**ASSIGNMENT – 4**

**Customer Segmentation Analysis**

| | |
|---|---|
| **ASSIGNMENT DATE** | 13-10-2022 |
| **STUDENT NAME** | Kabini R |
| **STUDENT ROLL NO.** | 913219104005 |
| **MAXIMUM MARK** | 2 Mark |

# Download the dataset

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

# Load the dataset into the tool

In [2]:

```
data=pd.read_csv("Mall_Customers.csv")
data
```

Out[2]:

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 |
| **1** | 2 | Male | 21 | 15 | 81 |
| **2** | 3 | Female | 20 | 16 | 6 |
| **3** | 4 | Female | 23 | 16 | 77 |
| **4** | 5 | Female | 31 | 17 | 40 |
| **...** | ... | ... | ... | ... | ... |
| **195** | 196 | Female | 35 | 120 | 79 |
| **196** | 197 | Female | 45 | 126 | 28 |
| **197** | 198 | Male | 32 | 126 | 74 |

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **198** | 199 | Male | 32 | 137 | 18 |
| **199** | 200 | Male | 30 | 137 | 83 |

200 rows × 5 columns

# Perform below visualizations

## Univariate Analysis

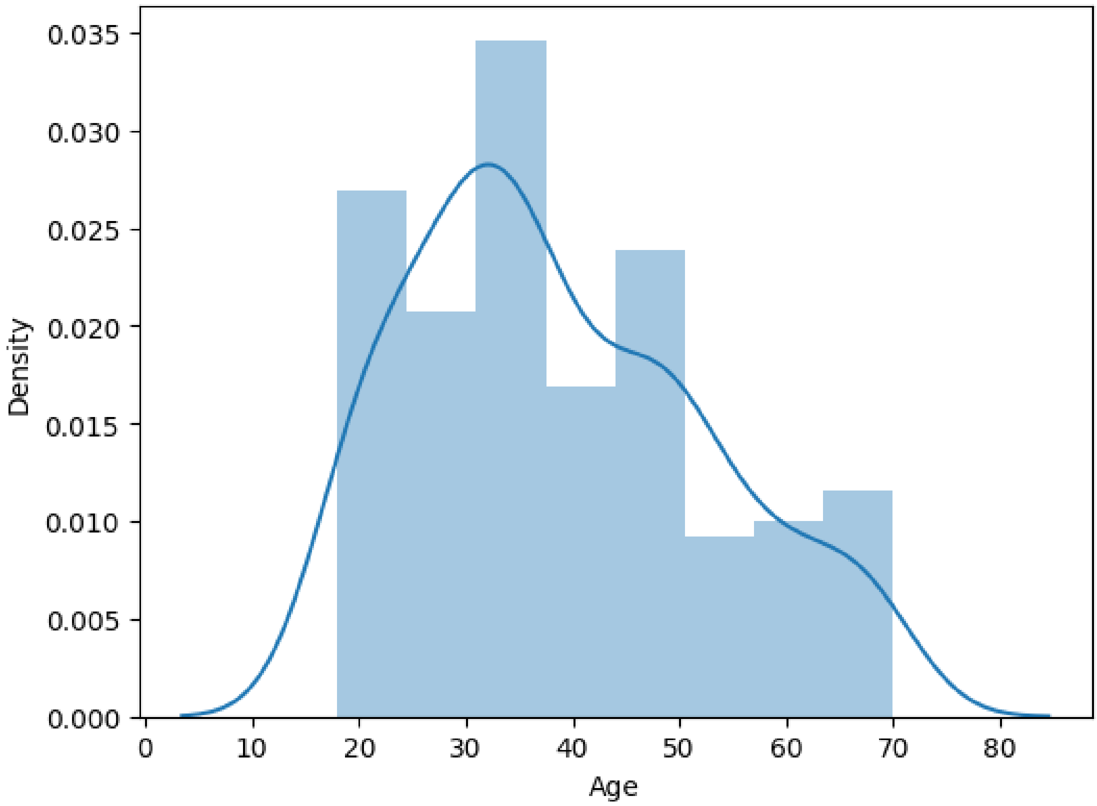
In [3]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
sns.distplot(data.Age)
```

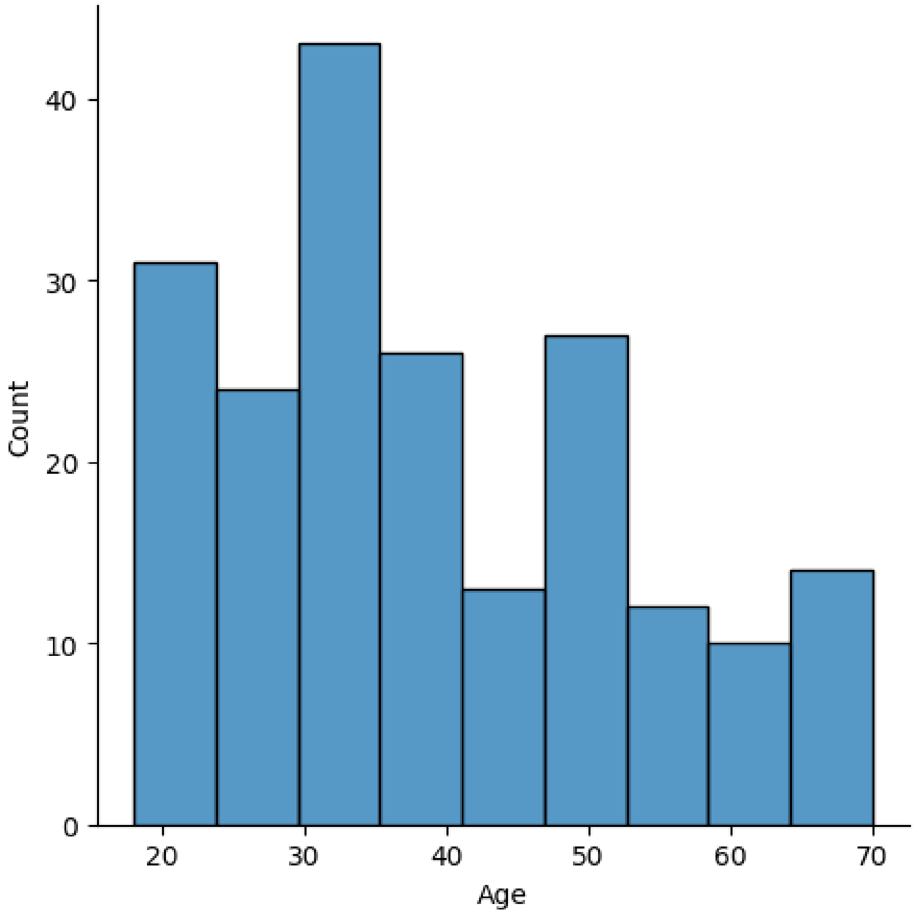
Out[4]:




In [5]:

```
sns.displot(data.Age)
```

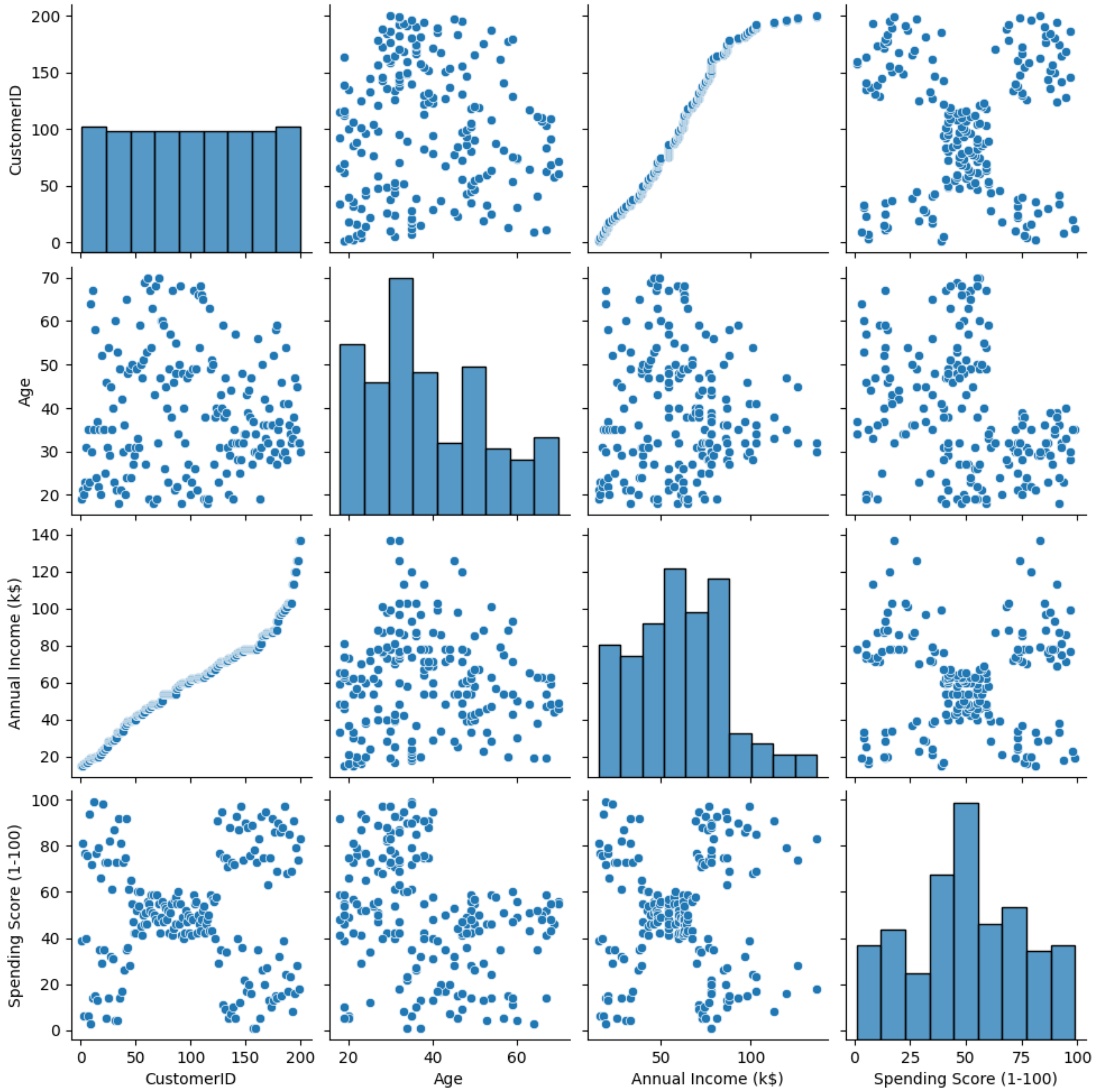
Out[5]:




In [6]:

```
sns.pairplot(data)
```

Out[6]:

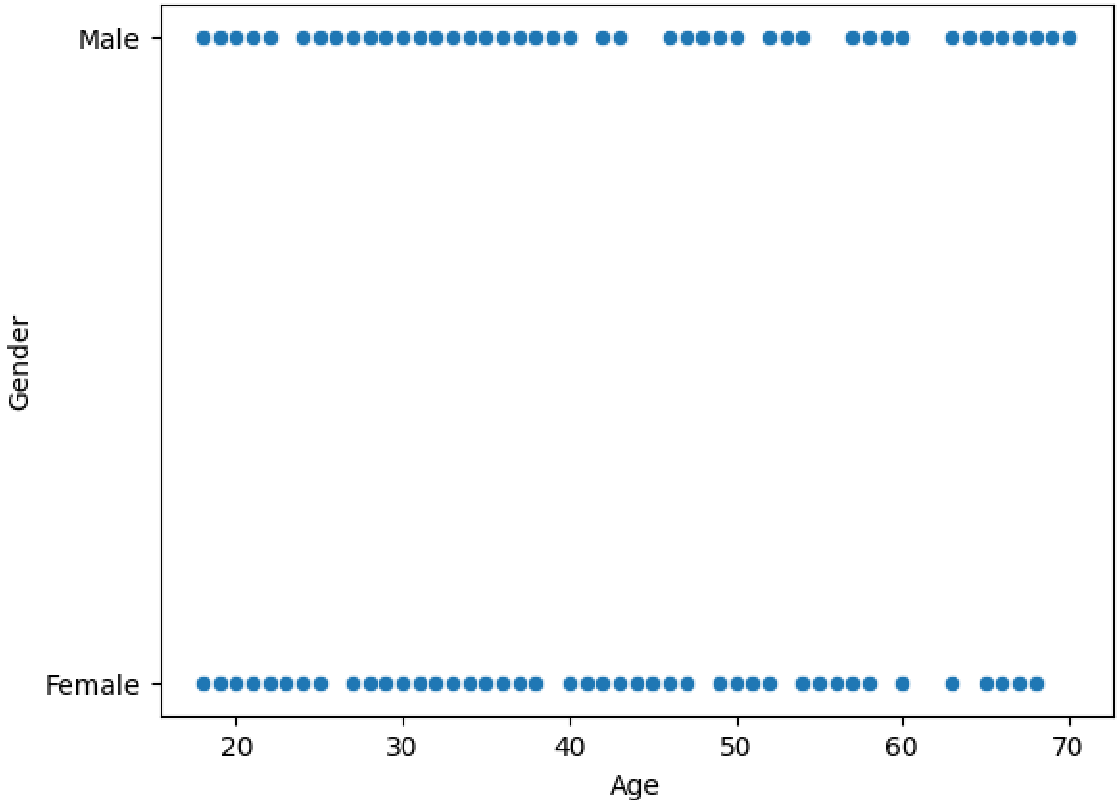



# Bivariate Analysis

In [7]:

```
sns.scatterplot(x=data.Age,y=data.Gender)
```

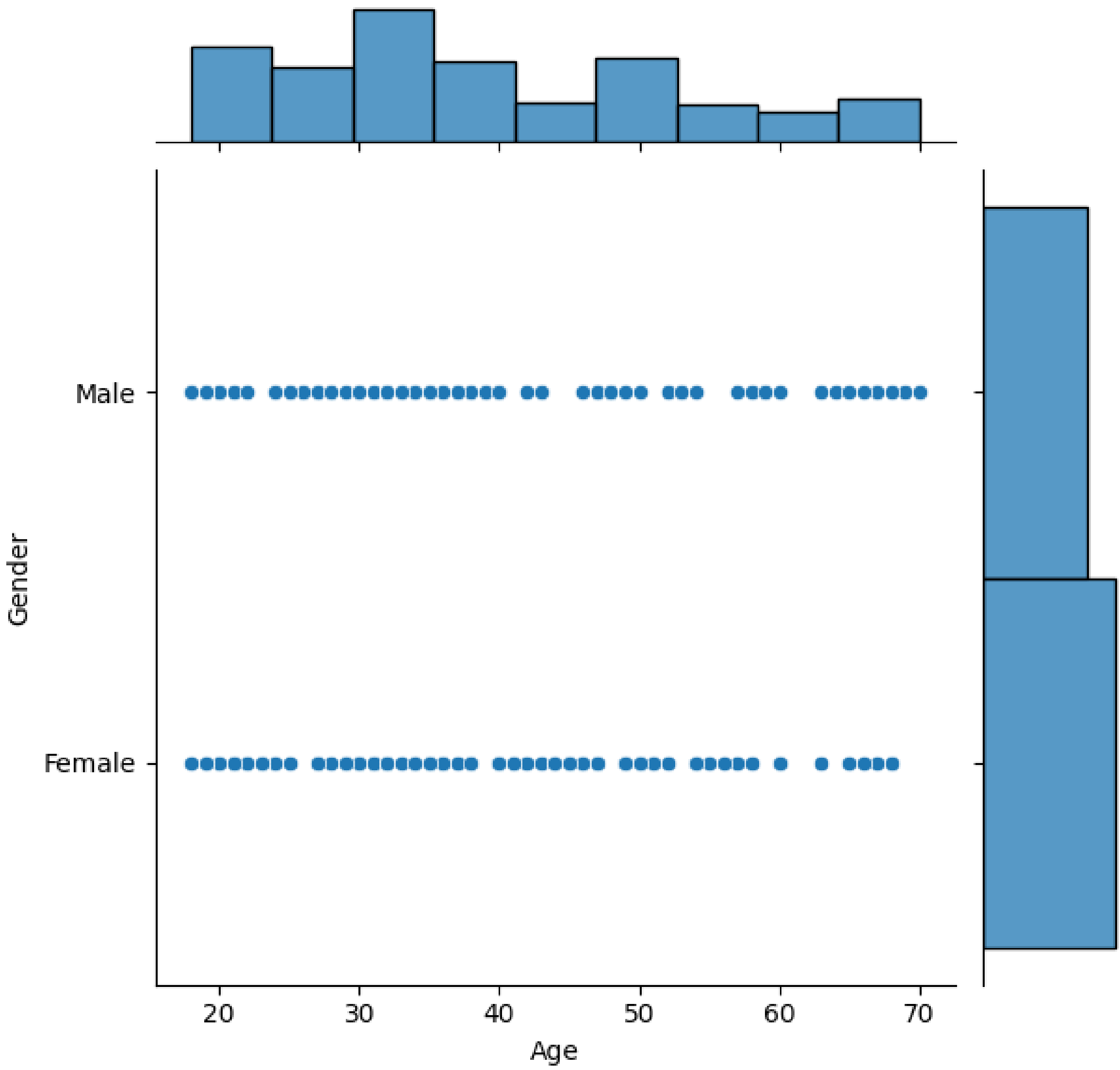
Out[7]:




In [8]:

```
sns.jointplot(x=data.Age,y=data.Gender)
```

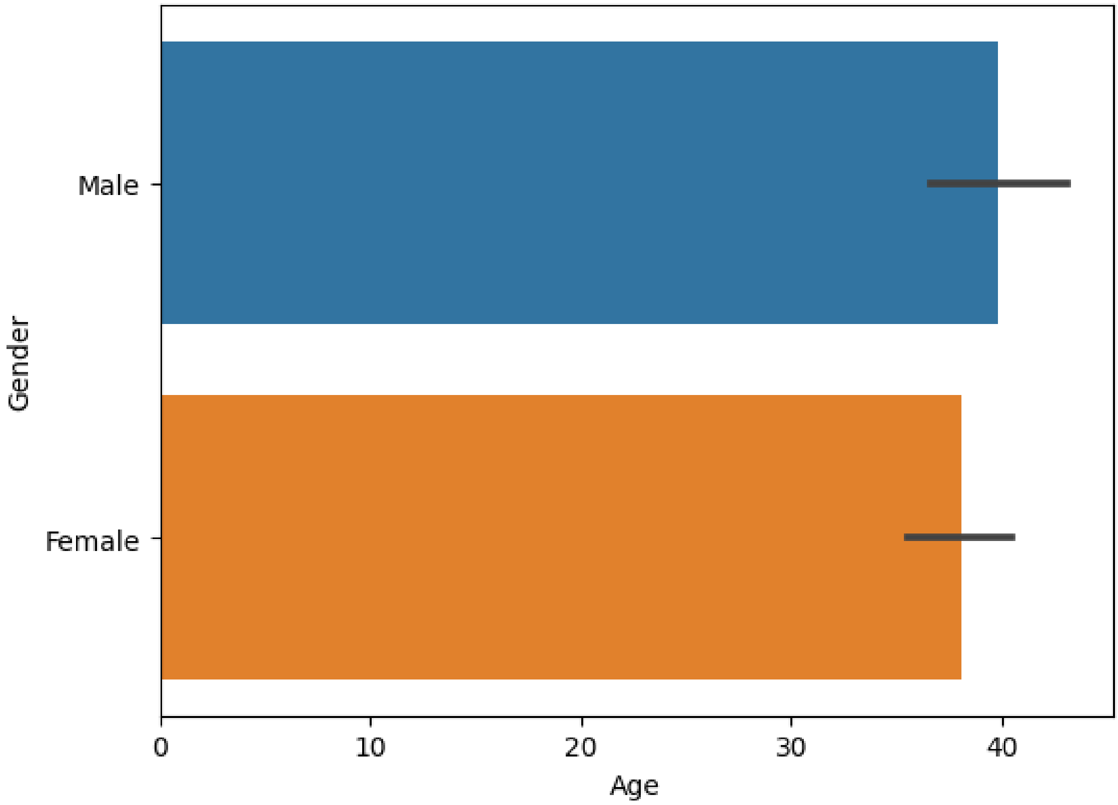
Out[8]:




In [9]:

```
sns.barplot(x=data.Age,y=data.Gender)
```

Out[9]:




# Perform the descriptive statistics on the dataset

In [10]:

```
data.head()
```

Out[10]:

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 |
| **1** | 2 | Male | 21 | 15 | 81 |
| **2** | 3 | Female | 20 | 16 | 6 |
| **3** | 4 | Female | 23 | 16 | 77 |
| **4** | 5 | Female | 31 | 17 | 40 |

In [11]:

```
data.info()
```

```
RangeIndex: 200 entries, 0 to 199
```

```
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

In [12]:

```
data.mean()
```

Out[12]:

```
CustomerID                100.50
Age                        38.85
Annual Income (k$)         60.56
Spending Score (1-100)     50.20
dtype: float64
```

In [13]:

```
data.shape
```

Out[13]:

```
(200, 5)
```

In [14]:

```
data.mode()
```

Out[14]:

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **0** | 1 | Female | 32.0 | 54.0 | 42.0 |
| **1** | 2 | NaN | NaN | 78.0 | NaN |
| **2** | 3 | NaN | NaN | NaN | NaN |
| **3** | 4 | NaN | NaN | NaN | NaN |
| **4** | 5 | NaN | NaN | NaN | NaN |
| **...** | ... | ... | ... | ... | ... |
| **195** | 196 | NaN | NaN | NaN | NaN |
| **196** | 197 | NaN | NaN | NaN | NaN |
| **197** | 198 | NaN | NaN | NaN | NaN |

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **198** | 199 | NaN | NaN | NaN | NaN |
| **199** | 200 | NaN | NaN | NaN | NaN |

200 rows × 5 columns

In [15]:

```
data.median()
```

Out[15]:

```
CustomerID                100.5
Age                        36.0
Annual Income (k$)         61.5
Spending Score (1-100)     50.0
dtype: float64
```

In [16]:

```
data.skew()
```

Out[16]:

```
CustomerID                0.000000
Age                       0.485569
Annual Income (k$)        0.321843
Spending Score (1-100)   -0.047220
dtype: float64
```

In [17]:

```
data.kurtosis()
```

Out[17]:

```
CustomerID               -1.200000
Age                      -0.671573
Annual Income (k$)       -0.098487
Spending Score (1-100)   -0.826629
dtype: float64
```

In [18]:

```
data.std()
```

Out[18]:

```
CustomerID                57.879185
Age                       13.969007
Annual Income (k$)        26.264721
Spending Score (1-100)    25.823522
dtype: float64
```

In [19]:

```
data.var()
```

Out[19]:

```
CustomerID                3350.000000
Age                        195.133166
Annual Income (k$)         689.835578
Spending Score (1-100)     666.854271
dtype: float64
```

# Check the missing values and deal with them

In [20]:

```
data.isnull().any()
```

Out[20]:

```
CustomerID                False
Gender                    False
Age                       False
Annual Income (k$)        False
Spending Score (1-100)    False
dtype: bool
```

In [21]:

```
data.isnull().sum()
```

Out[21]:

```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

In [22]:

```
data.dropna()
```

Out[22]:

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 |
| **1** | 2 | Male | 21 | 15 | 81 |
| **2** | 3 | Female | 20 | 16 | 6 |
| **3** | 4 | Female | 23 | 16 | 77 |
| **4** | 5 | Female | 31 | 17 | 40 |
| **...** | ... | ... | ... | ... | ... |
| **195** | 196 | Female | 35 | 120 | 79 |
| **196** | 197 | Female | 45 | 126 | 28 |
| **197** | 198 | Male | 32 | 126 | 74 |

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **198** | 199 | Male | 32 | 137 | 18 |
| **199** | 200 | Male | 30 | 137 | 83 |

200 rows × 5 columns

# Find the outliers and replace them outliers

In [23]:

```
qnt=data.quantile(q=[0.25,0.75])
qnt
```

Out[23]:

| | CustomerID | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|
| **0.25** | 50.75 | 28.75 | 41.5 | 34.75 |
| **0.75** | 150.25 | 49.00 | 78.0 | 73.00 |

# check the categorical columns and perform encoding

In [24]:

```
from sklearn.preprocessing import LabelEncoder
```

In [25]:

```
le=LabelEncoder()
```

In [26]:

```
data["Age"]=le.fit_transform(data['Age'])
data["Gender"]=le.fit_transform(data['Gender'])
data.head()
```

Out[26]:

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **0** | 1 | 1 | 1 | 15 | 39 |
| **1** | 2 | 1 | 3 | 15 | 81 |

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **2** | 3 | 0 | 2 | 16 | 6 |
| **3** | 4 | 0 | 5 | 16 | 77 |
| **4** | 5 | 0 | 13 | 17 | 40 |

# Split the dataset into dependent and independent variables

In [27]:

```
x=data.iloc[:,:-1].values
y=data.iloc[:,:-1].values
x
```

Out[27]:

```
array([[  1,   1,   1,  15],
       [  2,   1,   3,  15],
       [  3,   0,   2,  16],
       [  4,   0,   5,  16],
       [  5,   0,  13,  17],
       [  6,   0,   4,  17],
       [  7,   0,  17,  18],
       [  8,   0,   5,  18],
       [  9,   1,  44,  19],
       [ 10,   0,  12,  19],
       [ 11,   1,  47,  19],
       [ 12,   0,  17,  19],
       [ 13,   0,  40,  20],
       [ 14,   0,   6,  20],
       [ 15,   1,  19,  20],
       [ 16,   1,   4,  20],
       [ 17,   0,  17,  21],
       [ 18,   1,   2,  21],
       [ 19,   1,  34,  23],
       [ 20,   0,  17,  23],
       [ 21,   1,  17,  24],
       [ 22,   1,   7,  24],
       [ 23,   0,  28,  25],
       [ 24,   1,  13,  25],
       [ 25,   0,  36,  28],
       [ 26,   1,  11,  28],
       [ 27,   0,  27,  28],
       [ 28,   1,  17,  28],
       [ 29,   0,  22,  29],
       [ 30,   0,   5,  29],
       [ 31,   1,  42,  30],
       [ 32,   0,   3,  30],
       [ 33,   1,  35,  33],
```

```
[ 34,   1,   0,  33],
[ 35,   0,  31,  33],
[ 36,   0,   3,  33],
[ 37,   0,  24,  34],
[ 38,   0,  12,  34],
[ 39,   0,  18,  37],
[ 40,   0,   2,  37],
[ 41,   0,  45,  38],
[ 42,   1,   6,  38],
[ 43,   1,  30,  39],
[ 44,   0,  13,  39],
[ 45,   0,  31,  39],
[ 46,   0,   6,  39],
[ 47,   0,  32,  40],
[ 48,   0,   9,  40],
[ 49,   0,  11,  40],
[ 50,   0,  13,  40],
[ 51,   0,  31,  42],
[ 52,   1,  15,  42],
[ 53,   0,  13,  43],
[ 54,   1,  41,  43],
[ 55,   0,  32,  43],
[ 56,   1,  29,  43],
[ 57,   0,  33,  44],
[ 58,   1,  49,  44],
[ 59,   0,   9,  46],
[ 60,   1,  35,  46],
[ 61,   1,  50,  46],
[ 62,   1,   1,  46],
[ 63,   0,  47,  47],
[ 64,   0,  36,  47],
[ 65,   1,  43,  48],
[ 66,   1,   0,  48],
[ 67,   0,  25,  48],
[ 68,   0,  48,  48],
[ 69,   1,   1,  48],
[ 70,   0,  14,  48],
[ 71,   1,  50,  49],
[ 72,   0,  29,  49],
[ 73,   0,  42,  50],
[ 74,   0,  42,  50],
[ 75,   1,  41,  54],
[ 76,   1,   8,  54],
[ 77,   0,  27,  54],
[ 78,   1,  22,  54],
[ 79,   0,   5,  54],
[ 80,   0,  31,  54],
[ 81,   1,  39,  54],
[ 82,   1,  20,  54],
[ 83,   1,  47,  54],
[ 84,   0,  28,  54],
[ 85,   0,   3,  54],
[ 86,   1,  30,  54],
[ 87,   0,  37,  57],
[ 88,   0,   4,  57],
[ 89,   0,  16,  58],
[ 90,   0,  32,  58],
```

```
[ 91,   0,  48,  59],
[ 92,   1,   0,  59],
[ 93,   1,  30,  60],
[ 94,   0,  22,  60],
[ 95,   0,  14,  60],
[ 96,   1,   6,  60],
[ 97,   0,  29,  60],
[ 98,   0,   9,  60],
[ 99,   1,  30,  61],
[100,   1,   2,  61],
[101,   0,   5,  62],
[102,   0,  31,  62],
[103,   1,  47,  62],
[104,   1,   8,  62],
[105,   1,  31,  62],
[106,   0,   3,  62],
[107,   0,  46,  63],
[108,   1,  36,  63],
[109,   1,  48,  63],
[110,   1,  46,  63],
[111,   1,  45,  63],
[112,   0,   1,  63],
[113,   0,  20,  64],
[114,   1,   1,  64],
[115,   0,   0,  65],
[116,   0,   1,  65],
[117,   0,  43,  65],
[118,   0,  31,  65],
[119,   0,  33,  67],
[120,   0,  32,  67],
[121,   1,   9,  67],
[122,   0,  20,  67],
[123,   0,  22,  69],
[124,   1,  21,  69],
[125,   0,   5,  70],
[126,   0,  13,  70],
[127,   1,  25,  71],
[128,   1,  22,  71],
[129,   1,  41,  71],
[130,   1,  20,  71],
[131,   1,  29,  71],
[132,   1,  21,  71],
[133,   0,   7,  72],
[134,   0,  13,  72],
[135,   1,   2,  73],
[136,   0,  11,  73],
[137,   0,  26,  73],
[138,   1,  14,  73],
[139,   1,   1,  74],
[140,   0,  17,  74],
[141,   0,  39,  75],
[142,   1,  14,  75],
[143,   0,  10,  76],
[144,   0,  14,  76],
[145,   1,   7,  77],
[146,   1,  10,  77],
[147,   1,  30,  77],
```

```
       [148,   0,  14,  77],
       [149,   0,  16,  78],
       [150,   1,  16,  78],
       [151,   1,  25,  78],
       [152,   1,  21,  78],
       [153,   0,  26,  78],
       [154,   0,  20,  78],
       [155,   0,  29,  78],
       [156,   0,   9,  78],
       [157,   1,  19,  78],
       [158,   0,  12,  78],
       [159,   1,  16,  78],
       [160,   0,  12,  78],
       [161,   0,  38,  79],
       [162,   0,  11,  79],
       [163,   1,   1,  81],
       [164,   0,  13,  81],
       [165,   1,  32,  85],
       [166,   0,  18,  85],
       [167,   1,  24,  86],
       [168,   0,  15,  86],
       [169,   0,  18,  87],
       [170,   1,  14,  87],
       [171,   1,  22,  87],
       [172,   1,  10,  87],
       [173,   1,  18,  87],
       [174,   1,  18,  87],
       [175,   0,  34,  88],
       [176,   0,  12,  88],
       [177,   1,  40,  88],
       [178,   1,   9,  88],
       [179,   1,  41,  93],
       [180,   1,  17,  93],
       [181,   0,  19,  97],
       [182,   0,  14,  97],
       [183,   1,  28,  98],
       [184,   0,  11,  98],
       [185,   0,  23,  99],
       [186,   1,  12,  99],
       [187,   0,  36, 101],
       [188,   1,  10, 101],
       [189,   0,  23, 103],
       [190,   0,  18, 103],
       [191,   0,  16, 103],
       [192,   0,  14, 103],
       [193,   1,  15, 113],
       [194,   0,  20, 113],
       [195,   0,  29, 120],
       [196,   0,  17, 120],
       [197,   0,  27, 126],
       [198,   1,  14, 126],
       [199,   1,  14, 137],
       [200,   1,  12, 137]], dtype=int64)
```

In [28]:

```
y
```

Out[28]:

```
array([[  1,   1,   1,  15],
       [  2,   1,   3,  15],
       [  3,   0,   2,  16],
       [  4,   0,   5,  16],
       [  5,   0,  13,  17],
       [  6,   0,   4,  17],
       [  7,   0,  17,  18],
       [  8,   0,   5,  18],
       [  9,   1,  44,  19],
       [ 10,   0,  12,  19],
       [ 11,   1,  47,  19],
       [ 12,   0,  17,  19],
       [ 13,   0,  40,  20],
       [ 14,   0,   6,  20],
       [ 15,   1,  19,  20],
       [ 16,   1,   4,  20],
       [ 17,   0,  17,  21],
       [ 18,   1,   2,  21],
       [ 19,   1,  34,  23],
       [ 20,   0,  17,  23],
       [ 21,   1,  17,  24],
       [ 22,   1,   7,  24],
       [ 23,   0,  28,  25],
       [ 24,   1,  13,  25],
       [ 25,   0,  36,  28],
       [ 26,   1,  11,  28],
       [ 27,   0,  27,  28],
       [ 28,   1,  17,  28],
       [ 29,   0,  22,  29],
       [ 30,   0,   5,  29],
       [ 31,   1,  42,  30],
       [ 32,   0,   3,  30],
       [ 33,   1,  35,  33],
       [ 34,   1,   0,  33],
       [ 35,   0,  31,  33],
       [ 36,   0,   3,  33],
       [ 37,   0,  24,  34],
       [ 38,   0,  12,  34],
       [ 39,   0,  18,  37],
       [ 40,   0,   2,  37],
       [ 41,   0,  45,  38],
       [ 42,   1,   6,  38],
       [ 43,   1,  30,  39],
       [ 44,   0,  13,  39],
       [ 45,   0,  31,  39],
       [ 46,   0,   6,  39],
       [ 47,   0,  32,  40],
       [ 48,   0,   9,  40],
       [ 49,   0,  11,  40],
       [ 50,   0,  13,  40],
       [ 51,   0,  31,  42],
       [ 52,   1,  15,  42],
       [ 53,   0,  13,  43],
       [ 54,   1,  41,  43],
       [ 55,   0,  32,  43],
       [ 56,   1,  29,  43],
       [ 57,   0,  33,  44],
```

```
[ 58,   1,  49,  44],
[ 59,   0,   9,  46],
[ 60,   1,  35,  46],
[ 61,   1,  50,  46],
[ 62,   1,   1,  46],
[ 63,   0,  47,  47],
[ 64,   0,  36,  47],
[ 65,   1,  43,  48],
[ 66,   1,   0,  48],
[ 67,   0,  25,  48],
[ 68,   0,  48,  48],
[ 69,   1,   1,  48],
[ 70,   0,  14,  48],
[ 71,   1,  50,  49],
[ 72,   0,  29,  49],
[ 73,   0,  42,  50],
[ 74,   0,  42,  50],
[ 75,   1,  41,  54],
[ 76,   1,   8,  54],
[ 77,   0,  27,  54],
[ 78,   1,  22,  54],
[ 79,   0,   5,  54],
[ 80,   0,  31,  54],
[ 81,   1,  39,  54],
[ 82,   1,  20,  54],
[ 83,   1,  47,  54],
[ 84,   0,  28,  54],
[ 85,   0,   3,  54],
[ 86,   1,  30,  54],
[ 87,   0,  37,  57],
[ 88,   0,   4,  57],
[ 89,   0,  16,  58],
[ 90,   0,  32,  58],
[ 91,   0,  48,  59],
[ 92,   1,   0,  59],
[ 93,   1,  30,  60],
[ 94,   0,  22,  60],
[ 95,   0,  14,  60],
[ 96,   1,   6,  60],
[ 97,   0,  29,  60],
[ 98,   0,   9,  60],
[ 99,   1,  30,  61],
[100,   1,   2,  61],
[101,   0,   5,  62],
[102,   0,  31,  62],
[103,   1,  47,  62],
[104,   1,   8,  62],
[105,   1,  31,  62],
[106,   0,   3,  62],
[107,   0,  46,  63],
[108,   1,  36,  63],
[109,   1,  48,  63],
[110,   1,  46,  63],
[111,   1,  45,  63],
[112,   0,   1,  63],
[113,   0,  20,  64],
[114,   1,   1,  64],
```

```
[115,   0,   0,  65],
[116,   0,   1,  65],
[117,   0,  43,  65],
[118,   0,  31,  65],
[119,   0,  33,  67],
[120,   0,  32,  67],
[121,   1,   9,  67],
[122,   0,  20,  67],
[123,   0,  22,  69],
[124,   1,  21,  69],
[125,   0,   5,  70],
[126,   0,  13,  70],
[127,   1,  25,  71],
[128,   1,  22,  71],
[129,   1,  41,  71],
[130,   1,  20,  71],
[131,   1,  29,  71],
[132,   1,  21,  71],
[133,   0,   7,  72],
[134,   0,  13,  72],
[135,   1,   2,  73],
[136,   0,  11,  73],
[137,   0,  26,  73],
[138,   1,  14,  73],
[139,   1,   1,  74],
[140,   0,  17,  74],
[141,   0,  39,  75],
[142,   1,  14,  75],
[143,   0,  10,  76],
[144,   0,  14,  76],
[145,   1,   7,  77],
[146,   1,  10,  77],
[147,   1,  30,  77],
[148,   0,  14,  77],
[149,   0,  16,  78],
[150,   1,  16,  78],
[151,   1,  25,  78],
[152,   1,  21,  78],
[153,   0,  26,  78],
[154,   0,  20,  78],
[155,   0,  29,  78],
[156,   0,   9,  78],
[157,   1,  19,  78],
[158,   0,  12,  78],
[159,   1,  16,  78],
[160,   0,  12,  78],
[161,   0,  38,  79],
[162,   0,  11,  79],
[163,   1,   1,  81],
[164,   0,  13,  81],
[165,   1,  32,  85],
[166,   0,  18,  85],
[167,   1,  24,  86],
[168,   0,  15,  86],
[169,   0,  18,  87],
[170,   1,  14,  87],
[171,   1,  22,  87],
```

```
       [172,   1,  10,  87],
       [173,   1,  18,  87],
       [174,   1,  18,  87],
       [175,   0,  34,  88],
       [176,   0,  12,  88],
       [177,   1,  40,  88],
       [178,   1,   9,  88],
       [179,   1,  41,  93],
       [180,   1,  17,  93],
       [181,   0,  19,  97],
       [182,   0,  14,  97],
       [183,   1,  28,  98],
       [184,   0,  11,  98],
       [185,   0,  23,  99],
       [186,   1,  12,  99],
       [187,   0,  36, 101],
       [188,   1,  10, 101],
       [189,   0,  23, 103],
       [190,   0,  18, 103],
       [191,   0,  16, 103],
       [192,   0,  14, 103],
       [193,   1,  15, 113],
       [194,   0,  20, 113],
       [195,   0,  29, 120],
       [196,   0,  17, 120],
       [197,   0,  27, 126],
       [198,   1,  14, 126],
       [199,   1,  14, 137],
       [200,   1,  12, 137]], dtype=int64)
```

In [29]:

```
print(x.shape,y.shape)
```

```
(200, 4) (200, 4)
```

# split the data into training and testing

In [30]:

```
from sklearn.model_selection import train_test_split
```

In [31]:

```
x_train, x_test, y_train, y_test= train_test_split(x,y,test_size= 0.33,
random_state= 42)
```

In [32]:

```
x_train.shape
```

Out[32]:

```
(134, 4)
```

In [33]:

```
x_test.shape
```

Out[33]:

```
(66, 4)
```

# scaling the data

In [34]:

```
from sklearn.preprocessing import scale
```

In [35]:

```
x=scale(x)
x
```

Out[35]:

```
array([[-1.7234121 ,  1.12815215, -1.44683356, -1.73899919],
       [-1.70609137,  1.12815215, -1.29979763, -1.73899919],
       [-1.68877065, -0.88640526, -1.37331559, -1.70082976],
       [-1.67144992, -0.88640526, -1.15276169, -1.70082976],
       [-1.6541292 , -0.88640526, -0.56461797, -1.66266033],
       [-1.63680847, -0.88640526, -1.22627966, -1.66266033],
       [-1.61948775, -0.88640526, -0.27054611, -1.62449091],
       [-1.60216702, -0.88640526, -1.15276169, -1.62449091],
       [-1.5848463 ,  1.12815215,  1.71443895, -1.58632148],
       [-1.56752558, -0.88640526, -0.63813594, -1.58632148],
       [-1.55020485,  1.12815215,  1.93499284, -1.58632148],
       [-1.53288413, -0.88640526, -0.27054611, -1.58632148],
       [-1.5155634 , -0.88640526,  1.42036709, -1.54815205],
       [-1.49824268, -0.88640526, -1.07924373, -1.54815205],
       [-1.48092195,  1.12815215, -0.12351018, -1.54815205],
       [-1.46360123,  1.12815215, -1.22627966, -1.54815205],
       [-1.4462805 , -0.88640526, -0.27054611, -1.50998262],
       [-1.42895978,  1.12815215, -1.37331559, -1.50998262],
       [-1.41163905,  1.12815215,  0.9792593 , -1.43364376],
       [-1.39431833, -0.88640526, -0.27054611, -1.43364376],
       [-1.3769976 ,  1.12815215, -0.27054611, -1.39547433],
       [-1.35967688,  1.12815215, -1.00572576, -1.39547433],
       [-1.34235616, -0.88640526,  0.53815151, -1.3573049 ],
       [-1.32503543,  1.12815215, -0.56461797, -1.3573049 ],
       [-1.30771471, -0.88640526,  1.12629523, -1.24279661],
       [-1.29039398,  1.12815215, -0.7116539 , -1.24279661],
       [-1.27307326, -0.88640526,  0.46463354, -1.24279661],
       [-1.25575253,  1.12815215, -0.27054611, -1.24279661],
       [-1.23843181, -0.88640526,  0.09704371, -1.20462718],
       [-1.22111108, -0.88640526, -1.15276169, -1.20462718],
       [-1.20379036,  1.12815215,  1.56740302, -1.16645776],
       [-1.18646963, -0.88640526, -1.29979763, -1.16645776],
       [-1.16914891,  1.12815215,  1.05277726, -1.05194947],
       [-1.15182818,  1.12815215, -1.52035152, -1.05194947],
       [-1.13450746, -0.88640526,  0.7587054 , -1.05194947],
       [-1.11718674, -0.88640526, -1.29979763, -1.05194947],
       [-1.09986601, -0.88640526,  0.24407964, -1.01378004],
       [-1.08254529, -0.88640526, -0.63813594, -1.01378004],
       [-1.06522456, -0.88640526, -0.19702815, -0.89927175],
       [-1.04790384, -0.88640526, -1.37331559, -0.89927175],
       [-1.03058311, -0.88640526,  1.78795691, -0.86110232],
       [-1.01326239,  1.12815215, -1.07924373, -0.86110232],
       [-0.99594166,  1.12815215,  0.68518744, -0.82293289],
       [-0.97862094, -0.88640526, -0.56461797, -0.82293289],
       [-0.96130021, -0.88640526,  0.7587054 , -0.82293289],
       [-0.94397949, -0.88640526, -1.07924373, -0.82293289],
       [-0.92665877, -0.88640526,  0.83222337, -0.78476346],
       [-0.90933804, -0.88640526, -0.85868983, -0.78476346],
       [-0.89201732, -0.88640526, -0.7116539 , -0.78476346],
```

```
[-0.87469659, -0.88640526, -0.56461797, -0.78476346],
[-0.85737587, -0.88640526,  0.7587054 , -0.70842461],
[-0.84005514,  1.12815215, -0.41758204, -0.70842461],
[-0.82273442, -0.88640526, -0.56461797, -0.67025518],
[-0.80541369,  1.12815215,  1.49388505, -0.67025518],
[-0.78809297, -0.88640526,  0.83222337, -0.67025518],
[-0.77077224,  1.12815215,  0.61166947, -0.67025518],
[-0.75345152, -0.88640526,  0.90574133, -0.63208575],
[-0.73613079,  1.12815215,  2.08202878, -0.63208575],
[-0.71881007, -0.88640526, -0.85868983, -0.55574689],
[-0.70148935,  1.12815215,  1.05277726, -0.55574689],
[-0.68416862,  1.12815215,  2.15554674, -0.55574689],
[-0.6668479 ,  1.12815215, -1.44683356, -0.55574689],
[-0.64952717, -0.88640526,  1.93499284, -0.51757746],
[-0.63220645, -0.88640526,  1.12629523, -0.51757746],
[-0.61488572,  1.12815215,  1.64092098, -0.47940803],
[-0.597565  ,  1.12815215, -1.52035152, -0.47940803],
[-0.58024427, -0.88640526,  0.31759761, -0.47940803],
[-0.56292355, -0.88640526,  2.00851081, -0.47940803],
[-0.54560282,  1.12815215, -1.44683356, -0.47940803],
[-0.5282821 , -0.88640526, -0.49110001, -0.47940803],
[-0.51096138,  1.12815215,  2.15554674, -0.4412386 ],
[-0.49364065, -0.88640526,  0.61166947, -0.4412386 ],
[-0.47631993, -0.88640526,  1.56740302, -0.40306917],
[-0.4589992 , -0.88640526,  1.56740302, -0.40306917],
[-0.44167848,  1.12815215,  1.49388505, -0.25039146],
[-0.42435775,  1.12815215, -0.9322078 , -0.25039146],
[-0.40703703, -0.88640526,  0.46463354, -0.25039146],
[-0.3897163 ,  1.12815215,  0.09704371, -0.25039146],
[-0.37239558, -0.88640526, -1.15276169, -0.25039146],
[-0.35507485, -0.88640526,  0.7587054 , -0.25039146],
[-0.33775413,  1.12815215,  1.34684912, -0.25039146],
[-0.3204334 ,  1.12815215, -0.04999222, -0.25039146],
[-0.30311268,  1.12815215,  1.93499284, -0.25039146],
[-0.28579196, -0.88640526,  0.53815151, -0.25039146],
[-0.26847123, -0.88640526, -1.29979763, -0.25039146],
[-0.25115051,  1.12815215,  0.68518744, -0.25039146],
[-0.23382978, -0.88640526,  1.19981319, -0.13588317],
[-0.21650906, -0.88640526, -1.22627966, -0.13588317],
[-0.19918833, -0.88640526, -0.34406408, -0.09771374],
[-0.18186761, -0.88640526,  0.83222337, -0.09771374],
[-0.16454688, -0.88640526,  2.00851081, -0.05954431],
[-0.14722616,  1.12815215, -1.52035152, -0.05954431],
[-0.12990543,  1.12815215,  0.68518744, -0.02137488],
[-0.11258471, -0.88640526,  0.09704371, -0.02137488],
[-0.09526399, -0.88640526, -0.49110001, -0.02137488],
[-0.07794326,  1.12815215, -1.07924373, -0.02137488],
[-0.06062254, -0.88640526,  0.61166947, -0.02137488],
[-0.04330181, -0.88640526, -0.85868983, -0.02137488],
[-0.02598109,  1.12815215,  0.68518744,  0.01679455],
[-0.00866036,  1.12815215, -1.37331559,  0.01679455],
[ 0.00866036, -0.88640526, -1.15276169,  0.05496398],
[ 0.02598109, -0.88640526,  0.7587054 ,  0.05496398],
[ 0.04330181,  1.12815215,  1.93499284,  0.05496398],
[ 0.06062254,  1.12815215, -0.9322078 ,  0.05496398],
[ 0.07794326,  1.12815215,  0.7587054 ,  0.05496398],
[ 0.09526399, -0.88640526, -1.29979763,  0.05496398],
```

```
[ 0.11258471, -0.88640526,  1.86147488,  0.09313341],
[ 0.12990543,  1.12815215,  1.12629523,  0.09313341],
[ 0.14722616,  1.12815215,  2.00851081,  0.09313341],
[ 0.16454688,  1.12815215,  1.86147488,  0.09313341],
[ 0.18186761,  1.12815215,  1.78795691,  0.09313341],
[ 0.19918833, -0.88640526, -1.44683356,  0.09313341],
[ 0.21650906, -0.88640526, -0.04999222,  0.13130284],
[ 0.23382978,  1.12815215, -1.44683356,  0.13130284],
[ 0.25115051, -0.88640526, -1.52035152,  0.16947227],
[ 0.26847123, -0.88640526, -1.44683356,  0.16947227],
[ 0.28579196, -0.88640526,  1.64092098,  0.16947227],
[ 0.30311268, -0.88640526,  0.7587054 ,  0.16947227],
[ 0.3204334 , -0.88640526,  0.90574133,  0.24581112],
[ 0.33775413, -0.88640526,  0.83222337,  0.24581112],
[ 0.35507485,  1.12815215, -0.85868983,  0.24581112],
[ 0.37239558, -0.88640526, -0.04999222,  0.24581112],
[ 0.3897163 , -0.88640526,  0.09704371,  0.32214998],
[ 0.40703703,  1.12815215,  0.02352575,  0.32214998],
[ 0.42435775, -0.88640526, -1.15276169,  0.36031941],
[ 0.44167848, -0.88640526, -0.56461797,  0.36031941],
[ 0.4589992 ,  1.12815215,  0.31759761,  0.39848884],
[ 0.47631993,  1.12815215,  0.09704371,  0.39848884],
[ 0.49364065,  1.12815215,  1.49388505,  0.39848884],
[ 0.51096138,  1.12815215, -0.04999222,  0.39848884],
[ 0.5282821 ,  1.12815215,  0.61166947,  0.39848884],
[ 0.54560282,  1.12815215,  0.02352575,  0.39848884],
[ 0.56292355, -0.88640526, -1.00572576,  0.43665827],
[ 0.58024427, -0.88640526, -0.56461797,  0.43665827],
[ 0.597565  ,  1.12815215, -1.37331559,  0.4748277 ],
[ 0.61488572, -0.88640526, -0.7116539 ,  0.4748277 ],
[ 0.63220645, -0.88640526,  0.39111557,  0.4748277 ],
[ 0.64952717,  1.12815215, -0.49110001,  0.4748277 ],
[ 0.6668479 ,  1.12815215, -1.44683356,  0.51299713],
[ 0.68416862, -0.88640526, -0.27054611,  0.51299713],
[ 0.70148935, -0.88640526,  1.34684912,  0.55116656],
[ 0.71881007,  1.12815215, -0.49110001,  0.55116656],
[ 0.73613079, -0.88640526, -0.78517187,  0.58933599],
[ 0.75345152, -0.88640526, -0.49110001,  0.58933599],
[ 0.77077224,  1.12815215, -1.00572576,  0.62750542],
[ 0.78809297,  1.12815215, -0.78517187,  0.62750542],
[ 0.80541369,  1.12815215,  0.68518744,  0.62750542],
[ 0.82273442, -0.88640526, -0.49110001,  0.62750542],
[ 0.84005514, -0.88640526, -0.34406408,  0.66567484],
[ 0.85737587,  1.12815215, -0.34406408,  0.66567484],
[ 0.87469659,  1.12815215,  0.31759761,  0.66567484],
[ 0.89201732,  1.12815215,  0.02352575,  0.66567484],
[ 0.90933804, -0.88640526,  0.39111557,  0.66567484],
[ 0.92665877, -0.88640526, -0.04999222,  0.66567484],
[ 0.94397949, -0.88640526,  0.61166947,  0.66567484],
[ 0.96130021, -0.88640526, -0.85868983,  0.66567484],
[ 0.97862094,  1.12815215, -0.12351018,  0.66567484],
[ 0.99594166, -0.88640526, -0.63813594,  0.66567484],
[ 1.01326239,  1.12815215, -0.34406408,  0.66567484],
[ 1.03058311, -0.88640526, -0.63813594,  0.66567484],
[ 1.04790384, -0.88640526,  1.27333116,  0.70384427],
[ 1.06522456, -0.88640526, -0.7116539 ,  0.70384427],
[ 1.08254529,  1.12815215, -1.44683356,  0.78018313],
```

```
[ 1.09986601, -0.88640526, -0.56461797,  0.78018313],
[ 1.11718674,  1.12815215,  0.83222337,  0.93286085],
[ 1.13450746, -0.88640526, -0.19702815,  0.93286085],
[ 1.15182818,  1.12815215,  0.24407964,  0.97103028],
[ 1.16914891, -0.88640526, -0.41758204,  0.97103028],
[ 1.18646963, -0.88640526, -0.19702815,  1.00919971],
[ 1.20379036,  1.12815215, -0.49110001,  1.00919971],
[ 1.22111108,  1.12815215,  0.09704371,  1.00919971],
[ 1.23843181,  1.12815215, -0.78517187,  1.00919971],
[ 1.25575253,  1.12815215, -0.19702815,  1.00919971],
[ 1.27307326,  1.12815215, -0.19702815,  1.00919971],
[ 1.29039398, -0.88640526,  0.9792593 ,  1.04736914],
[ 1.30771471, -0.88640526, -0.63813594,  1.04736914],
[ 1.32503543,  1.12815215,  1.42036709,  1.04736914],
[ 1.34235616,  1.12815215, -0.85868983,  1.04736914],
[ 1.35967688,  1.12815215,  1.49388505,  1.23821628],
[ 1.3769976 ,  1.12815215, -0.27054611,  1.23821628],
[ 1.39431833, -0.88640526, -0.12351018,  1.390894  ],
[ 1.41163905, -0.88640526, -0.49110001,  1.390894  ],
[ 1.42895978,  1.12815215,  0.53815151,  1.42906343],
[ 1.4462805 , -0.88640526, -0.7116539 ,  1.42906343],
[ 1.46360123, -0.88640526,  0.17056168,  1.46723286],
[ 1.48092195,  1.12815215, -0.63813594,  1.46723286],
[ 1.49824268, -0.88640526,  1.12629523,  1.54357172],
[ 1.5155634 ,  1.12815215, -0.78517187,  1.54357172],
[ 1.53288413, -0.88640526,  0.17056168,  1.61991057],
[ 1.55020485, -0.88640526, -0.19702815,  1.61991057],
[ 1.56752558, -0.88640526, -0.34406408,  1.61991057],
[ 1.5848463 , -0.88640526, -0.49110001,  1.61991057],
[ 1.60216702,  1.12815215, -0.41758204,  2.00160487],
[ 1.61948775, -0.88640526, -0.04999222,  2.00160487],
[ 1.63680847, -0.88640526,  0.61166947,  2.26879087],
[ 1.6541292 , -0.88640526, -0.27054611,  2.26879087],
[ 1.67144992, -0.88640526,  0.46463354,  2.49780745],
[ 1.68877065,  1.12815215, -0.49110001,  2.49780745],
[ 1.70609137,  1.12815215, -0.49110001,  2.91767117],
[ 1.7234121 ,  1.12815215, -0.63813594,  2.91767117]])
```

# Build the model

In [39]:

```
from sklearn.linear_model import LinearRegression
```

In [40]:

```
regressor=LinearRegression()

regressor.fit(x_train,y_train)
```

Out[40]:

LinearRegression()

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**
**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

In [41]:

```
from sklearn.tree import DecisionTreeClassifier
```

In [42]:

```
model=DecisionTreeClassifier()
model.fit(x_train,y_train)
```

Out[42]:

DecisionTreeClassifier()
**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**
**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

# Train the model

In [84]:

```
from sklearn.multioutput import MultiOutputClassifier
```

In [85]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [86]:

```
knn=KNeighborsClassifier(n_neighbors=5)
```

In [87]:

```
knn.fit(x_train,y_train)
knn.fit(x_test,y_test)
```

Out[87]:

KNeighborsClassifier()
**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**
**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

# Measure the performance using evalution metrics

In [88]:

```
pred=knn.predict(x_test)

pred
```

Out[88]:

```
array([[ 79,   0,   3,  60],
       [  6,   0,   4,  17],
       [ 13,   0,  18,  20],
       [149,   0,  16,  78],
       [118,   0,  25,  71],
       [115,   0,   0,  65],
       [ 67,   0,   1,  48],
       [166,   1,  18,  87],
       [165,   1,  18,  87],
       [ 30,   0,   3,  29],
       [ 56,   0,  14,  48],
```

```
[171,   1,  18,  87],
[166,   1,  18,  87],
[ 69,   0,   1,  54],
[171,   1,  18,  87],
[170,   1,  18,  87],
[ 56,   0,  25,  48],
[144,   0,  16,  78],
[ 68,   0,  31,  54],
[ 66,   1,   0,  48],
[115,   0,   0,  65],
[ 10,   0,  17,  23],
[144,   0,  16,  78],
[ 94,   0,   9,  60],
[ 66,   1,   0,  48],
[ 56,   0,  29,  43],
[ 76,   0,   3,  54],
[ 57,   0,  33,  44],
[123,   0,   5,  70],
[125,   0,   5,  70],
[  6,   0,   4,  17],
[ 13,   0,  17,  23],
[ 39,   0,  14,  48],
[ 69,   0,   1,  54],
[144,   0,  16,  78],
[ 94,   0,  31,  60],
[126,   0,  14,  73],
[133,   0,  14,  73],
[153,   0,  18,  85],
[ 67,   0,  25,  54],
[ 67,   0,  25,  54],
[178,   1,   9,  88],
[ 30,   0,   3,  29],
[ 13,   0,  17,  23],
[118,   0,  13,  65],
[178,   1,   9,  88],
[ 16,   0,   3,  20],
[ 10,   0,  17,  19],
[136,   0,  14,  73],
[ 77,   0,  27,  54],
[ 98,   0,   0,  65],
[166,   1,  18,  87],
[  6,   0,   4,  17],
[118,   0,  13,  65],
[105,   0,  31,  62],
[ 68,   0,  31,  54],
[127,   1,  25,  71],
[ 87,   0,  22,  60],
[166,   1,  18,  87],
[ 87,   0,  22,  60],
[166,   1,  18,  87],
[ 85,   0,   0,  60],
[ 30,   0,   3,  29],
[ 13,   0,  17,  20],
[ 30,   0,   3,  29],
[118,   0,  22,  71]], dtype=int64)
```

# Perform any clustering algorithm

In [96]:

```
from sklearn import datasets
```

In [97]:

```
dir(datasets)
```

Out[97]:

```
['__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 '_arff_parser',
 '_base',
 '_california_housing',
 '_covtype',
 '_kddcup99',
 '_lfw',
 '_olivetti_faces',
 '_openml',
 '_rcv1',
 '_samples_generator',
 '_species_distributions',
 '_svmlight_format_fast',
 '_svmlight_format_io',
 '_twenty_newsgroups',
 'clear_data_home',
 'dump_svmlight_file',
 'fetch_20newsgroups',
 'fetch_20newsgroups_vectorized',
 'fetch_california_housing',
 'fetch_covtype',
 'fetch_kddcup99',
 'fetch_lfw_pairs',
 'fetch_lfw_people',
 'fetch_olivetti_faces',
 'fetch_openml',
 'fetch_rcv1',
 'fetch_species_distributions',
 'get_data_home',
 'load_boston',
 'load_breast_cancer',
 'load_diabetes',
 'load_digits',
 'load_files',
 'load_iris',
 'load_linnerud',
 'load_sample_image',
 'load_sample_images',
```

```
 'load_svmlight_file',
 'load_svmlight_files',
 'load_wine',
 'make_biclusters',
 'make_blobs',
 'make_checkerboard',
 'make_circles',
 'make_classification',
 'make_friedman1',
 'make_friedman2',
 'make_friedman3',
 'make_gaussian_quantiles',
 'make_hastie_10_2',
 'make_low_rank_matrix',
 'make_moons',
 'make_multilabel_classification',
 'make_regression',
 'make_s_curve',
 'make_sparse_coded_signal',
 'make_sparse_spd_matrix',
 'make_sparse_uncorrelated',
 'make_spd_matrix',
 'make_swiss_roll']
```

In [98]:

```
wine=datasets.load_wine()
```

In [99]:

```
print(wine)
```

```
{'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00
,
        1.065e+03],
       [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
        1.050e+03],
       [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
        1.185e+03],
       ...,
       [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
        8.350e+02],
       [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
        8.400e+02],
       [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
        5.600e+02]]), 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2]), 'frame': None, 'target_names': array(['class_0', 'class_1',
'class_2'], dtype='
```