Name: Saloni Vishwakarma

Roll No: C1-13

Multiple Regression

In [35]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing Dataset

In [36]:
```
dataset=pd.read_csv("Advertising_Sales.csv")
x=dataset.iloc[:,:-1].values
y=dataset.iloc[:, -1].values
```

In [37]:
```
dataset.describe()
```

Out[37]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

In [38]:
```
dataset.drop(['Unnamed: 0'],axis=1,inplace=True)
```

In [39]:
```
dataset.head()
```

Out[39]:

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

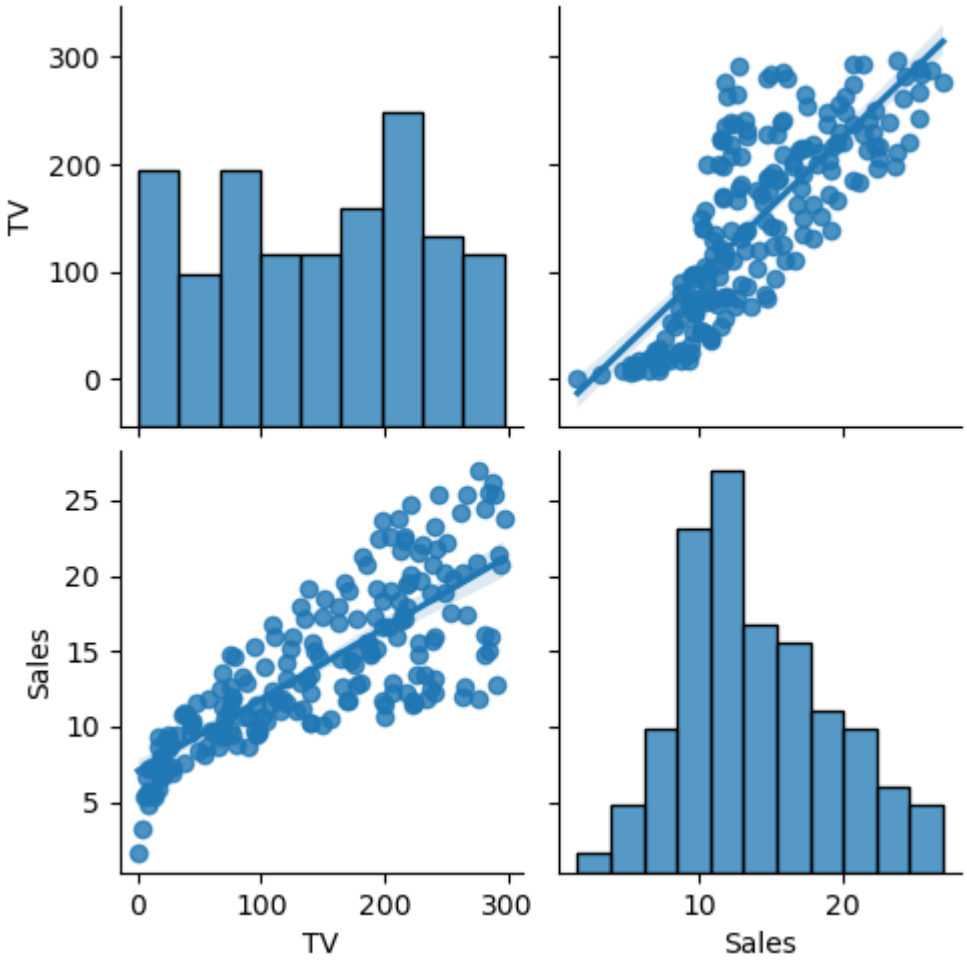
In [40]:
```
dataset.shape
```

Out[40]:
```
(200, 4)
```

In [41]:
```
import seaborn as sns
```

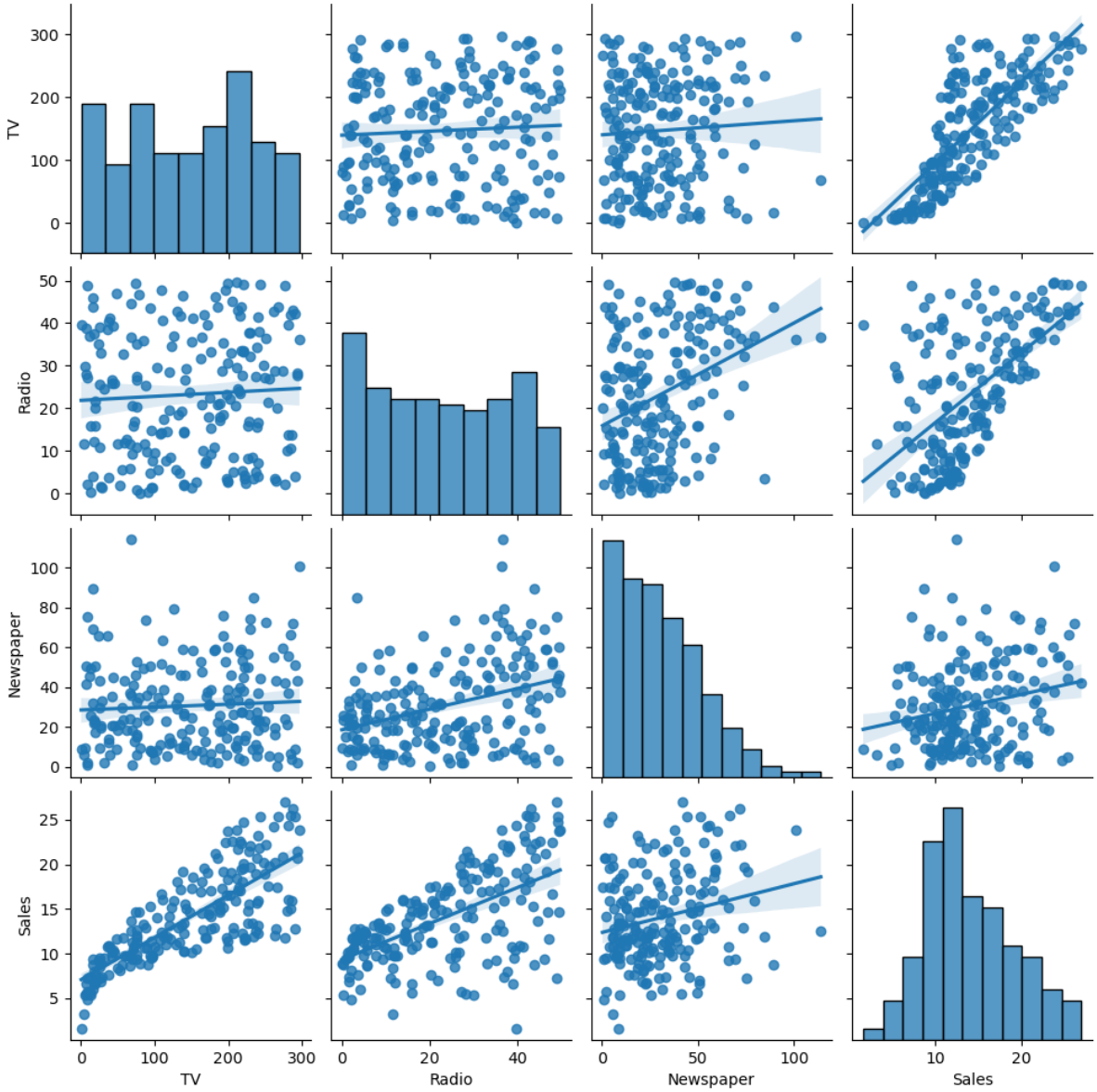
In [42]:
```
sns.pairplot(dataset,vars=['TV','Sales'],kind='reg')
```

Out[42]:
```
<seaborn.axisgrid.PairGrid at 0x19058a84250>
```



In [43]:
```
sns.pairplot(dataset,kind='reg')
```

Out[43]:
```
<seaborn.axisgrid.PairGrid at 0x190591d5b90>
```



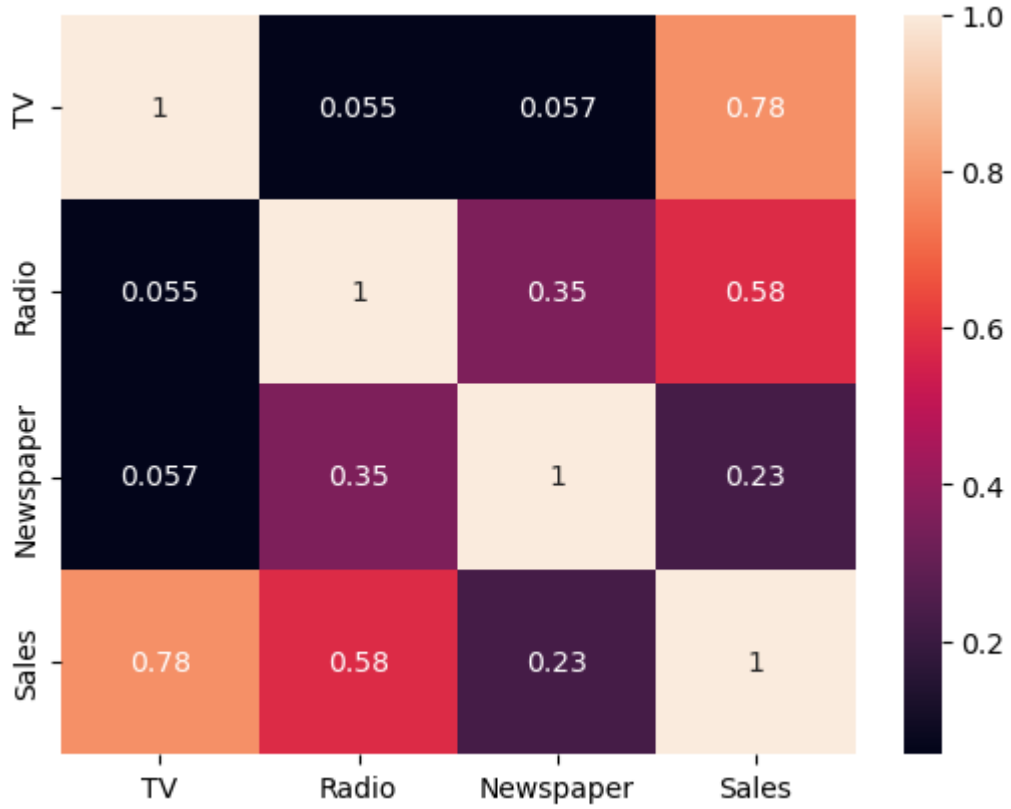
In [44]:
```
dataset.corr()
```

Out[44]:

	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.782224
Radio	0.054809	1.000000	0.354104	0.576223
Newspaper	0.056648	0.354104	1.000000	0.228299
Sales	0.782224	0.576223	0.228299	1.000000

In [45]:
```
sns.heatmap(dataset.corr(),annot=True)
```

Out[45]:
```
<Axes: >
```



In [46]:
```
x=dataset[['TV','Radio','Newspaper']]
```

In [47]:
```
print(x)
```

```
        TV  Radio  Newspaper
0    230.1   37.8       69.2
1     44.5   39.3       45.1
2     17.2   45.9       69.3
3    151.5   41.3       58.5
4    180.8   10.8       58.4
..     ...    ...        ...
195   38.2    3.7       13.8
196   94.2    4.9        8.1
197  177.0    9.3        6.4
198  283.6   42.0       66.2
199  232.1    8.6        8.7

[200 rows x 3 columns]
```

In [48]:
```
y=dataset.Sales
```

In [49]:
```
y
```

Out[49]:
```
0      22.1
1      10.4
2       9.3
3      18.5
4      12.9
       ...
195     7.6
196     9.7
197    12.8
198    25.5
199    13.4
Name: Sales, Length: 200, dtype: float64
```

Splitting the dataset into Training Set and Test Set

In [50]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=1)
```

In [51]:
```
print(x_train)
```

```
        TV  Radio  Newspaper
108   13.1    0.4       25.6
107   90.4    0.3       23.2
189   18.7   12.1       23.4
14   204.1   32.9       46.0
56     7.3   28.1       41.4
..     ...    ...        ...
133  219.8   33.5       45.1
137  273.7   28.9       59.7
72    26.8   33.0       19.3
140   73.4   17.0       12.9
37    74.7   49.4       45.7

[160 rows x 3 columns]
```

In [52]:
```
print("x_train")
print(x_train)
print("x_test")
print(x_test)
```

```
x_train
        TV  Radio  Newspaper
108   13.1    0.4       25.6
107   90.4    0.3       23.2
189   18.7   12.1       23.4
14   204.1   32.9       46.0
56     7.3   28.1       41.4
..     ...    ...        ...
133  219.8   33.5       45.1
137  273.7   28.9       59.7
72    26.8   33.0       19.3
140   73.4   17.0       12.9
37    74.7   49.4       45.7

[160 rows x 3 columns]
x_test
        TV  Radio  Newspaper
58   210.8   49.6       37.7
40   202.5   22.3       31.6
34    95.7    1.4        7.4
102  280.2   10.1       21.4
184  253.8   21.3       30.0
198  283.6   42.0       66.2
95   163.3   31.6       52.9
4    180.8   10.8       58.4
29    70.6   16.0       40.8
168  215.4   23.6       57.6
171  164.5   20.9       47.4
18    69.2   20.5       18.3
11   214.7   24.0        4.0
89   109.8   47.8       51.4
110  225.8    8.2       56.5
118  125.7   36.9       79.2
159  131.7   18.4       34.6
35   290.7    4.1        8.5
136   25.6   39.0        9.3
59   210.7   29.5        9.3
51   100.4    9.6        3.6
16    67.8   36.6      114.0
44    25.1   25.7       43.3
94   107.4   14.0       10.9
31   112.9   17.4       38.6
162  188.4   18.1       25.6
38    43.1   26.7       35.1
28   248.8   27.1       22.9
193  166.8   42.0        3.6
27   240.1   16.7       22.9
47   239.9   41.5       18.5
165  234.5    3.4       84.8
194  149.7   35.6        6.0
177  170.2    7.8       35.2
176  248.4   30.2       20.3
97   184.9   21.0       22.0
174  222.4    3.4       13.1
73   129.4    5.7       31.3
69   216.8   43.9       27.2
172   19.6   20.1       17.0
```

In [53]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(160, 3)
(40, 3)
(160,)
(40,)
```

In [54]:
```
print("x:",x)
```

```
x:          TV  Radio  Newspaper
0    230.1   37.8       69.2
1     44.5   39.3       45.1
2     17.2   45.9       69.3
3    151.5   41.3       58.5
4    180.8   10.8       58.4
..     ...    ...        ...
195   38.2    3.7       13.8
196   94.2    4.9        8.1
197  177.0    9.3        6.4
198  283.6   42.0       66.2
199  232.1    8.6        8.7

[200 rows x 3 columns]
```

In [55]:
```
print("x_train:",x_train)
print("x_test:",x_test)
```

```
x_train:         TV  Radio  Newspaper
108   13.1    0.4       25.6
107   90.4    0.3       23.2
189   18.7   12.1       23.4
14   204.1   32.9       46.0
56     7.3   28.1       41.4
..     ...    ...        ...
133  219.8   33.5       45.1
137  273.7   28.9       59.7
72    26.8   33.0       19.3
140   73.4   17.0       12.9
37    74.7   49.4       45.7

[160 rows x 3 columns]
x_test:         TV  Radio  Newspaper
58   210.8   49.6       37.7
40   202.5   22.3       31.6
34    95.7    1.4        7.4
102  280.2   10.1       21.4
184  253.8   21.3       30.0
198  283.6   42.0       66.2
95   163.3   31.6       52.9
4    180.8   10.8       58.4
29    70.6   16.0       40.8
168  215.4   23.6       57.6
171  164.5   20.9       47.4
18    69.2   20.5       18.3
11   214.7   24.0        4.0
89   109.8   47.8       51.4
110  225.8    8.2       56.5
118  125.7   36.9       79.2
159  131.7   18.4       34.6
35   290.7    4.1        8.5
136   25.6   39.0        9.3
59   210.7   29.5        9.3
51   100.4    9.6        3.6
16    67.8   36.6      114.0
44    25.1   25.7       43.3
94   107.4   14.0       10.9
31   112.9   17.4       38.6
162  188.4   18.1       25.6
38    43.1   26.7       35.1
28   248.8   27.1       22.9
193  166.8   42.0        3.6
27   240.1   16.7       22.9
47   239.9   41.5       18.5
165  234.5    3.4       84.8
194  149.7   35.6        6.0
177  170.2    7.8       35.2
176  248.4   30.2       20.3
97   184.9   21.0       22.0
174  222.4    3.4       13.1
73   129.4    5.7       31.3
69   216.8   43.9       27.2
172   19.6   20.1       17.0
```

In [56]:
```
from sklearn.preprocessing import StandardScaler
st=StandardScaler()
x_train=st.fit_transform(x_train)
x_test=st.transform(x_test)
```

In [57]:
```
print(x_train)
```

```
[[-1.47861420e+00 -1.50740896e+00 -2.07096465e-01]
 [-5.93912087e-01 -1.51399099e+00 -3.21290955e-01]
 [-1.41452194e+00 -7.37311102e-01 -3.11774748e-01]
 [ 7.07389725e-01  6.31751749e-01  7.63556698e-01]
 [-1.54499547e+00  3.15814168e-01  5.44683926e-01]
 [ 5.73110715e-02  3.93687845e-02 -5.16373209e-01]
 [-7.33541657e-01  1.54665433e+00  2.16374768e-01]
 [ 9.80926342e-01  9.47689330e-01  9.74221741e-02]
 [ 7.17690268e-01  1.43475977e+00 -4.92582690e-01]
 [ 9.98093912e-01  5.92259551e-01  2.10534195e+00]
 [ 1.68708573e+00  1.25046284e+00  1.01097809e+00]
 [-2.19659059e-01  7.43646309e-01 -8.35166159e-01]
 [-1.40651040e+00 -4.80611817e-01 -3.64113889e-01]
 [ 4.61321221e-01  1.50716213e+00  1.36783587e+00]
 [ 1.41126010e+00 -2.17330500e-01 -1.41089671e+00]
 [ 2.66755427e-01 -8.75533794e-01 -5.87744765e-01]
 [ 8.72198398e-01 -1.17830731e+00 -1.21450598e-01]
 [-1.34241814e+00 -8.09713464e-01 -1.20142119e-02]
 [ 1.02785103e+00 -9.67682255e-01 -1.01121600e+00]
 [-1.13640730e+00  1.71009443e-01 -4.49759756e-01]
 [-3.53938070e-02 -5.92506377e-01 -2.07096465e-01]
 [-3.42493023e-02 -5.79342311e-01 -9.39844442e-01]
 [-1.56674106e+00  4.34290761e-01 -9.77909272e-01]
 [ 1.73172141e+00  2.89486036e-01 -1.33952516e+00]
 [ 1.59209184e+00  1.07274796e+00  1.22985086e+00]
 [-1.52897240e+00  1.68487702e+00  2.14340678e+00]
 [-7.10651564e-01 -1.53373709e+00 -9.87425479e-01]
 [-5.02723677e-02  1.52032619e+00  1.38211018e+00]
 [ 9.71770305e-01 -4.93775883e-01  9.49122744e-01]
 [-1.26802534e+00  8.54430150e-02 -1.32049274e+00]
 [ 1.37806946e+00  3.61888398e-01 -6.68632528e-01]
 [ 8.14973165e-01  1.29653708e+00  1.83068041e-01]
 [ 1.66305113e+00  1.29653708e+00  1.99114746e+00]
 [-1.49463726e+00  8.95033066e-01  7.25491868e-01]
 [-1.42367797e+00  9.41107297e-01 -3.97420615e-01]
 [ 7.70337482e-01 -1.77838302e-01 -9.16053923e-01]
 [-1.35615220e+00  7.76556473e-01  1.71041934e+00]
 [-7.54142741e-01 -1.48108082e+00 -7.20971670e-01]
 [ 1.24301791e+00  8.68704934e-01  2.01493798e+00]
 [-1.13592089e-02  2.30247740e-01  7.73072905e-01]
 [ 5.20835464e-01 -1.44928138e-01 -9.73151168e-01]
 [-9.14745359e-02 -1.21121747e+00 -9.82667375e-01]
 [ 1.11597789e+00 -1.26387374e+00  3.30569257e-01]
 [ 9.35146155e-01 -1.37576830e+00 -6.82906840e-01]
 [-7.64443283e-01 -8.22877530e-01 -1.13968480e+00]
 [-5.12652255e-01 -1.03350258e+00 -1.08258755e+00]
 [-7.66732293e-01 -1.97584401e-01  1.21212693e-01]
 [-1.53240592e+00  2.56575871e-01 -1.32525084e+00]
 [ 5.58604118e-01  3.55306365e-01 -5.59196142e-01]
 [ 3.75483370e-01 -5.27796766e-02  7.36316554e-02]
 [-8.11739939e-02  1.21097065e+00  7.58798594e-01]
 [ 1.42613866e+00  1.34919334e+00 -1.18726584e+00]
 [-6.47703807e-01  8.22630704e-01  9.20574121e-01]
 [-3.19602930e-02 -1.39551440e+00 -1.59515428e-01]
 [ 3.31992193e-01  1.07932999e+00  3.68634087e-01]
 [ 5.21979969e-01 -4.01627422e-01 -5.73470454e-01]
 [-1.19134353e+00 -1.29020187e+00 -7.68552707e-01]
 [ 3.97228959e-01 -9.21608024e-01 -1.12065238e+00]
 [-1.19363254e+00  1.11882219e+00 -8.58956678e-01]
 [ 2.98801557e-01 -1.06641275e+00 -8.16133744e-01]
 [-1.43512302e+00  1.34261131e+00  2.82857372e+00]
 [ 1.08850978e+00 -1.19805341e+00 -3.07016644e-01]
 [-1.30121597e+00 -1.43500659e+00  1.45003212e-01]
 [ 6.32996922e-01 -1.23413118e-04 -7.49520292e-01]
```

```
[-1.43168951e+00  1.48741603e+00  1.87219487e+00]
[ 6.50164492e-01  4.80364991e-01  4.16215125e-01]
[ 5.82638716e-01 -3.22643027e-01  1.70090314e+00]
[ 1.58408030e+00 -6.18834509e-01  3.35327361e-01]
[-9.70454124e-01  6.25169716e-01 -3.07016644e-01]
[ 3.82350398e-01 -5.20104015e-01 -1.31097653e+00]
[ 1.00496094e+00  9.54271363e-01  1.86743677e+00]
[-1.58161962e+00 -7.70221267e-01 -1.15395911e+00]
[-2.49416181e-01  3.42142300e-01 -7.49520292e-01]
[-8.38836087e-01 -9.21608024e-01 -1.38234809e+00]
[ 6.47875482e-01  1.71778718e+00  1.42969122e+00]
[ 7.38291352e-01 -1.19147137e+00 -5.02098897e-01]
[-6.17946685e-01  1.44681311e-01  2.06727712e+00]
[-1.62053278e+00  1.07274796e+00 -1.01121600e+00]
[-6.01923620e-01 -8.82115826e-01  2.73472013e-01]
[ 1.11025537e+00 -5.13521982e-01 -1.26208702e-01]
[ 8.13828660e-01  8.54430150e-02 -8.01859433e-01]
[ 8.63042361e-01  6.71243946e-01  1.38211018e+00]
[ 1.15489105e+00  1.69145905e+00  6.82668934e-01]
[ 3.45726249e-01 -3.42389126e-01  3.55668255e-02]
[-4.31392424e-01 -1.15856121e+00  2.11616664e-01]
[-3.61577639e-01  1.13856828e+00  1.58195054e+00]
[ 1.63750006e-01 -1.36260423e+00 -1.03024841e+00]
[-2.28042556e-02 -1.40867846e+00 -9.96941687e-01]
[-2.87184835e-01 -5.66178246e-01 -1.16823342e+00]
[-8.72026723e-01 -1.15197918e+00 -2.73709918e-01]
[-5.16085769e-01 -1.43500659e+00  2.26009928e-03]
[-8.45703115e-01  1.39526757e+00  2.68713909e-01]
[-9.85332685e-01 -1.15856121e+00 -1.20142119e-02]
[ 9.16834081e-01 -1.25070967e+00  9.44364640e-01]
[-6.30536237e-01 -7.57057201e-01 -1.92822154e-01]
[ 9.84359856e-01 -4.21373521e-01 -1.78547843e-01]
[-1.53927295e+00  1.02667372e+00  9.82429470e-01]
[ 6.58176025e-01 -1.36260423e+00 -4.16453030e-01]
[-1.05629197e+00 -7.70221267e-01 -5.49679935e-01]
[ 1.09766582e+00  7.23900210e-01 -1.17299152e+00]
[-2.43693657e-01 -9.80846321e-01  8.92025499e-01]
[-1.07574855e+00  1.55981839e+00 -1.02073221e+00]
[ 1.13772348e+00  9.67435428e-01 -3.21290955e-01]
[ 1.64473905e+00 -6.18834509e-01 -1.24912118e+00]
[-1.28098685e-01  1.28337301e+00 -5.00790419e-02]
[ 6.07817819e-01  1.60589262e+00  1.09186586e+00]
[ 3.97228959e-01  6.64661913e-01  4.16215125e-01]
[ 2.42720828e-01  8.88451033e-01 -1.07307135e+00]
[ 1.29452062e+00  2.36829773e-01 -1.16347532e+00]
[-7.54142741e-01  2.23665707e-01 -3.64113889e-01]
[ 8.48163800e-01  1.21097065e+00  4.59038059e-01]
[ 6.95286589e-03  3.94798563e-01 -8.25649952e-01]
[-1.31380553e+00 -1.42842456e+00 -4.40243549e-01]
[ 7.39435856e-01 -9.80846321e-01 -1.69031635e-01]
[-9.15517900e-01 -7.04400937e-01 -5.54438039e-01]
[ 1.05380268e-01  1.18464252e+00  1.35831967e+00]
[ 1.62528247e+00 -8.36041596e-01 -1.12065238e+00]
[-8.62870685e-01 -7.63639234e-01  3.25811154e-01]
[-1.11695072e+00  1.64427410e-01 -4.45001652e-01]
[ 1.40668208e+00 -1.34285813e+00  6.20813586e-01]
[-1.17646497e+00  1.17147845e+00 -1.14920101e+00]
[ 1.72370987e+00  3.28978234e-01  6.30329793e-01]
[-1.41337743e+00 -1.05435940e-01  9.72913263e-01]
[-7.69021302e-01  7.69974440e-01  1.08234965e+00]
[ 1.54058913e+00  1.68487702e+00  5.63716341e-01]
[-1.53011691e+00 -1.39551440e+00 -1.37758999e+00]
[-1.01623431e+00 -1.40209643e+00 -4.06936823e-01]
[ 8.92799482e-01  1.69145905e+00 -1.27291170e+00]
```

```
[-1.11923973e+00  1.05300186e+00  7.20733764e-01]
[-6.97289472e-02 -2.69986763e-01 -6.35325802e-01]
[-1.47746969e+00 -4.87193850e-01  9.34848433e-01]
[ 1.36204640e+00  1.27679098e+00  1.17751172e+00]
[-4.53138012e-01  4.14544662e-01 -1.02549031e+00]
[-2.52849695e-01 -2.43658632e-01 -8.73230989e-01]
[-5.50420910e-01 -1.21121747e+00 -1.03976462e+00]
[ 5.88361240e-01  7.96302572e-01  2.17195541e+00]
[-7.55287246e-01  2.76321970e-01 -6.63874425e-01]
[ 6.32996922e-01 -1.30336593e+00 -1.14444290e+00]
[ 1.08850978e+00  2.76321970e-01 -9.01779612e-01]
[ 6.58176025e-01 -1.32969407e+00  2.21132871e-01]
[ 8.95088492e-01  6.51497847e-01  3.78150295e-01]
[ 8.59236883e-02 -1.44817066e+00 -2.68951814e-01]
[-5.53854424e-01  1.32944724e+00  9.77671366e-01]
[ 4.87644828e-01  1.35577537e+00 -1.34428326e+00]
[ 1.11941141e+00 -1.05324868e+00 -1.01121600e+00]
[ 1.76376754e+00  8.55540868e-01  3.37575565e+00]
[ 8.71053894e-01  2.89486036e-01  1.11565637e+00]
[ 1.53830012e+00 -1.38235033e+00 -2.97500436e-01]
[ 1.38035847e+00 -1.30336593e+00 -4.97340794e-01]
[-1.20622209e+00  1.00692763e+00  1.69614503e+00]
[-3.71878181e-01 -5.92506377e-01  8.31478629e-02]
[-9.46419526e-01 -7.43893135e-01  6.25571689e-01]
[-5.27530816e-01 -5.59596213e-01  4.25731332e-01]
[-1.43168951e+00 -1.26387374e+00  7.83897591e-02]
[-3.00918891e-01 -1.02692055e+00 -3.26049059e-01]
[ 8.87076959e-01  6.71243946e-01  7.20733764e-01]
[ 1.50396498e+00  3.68470431e-01  1.41541691e+00]
[-1.32181706e+00  6.38333782e-01 -5.06857001e-01]
[-7.88477881e-01 -4.14791488e-01 -8.11375641e-01]
[-7.73599321e-01  1.71778718e+00  7.49282387e-01]]
```

In [58]:
```
print(x_test)
```

```
[[ 7.84071539e-01  1.73095125e+00  3.68634087e-01]
 [ 6.89077651e-01 -6.59437425e-02  7.83897591e-02]
 [-5.33253340e-01 -1.44158863e+00 -1.07307135e+00]
 [ 1.57835778e+00 -8.68951761e-01 -4.06936823e-01]
 [ 1.27620855e+00 -1.31764072e-01  2.26009928e-03]
 [ 1.61727094e+00  1.23071675e+00  1.72469365e+00]
 [ 2.40431819e-01  5.46185321e-01  1.09186586e+00]
 [ 4.40720137e-01 -8.22877530e-01  1.35356156e+00]
 [-8.20524012e-01 -4.80611817e-01  5.16135303e-01]
 [ 8.36718753e-01  1.96226857e-02  1.31549673e+00]
 [ 2.54165875e-01 -1.58092204e-01  8.30170150e-01]
 [-8.36547078e-01 -1.84420335e-01 -5.54438039e-01]
 [ 8.28707221e-01  4.59508174e-02 -1.23484687e+00]
 [-3.71878181e-01  1.61247466e+00  1.02049430e+00]
 [ 9.55747239e-01 -9.94010386e-01  1.26315759e+00]
 [-1.89901938e-01  8.95033066e-01  2.34324714e+00]
 [-1.21231657e-01 -3.22643027e-01  2.21132871e-01]
 [ 1.69853077e+00 -1.26387374e+00 -1.02073221e+00]
 [-1.33555111e+00  1.03325576e+00 -9.82667375e-01]
 [ 7.82927034e-01  4.07962629e-01 -9.82667375e-01]
 [-4.79461620e-01 -9.01861925e-01 -1.25387929e+00]
 [-8.52570143e-01  8.75286967e-01  3.99906724e+00]
 [-1.34127364e+00  1.57845377e-01  6.35087897e-01]
 [-3.99346293e-01 -6.12252476e-01 -9.06537715e-01]
 [-3.36398536e-01 -3.88463356e-01  4.11457021e-01]
 [ 5.27702492e-01 -3.42389126e-01 -2.07096465e-01]
 [-1.13526280e+00  2.23665707e-01  2.44923390e-01]
 [ 1.21898331e+00  2.49993838e-01 -3.35565266e-01]
 [ 2.80489483e-01  1.23071675e+00 -1.25387929e+00]
 [ 1.11941141e+00 -4.34537587e-01 -3.35565266e-01]
 [ 1.11712240e+00  1.19780658e+00 -5.44921831e-01]
 [ 1.05531915e+00 -1.30994797e+00  2.60970095e+00]
 [ 8.47791836e-02  8.09466638e-01 -1.13968480e+00]
 [ 3.19402641e-01 -1.02033852e+00  2.49681494e-01]
 [ 1.21440530e+00  4.54036859e-01 -4.59275964e-01]
 [ 4.87644828e-01 -1.51510171e-01 -3.78388200e-01]
 [ 9.16834081e-01 -1.30994797e+00 -8.01859433e-01]
 [-1.47555265e-01 -1.15856121e+00  6.41154479e-02]
 [ 8.52741819e-01  1.35577537e+00 -1.30966805e-01]
 [-1.40422140e+00 -2.10748467e-01 -6.16293387e-01]]
```

In [59]:
```
from sklearn.linear_model import LinearRegression
linreg=LinearRegression()
linreg.fit(x_train,y_train)
```

Out[59]:
```
▾ LinearRegression

LinearRegression()
```

In [60]:
```
linreg.intercept_
```

Out[60]:
```
13.811250000000001
```

In [61]:
```
linreg.coef_
```

Out[61]:
```
array([4.09287129, 2.7126018 , 0.05435329])
```

In [62]:
```
y_pred=linreg.predict(x_test)
```

In [63]:
```
y_pred
```

Out[63]:
```
array([21.73577184, 16.45693776,  7.65993185, 17.89202679, 18.67730671,
       23.86271904, 16.33623628, 13.45649226,  9.177296  , 17.36056228,
       14.4677995 ,  9.85697601, 17.26057027, 16.71866935, 15.09530285,
       15.58923732, 12.45188167, 17.27925151, 11.0944114 , 18.06889853,
        9.33433055, 12.91345761,  8.7842804 , 10.46670654, 11.40303174,
       15.03104665,  9.78479388, 19.46028647, 18.22954934, 17.1958903 ,
       21.60304218, 14.71901407, 16.29205532, 12.36432281, 19.98831261,
       15.37556411, 13.96678297, 10.06809496, 20.97197274,  7.45877832])
```

In [64]:
```
df=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
print(df)
```

```
     Actual  Predicted
58     23.8  21.735772
40     16.6  16.456938
34      9.5   7.659932
102    14.8  17.892027
184    17.6  18.677307
198    25.5  23.862719
95     16.9  16.336236
4      12.9  13.456492
29     10.5   9.177296
168    17.1  17.360562
171    14.5  14.467799
18     11.3   9.856976
11     17.4  17.260570
89     16.7  16.718669
110    13.4  15.095303
118    15.9  15.589237
159    12.9  12.451882
35     12.8  17.279252
136     9.5  11.094411
59     18.4  18.068899
51     10.7   9.334331
16     12.5  12.913458
44      8.5   8.784280
94     11.5  10.466707
31     11.9  11.403032
162    14.9  15.031047
38     10.1   9.784794
28     18.9  19.460286
193    19.6  18.229549
27     15.9  17.195890
47     23.2  21.603042
165    11.9  14.719014
194    17.3  16.292055
177    11.7  12.364323
176    20.2  19.988313
97     15.5  15.375564
174    11.5  13.966783
73     11.0  10.068095
69     22.3  20.971973
172     7.6   7.458778
```

In [65]:
```
from sklearn import metrics
print('Mean Absolute Error: ',metrics.mean_absolute_error(y_test,y_pred))
```

```
Mean Absolute Error:  1.0402154012924714
```

In [66]:
```
print('Mean Squared Error: ',metrics.mean_squared_error(y_test,y_pred))
```

```
Mean Squared Error:  1.9918855518287883
```

In [67]:
```
print('Root Mean Square Error: ',np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
Root Mean Square Error:  1.411341755858158
```

In [68]:
```
metrics.r2_score(y_test,y_pred)
```

Out[68]:
```
0.8927605914615385
```