Practical-3

Aim- Implement Linear Regression in R or Python

Import Libraries

```
import pandas as pd import numpy as np

from sklearn import linear_model

from sklearn.linear_model import LinearRegression

from sklearn.model_selection import train_test_split from sklearn.metrics import mean_squared_error

from sklearn.metrics import r2_score import seaborn as sns

from sklearn import metrics

import matplotlib.pyplot as plt

%matplotlib inline
```

Load Data

```
df  =  pd.read_csv(' /content/Advertising.csv') df
```

```
200 rows × 5 columns
```

Data info:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'> RangeIndex: 200 entries, 0 to 199

Data columns (total 5 columns):

#          Column        Non-Null Count      Dtype

0          Unnamed: 0    200 non-null        int64

1          TV            200 non-null        float64

2          Radio         200 non-null        float64

3          Newspaper     200 non-null        float64

4          Sales         200 non-null        float64

dtypes: float64(4), int64(1)    memory usage: 7.9 KB
```

Simple Linear Regression:

Taking X = Tv and Y = Sales for Linear Regression

```
y =
df['Sales'].values.reshap
e(-1,  1)  x =
df['TV'].values.reshape(
-1, 1)
```

Split Data:

Train And Test Data(80/20)

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state =
42)
```

Create a model and fit it for TV and Sales:

```
model = LinearRegression().fit(x, y)
```

The General Equation of Linear regression model is y = b_0 + b_1*x

Here b_0 =
intercept b_1
= slope

```
print("The value of intercept is :
",model.intercept_) print("The
value of coefficient is :
",model.coef_)
```

```
The value of intercept is : [7.03259355]
The value of coefficient is : [[0.04753664]]
```

Predict y:

```
y_predict = model.predict(x_test)
```

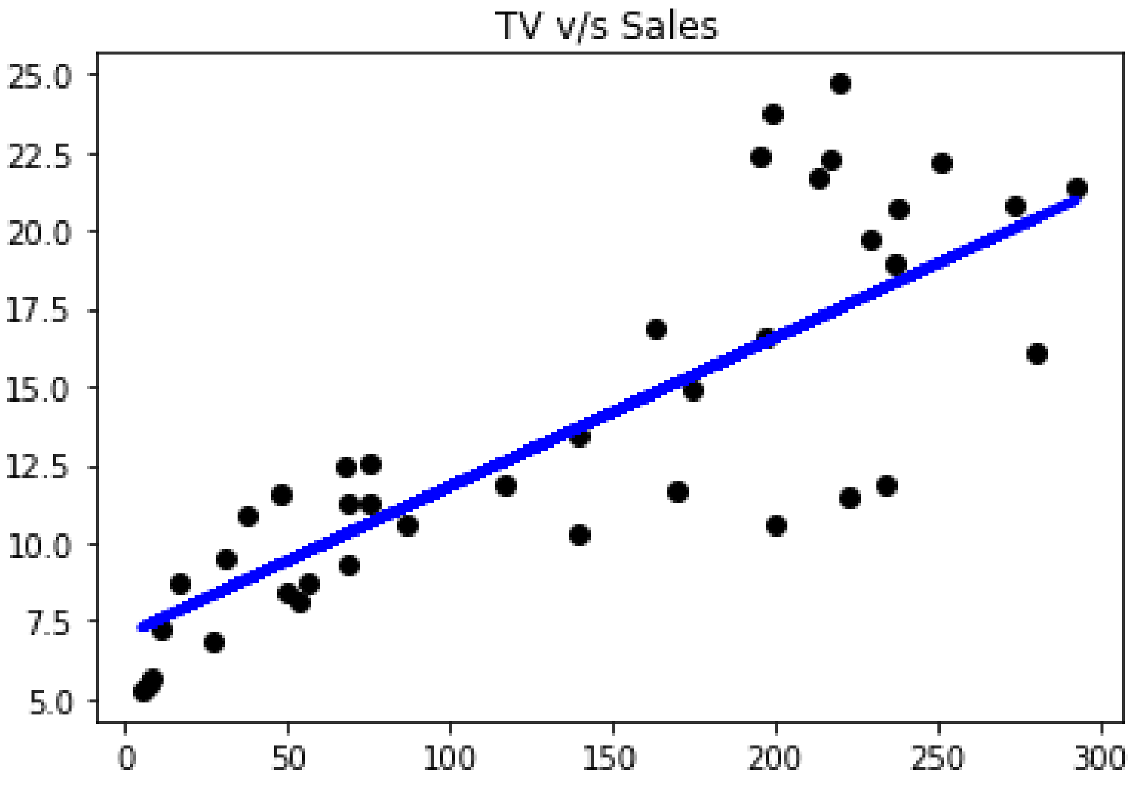
MSE And R_2 value of data:

```
mse =
mean_squared_error(y_test,y_
predict) R_2 =
r2_score(y_test, y_predict)
print(f'R-Square value: {R_2:.2f}')
print(f'Mean squared error: {mse:.2f}')
```

```
R-Square value: 0.68
Mean squared error: 10.10
```

Displaying the visulaization:

```
plt.scatter(x_test, y_test, color="black")
plt.plot(x_test, y_predict,
color="blue", linewidth=3)
plt.title("TV v/s Sales")
plt.show()
```

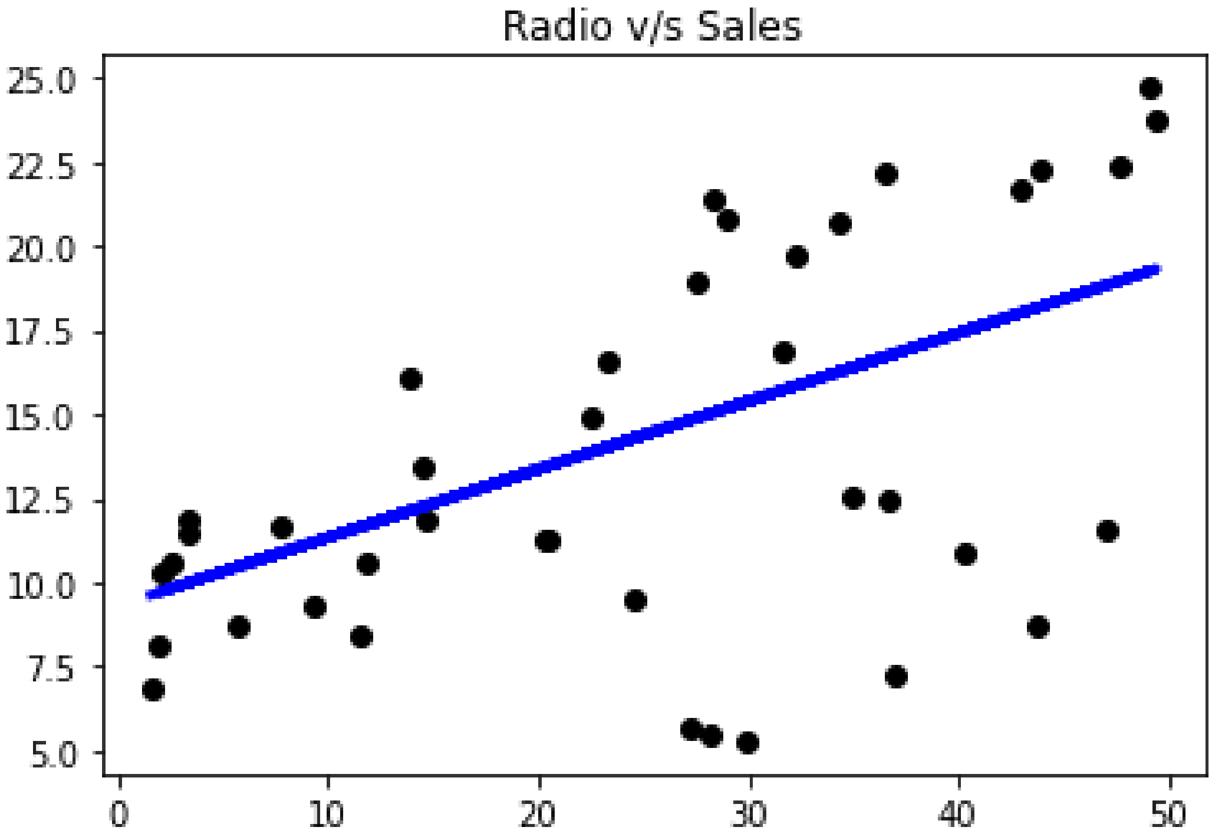


Similar Process for Radio v/s Sales:

```
y =  df['Sales'].values.reshape(-1,  1)  x = df['Radio'].values.reshape(-1, 1)

#train and test of model

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2,

random_state = 42)


#Fit model:

model = LinearRegression().fit(x, y) #value of b_0


print("The value of intercept is : ",model.intercept_)


#value of b_1

print("\nThe value of coefficient is : ",model.coef_)


#predict of y:

y_predict = model.predict(x_test)


#MSE:

mse = mean_squared_error(y_test,y_predict) print(f'\nMean  squared  error:  {mse:.2f}')


#R_2

R_2 = r2_score(y_test, y_predict)

print(f'\nR-Square  value:  {R_2:.2f}') print("\n")


#displaying the visulaization
```

```
plt.scatter(x_test, y_test, color="black")

plt.plot(x_test, y_predict, color="blue", linewidth=3) plt.title("Radio v/s Sales")

plt.show()
```

```
The value of intercept is : [9.3116381]

The value of coefficient is : [[0.20249578]]

Mean squared error: 23.02

R-Square value: 0.27
```



Similar Process for Newspaper v/s Sales:

```
y = df['Sales'].values.reshape(-1, 1)
x = df['Newspaper'].values.reshape(-1, 1)
```

#train and test of model:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2,
random_state = 42)
```

#Fit model:

```
model = LinearRegression().fit(x, y)

#value of b_0
print("The value of intercept is : ",model.intercept_)

#value of b_1
print("\nThe value of coefficient is : ",model.coef_)
```

#**predict of y:**

```
y_predict = model.predict(x_test)
```

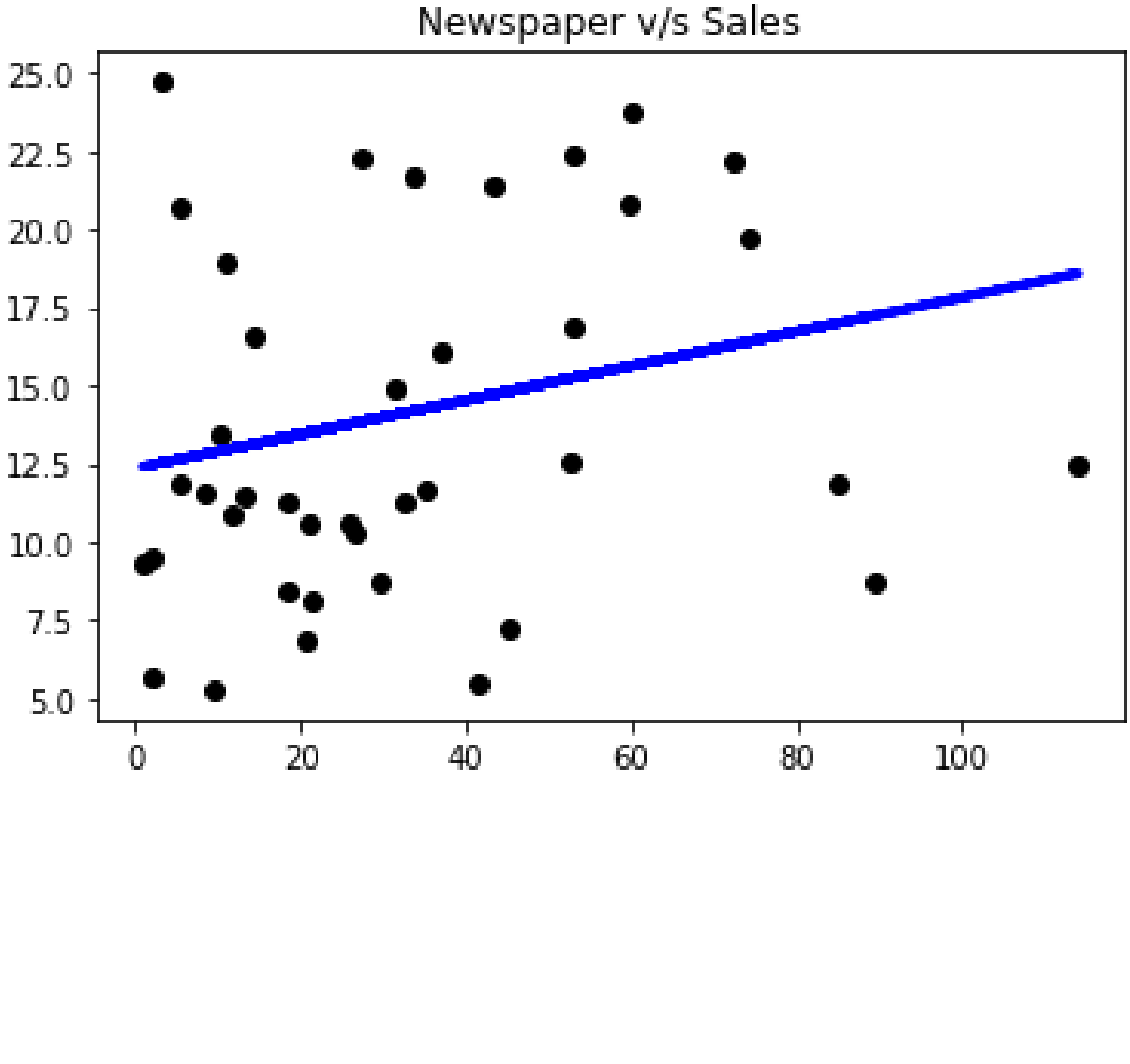
#MSE:

```
mse = mean_squared_error(y_test,y_predict) print(f'\nMean squared error: {mse:.2f}')

#R_2
R_2 = r2_score(y_test, y_predict)
print(f'\nR-Square value: {R_2:.2f}') print("\n")
```

#**displaying the visualization:**

```
plt.scatter(x_test, y_test, color="black")
plt.plot(x_test, y_predict, color="blue", linewidth=3) plt.title("Newspaper v/s Sales")
plt.show()
```

```
The value of intercept is : [12.35140707]

The value of coefficient is : [[0.0546931]]

Mean squared error: 30.40
 R-Square value: 0.04
```



Multiple Linear Regression:

Intializing:

```
y = df['Sales']
x = df[['TV', 'Radio','Newspaper']]
```

Train And Test Data:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

Fit Model:

```
model = LinearRegression().fit(x, y)
```

Value of Coeficient & Intercept:
here general equation:
y = b_0 + b_1* x_1 + b_2* x_2 + b_3*x_3
where
b_0 = intercept
b_1 = coeficient of x_1 b_2 = coeficient of x_2 b_3 = coeficient of x_3

```
#value of b_0
print("The value of intercept is : ",model.intercept_)

#coefficient of every column i.e. TV(b_1), Radio(b_2), Newspaper(b_3) columns_name
= x.columns
model_coefficients = model.coef_

coefficients_df = pd.DataFrame(data = model_coefficients,
index  =  columns_name,
columns = ['Coefficient value'])
print("\n",coefficients_df)
```

```
The value of intercept is : 2.938889369459412

TV     Coefficient value
0.045765
Radio 0.188530
Newspaper   -0.001037
```

Predict Y:

```
y_predict = model.predict(x_test)
results = pd.DataFrame({'Actual': y_test, 'Predicted': y_predict}, index=None)
print(results)
```

```
     Actual    Predicted
95   16.9   16.314921
15   22.4   20.819300
30   21.4   21.633934
158  7.3    10.384199
128  24.7   22.255492
115  12.6   12.919689
69   22.3   21.108912
170  8.4    7.394980
174  11.5   13.744357
45   14.9   15.161523
66   9.5    9.016032
182  8.7    6.554670
165  11.9   14.223721
78   5.3    8.813314
186  10.3   9.691373
177  11.7   12.162046
56   5.5    8.527713
152  16.6   16.360000
82   11.3   10.178408
68   18.9   18.976579
124  19.7   19.454413
16   12.5   12.823657
148  10.9   12.263359
93   22.2   21.227574
65   9.3    7.849045
60   8.1    5.742156
84   21.7   20.781365
67   13.4   12.037007
125  10.6   9.127350
132  5.7    8.449150
9    10.6   12.550849
18   11.3   9.951682
55   23.7   21.292611
75   8.7    11.858322
150  16.1   18.367205
104  20.7   20.301109
135  11.6   14.001414
137  20.8   20.851252
164  11.9   11.068295
76   6.9    4.477589
```

Value of MSE:

```
mse = mean_squared_error(y_test,y_predict) print(f'Mean  squared  error:  {mse:.2f}')
```

```
Mean squared error: 3.04
```

Value of R² :

```
R_2 = r2_score(y_test, y_predict) print(f'R²:  {R_2:.2f}')
```

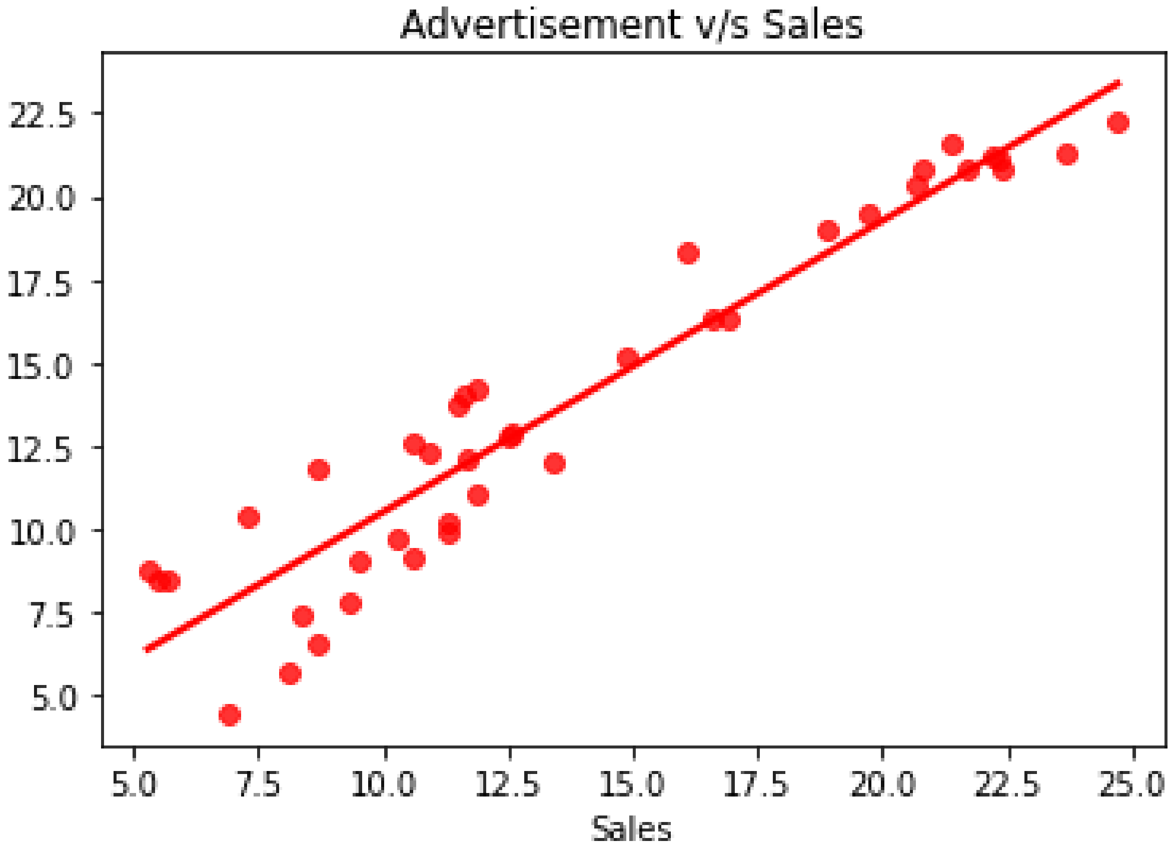
```
R²: 0.90
```

Visulaization Of data:

where
X = [TV, RADIO, NEWSPAPER]
Y = Sales

```
sns.regplot(x=y_test,y=y_predict,ci=None,color  ='red').set(title='Advertisement  v/s
Sales')
```

```
[Text(0.5, 1.0, 'Advertisement v/s Sales')]
```


Advertisement v/s Sales
Sales