```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Datasets/Advertising.csv')
df
```

	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	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

200 rows × 4 columns

```
df.shape
```

```
(200, 4)
```

```
df.describe
```

```
<bound method NDFrame.describe of         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   12.0
3    151.5   41.3       58.5   16.5
4    180.8   10.8       58.4   17.9
..     ...    ...        ...    ...
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1   14.0
197  177.0    9.3        6.4   14.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   18.4

[200 rows x 4 columns]>
```

```
df.isna().sum()
```

```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

```
x=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
x
```

```
array([[230.1,  37.8,  69.2],
       [ 44.5,  39.3,  45.1],
       [ 17.2,  45.9,  69.3],
       [151.5,  41.3,  58.5],
       [180.8,  10.8,  58.4],
       [  8.7,  48.9,  75. ],
       [ 57.5,  32.8,  23.5],
       [120.2,  19.6,  11.6],
       [  8.6,   2.1,   1. ],
       [199.8,   2.6,  21.2],
       [ 66.1,   5.8,  24.2],
       [214.7,  24. ,   4. ],
```

```
            [ 23.8,  35.1,  65.9],
            [ 97.5,   7.6,   7.2],
            [204.1,  32.9,  46. ],
            [195.4,  47.7,  52.9],
            [ 67.8,  36.6, 114. ],
            [281.4,  39.6,  55.8],
            [ 69.2,  20.5,  18.3],
            [147.3,  23.9,  19.1],
            [218.4,  27.7,  53.4],
            [237.4,   5.1,  23.5],
            [ 13.2,  15.9,  49.6],
            [228.3,  16.9,  26.2],
            [ 62.3,  12.6,  18.3],
            [262.9,   3.5,  19.5],
            [142.9,  29.3,  12.6],
            [240.1,  16.7,  22.9],
            [248.8,  27.1,  22.9],
            [ 70.6,  16. ,  40.8],
            [292.9,  28.3,  43.2],
            [112.9,  17.4,  38.6],
            [ 97.2,   1.5,  30. ],
            [265.6,  20. ,   0.3],
            [ 95.7,   1.4,   7.4],
            [290.7,   4.1,   8.5],
            [266.9,  43.8,   5. ],
            [ 74.7,  49.4,  45.7],
            [ 43.1,  26.7,  35.1],
            [228. ,  37.7,  32. ],
            [202.5,  22.3,  31.6],
            [177. ,  33.4,  38.7],
            [293.6,  27.7,   1.8],
            [206.9,   8.4,  26.4],
            [ 25.1,  25.7,  43.3],
            [175.1,  22.5,  31.5],
            [ 89.7,   9.9,  35.7],
            [239.9,  41.5,  18.5],
            [227.2,  15.8,  49.9],
            [ 66.9,  11.7,  36.8],
            [199.8,   3.1,  34.6],
            [100.4,   9.6,   3.6],
            [216.4,  41.7,  39.6],
            [182.6,  46.2,  58.7],
            [262.7,  28.8,  15.9],
            [198.9,  49.4,  60. ],
            [  7.3,  28.1,  41.4],
            [136.2,  19.2,  16.6],
```

```
y
```

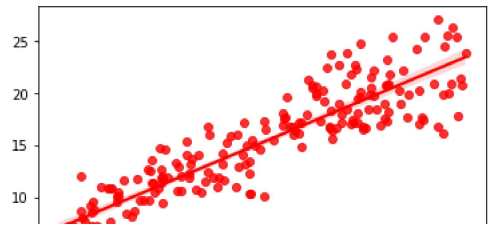
```
array([22.1, 10.4, 12. , 16.5, 17.9,  7.2, 11.8, 13.2,  4.8, 15.6, 12.6,
       17.4,  9.2, 13.7, 19. , 22.4, 12.5, 24.4, 11.3, 14.6, 18. , 17.5,
        5.6, 20.5,  9.7, 17. , 15. , 20.9, 18.9, 10.5, 21.4, 11.9, 13.2,
       17.4, 11.9, 17.8, 25.4, 14.7, 10.1, 21.5, 16.6, 17.1, 20.7, 17.9,
        8.5, 16.1, 10.6, 23.2, 19.8,  9.7, 16.4, 10.7, 22.6, 21.2, 20.2,
       23.7,  5.5, 13.2, 23.8, 18.4,  8.1, 24.2, 20.7, 14. , 16. , 11.3,
       11. , 13.4, 18.9, 22.3, 18.3, 12.4,  8.8, 11. , 17. ,  8.7,  6.9,
       14.2,  5.3, 11. , 11.8, 17.3, 11.3, 13.6, 21.7, 20.2, 12. , 16. ,
       12.9, 16.7, 14. ,  7.3, 19.4, 22.2, 11.5, 16.9, 16.7, 20.5, 25.4,
       17.2, 16.7, 23.8, 19.8, 19.7, 20.7, 15. ,  7.2, 12. ,  5.3, 19.8,
       18.4, 21.8, 17.1, 20.9, 14.6, 12.6, 12.2,  9.4, 15.9,  6.6, 15.5,
        7. , 16.6, 15.2, 19.7, 10.6,  6.6, 11.9, 24.7,  9.7,  1.6, 17.7,
        5.7, 19.6, 10.8, 11.6,  9.5, 20.8,  9.6, 20.7, 10.9, 19.2, 20.1,
       10.4, 12.3, 10.3, 18.2, 25.4, 10.9, 10.1, 16.1, 11.6, 16.6, 16. ,
       20.6,  3.2, 15.3, 10.1,  7.3, 12.9, 16.4, 13.3, 19.9, 18. , 11.9,
       16.9,  8. , 17.2, 17.1, 20. ,  8.4, 17.5,  7.6, 16.7, 16.5, 27. ,
       20.2, 16.7, 16.8, 17.6, 15.5, 17.2,  8.7, 26.2, 17.6, 22.6, 10.3,
       17.3, 20.9,  6.7, 10.8, 11.9,  5.9, 19.6, 17.3,  7.6, 14. , 14.8,
       25.5, 18.4])
```

```
df.columns
```

```
Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')
```

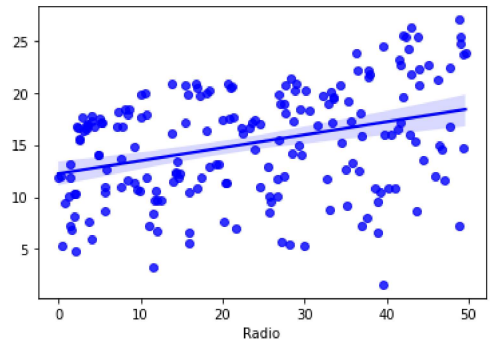
```
#Regression Plot Tv
sns.regplot(x=df['TV'],y=y,color='red')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1393092290>
```



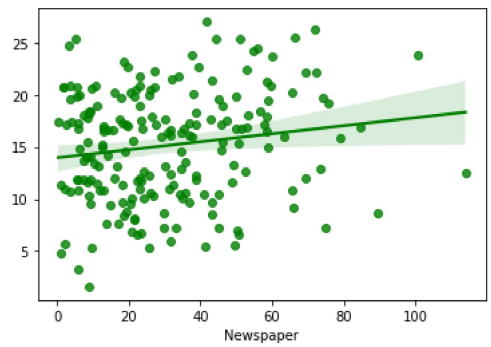
```
sns.regplot(x=df['Radio'],y=y,color='blue')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1392fd4f50>
```




```
sns.regplot(x=df['Newspaper'],y=y,color='green')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1392b0f910>
```




```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=1)
x_train
```

```
array([[ 13.1,   0.4,  25.6],
       [ 90.4,   0.3,  23.2],
       [ 18.7,  12.1,  23.4],
       [204.1,  32.9,  46. ],
       [  7.3,  28.1,  41.4],
       [147.3,  23.9,  19.1],
       [ 78.2,  46.8,  34.5],
       [228. ,  37.7,  32. ],
       [205. ,  45.1,  19.6],
       [229.5,  32.3,  74.2],
       [289.7,  42.3,  51.2],
       [123.1,  34.6,  12.4],
       [ 19.4,  16. ,  22.3],
       [182.6,  46.2,  58.7],
       [265.6,  20. ,   0.3],
       [165.6,  10. ,  17.6],
       [218.5,   5.4,  27.4],
       [ 25. ,  11. ,  29.7],
       [232.1,   8.6,   8.7],
       [ 43. ,  25.9,  20.5],
       [139.2,  14.3,  25.6],
       [139.3,  14.5,  10.2],
       [  5.4,  29.9,   9.4],
       [293.6,  27.7,   1.8],
       [281.4,  39.6,  55.8],
       [  8.7,  48.9,  75. ],
       [ 80.2,   0. ,   9.2],
       [137.9,  46.4,  59. ],
```

```
            [227.2,  15.8,  49.9],
            [ 31.5,  24.6,   2.2],
            [262.7,  28.8,  15.9],
            [213.5,  43. ,  33.8],
            [287.6,  43. ,  71.8],
            [ 11.7,  36.9,  45.2],
            [ 17.9,  37.6,  21.6],
            [209.6,  20.6,  10.7],
            [ 23.8,  35.1,  65.9],
            [ 76.4,   0.8,  14.8],
            [250.9,  36.5,  72.3],
            [141.3,  26.8,  46.2],
            [187.8,  21.1,   9.5],
            [134.3,   4.9,   9.3],
            [239.8,   4.1,  36.9],
            [224. ,   2.4,  15.6],
            [ 75.5,  10.8,   6. ],
            [ 97.5,   7.6,   7.2],
            [ 75.3,  20.3,  32.5],
            [  8.4,  27.2,   2.1],
            [191.1,  28.7,  18.2],
            [175.1,  22.5,  31.5],
            [135.2,  41.7,  45.9],
            [266.9,  43.8,   5. ],
            [ 85.7,  35.8,  49.3],
            [139.5,   2.1,  26.6],
            [171.3,  39.7,  37.7],
            [187.9,  17.2,  17.9],
            [ 38.2,   3.7,  13.8],
            [177. ,   9.3,   6.4]]
```

```
y_train
```

```
array([ 5.3, 12. ,  6.7, 19. ,  5.5, 14.6, 14.6, 21.5, 22.6, 19.7, 25.4,
       15.2,  6.6, 21.2, 17.4, 17.6, 17.2,  7.2, 18.4,  9.6, 12.2, 13.4,
        5.3, 20.7, 24.4,  7.2, 11.9, 15. , 19.8, 11. , 20.2, 21.7, 26.2,
        7.3,  8. , 20.9,  9.2,  9.4, 22.2, 15.5, 20.6, 14. , 17.3, 16.6,
       11.9, 13.7, 11.3,  5.7, 17.3, 16.1, 17.2, 25.4, 13.3, 10.3, 16. ,
       19.7,  7.6, 14.8, 10.9, 16.7,  8.7, 17.5,  7.3, 16.6, 12. , 18.3,
       20.2, 16.1, 11.8, 17.1, 22.1,  3.2, 14.2, 11.3, 23.7, 17.2, 12.9,
        1.6, 10.6, 20.7, 17. , 19.4, 25.4, 16.4, 10.4, 16. , 15.5, 10.3,
       11.9, 12.6, 13.2, 13.6,  8.7, 16.7, 10.6, 20.5,  6.6, 15.6,  8.4,
       20.7, 11.6, 11.6, 21.8, 20.9, 16. , 22.4, 17.1, 18. , 19.8, 11.8,
       22.6, 15. ,  6.9, 17.9,  9.7, 16.5, 20. ,  9.7, 10.1, 17.7, 10.8,
       21.4,  7. , 12.6, 27. ,  4.8,  8.1, 24.7, 10.4, 13.2,  5.6, 24.2,
       14. , 13.2, 14. , 19.2, 12. , 16.7, 18.9, 16.4, 20.1, 10.1, 15.3,
       20.7, 18.2, 23.8, 18. , 16.8, 17. , 10.8, 12.4,  9.7, 12.3,  5.9,
       11. , 19.6, 20.8,  8.8, 10.9, 14.7])
```

```
#model creations
from sklearn.linear_model import LinearRegression
mlr=LinearRegression()
mlr.fit(x_train,y_train)
y_pred=mlr.predict(x_test)
y_pred
```

```
array([21.32727775, 18.06138419, 10.04630254, 21.0925422 , 20.78527508,
       24.52786989, 16.84180311, 15.656542  , 10.13878037, 18.88248026,
       15.80983753, 10.54583142, 18.93346094, 15.56643436, 17.86877073,
       15.29349959, 13.75707845, 21.06397901, 10.05959685, 19.27534125,
       11.15389873, 12.04216022,  8.63037961, 11.98644768, 12.61490963,
       16.85722247,  9.73227033, 21.11417665, 18.15109551, 19.56290183,
       22.11237483, 17.82764148, 16.54733981, 14.78435804, 21.41405363,
       16.96663966, 17.22580207, 12.32418381, 21.07962358,  7.77386767])
```

```
print('Intersept',mlr.intercept_)
print('slope',mlr.coef_)
var=['TV','Radio','Newspaper']
print(var,mlr.coef_)
```

```
Intersept 4.637624442397913
slope [ 0.05507865  0.10308563 -0.00090115]
['TV', 'Radio', 'Newspaper'] [ 0.05507865  0.10308563 -0.00090115]
```

```
df1=pd.DataFrame({"Actual_value":y_test,"Predicted_value":y_pred})
df1
```

	Actual_value	Predicted_value
0	23.8	21.327278
1	16.6	18.061384
2	11.9	10.046303
3	19.8	21.092542
4	17.6	20.785275
5	25.5	24.527870
6	16.9	16.841803
7	17.9	15.656542
8	10.5	10.138780
9	17.1	18.882480
10	17.5	15.809838
11	11.3	10.545831
12	17.4	18.933461
13	16.7	15.566434
14	18.4	17.868771
15	15.9	15.293500
16	12.9	13.757078
17	17.8	21.063979
18	9.5	10.059597
19	18.4	19.275341
20	10.7	11.153899
21	12.5	12.042160
22	8.5	8.630380
23	11.5	11.986448
24	11.9	12.614910
25	19.9	16.857222
26	10.1	9.732270
27	18.9	21.114177
28	19.6	18.151096

```
#Mean Absolute Error
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error
print("Mean Absolute Error is ",mean_absolute_error(y_test,y_pred))
print("Error Percentage ",mean_absolute_percentage_error(y_test,y_pred))
```

```
Mean Absolute Error is  1.275439091293969
Error Percentage  0.07544206161715512
```

```
#Mean Squared Error
from sklearn.metrics import mean_squared_error
print("MSE",mean_squared_error(y_test,y_pred))
```

```
MSE 2.4093336128923704
```

```
#Root Mean Squared Error
print("Root Mean Squared Error",np.sqrt(mean_squared_error(y_test,y_pred)))
```

```
Root Mean Squared Error 1.5522028259516765
```