Practical 1

Set A

Q.1. **Create ‘sales’ Data set having 5 columns namely: ID, TV, Radio, Newspaper and Sales.(random 500 entries).**
Build a linear regression model by identifying independent and target variable.
Split the variables into training and testing sets then divide the training and testing sets into a 7:3 ratio, respectively and print them.
Build a simple linear regression model.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_squared_error


ID=random.sample(range(0,500),500)
TV=random.sample(range(200,800),500)
Radio=random.sample(range(100,900),500)
Newspaper=random.sample(range(100,600),500)
Sales=random.sample(range(100,900),500)
data=list(zip(ID,TV,Radio,Newspaper,Sales))
df=pd.DataFrame(data,columns=['ID','TV','Radio','Newspaper','Sales'])
print(df)


X = np.array(df[['TV']])
                 # Store‘TV’column as a numpy array into ‘X’ variable
y = np.array(df[['Sales']])
                 # Store ‘Sales’ column as a numpy array into ‘y’ variable
print(X.shape) # Vewing the shape of X
print(y.shape) # Vewing the shape of y

plt.scatter(X,y,color="red")    # Plot a graph X vs y
plt.title('TV vs Sales')        # Title of the graph as'TV vs Sales'
plt.xlabel('TV')                # X label as 'TV'
plt.ylabel('Sales')             # y label as 'Sales'
plt.show()
```

```
# Splitting our Data set in Dependent and Independent variables.
# Spliting into train & test dataset
# Here we split our ‘X’ and ‘y’ dataset into ‘X_train’, ‘X_test’ and ‘y_train’, ‘y_test’.
# Here we take 30% data as test dataset and remaining as train dataset. We take the random_state value as 15 for our better prediction.

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.30,random_state=15)



regressor = LinearRegression()     # Creating a regressor
regressor.fit(X_train,y_train)     # Fiting the dataset into the model


plt.scatter(X_test,y_test,color="green") # Plot a graph with X_test vs y_test
plt.plot(X_train,regressor.predict(X_train),color="red",linewidth=3) # Regressior line showing
plt.title('Regression(Test Set)')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.show()



plt.scatter(X_train,y_train,color="blue") # Plot a graph with X_train vs y_train
plt.plot(X_train,regressor.predict(X_train),color="red",linewidth=3) # Regressior line showing
plt.title('Regression(training Set)')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.show()



y_pred = regressor.predict(X_test)
print('R2 score: %.2f' % r2_score(y_test,y_pred)) # Priniting R2 Score
print('Mean Error :',mean_squared_error(y_test,y_pred)) # Priniting the me
an error
```

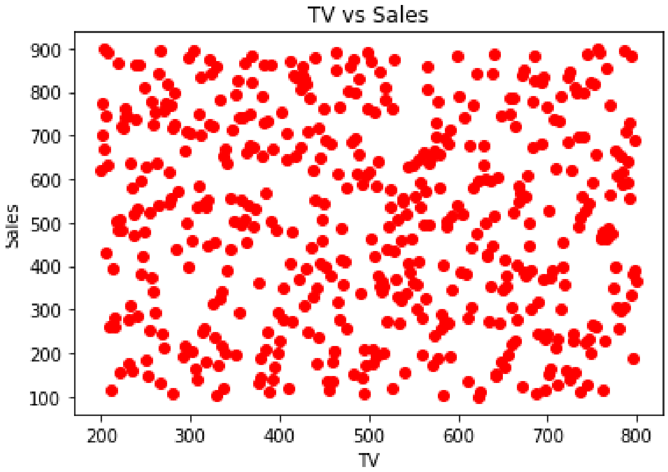
```
def TV_price(tv):
  result=regressor.predict(np.array(tv).reshape(1, -1))
  return(result[0,0])
TV_unit = int(input('Enter number of TVs : '))
print('This TV Price will be : ',int(TV_price(TV_unit))*10,'□')
```

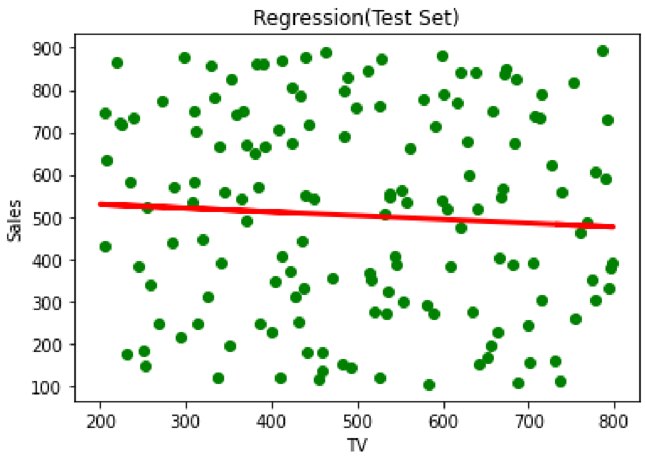
Output :-

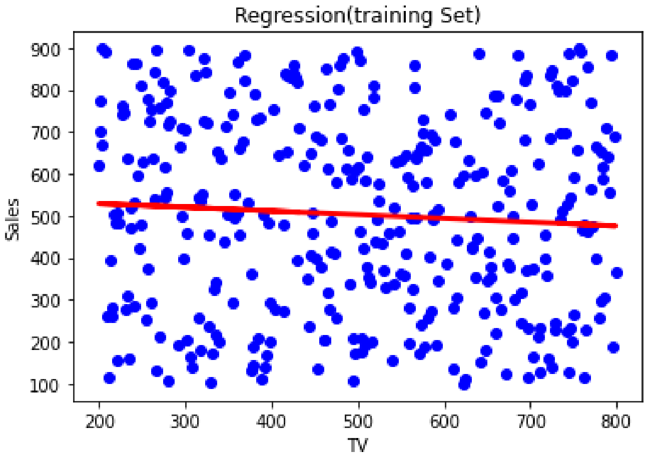
```
           ID    TV   Radio  Newspaper  Sales

    0     356   525    149         239    575
    1     373   748    730         332    824
    2      32   631    362         564    600
    3     176   645    196         569    606
    4     426   412    688         441    408
    ..    ...   ...    ...         ...    ...
    495   135   763    372         219    116
    496   409   662    233         297    525
    497   421   469    240         483    413
    498   436   459    114         209    137
    499   475   411    395         319    872

    [500 rows x 5 columns]
    (500, 1)
    (500, 1)
```







```
R2 score: 0.00

Mean Error : 57460.66887700949
```