  \
fare_amount       -0.023605  0.004552  0.030815  0.141271   0.013664

pickup_longitude   0.011590 -0.003194  0.001168  0.010193  -0.024645

pickup_latitude    0.029691 -0.001544  0.001561 -0.014247  -0.042304

dropoff_longitude -0.046560 -0.004008  0.002392  0.011347  -0.003337

dropoff_latitude   0.019765 -0.003498 -0.001191 -0.009595  -0.031932

passenger_count    0.020260  0.002699  0.010353 -0.009743   0.048542

hour               1.000000  0.004664 -0.003924  0.002162  -0.086956
```

```
day                0.004664  1.000000 -0.017358 -0.012165   0.005609

month             -0.003924 -0.017358  1.000000 -0.115860  -0.008785

year               0.002162 -0.012165 -0.115860  1.000000   0.006116

dayofweek         -0.086956  0.005609 -0.008785  0.006116   1.000000

dist_travel_km    -0.035679  0.001738  0.010046  0.022282   0.030403
```

```
                   dist_travel_km
fare_amount              0.786381
pickup_longitude         0.048423
pickup_latitude         -0.073385
dropoff_longitude        0.155200
dropoff_latitude        -0.052657
passenger_count          0.009916
hour                    -0.035679
day                      0.001738
month                    0.010046
year                     0.022282
dayofweek                0.030403
dist_travel_km           1.000000
```

```
fig,axis = plt.subplots(figsize = (10,6))
sns.heatmap(df.corr(),annot = True)
```

```
<Axes: >
```



```
x = df[['pickup_longitude','pickup_latitude','dropoff_longitude',
        'dropoff_latitude','passenger_count','hour','day','month',
        'year', 'dayofweek', 'dist_travel_km']]
y = df['fare_amount']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.33)
regression = LinearRegression()
regression.fit(x_train,y_train)

LinearRegression()

regression.intercept_

3642.245916151823

regression.coef_

array([ 2.59328347e+01, -7.47376142e+00,  1.94138306e+01, -
1.76412371e+01,
        6.72633282e-02,  5.33191567e-03,  3.33385157e-03,
5.89960478e-02,
        3.67958475e-01, -3.39071893e-02,  1.85104129e+00])
```

```
prediction = regression.predict(x_test)
print('Prediction for x:\n', prediction,'\n')
print('Fare Amount test data:\n', y_test)
```

```
Prediction for x:
 [11.84624141  8.13921591  8.99398439 ...  5.37209863  5.67245379
  8.29924635]

Fare Amount test data:
 27708     12.5
6277      10.9
196687     9.7
118237     8.5
194994     6.1
          ...
39133     11.7
93332     15.0
190275     6.9
169173     4.1
18935     12.0
Name: fare_amount, Length: 66000, dtype: float64
```

```
print('R2 Score:\n',r2_score(y_test, prediction))
```

```
R2 Score:
 0.6616183907981845
```

```
MSE = mean_squared_error(y_test, prediction)
print('Mean Squared Error:\n', MSE)
```

```
Mean Squared Error:
 10.028547305604478
```

```
RMSE = root_mean_squared_error(y_test, prediction)
print('Root Mean Squared Error:\n', RMSE)
```

```
Root Mean Squared Error:
 3.1667881687293953
```

```
rf = RandomForestRegressor(n_estimators=100)
rf.fit(x_train, y_train)
```

```
y_pred = rf.predict(x_test)
print('Predictions for Fare Amount:\n', y_pred)
```

```
Predictions for Fare Amount:
 [ 8.782   9.705   8.0947 ... 17.314   6.78   11.53  ]
```

```
R2_Random = r2_score(y_test, y_pred)
print('Random R2 Score:\n', R2_Random)
```

```
Random R2 Score:
 0.7948958964943291
```

```python
MSE_Random = mean_squared_error(y_test, y_pred)
print('Random Mean Squared Error:\n', MSE_Random)
```

```
Random Mean Squared Error:
 6.071112258179303
```

```python
RMSE_Random = root_mean_squared_error(y_test, y_pred)
print('Random Root Mean Squared Error:\n', RMSE_Random)
```

```
Random Root Mean Squared Error:
 2.463962714445838
```