Model creation of advertising.csv

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt #-->plotting visualization
#%matplotlib inline#-->spcl function
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df=pd.read_csv("advertising.csv")
df.head(10)
```

Out[2]:

	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
5	8.7	48.9	75.0	7.2
6	57.5	32.8	23.5	11.8
7	120.2	19.6	11.6	13.2
8	8.6	2.1	1.0	4.8
9	199.8	2.6	21.2	15.6

In [3]:

```
df.shape
```

Out[3]:

```
(200, 4)
```

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

In [5]:
```
df.describe()
```

Out[5]:

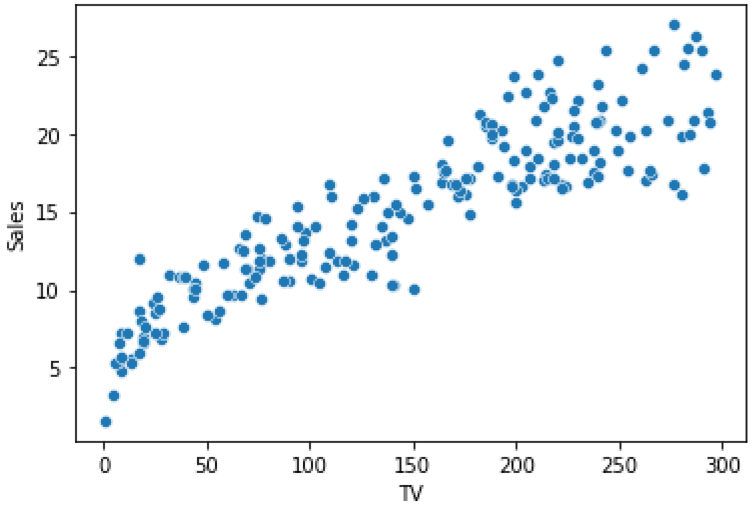
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

In [7]:
```
sns.scatterplot(x='TV',y='Sales',data=df)
```

Out[7]:
```
<AxesSubplot:xlabel='TV', ylabel='Sales'>
```



In [9]:
```
from sklearn.preprocessing import StandardScaler
```

In [10]:
```
se=StandardScaler()
```

```
In [12]: se.fit_transform(df[['TV']])
```

```
           [ 0.28325186],
           [ 0.47592034],
           [-1.66912209],
           [-0.62053847],
           [ 0.03219899],
           [-1.58037782],
           [-0.1791525 ],
           [ 0.29726411],
           [-0.71628887],
           [ 0.48292647],
           [ 0.19217221],
           [-0.34846722],
           [ 1.02123053],
           [-1.50798117],
           [ 0.69778102],

           [ 0.79820216],
           [ 1.60273904],
           [-1.1331534 ],
           [ 0.20384909],
           [ 1.48813948]
```

```
In [13]: from sklearn.preprocessing import MinMaxScaler
```

```
In [16]: min_max=MinMaxScaler()
```

```
In [19]: min_max.fit_transform(df[['TV']])
```

```
Out[19]: array([[0.77578627],
                [0.1481231 ],
                [0.0557998 ],
                [0.50997633],
                [0.60906324],
                [0.02705445],
                [0.19208657],
                [0.4041258 ],
                [0.02671627],
                [0.67331755],
                [0.2211701 ],
                [0.72370646],
                [0.07811972],
                [0.32735881],
                [0.68785932],
                [0.65843761],
                [0.22691917],
                [0.94927291],
                [0.2316537 ],
                [0.49577374]
```

```
In [20]: x=df.iloc[:,:-3]
```

In [21]:
```
x
```

Out[21]:

	TV
0	230.1
1	44.5
2	17.2
3	151.5
4	180.8
...	...
195	38.2
196	94.2
197	177.0
198	283.6
199	232.1

200 rows × 1 columns

In [22]:
```
y=df.iloc[:,-1]
```

In [23]:
```
y
```

Out[23]:
```
0      22.1
1      10.4
2      12.0
3      16.5
4      17.9
       ...
195     7.6
196    14.0
197    14.8
198    25.5
199    18.4
Name: Sales, Length: 200, dtype: float64
```

In [24]:
```
from sklearn.model_selection import train_test_split
```

In [25]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=12
```

In [26]:
```
x_train.shape
```

Out[26]:
```
(160, 1)
```

In [27]:

```
y_train.shape
```

Out[27]:

```
(160,)
```

In [28]:

```
x_test.shape
```

Out[28]:

```
(40, 1)
```

In [29]:

```
y_test.shape
```

Out[29]:

```
(40,)
```

Module building

In [30]:

```
from sklearn.linear_model import LinearRegression
```

In [31]:

```
reg=LinearRegression()
```

In [32]:

```
reg.fit(x_train,y_train)
```

Out[32]:

```
▾ LinearRegression
LinearRegression()
```

In [33]:

```
y_train_pred=reg.predict(x_train)
```

In [34]:

```
y_test_pred=reg.predict(x_test)
```

In [35]:
```
y_train_pred
```

Out[35]:
```
array([10.46165962, 22.26596879, 11.32103927, 21.44982837, 19.43379937,
       15.98547096, 16.6610839 , 19.11491007, 17.77449401, 13.02898877,
       14.6342451 , 19.29867679, 22.96320134, 14.73153336, 11.17510688,
       11.72640703, 10.66704595, 22.06058246, 13.60731344, 20.02833876,
       20.05536327, 11.98043749, 16.59082015, 14.67748432,  7.55922646,
       13.63433795, 17.24481347, 14.35319012, 21.52009211, 12.64524062,
       20.54180459, 12.36418564, 14.21266263, 11.22375101, 12.16960911,
        8.45103553,  8.02404816, 15.19095015,  7.13223908,  9.98602811,
       13.59109873, 22.92536701, 16.29355046,  7.54841665, 10.91026661,
       22.80645914, 19.53108764, 22.55242867, 16.55839073, 21.88762555,
       18.03933428, 18.48253636, 20.23912999, 11.94260317, 11.80748058,
        7.56463136, 10.75892931, 22.46054532, 10.82378815, 18.62846876,
       21.21741752,  9.15907788,  9.42391815, 20.07157798, 21.29308617,
       12.74793379,  9.14826808, 18.90411883, 15.93142193, 19.0122169 ,
       13.0776329 , 17.89340189, 10.20222425, 16.41786324, 14.61803039,
       19.20138853, 19.76890339, 12.18582383, 18.27174513, 17.56370277,
       13.02898877, 19.92564559, 18.6987325 , 14.62343529, 15.28283351,
       11.2183461 , 18.73656683,  8.64020715,  8.47806005, 19.37434544,
       11.22375101, 12.52092784,  9.4995868 , 10.83459796,  8.38077178,
       20.52018497,  7.48896271, 21.30389597, 23.11453864, 18.27715003,
       10.31572723, 17.25021837, 18.12581274,  9.08881414, 11.15348726,
       19.63918571, 17.85556756, 17.65558613, 20.07157798,  7.80785202,
        9.41851325, 18.48794127,  8.00783345, 14.1802332 , 18.81223548,
       16.10978374, 21.42820875,  8.58075321, 12.26689738,  7.3160058 ,
       12.89927108, 13.88836842, 17.08807127, 22.23894427, 18.86087961,
       19.49865822, 14.40183425, 14.54776664,  9.70497313, 17.42317528,
       16.1962622 , 16.6610839 , 19.41758466, 18.63387366,  8.02404816,
       17.84475776, 10.71028518, 19.92564559,  9.22934163, 13.42895162,
        7.38626955, 15.18554525, 16.3530044 , 20.15805644, 19.11491007,
       12.34797093, 14.08834985, 20.06076818, 22.42271099, 18.42308243,
       17.77449401, 14.45588328, 13.74784093,  8.44563063, 10.79135873,
       22.30380312, 22.7524101 ,  8.79694935,  7.51598723, 20.89852821])
```

```
In [36]: y_train_pred
```

```
Out[36]: array([10.46165962, 22.26596879, 11.32103927, 21.44982837, 19.43379937,
                15.98547096, 16.6610839 , 19.11491007, 17.77449401, 13.02898877,
                14.6342451 , 19.29867679, 22.96320134, 14.73153336, 11.17510688,
                11.72640703, 10.66704595, 22.06058246, 13.60731344, 20.02833876,
                20.05536327, 11.98043749, 16.59082015, 14.67748432,  7.55922646,
                13.63433795, 17.24481347, 14.35319012, 21.52009211, 12.64524062,
                20.54180459, 12.36418564, 14.21266263, 11.22375101, 12.16960911,
                 8.45103553,  8.02404816, 15.19095015,  7.13223908,  9.98602811,
                13.59109873, 22.92536701, 16.29355046,  7.54841665, 10.91026661,
                22.80645914, 19.53108764, 22.55242867, 16.55839073, 21.88762555,
                18.03933428, 18.48253636, 20.23912999, 11.94260317, 11.80748058,
                 7.56463136, 10.75892931, 22.46054532, 10.82378815, 18.62846876,
                21.21741752,  9.15907788,  9.42391815, 20.07157798, 21.29308617,
                12.74793379,  9.14826808, 18.90411883, 15.93142193, 19.0122169 ,
                13.0776329 , 17.89340189, 10.20222425, 16.41786324, 14.61803039,
                19.20138853, 19.76890339, 12.18582383, 18.27174513, 17.56370277,
                13.02898877, 19.92564559, 18.6987325 , 14.62343529, 15.28283351,
                11.2183461 , 18.73656683,  8.64020715,  8.47806005, 19.37434544,
                11.22375101, 12.52092784,  9.4995868 , 10.83459796,  8.38077178,
                20.52018497,  7.48896271, 21.30389597, 23.11453864, 18.27715003,
                10.31572723, 17.25021837, 18.12581274,  9.08881414, 11.15348726,
                19.63918571, 17.85556756, 17.65558613, 20.07157798,  7.80785202,
                 9.41851325, 18.48794127,  8.00783345, 14.1802332 , 18.81223548,
                16.10978374, 21.42820875,  8.58075321, 12.26689738,  7.3160058 ,
                12.89927108, 13.88836842, 17.08807127, 22.23894427, 18.86087961,
                19.49865822, 14.40183425, 14.54776664,  9.70497313, 17.42317528,
                16.1962622 , 16.6610839 , 19.41758466, 18.63387366,  8.02404816,
                17.84475776, 10.71028518, 19.92564559,  9.22934163, 13.42895162,
                 7.38626955, 15.18554525, 16.3530044 , 20.15805644, 19.11491007,
                12.34797093, 14.08834985, 20.06076818, 22.42271099, 18.42308243,
                17.77449401, 14.45588328, 13.74784093,  8.44563063, 10.79135873,
                22.30380312, 22.7524101 ,  8.79694935,  7.51598723, 20.89852821])
```

```
In [37]: reg.intercept_
```

```
Out[37]: 7.094404759150406
```

```
In [38]: reg.coef_
```

```
Out[38]: array([0.05404903])
```

`Accuracy

```
In [39]: from sklearn.metrics import r2_score
```

```
In [40]: train=r2_score(y_train,y_train_pred)
```

```
In [41]: test=r2_score(y_test,y_test_pred)
```

In [42]:
```
train
```

Out[42]:
```
0.8037445271259666
```

In [43]:
```
test
```

Out[43]:
```
0.8353708570111553
```

In [ ]:
```

```