In [1]:

```
#importing dependenies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#loading the dataset to a pandas dataframe
data=pd.read_csv(r"C:\Users\Kuhal Soni\Desktop\ML\data.csv",header=None)
```

In [73]:

```
data.head()

#Checking the total no. of empty cells
data.isnull().sum() #We found that there are no empty values in the dataset
```

Out[73]:

```
0    0
1    0
2    0
3    0
4    0
5    0
dtype: int64
```

In [4]:

```
#No. of rows and colums
data.shape
```

Out[4]:

```
(151, 6)
```

In [5]:

```
#Desribing statistical measures of data
data.describe()
```

Out[5]:

	0	1	2	3	4	5
count	151.000000	151.000000	151.000000	151.000000	151.000000	151.000000
mean	1.807947	13.642384	8.105960	1.847682	27.867550	2.019868
std	0.395225	6.825779	7.023914	0.360525	12.893758	0.820327
min	1.000000	1.000000	1.000000	1.000000	3.000000	1.000000
25%	2.000000	8.000000	3.000000	2.000000	19.000000	1.000000
50%	2.000000	13.000000	4.000000	2.000000	27.000000	2.000000
75%	2.000000	20.000000	15.000000	2.000000	37.000000	3.000000
max	2.000000	25.000000	26.000000	2.000000	66.000000	3.000000

In [6]:

```
#separating data and labels/Classifying dependent and independent attribute
x=data.drop(columns=5,axis=1)
y=data[[5]]

print(x)
print(y)
```

```
     0   1   2  3   4
0    1  23   3  1  19
1    2  15   3  1  17
2    1  23   3  2  49
3    1   5   2  2  33
4    2   7  11  2  55
..  ..  ..  .. ..  ..
146  2   3   2  2  26
147  2  10   3  2  12
148  1  18   7  2  48
149  2  22   1  2  51
150  2   2  10  2  27

[151 rows x 5 columns]
     5
0    3
1    3
2    3
3    3
4    3
..  ..
146  1
147  1
148  1
149  1
150  1

[151 rows x 1 columns]
```

In [72]:

```
#Checking shape to data

print(x.shape,x_train.shape,x_test.shape)
print(y.shape,y_train.shape,y_test.shape)



#Training and testing data
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.13)
```

```
(151, 5) (131, 5) (20, 5)
(151, 1) (131, 1) (20, 1)
```

In [62]:

```
#Scaling of Data

from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
sc.fit(x_train)
x_train_sc=sc.transform(x_train)
x_test_sc=sc.transform(x_test)
```

In [63]:

```
#Applying ML algorithm to train dataset

from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train_sc,y_train)
```

```
C:\Users\Kuhal Soni\anaconda3\lib\site-packages\sklearn\utils\validation.
py:993: DataConversionWarning: A column-vector y was passed when a 1d arr
ay was expected. Please change the shape of y to (n_samples, ), for examp
le using ravel().
  y = column_or_1d(y, warn=True)
```

Out[63]:

```
LogisticRegression()
```

In [64]:

```
#Checking model accuracy

model.score(x_test_sc,y_test)
```

Out[64]:

```
0.75
```

In [65]:

```
y_pred=model.predict(x_test_sc)
y_pred[0:5] #Predicted values
```

Out[65]:

```
array([3, 1, 2, 1, 1], dtype=int64)
```

In [66]:

```
y_test[0:5] #Actual values
```

Out[66]:

	5
84	3
147	1
15	2
68	1
99	2

In [68]:

```
#Taking a set of data to predict score
new_data=np.array([2,9,5,2,19]).reshape(1,-1)
new_data
```

Out[68]:

```
array([[ 2,  9,  5,  2, 19]])
```

In [70]:

```
#Predicting the score of new_data
prediction=model.predict(new_data)
prediction
```

Out[70]:

```
array([3], dtype=int64)
```

In [71]:

```
#Defining the mathematical score in three categories
if prediction[0]==1:
 print('Low score')
elif prediction[0]==2:
 print('Medium score')
elif prediction[0]==3:
 print('High score')
```

```
High score
```

In [ ]:

```
# We have successfully predicted the score fromthe given attributes
```