#Include library#

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

import datetime

from sklearn.metrics.pairwise import haversine\_distances

from math import radians

from sklearn.model\_selection import train

#For colab to upload csv file#

from google.colab import files

uploaded = files.upload()

#To read csv file#

df = pd.read\_csv('NYCTaxiFares.csv')

#To display first five records#

df.head()

#To display number of columns and rows#

df.shape

#To describe the data#

df.describe()

#To covert given time into Indian time#

df.pickup\_datetime = pd.to\_datetime(df.pickup\_datetime) - datetime.timedelta(hours=4)

#Display records after change in time#

df.head()

#To find the day month year hour minute#

df['day'] = df.pickup\_datetime.dt.day

df['month'] = df.pickup\_datetime.dt.month

df['year'] = df.pickup\_datetime.dt.year

df['hour'] = df.pickup\_datetime.dt.hour

df['minute'] = df.pickup\_datetime.dt.minute

df.drop('pickup\_datetime', axis=1, inplace=True)

df.head()

df.columns

#To find data type of each column#

df.dtypes

#To find the distance using given data#

def findDist(pickup\_longitude,pickup\_latitude,dropoff\_longitude,dropoff\_latitude):

    pickup\_long\_rad = radians(pickup\_longitude)

    pickup\_lat\_rad = radians(pickup\_latitude)

    drop\_long\_rad = radians(dropoff\_longitude)

    drop\_lat\_rad = radians(dropoff\_latitude)

    dist = haversine\_distances([[pickup\_lat\_rad,pickup\_long\_rad],[drop\_lat\_rad,drop\_long\_rad]])

    return dist \* 6371

# Creating Distance column#

rad = map(findDist, df.pickup\_longitude,df.pickup\_latitude,df.dropoff\_longitude,df.dropoff\_latitude)

distmat = list(rad)

distance = []

for i in distmat:

    distance.append(i[0][1])

df['distance'] = distance

df.head()

#To drop the unwanted column#

dist\_data = df.drop(['pickup\_longitude', 'pickup\_latitude', 'dropoff\_longitude', 'dropoff\_latitude', 'day', 'month', 'year', 'hour', 'minute', 'passenger\_count'], axis=1)

dist\_data.head()

sns.barplot(df.day.sort\_index(), df.day.value\_counts().sort\_index())

plt.plot(df.day.value\_counts().sort\_index())

#Visualization#

sns.pairplot(dist\_data, hue='fare\_class')

sns.heatmap(dist\_data.corr(), annot=True)

Regression

X = dist\_data.drop('fare\_amount', axis=1)

y = dist\_data['fare\_amount']

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

from sklearn.linear\_model import LinearRegression

lin\_reg = LinearRegression(normalize=True, n\_jobs=-1)

lin\_reg.fit(X\_train,y\_train)

print('Score:', lin\_reg.score(X\_test,y\_test)\*100)