Practical 1

Set A

Q.2. **Create 'realestate' Data set having 4 columns namely: ID,flat, houses and purchases (random 500 entries).**
Build a linear regression model by identifying independent and target variable.
Split the variables into training and testing sets and print them.
Build a simple linear regression model for predicting purchases.

➔

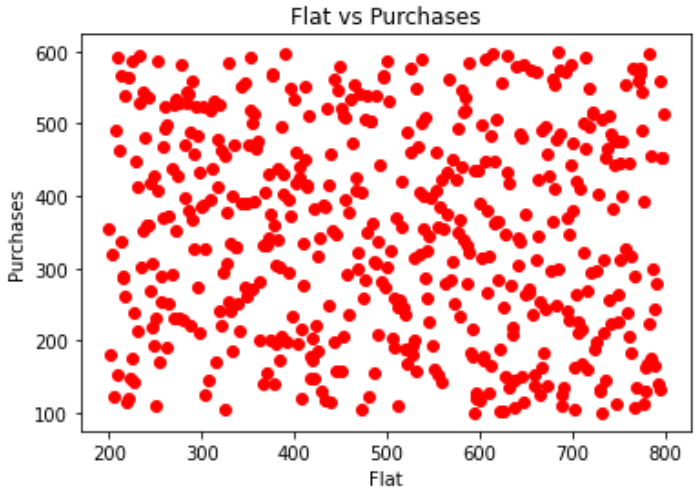
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_squared_error

ID=random.sample(range(0,500),500)
Flat=random.sample(range(200,800),500)
Houses=random.sample(range(100,900),500)
Purchases=random.sample(range(100,600),500)

data=list(zip(ID,Flat,Houses,Purchases))
df_realestate=pd.DataFrame(data,columns=['ID','Flat','Houses','Purchases'])
print(df_realestate)

X = np.array(df_realestate[['Flat']])
# Store'Flat'column as a numpy array into 'X' variable
y = np.array(df_realestate[['Purchases']])
# Store 'Purchases' 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('Flat vs Purchases')  #Title of the graph as 'Flat vs Purchases'
plt.xlabel('Flat')                # X label as 'Flat'
plt.ylabel('Purchases')             # y label as 'Purchases'
plt.show()

# Splitting our Data set in Dependent and Independent variables.
# Spliting into train & test dataset
```

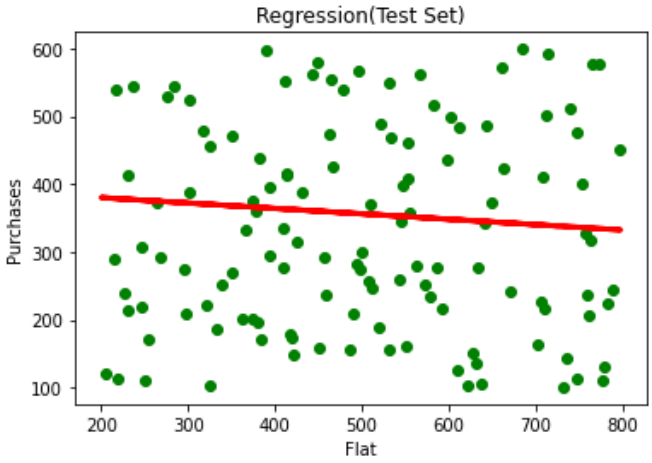
```
# Here we split our 'X' and 'y' dataset into 'X_train', 'X_test' and 'y_tr
ain', 'y_test'.
# Here we take 25% data as test dataset and remaining as train dataset. We
 take the random_state value as 15 for our better prediction.

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.25,rand
om_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_t
est
plt.plot(X_train,regressor.predict(X_train),color="red",linewidth=3) # Reg
ressior line showing
plt.title('Regression(Test Set)')
plt.xlabel('Flat')
plt.ylabel('Purchases')
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) # Reg
ressior line showing
plt.title('Regression(training Set)')
plt.xlabel('Flat')
plt.ylabel('Purchases')
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
```

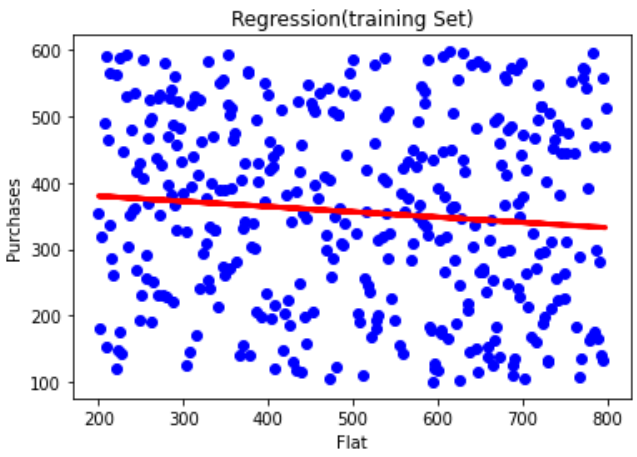
Output :-

```
ID  Flat  Houses  Purchases
0      4   225     255        175
1    344   506     605        203
2    402   774     467        577
3    407   719     453        495
4    456   269     429        291
..   ...   ...     ...        ...
495   88   446     488        158
496  461   761     187        446
497  204   319     616        241
```

```
498   209    325      419         455
499   174    385      846         302

[500 rows x 4 columns]
(500, 1)
(500, 1)
```







R2 score: -0.04

Mean Error : 22555.74264927693