import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

energy\_df=pd.read\_csv('energydata\_complete (1).csv')

X=energy\_df.iloc[:,5].values

Y=energy\_df.iloc[:,13].values

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(X,Y,test\_size=1/3,random\_state=0)

from sklearn.linear\_model import LinearRegression

regressor=LinearRegression()

X\_train=X\_train.reshape(-1,1)

Y\_train=Y\_train.reshape(-1,1)

regressor.fit(X\_train,Y\_train)

#predicting test result

X\_test=X\_test.reshape(-1,1)

Y\_pred=regressor.predict(X\_test)

from sklearn.metrics import r2\_score

r2=r2\_score(Y\_pred,Y\_test)

round(r2,2)

# r2\_score=0.47

# question no 13

#Firstly, we normalise our dataset to a common scale using the min max scaler

from sklearn.preprocessing import MinMaxScaler

scaler=MinMaxScaler()

# We have to remove the date string as the MinMaxScaler does not work with strings

new\_energy\_df=energy\_df.drop(columns=['date','lights'])

normalized\_df=pd.DataFrame(scaler.fit\_transform(new\_energy\_df))

features\_df=new\_energy\_df.drop('Appliances', axis=1)

target\_value=normalized\_df[[0]]

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(features\_df, target\_value, test\_size= 0.3 ,random\_state= 42)

from sklearn.linear\_model import LinearRegression

linear\_model=LinearRegression()

linear\_model.fit(x\_train,y\_train)

predicted\_values=linear\_model.predict(x\_test)

# predicting the model accuracy using mae

from sklearn.metrics import mean\_absolute\_error

mae=mean\_absolute\_error(predicted\_values,y\_test)

round(mae,2)

# mae=0.05

# Residual sum of squares

import numpy as np

rss=np.sum(np.square(predicted\_values-y\_test))

round(rss,2)

# using rmse

from sklearn.metrics import mean\_squared\_error

rmse=np.sqrt(mean\_squared\_error(predicted\_values, y\_test))

round(rmse,3)

# using r-squared

from sklearn.metrics import r2\_score

rs=r2\_score(predicted\_values,y\_test)

round(rs,2)