*#Shubhada Jadhav #COBA55*

import pandas as pd import numpy as np

import matplotlib.pyplot as plt import seaborn as sns

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import r2\_score, mean\_squared\_error data = pd.read\_csv("Uber.csv")

data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | pickup\_datetime 2015-05-07 19:52:06 UTC | | | pickup\_longitude  -73.999817 | pickup\_latitude \  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]

data["pickup\_datetime"] = pd.to\_datetime(data["pickup\_datetime"]) missing\_values = data.isnull().sum()

print("Missing values in the dataset:") print(missing\_values) data.dropna(inplace=True) missing\_values = data.isnull().sum() print("Missing values after handling:") print(missing\_values) sns.boxplot(x=data["fare\_amount"]) plt.show()

Missing values in the dataset:

|  |  |
| --- | --- |
| 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 |  |

Missing values after handling: key 0

fare\_amount 0

pickup\_datetime 0

pickup\_longitude 0

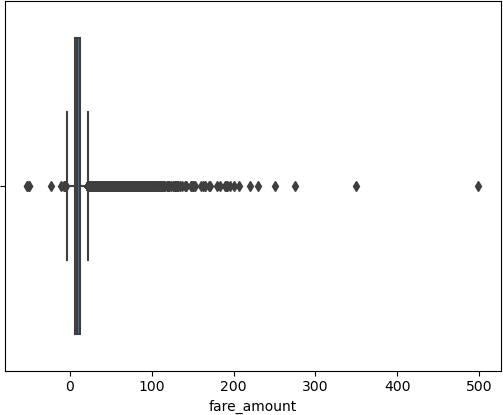
pickup\_latitude 0

dropoff\_longitude 0

dropoff\_latitude 0

passenger\_count 0

dtype: int64



Q1 = data["fare\_amount"].quantile(0.25) Q3 = data["fare\_amount"].quantile(0.75)

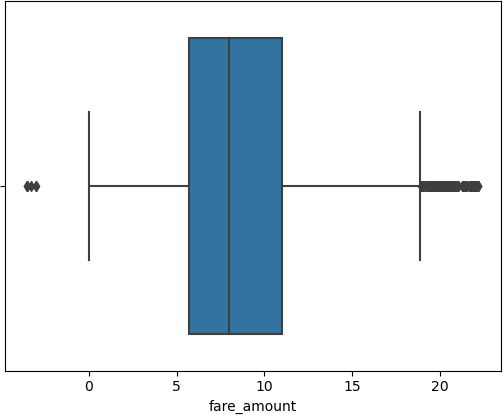
IQR = Q3 - Q1

threshold = 1.5

lower\_bound = Q1 - threshold \* IQR upper\_bound = Q3 + threshold \* IQR

data\_no\_outliers = data[(data["fare\_amount"] >= lower\_bound) & (data["fare\_amount"] <= upper\_bound)] sns.boxplot(x=data\_no\_outliers["fare\_amount"])

plt.show()

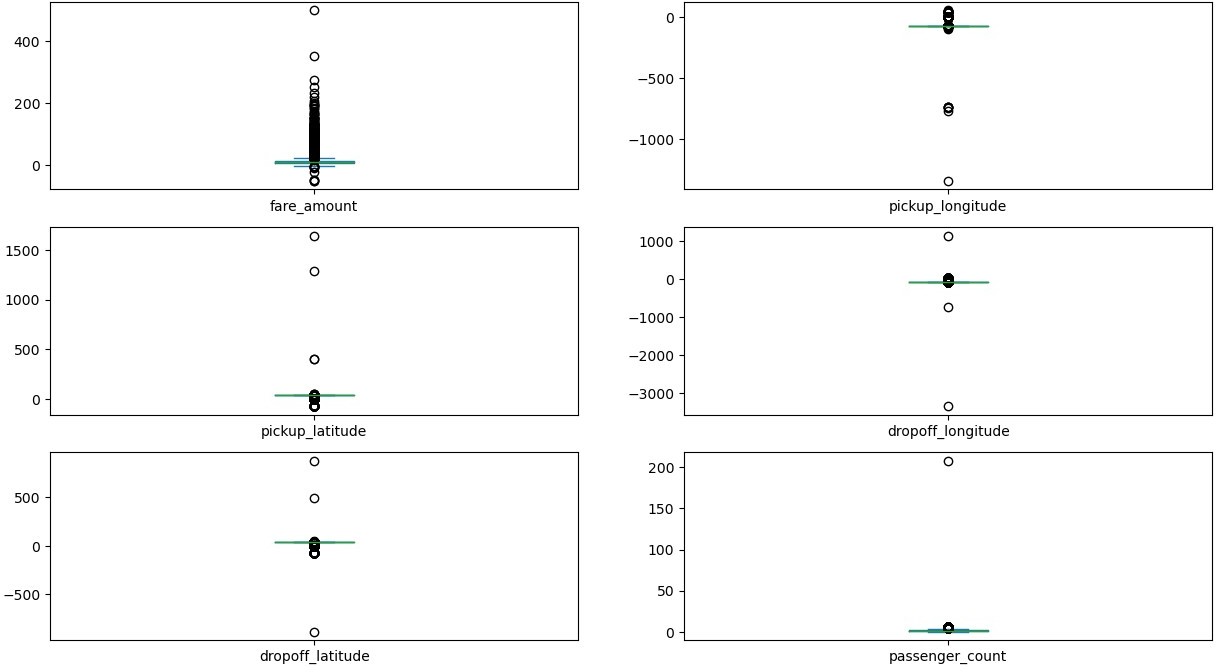


fare\_amount pickup\_longitude pickup\_latitude dropoff\_longitude dropoff\_latitude passenger\_count

AxesSubplot(0.125,0.786098;0.352273x0.0939024) AxesSubplot(0.547727,0.786098;0.352273x0.0939024) AxesSubplot(0.125,0.673415;0.352273x0.0939024) AxesSubplot(0.547727,0.673415;0.352273x0.0939024) AxesSubplot(0.125,0.560732;0.352273x0.0939024) AxesSubplot(0.547727,0.560732;0.352273x0.0939024)

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

dtype: object



correlation\_matrix = data.corr() sns.heatmap(correlation\_matrix, annot=True) plt.show()

X = data[['pickup\_longitude', 'pickup\_latitude', 'dropoff\_longitude', 'dropoff\_latitude', 'passenger\_count']]

y = data['fare\_amount']

... 199995 3.0

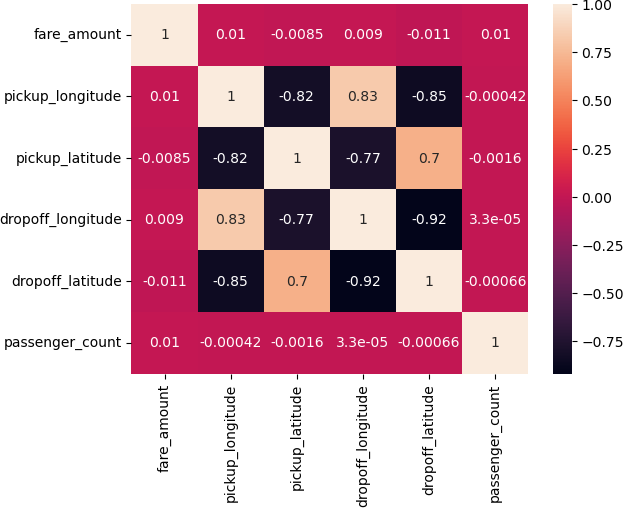
199996 7.5

199997 30.9

199998 14.5

199999 14.1

Name: fare\_amount, Length: 199999, dtype: float64



|  |  |
| --- | --- |
| y |  |
| 0 | 7.5 |
| 1 | 7.7 |
| 2 | 12.9 |
| 3 | 5.3 |
| 4 | 16.0 |

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

lr\_model = LinearRegression() lr\_model.fit(X\_train, y\_train)

LinearRegression()

rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42) rf\_model.fit(X\_train, y\_train)

RandomForestRegressor(random\_state=42)

y\_pred\_lr = lr\_model.predict(X\_test) y\_pred\_lr

print("Linear Model:",y\_pred\_lr) y\_pred\_rf = rf\_model.predict(X\_test) print("Random Forest Model:", y\_pred\_rf)

Linear Model: [11.29237916 11.29171388 11.5718662 ... 11.29183291

11.43252639

11.29190248]

Random Forest Model: [ 9.262 5.043 12.547 ... 6.8087 11.279

8.315 ]

r2\_lr = r2\_score(y\_test, y\_pred\_lr)

rmse\_lr = np.sqrt(mean\_squared\_error(y\_test, y\_pred\_lr))

print("Linear Regression - R2:", r2\_lr) print("Linear Regression - RMSE:", rmse\_lr)

Linear Regression - R2: 0.00034152697863043535 Linear Regression - RMSE: 10.197470623964248

r2\_rf = r2\_score(y\_test, y\_pred\_rf)

rmse\_rf = np.sqrt(mean\_squared\_error(y\_test, y\_pred\_rf)) print("Random Forest Regression R2:", r2\_rf) print("Random Forest Regression RMSE:",rmse\_rf)

Random Forest Regression R2: 0.7011790407391916 Random Forest Regression RMSE: 5.575350372469675