LINEAR REGRESSION

ANAKHA R MENON

CH.EN.U4CSE20103

CSE-B

AIM : To understand and implement the concepts of Linear and Logistic regression in python programming.

CODE : LINEAR REGRESSION

In [ ]:
```
#Importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
#Reading the dataset
dataset = pd.read_csv("Advertising.csv")
dataset.head()
#Dropping the unnecessary columns
dataset.drop(columns=['Radio', 'Newspaper'], inplace = True)
dataset.head()
```

Out[ ]:

	Unnamed: 0	TV	Sales
0	1	230.1	22.1
1	2	44.5	10.4
2	3	17.2	9.3
3	4	151.5	18.5
4	5	180.8	12.9

In [ ]:
```
#Setting the value for X and Y
x = dataset[['TV']]
y = dataset['Sales']
#Splitting the dataset
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_s
```

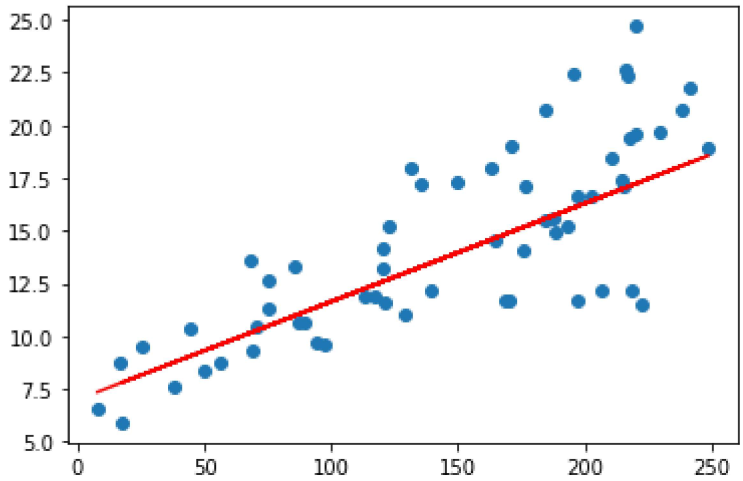
In [ ]:
```
#Fitting the Linear Regression model
from sklearn.linear_model import LinearRegression
slr = LinearRegression()
slr.fit(x_train, y_train)
#Intercept and Coefficient
print("Intercept: ", slr.intercept_)
print("Coefficient: ", slr.coef_)
```

```
#Prediction of test set
y_pred_slr= slr.predict(x_test)
#Predicted values
print("Prediction for test set: {}".format(y_pred_slr))
```

```
Intercept:  6.98966585741168
Coefficient:  [0.04649736]
Prediction for test set: [ 7.35234526 18.06533671 13.27610876 17.11214086 18.22807
747 16.60531965
 13.4620982  16.17754395 17.05169429 17.07029323 12.4391563  17.66080969
  9.60281742 15.72186983 11.04423554 11.36971705 13.95032046 14.90351632
 14.59198401 12.23921766 16.97264878 13.00642408 16.07524976 15.21969836
 15.58702749 17.23303399 17.20978531 10.49091697 15.58702749 12.71349072
 10.1700852  10.19798361 12.61584627 15.74976825  9.31453379 12.59259759
 11.50920913 14.81982107 17.33067844 15.97295557 17.00519693 15.15925179
 14.63848137 17.14933874 12.57864838 11.16047894  7.77547122 18.55820871
 10.27237939  8.76586496 16.405381   14.95466341 10.4816175  13.08546959
 16.78665935  9.05879832  7.78942043  8.17999824 16.17754395 10.9744895 ]
```

In [ ]:

```
#Actual value and the predicted value
slr_diff = pd.DataFrame({'Actual value': y_test, 'Predicted value': y_pred_slr})
slr_diff.head()
#Line of best fit
plt.scatter(x_test,y_test)
plt.plot(x_test, y_pred_slr, 'Red')
plt.show()
```



In [ ]:

```
#Model Evaluation
from sklearn import metrics
meanAbErr = metrics.mean_absolute_error(y_test, y_pred_slr)
meanSqErr = metrics.mean_squared_error(y_test, y_pred_slr)
rootMeanSqErr = np.sqrt(metrics.mean_squared_error(y_test, y_pred_slr))
print('R squared: {:.2f}'.format(slr.score(x,y)*100))
print('Mean Absolute Error:', meanAbErr)
print('Mean Square Error:', meanSqErr)
print('Root Mean Square Error:', rootMeanSqErr)
```

```
R squared: 61.02
Mean Absolute Error: 2.161984932672072
Mean Square Error: 7.975798532854851
Root Mean Square Error: 2.8241456288327007
```