Oasis infotech :data Science

Task3: A product and sevice-based business always need their data science to predict their future sales with every step they take to manipulate the cost of advertising theri product .so let's start the task of sales prediction with machine learning using python

Author: Sujata Gaikwad

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
data=pd.read_csv(r"C:\Users\HP\Downloads\Advertising.csv")
data
```

Out[2]:

	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

In [3]:

```
data.head()
```

Out[3]:

	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

```
In [4]: data.pop("Unnamed: 0")
```

```
Out[4]: 0        1
        1        2
        2        3
        3        4
        4        5
              ...
        195    196
        196    197
        197    198
        198    199
        199    200
        Name: Unnamed: 0, Length: 200, dtype: int64
```

```
In [5]: data.isnull().sum()
```

```
Out[5]: TV           0
        Radio        0
        Newspaper    0
        Sales        0
        dtype: int64
```

```
In [6]: data.describe()
```

Out[6]:

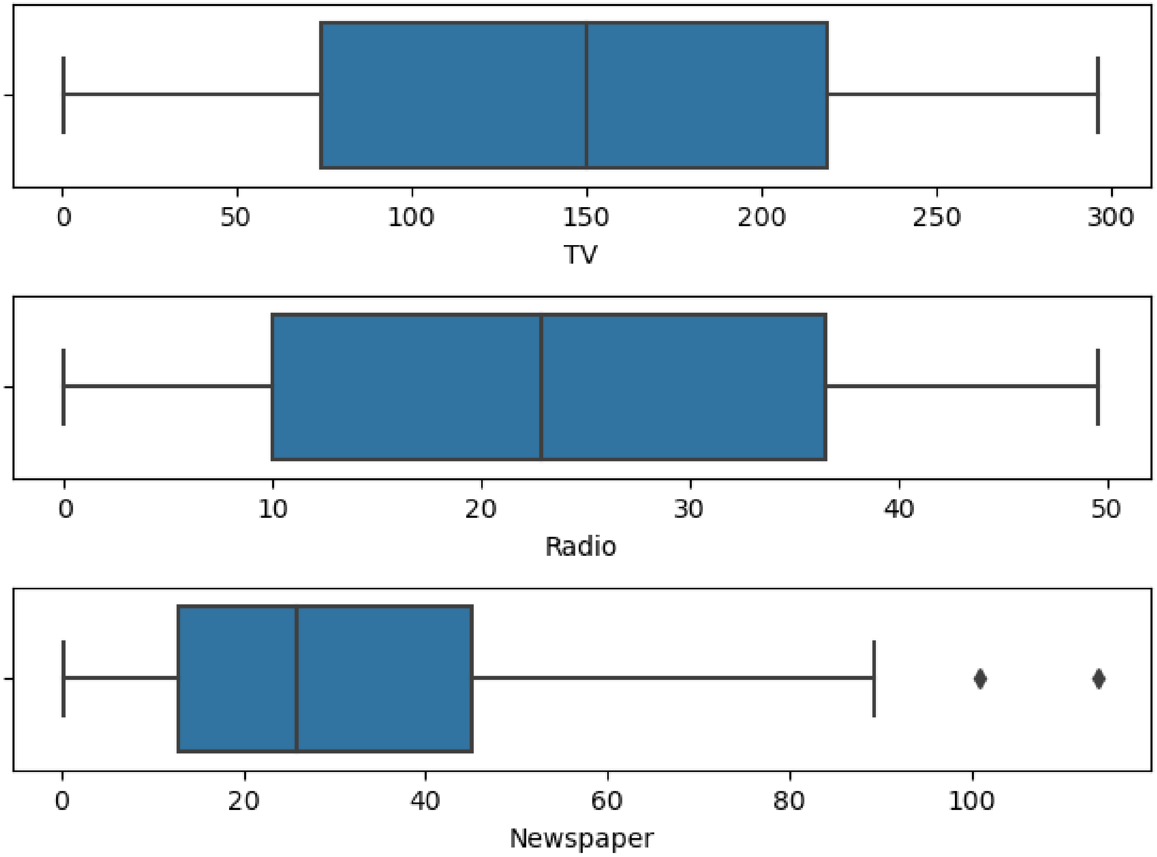
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	14.022500
std	85.854236	14.846809	21.778621	5.217457
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	10.375000
50%	149.750000	22.900000	25.750000	12.900000
75%	218.825000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

Outlier Analysis

In [7]:

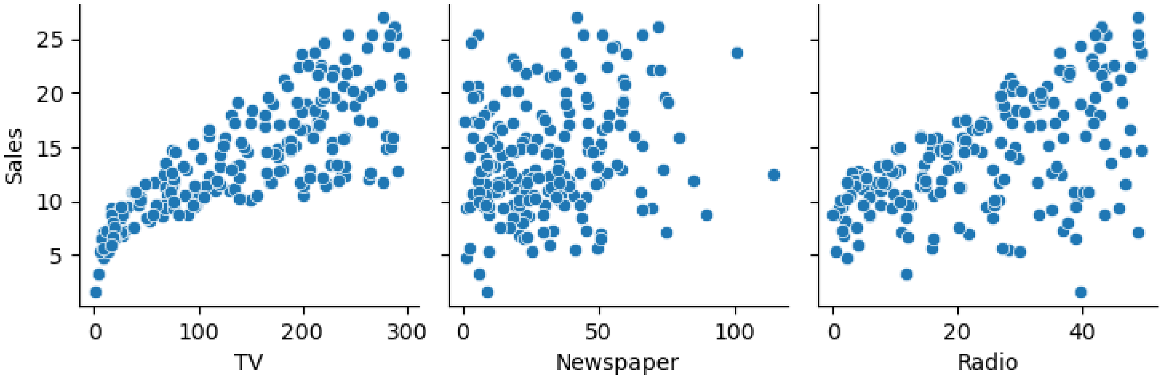
```
import matplotlib.pyplot as plt
import seaborn as sns
fig,axs=plt.subplots(3)
plt1=sns.boxplot(data["TV"],ax=axs[0])
plt1=sns.boxplot(data["Radio"],ax=axs[1])
plt1=sns.boxplot(data["Newspaper"],ax=axs[2])
plt.tight_layout()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Data Visualization

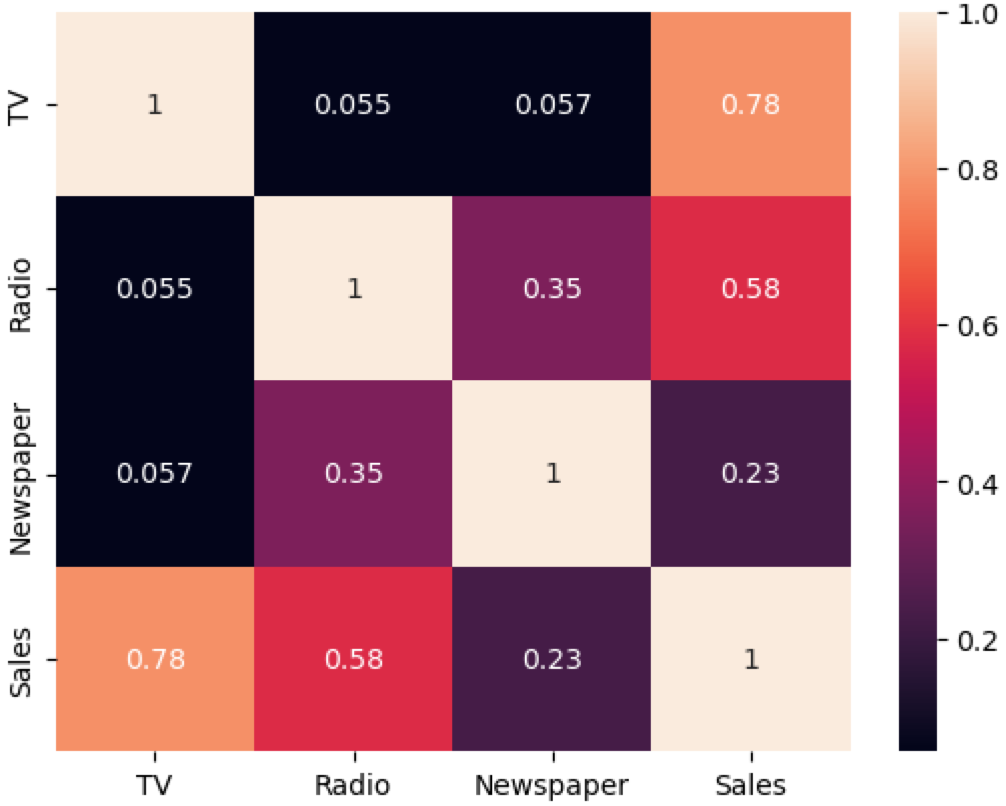
In [8]:
```
sns.pairplot(data,x_vars=["TV","Newspaper","Radio"],y_vars="Sales",kind="scatt
plt.show()
```



Correlation Coefficient

In [9]:
```
sns.heatmap(data.corr(),annot=True)
```

Out[9]:
```
<AxesSubplot:>
```



In [10]:

```
feature=["TV","Radio","Newspaper"]
x=data[feature]
y=data["Sales"]
```

In [11]:

```
### train test split
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=
```

In [12]:

```
from sklearn.ensemble import GradientBoostingRegressor
g1= GradientBoostingRegressor()
g1.fit(x_train,y_train)
```

Out[12]:

```
GradientBoostingRegressor()
```

In [13]:

```
y_pred=g1.predict(x_test)
y_pred
```

Out[13]:

```
array([19.0501368 ,  8.34995302, 18.91907149, 24.89240076, 21.43411011,
       14.88789608, 15.03640137, 22.45211222, 22.6249678 ,  8.31531013,
       24.91707413,  9.27889453,  7.92296456, 20.15198396, 20.3625618 ,
       12.41645511, 18.32359995,  5.01606724, 22.33523029, 21.95620665,
       15.51726858,  6.29199055, 24.71264554, 15.87351209, 13.99196387,
        7.05851741,  9.20675313, 10.60592285, 22.39156378,  6.93172934,
       13.15162106, 22.04524499,  7.60184604,  7.72737772, 12.83763538,
       12.21761435,  8.89042133, 14.2067268 ,  9.91120682, 11.86999585])
```

In [14]:

```
from sklearn.metrics import r2_score
print(g1,"","Test case R2 score in %:",r2_score(y_test,y_pred)*100)
print("\n")
```

```
GradientBoostingRegressor()  Test case R2 score in %: 97.48729257964301
```

Thank you