

TASK3:-SALES PREDICTION USING PYTHON

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

```
df=pd.read_csv("C:/Users/HP/Desktop/oasis/Advertising.csv")
df
```

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

200 rows × 5 columns

```
df.head(10)
```

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
5	6	8.7	48.9	75.0	7.2
6	7	57.5	32.8	23.5	11.8
7	8	120.2	19.6	11.6	13.2
8	9	8.6	2.1	1.0	4.8
9	10	199.8	2.6	21.2	10.6

```
df.tail(10)
```

	Unnamed: 0	TV	Radio	Newspaper	Sales
190	191	39.5	41.1	5.8	10.8
191	192	75.5	10.8	6.0	9.9
192	193	17.2	4.1	31.6	5.9
193	194	166.8	42.0	3.6	19.6
194	195	149.7	35.6	6.0	17.3
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

```
df.shape
```

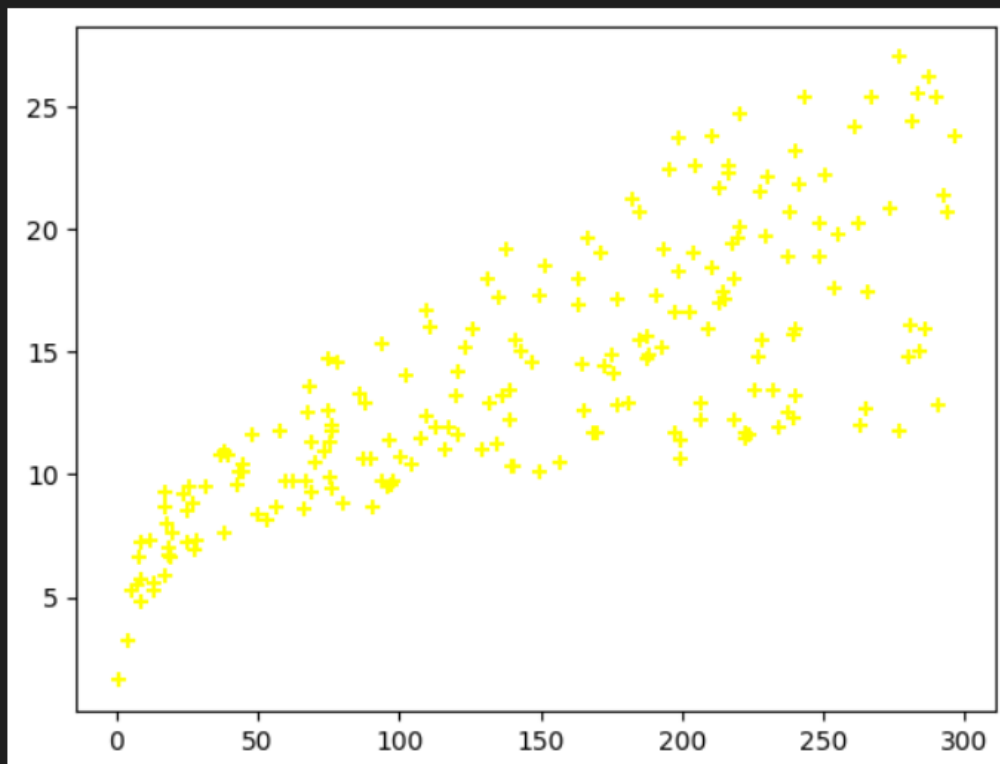
```
(200, 5)
```

```
df.describe()
```

	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

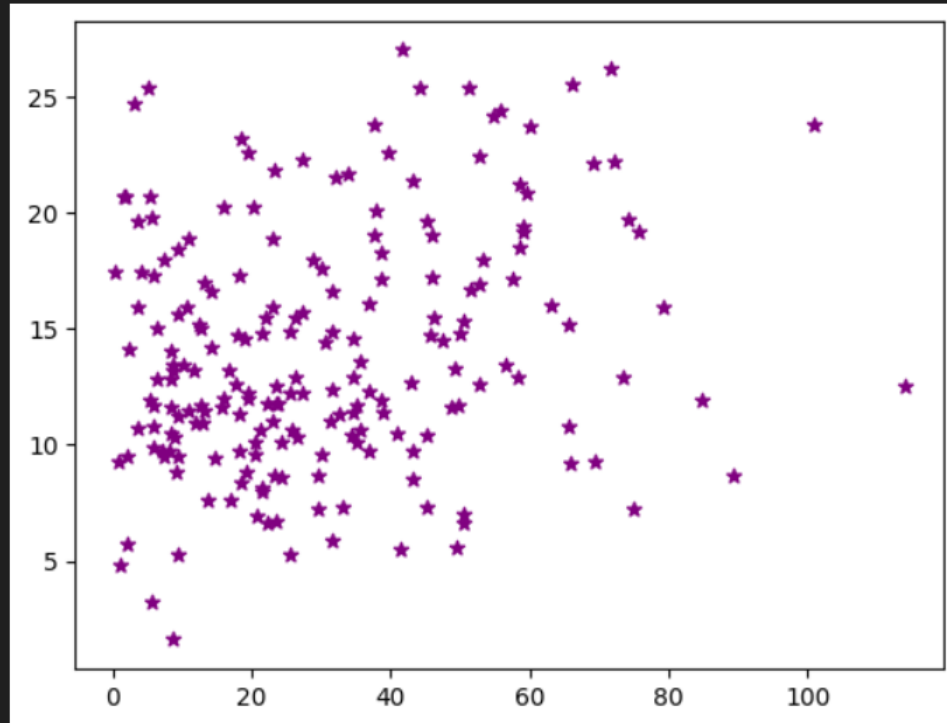
```
plt.scatter(df['TV'],df['Sales'],marker='+',color="yellow")
```

```
<matplotlib.collections.PathCollection at 0x224fb5474c0>
```



```
plt.scatter(df['Newspaper'],df['Sales'],marker='*',color="purple")
```

```
<matplotlib.collections.PathCollection at 0x224fd89cf10>
```



```
x=df.iloc[:,0:-1]  
x
```

	Unnamed: 0	TV	Radio	Newspaper
0	1	230.1	37.8	69.2
1	2	44.5	39.3	45.1
2	3	17.2	45.9	69.3
3	4	151.5	41.3	58.5
4	5	180.8	10.8	58.4
...
195	196	38.2	3.7	13.8
196	197	94.2	4.9	8.1
197	198	177.0	9.3	6.4
198	199	283.6	42.0	66.2
199	200	232.1	8.6	8.7

200 rows × 4 columns

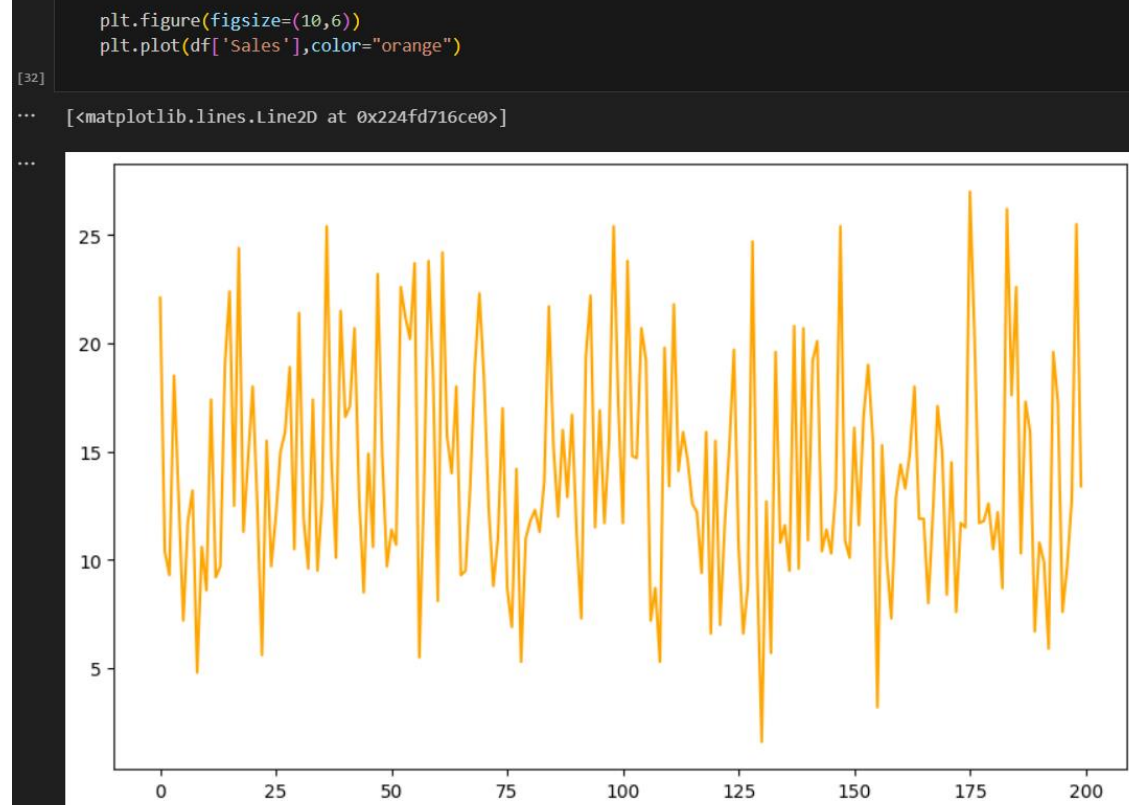
```

y=df.iloc[:,-1]
y
[10]
...
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

df.isnull().sum()
[30]
...
Unnamed: 0      0
TV              0
Radio           0
Newspaper       0
Sales           0
dtype: int64

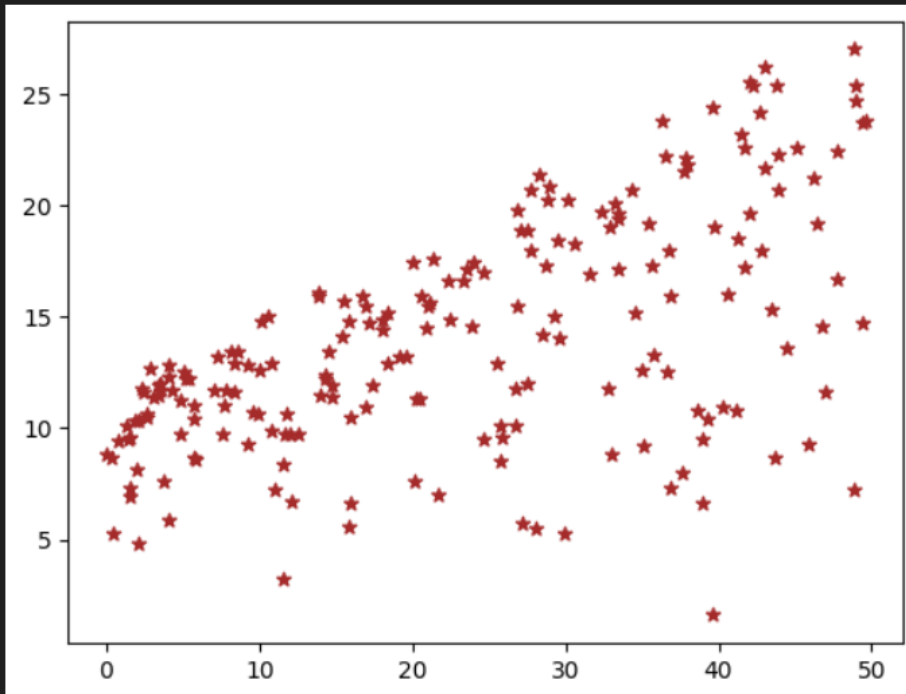
df.columns
[31]
...
Index(['Unnamed: 0', 'TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')

```



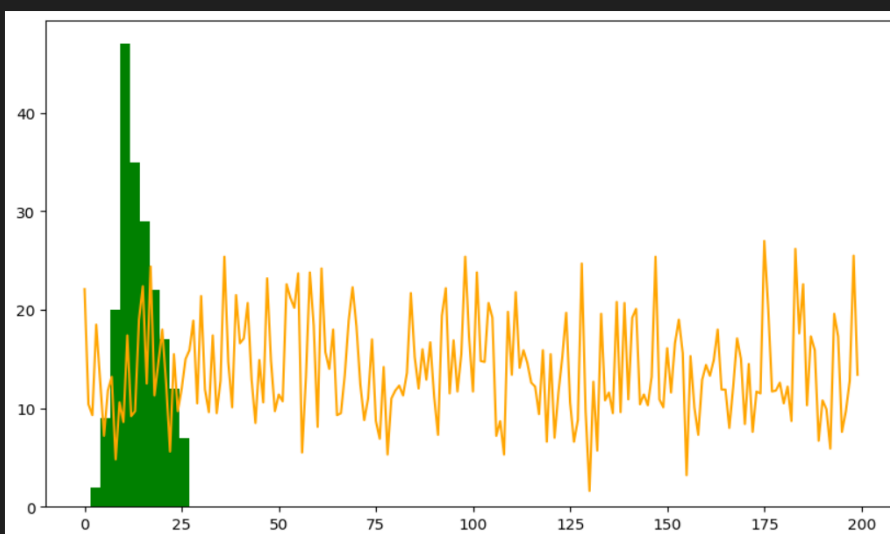
```
plt.scatter(df['Radio'],df['Sales'],marker='*',color="brown")
```

```
<matplotlib.collections.PathCollection at 0x224fd7b46a0>
```



```
plt.figure(figsize=(10,6))
plt.plot(df['Sales'],color="orange")
plt.hist(df['Sales'],color="green")
```

```
(array([ 2.,  9., 20., 47., 35., 29., 22., 17., 12.,  7.]),
 array([ 1.6,  4.14,  6.68,  9.22, 11.76, 14.3, 16.84, 19.38, 21.92,
        24.46, 27.  ]),
 <BarContainer object of 10 artists>)
```

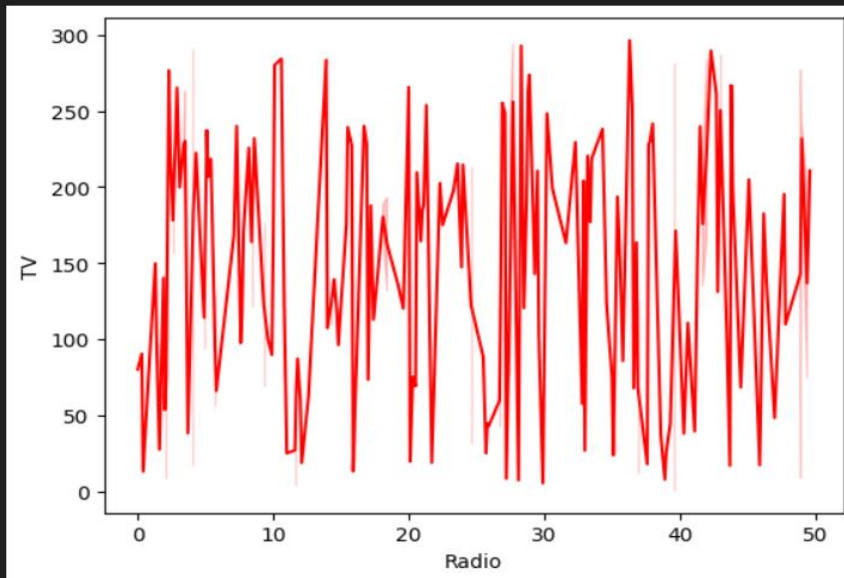


```
sns.lineplot(x='Radio',y='TV',data=df,color='red')
```

45]

```
<AxesSubplot:xlabel='Radio', ylabel='TV'>
```

...

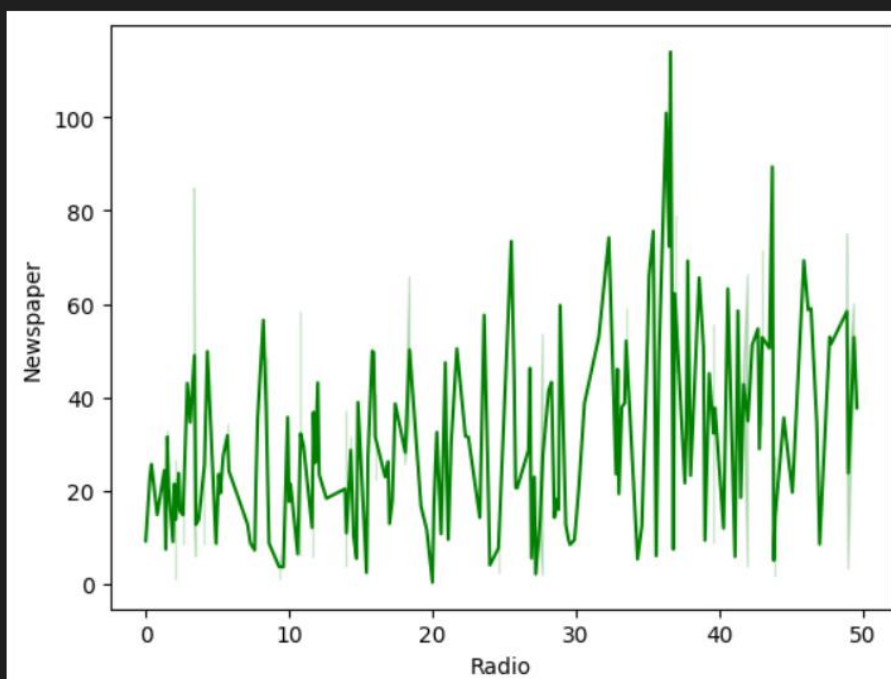


```
sns.lineplot(x='Radio',y='Newspaper',data=df,color='green')
```

7]

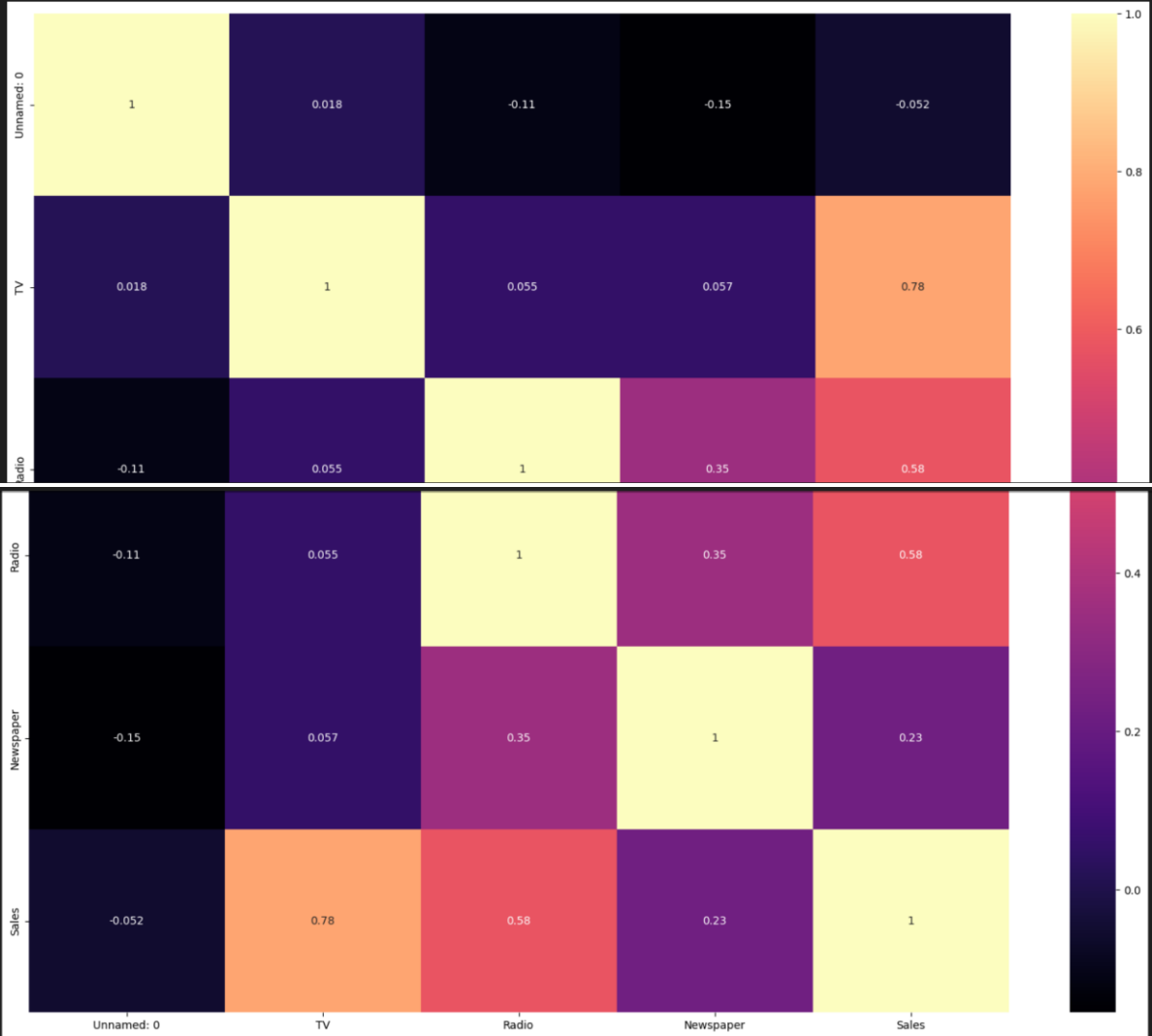
```
<AxesSubplot:xlabel='Radio', ylabel='Newspaper'>
```

...



```
plt.figure(figsize=(20,15))
correlations=df.corr()
sns.heatmap(correlations,cmap="magma",annot=True)
plt.show()
```

Python



```
[14] from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix

[16] x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=43)

[17] x_train
...

```

	Unnamed: 0	TV	Radio	Newspaper
22	23	13.2	15.9	49.6
129	130	59.6	12.0	43.1
105	106	137.9	46.4	59.0
102	103	280.2	10.1	21.4
159	160	131.7	18.4	34.6
...
58	59	210.8	49.6	37.7
21	22	237.4	5.1	23.5
49	50	66.9	11.7	36.8
64	65	131.1	42.8	28.9
68	69	237.4	27.5	11.0

150 rows × 4 columns

```
[18] x_test
...

```

	Unnamed: 0	TV	Radio	Newspaper
56	57	7.3	28.1	41.4
37	38	74.7	49.4	45.7
67	68	139.3	14.5	10.2
79	80	116.0	7.7	23.1
80	81	76.4	26.7	22.3
188	189	286.0	13.9	3.7
183	184	287.6	43.0	71.8
10	11	66.1	5.8	24.2
128	129	220.3	49.0	3.2
62	63	239.3	15.5	27.3
65	66	69.0	9.3	0.9
17	18	281.4	39.6	55.8
133	134	219.8	33.5	45.1
195	196	38.2	3.7	13.8
146	147	240.1	7.3	8.7
38	39	43.1	26.7	35.1
173	174	168.4	7.1	12.8
149	150	44.7	25.8	20.6
93	94	250.9	36.5	72.3
29	30	70.6	16.0	40.8
0	1	230.1	37.8	69.2
2	3	17.2	45.9	69.3
122	123	224.0	2.4	15.6
180	181	156.6	2.6	8.3

180	181	156.6	2.6	8.3
95	96	163.3	31.6	52.9
121	122	18.8	21.7	50.4
185	186	205.0	45.1	19.6
39	40	228.0	37.7	32.0
66	67	31.5	24.6	2.2
19	20	147.3	23.9	19.1
11	12	214.7	24.0	4.0
45	46	175.1	22.5	31.5
41	42	177.0	33.4	38.7
92	93	217.7	33.5	59.0
168	169	215.4	23.6	57.6
1	2	44.5	39.3	45.1
57	58	136.2	19.2	16.6
189	190	18.7	12.1	23.4
151	152	121.0	8.4	48.7
167	168	206.8	5.2	19.4
116	117	139.2	14.3	25.6
138	139	43.0	25.9	20.5
155	156	4.1	11.6	5.7
82	83	75.3	20.3	32.5
160	161	172.5	18.1	30.7
181	182	218.5	5.4	27.4
171	172	164.5	20.9	47.4
12	13	23.8	35.1	65.9
55	56	198.9	49.4	60.0
77	78	120.5	28.5	14.2

```

y_train
[9]
. 22      5.6
   129     9.7
   105    19.2
   102    14.8
   159    12.9
   ...
   58    23.8
   21    12.5
   49     9.7
   64    18.0
   68    18.9
Name: Sales, Length: 150, dtype: float64

```

y_test

```
56      5.5
37     14.7
67     13.4
79     11.0
80     11.8
188     15.9
183     26.2
10      8.6
128     24.7
62     15.7
65      9.3
17     24.4
133     19.6
195      7.6
146     13.2
38     10.1
173     11.7
149     10.1
93     22.2
29     10.5
0      22.1
2       9.3
122     11.6
180     10.5
95     16.9
...
171     14.5
12       9.2
55     23.7
77     14.2
```

Name: Sales, dtype: float64

```
x_train.astype(int)
x_test.astype(int)
y_train.astype(int)
y_test.astype(int)
from sklearn.preprocessing import StandardScaler
Sc=StandardScaler()
x_train_scaled=Sc.fit_transform(x_train)
x_test_scaled=Sc.fit_transform(x_test)
```

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train_scaled,y_train)
```

▼ LinearRegression

LinearRegression()

```
y_pred=lr.predict(x_test_scaled)
```

```
from sklearn.metrics import r2_score
```

```
r2_score(y_test,y_pred)
```

0.9328733433614231

```
plt.scatter(y_test,y_pred,c='skyblue')
```

```
<matplotlib.collections.PathCollection at 0x224f935ed10>
```

