# Assignment no: 1

```
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 RandomForestClassifier


from sklearn.impute import SimpleImputer


from sklearn.metrics import r2_score, mean_squared_error


from scipy import stats
```

In [124]:

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

In [125]:

```
df.head()
```

Out[125]:

| | Unnamed: 0 | key | fare_amount | pickup_datetime | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_latitude | passenger_count |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 24238194 | 2015-05-07 19:52:06.0000003 | 7.5 | 2015-05-07 19:52:06 UTC | -73.999817 | 40.738354 | -73.999512 | 40.723217 | 1 |

| | Unnamed: 0 | key | fare_amount | pickup_datetime | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_latitude | passenger_count |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 27835199 | 2009-07-17 20:04:56.0000002 | 7.7 | 2009-07-17 20:04:56 UTC | -73.994355 | 40.728225 | -73.994710 | 40.750325 | 1 |
| **2** | 44984355 | 2009-08-24 21:45:00.00000061 | 12.9 | 2009-08-24 21:45:00 UTC | -74.005043 | 40.740770 | -73.962565 | 40.772647 | 1 |
| **3** | 25894730 | 2009-06-26 08:22:21.0000001 | 5.3 | 2009-06-26 08:22:21 UTC | -73.976124 | 40.790844 | -73.965316 | 40.803349 | 3 |
| **4** | 17610152 | 2014-08-28 17:47:00.000000188 | 16.0 | 2014-08-28 17:47:00 UTC | -73.925023 | 40.744085 | -73.973082 | 40.761247 | 5 |

In [126]:

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

Out[126]:

```
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 [127]:

```
df['pickup_datetime']=pd.to_datetime(df['pickup_datetime'])
```

In [128]:

```
numeric_columns = df.select_dtypes(include=[np.number]).columns

imputer = SimpleImputer(strategy='mean')

df[numeric_columns] = imputer.fit_transform(df[numeric_columns])
```

In [129]:

```
df.dropna(subset=['fare_amount'],inplace=True)
```

In [130]:

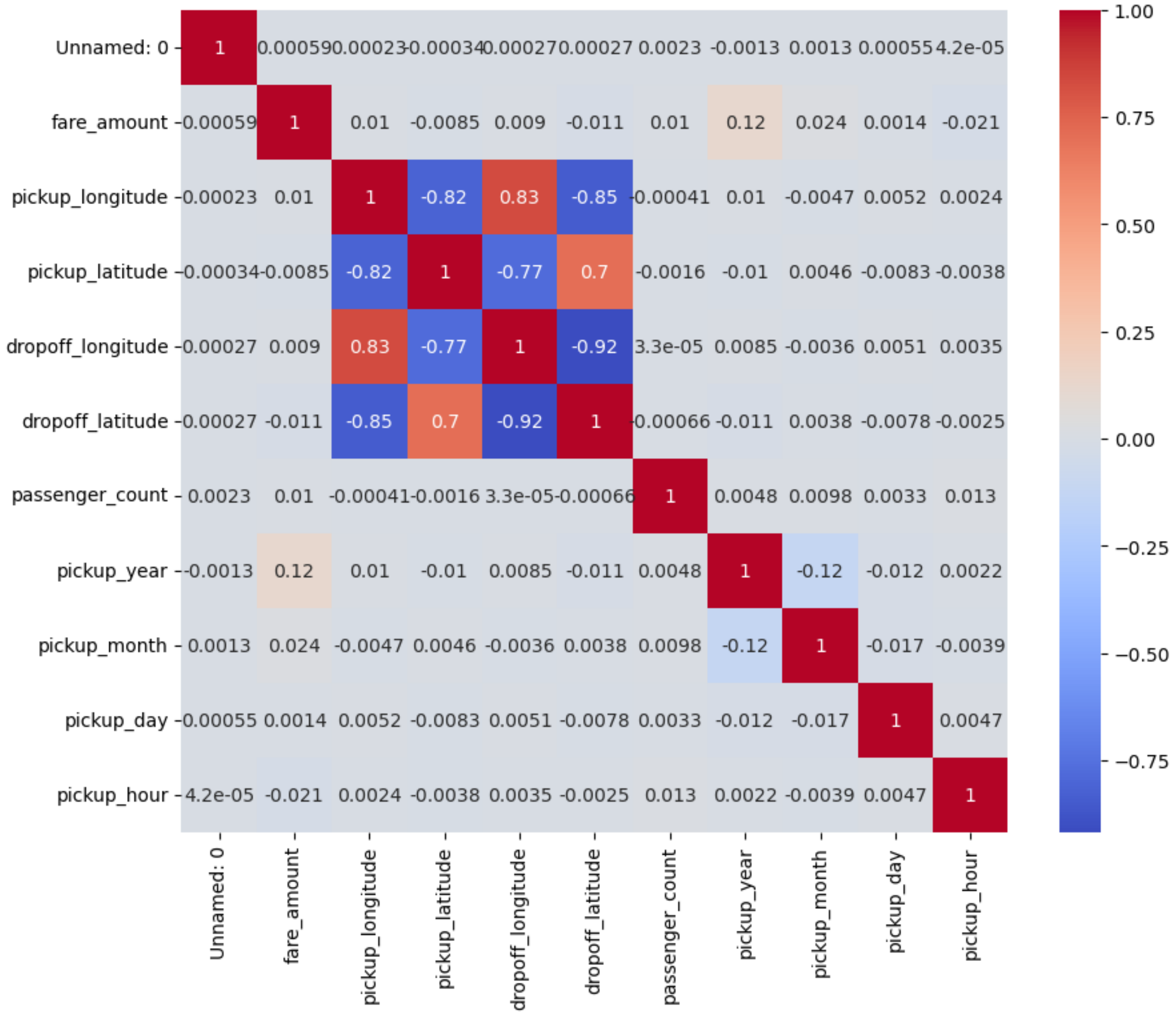
```
df['pickup_year']=df['pickup_datetime'].dt.year
df['pickup_month']=df['pickup_datetime'].dt.month
df['pickup_day']=df['pickup_datetime'].dt.day
df['pickup_hour']=df['pickup_datetime'].dt.hour
```

In [131]:

```
df.drop(columns=['pickup_datetime','key'],inplace=True)
```

In [132]:

```
corr_matrix=df.corr()
plt.figure(figsize=(10,8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()
```



In [133]:

```
X = [[1], [2], [3], [4]]
y = [1, 2, 3, 4]
```

In [134]:

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [135]:

```
lr_model=LinearRegression()
```

In [136]:

```
lr_model.fit(x_train, y_train)
y_pred_lr = lr_model.predict(x_test)
```

In [137]:

```
rf_model = RandomForestClassifier()
```

In [138]:

```
rf_model.fit(x_train, y_train)
y_pred_rf = rf_model.predict(x_test)
```

In [139]:

```
r2_lr = r2_score(y_test, y_pred_lr)
rmse_lr = np.sqrt(mean_squared_error(y_test, y_pred_lr))


r2_rf = r2_score(y_test, y_pred_rf)
rmse_rf = np.sqrt(mean_squared_error(y_test, y_pred_rf))
```

```
C:\Users\hp\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:996: UndefinedMetricWarning: R^2 score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
C:\Users\hp\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:996: UndefinedMetricWarning: R^2 score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
```

In [140]:

```
print("Linear Regression R2:", r2_lr, " RMSE:", rmse_lr)
print("Random Forest R2:", r2_rf, " RMSE:", rmse_rf)
```

```
Linear Regression R2: nan  RMSE: 0.0
Random Forest R2: nan  RMSE: 1.0
```

In [ ]: