Sales Prediction:

The aim to Build a model which predicts sales based on the Money spent on different platfoms such as TV, Radio and Newspaper for marketing

```
# Importing the Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Reading the dataset
df=pd.read_csv("advertising.csv")
df.head()
```

	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

- ## Data Pre-Processing

1) Checking for missing values

```
# Checking for missing values
df.isna().sum()
```

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

Conclusion:-The dataset Does not have Missing values

2) Checking the Duplicate Values

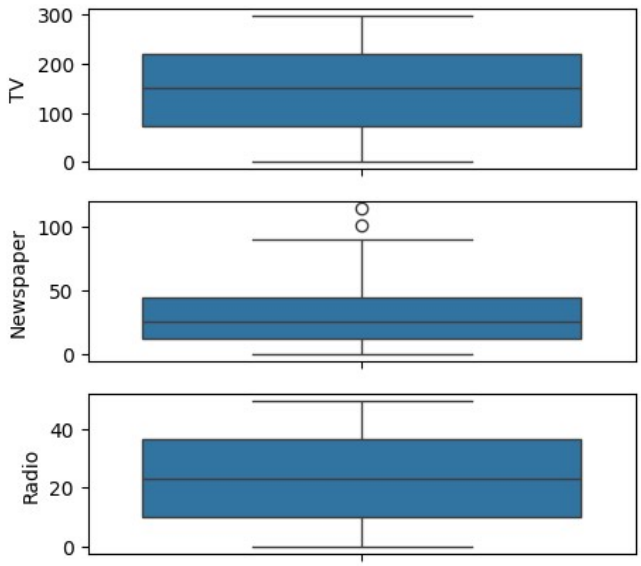
```
# Checking the Duplicate Values
df.duplicated().any()
```

```
np.False_
```

Conclusion:- There are no duplicate rows present in the dataset

3) Checking the outlier

```
#Checking the outlier
fig,axs=plt.subplots(3,figsize=(5,5))
plt1=sns.boxplot(df['TV'],ax=axs[0])
plt2=sns.boxplot(df['Newspaper'],ax=axs[1])
plt3=sns.boxplot(df['Radio'],ax=axs[2])
```

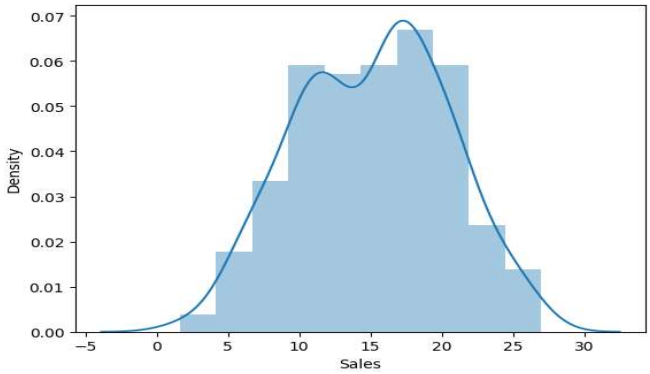


Conclusion :- There are not Extreme Values Present in the dataset

Exploratory Data Analysis

1) Distribution of the target variable

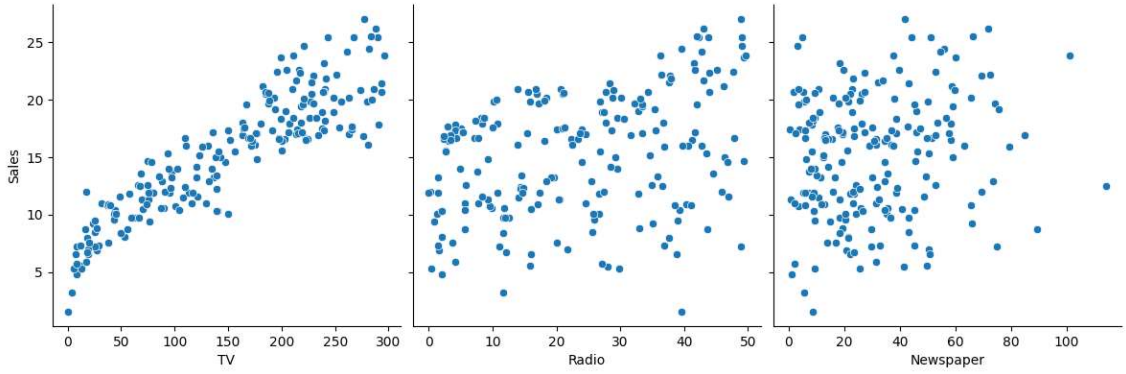
```
sns.distplot(df['Sales'])
```



Conclusion :- It is Normally Distributed

2) How Sales are related with each other

```
sns.pairplot(df,x_vars=['TV','Radio','Newspaper'],y_vars='Sales',height=4,aspect=1,kind='scatter')
```

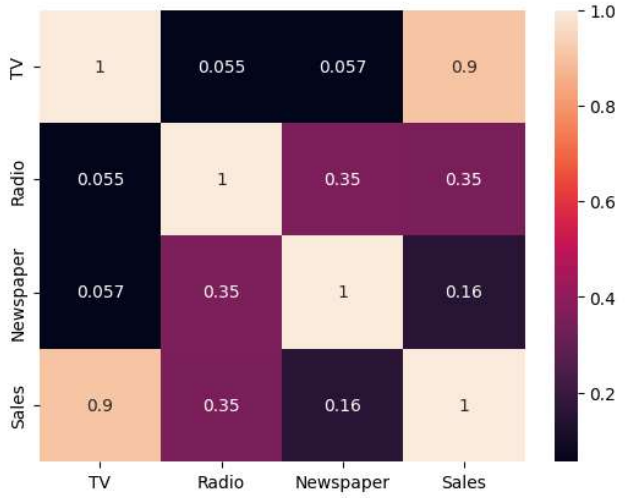


Conclusion :- TV is strongly ,positively, linearly corelated with the target variable.

Radio and Newspaper are seems to uncorrelated

3) Heatmap

```
#3)Heatmap
sns.heatmap(df.corr(numeric_only=True),annot=True)
```



Conclusion :- TV seems to be most correlated with sales as 0.9 very close to 1

Model Building

1)Simple Linear Regression

```
# Model Building
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

2) Setting the values for X and Y

```
x=df[['TV']]
y=df['Sales']
```

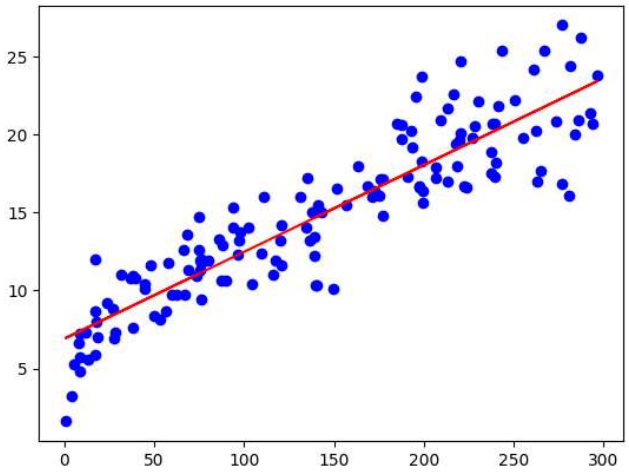
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(xtrain,ytrain)
ypred=lr.predict(xtest)
```

```
# Priniting the model Coefficients
print("Intercept",lr.intercept_)
print("Coeffcient:",lr.coef_)
```

```
Intercept 6.904032471762278
Coeffcient: [0.05566076]
```

3) Line of Best Fit

```
plt.scatter(xtrain,ytrain,color='Blue')
plt.plot(xtrain,lr.predict(xtrain),color='Red')
```




```
# Predict for any Value
slr_diff=pd.DataFrame({'Actual Value':ytest,'predicated Value' :ypred})
```

+ Code + Markdown

```
1  slr_diff
```

	Actual Value	predicated Value
58	23.8	18.637321
40	16.6	18.175337
34	11.9	12.230768
102	19.8	22.500178
184	17.6	21.030734
198	25.5	22.689425
95	16.9	15.993435
4	17.9	16.967499
29	10.5	10.833682
168	17.1	18.893361
171	17.5	16.060228
18	11.3	10.755757
11	17.4	18.854398
89	16.7	13.015584
110	18.4	19.472233
118	15.9	13.900590
159	12.9	14.234555
35	17.8	23.084616
136	9.5	8.328948
59	18.4	18.631755
51	10.7	12.492373
16	12.5	10.677832
44	8.5	8.301118

```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

```
mae=mean_absolute_error(ytest,ypred)
mse=mean_squared_error(ytest,ypred)
rmse=np.sqrt(mse) # root mean squared error
r2score=r2_score(ytest,ypred)
print(f'MAE:{mae}\nMSE:{mse}\nRMSE:{rmse}\nR2_Score:{r2score}')
```

```
MAE:1.86239036505223
MSE:5.143558863773587
RMSE:2.267941547697733
R2_Score:0.7984701565892568
```