Assignment -4

Assignment Date	18 October 2022
Student Name	Gita Prakash N
Student Roll Number	811519104032
Maximum Marks	2 Marks

[1]:
```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
import pickle
import numpy as np
from sklearn.preprocessing import StandardScaler
```

[2]:
```
df=pd.read_csv(r"C:\Users\Admin\Downloads\Mall_Customers.csv")
```

[3]:
```
df.head()
```

[3]:

	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

[4]:
```
df.shape
```

[4]: (200, 5)

Checking the Null Values

[5]:
```
df.isnull().sum()
```

[5]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

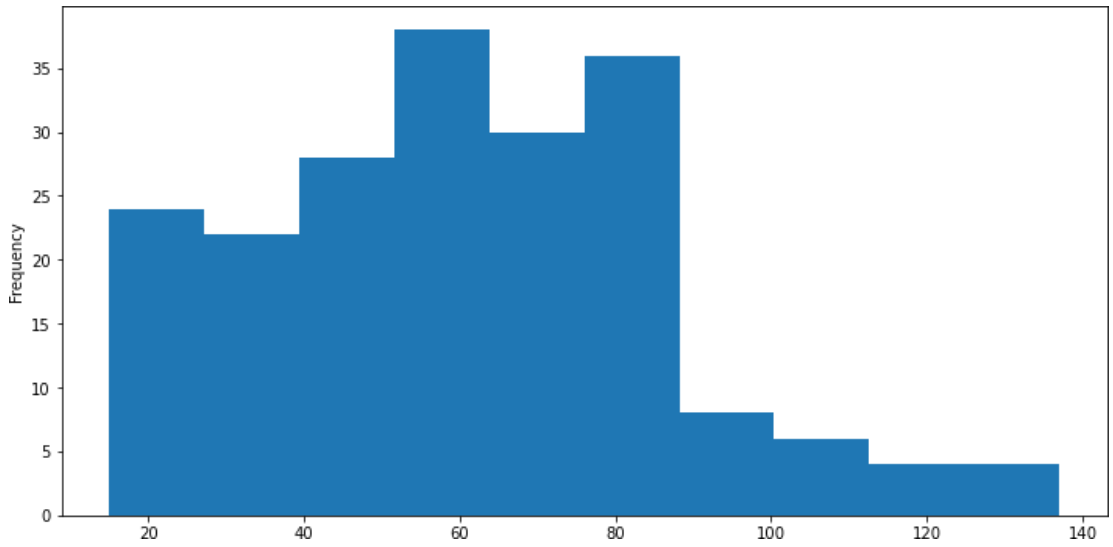
Renaming The Columns

[6]:
```
df.columns=['CustomerID','Gender','Age','Annual_Income','Spending_Score']
```

Univariant Plot

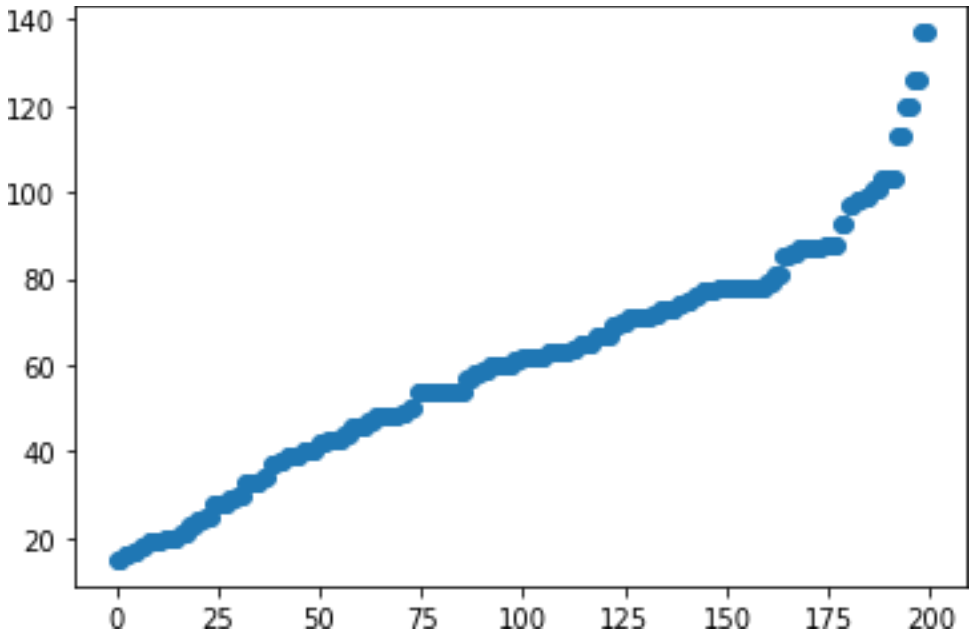
[7]:
```
plt.figure()
df.Annual_Income.plot(kind='hist' ,figsize=(12,6))
```

[7] : <AxesSubplot:ylabel='Frequency'>



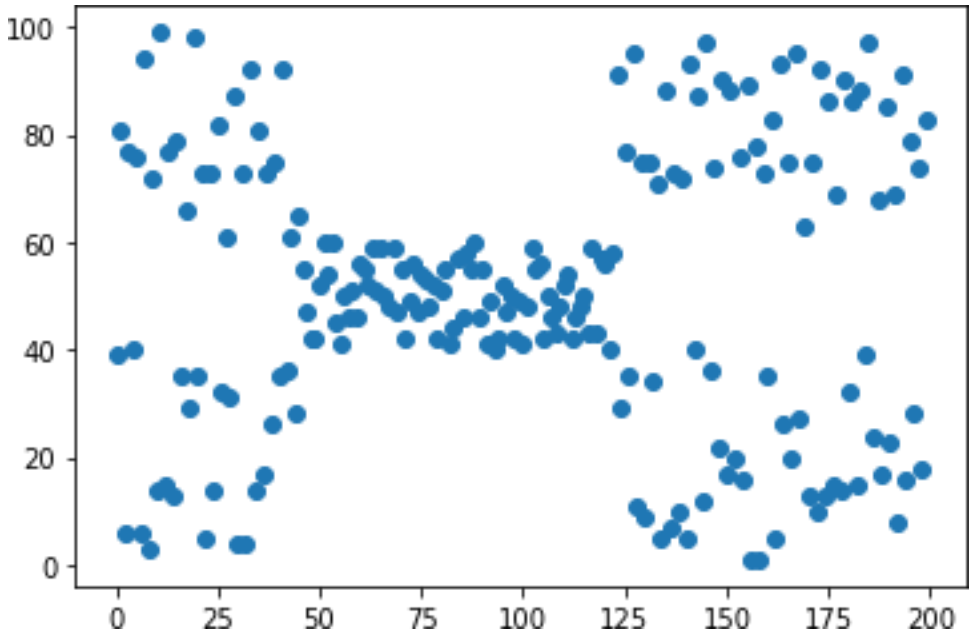
[8]:
```
plt.scatter(df.index,df['Annual_Income'])
```

[8] : <matplotlib.collections.PathCollection at 0x15b1393a340>



[9]:
```
plt.scatter(df.index,df["Spending_Score"])
```

[9] : <matplotlib.collections.PathCollection at 0x15b139bb7f0>

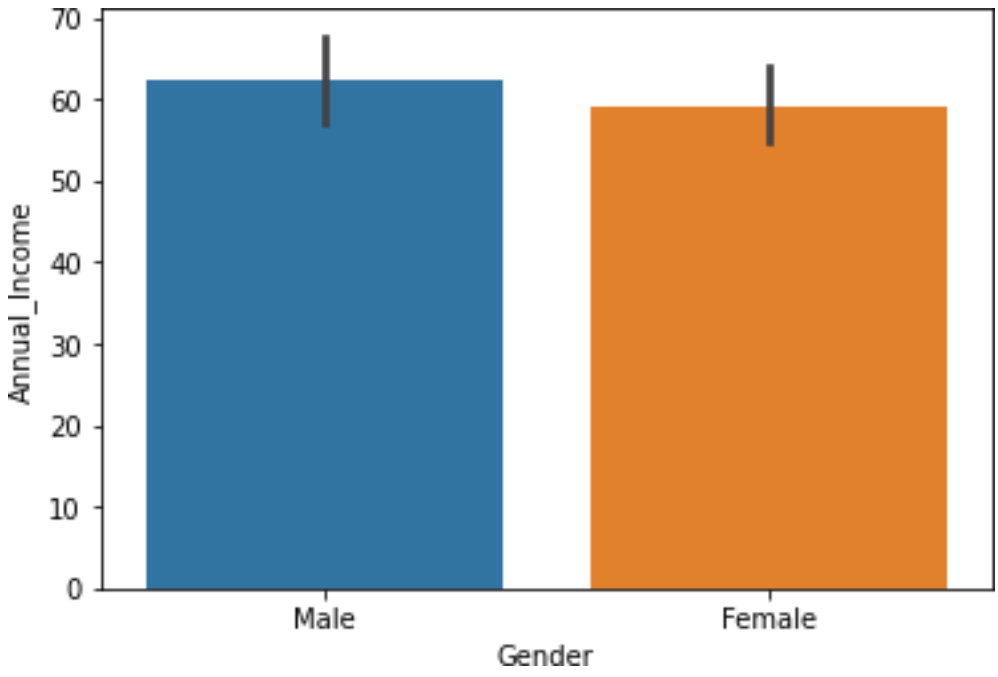


[10]:
```
# z score computation
outliers=[]
def detect_outliers(data):
        threshold=3
        mean=np.mean(data)
        std=np.std(data)

        for i in data:
            z_score=(i-mean)/std
            if np.abs(z_score)>threshold:
                outliers.append(i)

        return outliers
```

Bivariant Plot

[11]:
```
sns.barplot(x="Gender",y="Annual_Income",data=df)
```

[11] : <AxesSubplot:xlabel='Gender', ylabel='Annual_Income'>



[12]:
```
df
```

[12] :

	CustomerID	Gender	Age	Annual_Income	Spending_Score
0	1	Male	19	15	39
1	2	Male	21	15	81

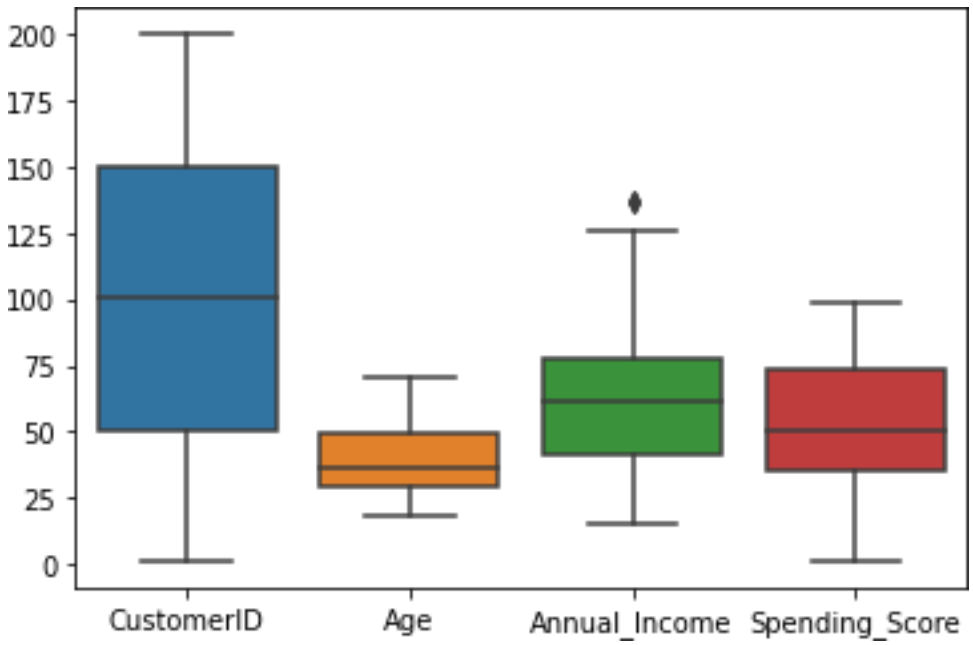
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
..	...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

[200 rows x 5 columns]

Checking Outliers

[13]:
```
sns.boxplot(data=df)
```

[13] : <AxesSubplot:>



Description About the Dataset

[14]:
```
df.describe()
```

[14] :

	CustomerID	Age	Annual_Income	Spending_Score
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000

25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

[15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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   200 non-null    int64 
 4   Spending_Score  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

Label Encoding for Gender

[16]:
```
df["Gender"]=df["Gender"].replace(["Male","Female"],[0,1])
```

[17]:
```
df.head()
```

[17]:

	CustomerID	Gender	Age	Annual_Income	Spending_Score
0	1	0	19	15	39
1	2	0	21	15	81
2	3	1	20	16	6
3	4	1	23	16	77
4	5	1	31	17	40

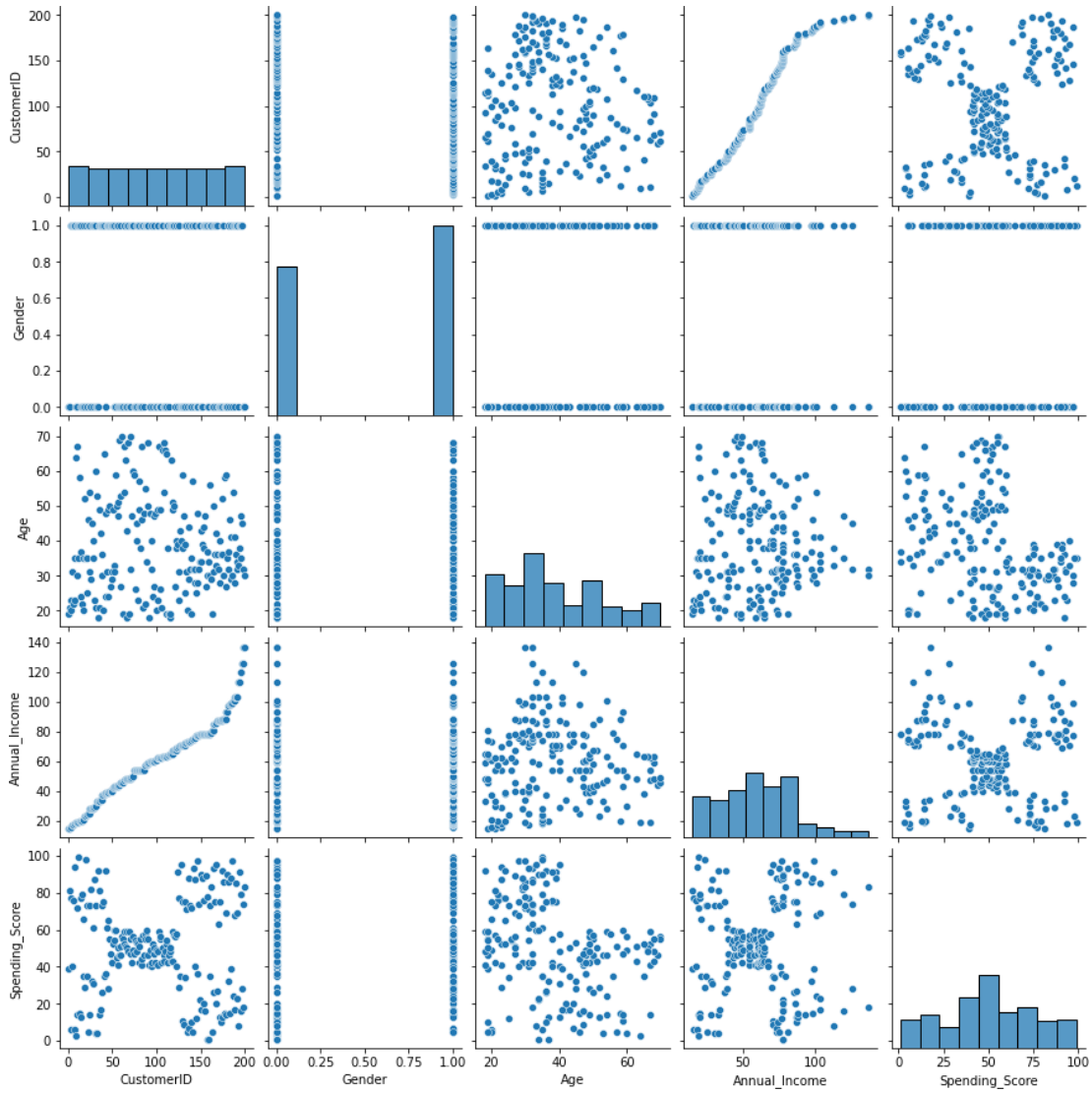
[18]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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    int64 
 2   Age             200 non-null    int64 
 3   Annual_Income   200 non-null    int64 
 4   Spending_Score  200 non-null    int64 
dtypes: int64(5)
memory usage: 7.9 KB
```

Multivariant Plot

[19]:
```
sns.pairplot(df)
```

[19] : <seaborn.axisgrid.PairGrid at 0x15b13bb63a0>



Scaling The Data

[20]:
```
scaler = StandardScaler()
scaler.fit(df)
```

[20] : StandardScaler()

[21]:
```
print(scaler.transform(df))
```

```
[[-1.7234121  -1.12815215 -1.42456879 -1.73899919 -0.43480148]
```

```
[-1.70609137 -1.12815215 -1.28103541 -1.73899919  1.19570407]
[-1.68877065  0.88640526 -1.3528021  -1.70082976 -1.71591298]
[-1.67144992  0.88640526 -1.13750203 -1.70082976  1.04041783]
[-1.6541292   0.88640526 -0.56336851 -1.66266033 -0.39597992]
[-1.63680847  0.88640526 -1.20926872 -1.66266033  1.00159627]
[-1.61948775  0.88640526 -0.27630176 -1.62449091 -1.71591298]
[-1.60216702  0.88640526 -1.13750203 -1.62449091  1.70038436]
[-1.5848463  -1.12815215  1.80493225 -1.58632148 -1.83237767]
[-1.56752558  0.88640526 -0.6351352  -1.58632148  0.84631002]
[-1.55020485 -1.12815215  2.02023231 -1.58632148 -1.4053405 ]
[-1.53288413  0.88640526 -0.27630176 -1.58632148  1.89449216]
[-1.5155634   0.88640526  1.37433211 -1.54815205 -1.36651894]
[-1.49824268  0.88640526 -1.06573534 -1.54815205  1.04041783]
[-1.48092195 -1.12815215 -0.13276838 -1.54815205 -1.44416206]
[-1.46360123 -1.12815215 -1.20926872 -1.54815205  1.11806095]
[-1.4462805   0.88640526 -0.27630176 -1.50998262 -0.59008772]
[-1.42895978 -1.12815215 -1.3528021  -1.50998262  0.61338066]
[-1.41163905 -1.12815215  0.94373197 -1.43364376 -0.82301709]
[-1.39431833  0.88640526 -0.27630176 -1.43364376  1.8556706 ]
[-1.3769976  -1.12815215 -0.27630176 -1.39547433 -0.59008772]
[-1.35967688 -1.12815215 -0.99396865 -1.39547433  0.88513158]
[-1.34235616  0.88640526  0.51313183 -1.3573049  -1.75473454]
[-1.32503543 -1.12815215 -0.56336851 -1.3573049   0.88513158]
[-1.30771471  0.88640526  1.08726535 -1.24279661 -1.4053405 ]
[-1.29039398 -1.12815215 -0.70690189 -1.24279661  1.23452563]
[-1.27307326  0.88640526  0.44136514 -1.24279661 -0.7065524 ]
[-1.25575253 -1.12815215 -0.27630176 -1.24279661  0.41927286]
[-1.23843181  0.88640526  0.08253169 -1.20462718 -0.74537397]
[-1.22111108  0.88640526 -1.13750203 -1.20462718  1.42863343]
[-1.20379036 -1.12815215  1.51786549 -1.16645776 -1.7935561 ]
[-1.18646963  0.88640526 -1.28103541 -1.16645776  0.88513158]
[-1.16914891 -1.12815215  1.01549866 -1.05194947 -1.7935561 ]
[-1.15182818 -1.12815215 -1.49633548 -1.05194947  1.62274124]
[-1.13450746  0.88640526  0.7284319  -1.05194947 -1.4053405 ]
[-1.11718674  0.88640526 -1.28103541 -1.05194947  1.19570407]
[-1.09986601  0.88640526  0.22606507 -1.01378004 -1.28887582]
[-1.08254529  0.88640526 -0.6351352  -1.01378004  0.88513158]
[-1.06522456  0.88640526 -0.20453507 -0.89927175 -0.93948177]
[-1.04790384  0.88640526 -1.3528021  -0.89927175  0.96277471]
[-1.03058311  0.88640526  1.87669894 -0.86110232 -0.59008772]
[-1.01326239 -1.12815215 -1.06573534 -0.86110232  1.62274124]
[-0.99594166 -1.12815215  0.65666521 -0.82293289 -0.55126616]
[-0.97862094  0.88640526 -0.56336851 -0.82293289  0.41927286]
[-0.96130021  0.88640526  0.7284319  -0.82293289 -0.86183865]
[-0.94397949  0.88640526 -1.06573534 -0.82293289  0.5745591 ]
[-0.92665877  0.88640526  0.80019859 -0.78476346  0.18634349]
[-0.90933804  0.88640526 -0.85043527 -0.78476346 -0.12422899]
[-0.89201732  0.88640526 -0.70690189 -0.78476346 -0.3183368 ]
```

```
[-0.87469659  0.88640526 -0.56336851 -0.78476346 -0.3183368 ]
[-0.85737587  0.88640526  0.7284319  -0.70842461  0.06987881]
[-0.84005514 -1.12815215 -0.41983513 -0.70842461  0.38045129]
[-0.82273442  0.88640526 -0.56336851 -0.67025518  0.14752193]
[-0.80541369 -1.12815215  1.4460988  -0.67025518  0.38045129]
[-0.78809297  0.88640526  0.80019859 -0.67025518 -0.20187212]
[-0.77077224 -1.12815215  0.58489852 -0.67025518 -0.35715836]
[-0.75345152  0.88640526  0.87196528 -0.63208575 -0.00776431]
[-0.73613079 -1.12815215  2.16376569 -0.63208575 -0.16305055]
[-0.71881007  0.88640526 -0.85043527 -0.55574689  0.03105725]
[-0.70148935 -1.12815215  1.01549866 -0.55574689 -0.16305055]
[-0.68416862 -1.12815215  2.23553238 -0.55574689  0.22516505]
[-0.6668479  -1.12815215 -1.42456879 -0.55574689  0.18634349]
[-0.64952717  0.88640526  2.02023231 -0.51757746  0.06987881]
[-0.63220645  0.88640526  1.08726535 -0.51757746  0.34162973]
[-0.61488572 -1.12815215  1.73316556 -0.47940803  0.03105725]
[-0.597565   -1.12815215 -1.49633548 -0.47940803  0.34162973]
[-0.58024427  0.88640526  0.29783176 -0.47940803 -0.00776431]
[-0.56292355  0.88640526  2.091999   -0.47940803 -0.08540743]
[-0.54560282 -1.12815215 -1.42456879 -0.47940803  0.34162973]
[-0.5282821   0.88640526 -0.49160182 -0.47940803 -0.12422899]
[-0.51096138 -1.12815215  2.23553238 -0.4412386   0.18634349]
[-0.49364065  0.88640526  0.58489852 -0.4412386  -0.3183368 ]
[-0.47631993  0.88640526  1.51786549 -0.40306917 -0.04658587]
[-0.4589992   0.88640526  1.51786549 -0.40306917  0.22516505]
[-0.44167848 -1.12815215  1.4460988  -0.25039146 -0.12422899]
[-0.42435775 -1.12815215 -0.92220196 -0.25039146  0.14752193]
[-0.40703703  0.88640526  0.44136514 -0.25039146  0.10870037]
[-0.3897163  -1.12815215  0.08253169 -0.25039146 -0.08540743]
[-0.37239558  0.88640526 -1.13750203 -0.25039146  0.06987881]
[-0.35507485  0.88640526  0.7284319  -0.25039146 -0.3183368 ]
[-0.33775413 -1.12815215  1.30256542 -0.25039146  0.03105725]
[-0.3204334  -1.12815215 -0.06100169 -0.25039146  0.18634349]
[-0.30311268 -1.12815215  2.02023231 -0.25039146 -0.35715836]
[-0.28579196  0.88640526  0.51313183 -0.25039146 -0.24069368]
[-0.26847123  0.88640526 -1.28103541 -0.25039146  0.26398661]
[-0.25115051 -1.12815215  0.65666521 -0.25039146 -0.16305055]
[-0.23382978  0.88640526  1.15903204 -0.13588317  0.30280817]
[-0.21650906  0.88640526 -1.20926872 -0.13588317  0.18634349]
[-0.19918833  0.88640526 -0.34806844 -0.09771374  0.38045129]
[-0.18186761  0.88640526  0.80019859 -0.09771374 -0.16305055]
[-0.16454688  0.88640526  2.091999   -0.05954431  0.18634349]
[-0.14722616 -1.12815215 -1.49633548 -0.05954431 -0.35715836]
[-0.12990543 -1.12815215  0.65666521 -0.02137488 -0.04658587]
[-0.11258471  0.88640526  0.08253169 -0.02137488 -0.39597992]
[-0.09526399  0.88640526 -0.49160182 -0.02137488 -0.3183368 ]
[-0.07794326 -1.12815215 -1.06573534 -0.02137488  0.06987881]
[-0.06062254  0.88640526  0.58489852 -0.02137488 -0.12422899]
```

```
[-0.04330181  0.88640526 -0.85043527  -0.02137488 -0.00776431]
[-0.02598109  -1.12815215  0.65666521  0.01679455 -0.3183368 ]
[-0.00866036  -1.12815215 -1.3528021   0.01679455 -0.04658587]
[ 0.00866036  0.88640526 -1.13750203  0.05496398 -0.35715836]
[ 0.02598109  0.88640526  0.7284319   0.05496398 -0.08540743]
[ 0.04330181  -1.12815215  2.02023231  0.05496398  0.34162973]
[ 0.06062254  -1.12815215 -0.92220196  0.05496398  0.18634349]
[ 0.07794326  -1.12815215  0.7284319   0.05496398  0.22516505]
[ 0.09526399  0.88640526 -1.28103541  0.05496398 -0.3183368 ]
[ 0.11258471  0.88640526  1.94846562  0.09313341 -0.00776431]
[ 0.12990543  -1.12815215  1.08726535  0.09313341 -0.16305055]
[ 0.14722616  -1.12815215   2.091999   0.09313341 -0.27951524]
[ 0.16454688  -1.12815215  1.94846562  0.09313341 -0.08540743]
[ 0.18186761  -1.12815215  1.87669894  0.09313341  0.06987881]
[ 0.19918833  0.88640526 -1.42456879  0.09313341  0.14752193]
[ 0.21650906  0.88640526 -0.06100169  0.13130284 -0.3183368 ]
[ 0.23382978  -1.12815215 -1.42456879  0.13130284 -0.16305055]
[ 0.25115051  0.88640526 -1.49633548  0.16947227 -0.08540743]
[ 0.26847123  0.88640526 -1.42456879  0.16947227 -0.00776431]
[ 0.28579196  0.88640526  1.73316556  0.16947227 -0.27951524]
[ 0.30311268  0.88640526  0.7284319   0.16947227  0.34162973]
[ 0.3204334   0.88640526  0.87196528  0.24581112 -0.27951524]
[ 0.33775413  0.88640526  0.80019859  0.24581112  0.26398661]
[ 0.35507485  -1.12815215 -0.85043527  0.24581112  0.22516505]
[ 0.37239558  0.88640526 -0.06100169  0.24581112 -0.39597992]
[ 0.3897163   0.88640526  0.08253169  0.32214998  0.30280817]
[ 0.40703703  -1.12815215   0.010765   0.32214998  1.58391968]
[ 0.42435775  0.88640526 -1.13750203  0.36031941 -0.82301709]
[ 0.44167848  0.88640526 -0.56336851  0.36031941  1.04041783]
[ 0.4589992   -1.12815215  0.29783176  0.39848884 -0.59008772]
[ 0.47631993  -1.12815215  0.08253169  0.39848884  1.73920592]
[ 0.49364065  -1.12815215   1.4460988   0.39848884 -1.52180518]
[ 0.51096138  -1.12815215 -0.06100169  0.39848884  0.96277471]
[ 0.5282821   -1.12815215  0.58489852  0.39848884 -1.5994483 ]
[ 0.54560282  -1.12815215   0.010765   0.39848884  0.96277471]
[ 0.56292355  0.88640526 -0.99396865  0.43665827 -0.62890928]
[ 0.58024427  0.88640526 -0.56336851  0.43665827  0.80748846]
[ 0.597565    -1.12815215 -1.3528021   0.4748277  -1.75473454]
[ 0.61488572  0.88640526 -0.70690189   0.4748277   1.46745499]
[ 0.63220645  0.88640526  0.36959845   0.4748277  -1.67709142]
[ 0.64952717  -1.12815215 -0.49160182   0.4748277   0.88513158]
[ 0.6668479   -1.12815215 -1.42456879  0.51299713 -1.56062674]
[ 0.68416862  0.88640526 -0.27630176  0.51299713  0.84631002]
[ 0.70148935  0.88640526  1.30256542  0.55116656 -1.75473454]
[ 0.71881007  -1.12815215 -0.49160182  0.55116656   1.6615628 ]
[ 0.73613079  0.88640526 -0.77866858  0.58933599 -0.39597992]
[ 0.75345152  0.88640526 -0.49160182  0.58933599  1.42863343]
[ 0.77077224  -1.12815215 -0.99396865  0.62750542 -1.48298362]
```

```
[ 0.78809297 -1.12815215 -0.77866858  0.62750542  1.81684904]
[ 0.80541369 -1.12815215  0.65666521  0.62750542 -0.55126616]
[ 0.82273442  0.88640526 -0.49160182  0.62750542  0.92395314]
[ 0.84005514  0.88640526 -0.34806844  0.66567484 -1.09476801]
[ 0.85737587 -1.12815215 -0.34806844  0.66567484  1.54509812]
[ 0.87469659 -1.12815215  0.29783176  0.66567484 -1.28887582]
[ 0.89201732 -1.12815215   0.010765   0.66567484  1.46745499]
[ 0.90933804  0.88640526  0.36959845  0.66567484 -1.17241113]
[ 0.92665877  0.88640526 -0.06100169  0.66567484  1.00159627]
[ 0.94397949  0.88640526  0.58489852  0.66567484 -1.32769738]
[ 0.96130021  0.88640526 -0.85043527  0.66567484  1.50627656]
[ 0.97862094 -1.12815215 -0.13276838  0.66567484 -1.91002079]
[ 0.99594166  0.88640526 -0.6351352   0.66567484  1.07923939]
[ 1.01326239 -1.12815215 -0.34806844  0.66567484 -1.91002079]
[ 1.03058311  0.88640526 -0.6351352   0.66567484  0.88513158]
[ 1.04790384  0.88640526  1.23079873  0.70384427 -0.59008772]
[ 1.06522456  0.88640526 -0.70690189  0.70384427  1.27334719]
[ 1.08254529 -1.12815215 -1.42456879  0.78018313 -1.75473454]
[ 1.09986601  0.88640526 -0.56336851  0.78018313  1.6615628 ]
[ 1.11718674 -1.12815215  0.80019859  0.93286085 -0.93948177]
[ 1.13450746  0.88640526 -0.20453507  0.93286085  0.96277471]
[ 1.15182818 -1.12815215  0.22606507  0.97103028 -1.17241113]
[ 1.16914891  0.88640526 -0.41983513  0.97103028  1.73920592]
[ 1.18646963  0.88640526 -0.20453507  1.00919971 -0.90066021]
[ 1.20379036 -1.12815215 -0.49160182  1.00919971  0.49691598]
[ 1.22111108 -1.12815215  0.08253169  1.00919971 -1.44416206]
[ 1.23843181 -1.12815215 -0.77866858  1.00919971  0.96277471]
[ 1.25575253 -1.12815215 -0.20453507  1.00919971 -1.56062674]
[ 1.27307326 -1.12815215 -0.20453507  1.00919971  1.62274124]
[ 1.29039398  0.88640526  0.94373197  1.04736914 -1.44416206]
[ 1.30771471  0.88640526 -0.6351352   1.04736914  1.38981187]
[ 1.32503543 -1.12815215  1.37433211  1.04736914 -1.36651894]
[ 1.34235616 -1.12815215 -0.85043527  1.04736914  0.72984534]
[ 1.35967688 -1.12815215   1.4460988  1.23821628 -1.4053405 ]
[ 1.3769976  -1.12815215 -0.27630176  1.23821628  1.54509812]
[ 1.39431833  0.88640526 -0.13276838  1.390894   -0.7065524 ]
[ 1.41163905  0.88640526 -0.49160182  1.390894    1.38981187]
[ 1.42895978 -1.12815215  0.51313183  1.42906343 -1.36651894]
[ 1.4462805   0.88640526 -0.70690189  1.42906343  1.46745499]
[ 1.46360123  0.88640526  0.15429838  1.46723286 -0.43480148]
[ 1.48092195 -1.12815215 -0.6351352   1.46723286  1.81684904]
[ 1.49824268  0.88640526  1.08726535  1.54357172 -1.01712489]
[ 1.5155634  -1.12815215 -0.77866858  1.54357172  0.69102378]
[ 1.53288413  0.88640526  0.15429838  1.61991057 -1.28887582]
[ 1.55020485  0.88640526 -0.20453507  1.61991057  1.35099031]
[ 1.56752558  0.88640526 -0.34806844  1.61991057 -1.05594645]
[ 1.5848463   0.88640526 -0.49160182  1.61991057  0.72984534]
[ 1.60216702 -1.12815215 -0.41983513  2.00160487 -1.63826986]
```

```
[ 1.61948775  0.88640526 -0.06100169  2.00160487  1.58391968]
[ 1.63680847  0.88640526  0.58489852  2.26879087 -1.32769738]
[ 1.6541292   0.88640526 -0.27630176  2.26879087  1.11806095]
[ 1.67144992  0.88640526  0.44136514  2.49780745 -0.86183865]
[ 1.68877065 -1.12815215 -0.49160182  2.49780745  0.92395314]
[ 1.70609137 -1.12815215 -0.49160182  2.91767117 -1.25005425]
[ 1.7234121  -1.12815215 -0.6351352   2.91767117  1.27334719]]
```

[22]:
```
X = df.iloc[:, [3, 4]].values
```

[23]:
```
X
```

[23]:
```
array([[ 15,  39],
       [ 15,  81],
       [ 16,   6],
       [ 16,  77],
       [ 17,  40],
       [ 17,  76],
       [ 18,   6],
       [ 18,  94],
       [ 19,   3],
       [ 19,  72],
       [ 19,  14],
       [ 19,  99],
       [ 20,  15],
       [ 20,  77],
       [ 20,  13],
       [ 20,  79],
       [ 21,  35],
       [ 21,  66],
       [ 23,  29],
       [ 23,  98],
       [ 24,  35],
       [ 24,  73],
       [ 25,   5],
       [ 25,  73],
       [ 28,  14],
       [ 28,  82],
       [ 28,  32],
       [ 28,  61],
       [ 29,  31],
       [ 29,  87],
       [ 30,   4],
       [ 30,  73],
       [ 33,   4],
       [ 33,  92],
       [ 33,  14],
```

```
[ 33,  81],
[ 34,  17],
[ 34,  73],
[ 37,  26],
[ 37,  75],
[ 38,  35],
[ 38,  92],
[ 39,  36],
[ 39,  61],
[ 39,  28],
[ 39,  65],
[ 40,  55],
[ 40,  47],
[ 40,  42],
[ 40,  42],
[ 42,  52],
[ 42,  60],
[ 43,  54],
[ 43,  60],
[ 43,  45],
[ 43,  41],
[ 44,  50],
[ 44,  46],
[ 46,  51],
[ 46,  46],
[ 46,  56],
[ 46,  55],
[ 47,  52],
[ 47,  59],
[ 48,  51],
[ 48,  59],
[ 48,  50],
[ 48,  48],
[ 48,  59],
[ 48,  47],
[ 49,  55],
[ 49,  42],
[ 50,  49],
[ 50,  56],
[ 54,  47],
[ 54,  54],
[ 54,  53],
[ 54,  48],
[ 54,  52],
[ 54,  42],
[ 54,  51],
[ 54,  55],
```

[54, 41],
[54, 44],
[54, 57],
[54, 46],
[57, 58],
[57, 55],
[58, 60],
[58, 46],
[59, 55],
[59, 41],
[60, 49],
[60, 40],
[60, 42],
[60, 52],
[60, 47],
[60, 50],
[61, 42],
[61, 49],
[62, 41],
[62, 48],
[62, 59],
[62, 55],
[62, 56],
[62, 42],
[63, 50],
[63, 46],
[63, 43],
[63, 48],
[63, 52],
[63, 54],
[64, 42],
[64, 46],
[65, 48],
[65, 50],
[65, 43],
[65, 59],
[67, 43],
[67, 57],
[67, 56],
[67, 40],
[69, 58],
[69, 91],
[70, 29],
[70, 77],
[71, 35],
[71, 95],
[71, 11],

[71, 75],
[71,   9],
[71, 75],
[72, 34],
[72, 71],
[73,   5],
[73, 88],
[73,   7],
[73, 73],
[74, 10],
[74, 72],
[75,   5],
[75, 93],
[76, 40],
[76, 87],
[77, 12],
[77, 97],
[77, 36],
[77, 74],
[78, 22],
[78, 90],
[78, 17],
[78, 88],
[78, 20],
[78, 76],
[78, 16],
[78, 89],
[78,   1],
[78, 78],
[78,   1],
[78, 73],
[79, 35],
[79, 83],
[81,   5],
[81, 93],
[85, 26],
[85, 75],
[86, 20],
[86, 95],
[87, 27],
[87, 63],
[87, 13],
[87, 75],
[87, 10],
[87, 92],
[88, 13],
[88, 86],

```
[ 88,  15],
[ 88,  69],
[ 93,  14],
[ 93,  90],
[ 97,  32],
[ 97,  86],
[ 98,  15],
[ 98,  88],
[ 99,  39],
[ 99,  97],
[101,  24],
[101,  68],
[103,  17],
[103,  85],
[103,  23],
[103,  69],
[113,   8],
[113,  91],
[120,  16],
[120,  79],
[126,  28],
[126,  74],
[137,  18],
[137,  83]], dtype=int64)
```

DBSCAN Clustering Algorithm

[24]:
```
from sklearn.cluster import DBSCAN
dbscan=DBSCAN(eps=3,min_samples=4)
```

[25]:
```
model=dbscan.fit(X)
labels=model.labels_
```

[26]:
```
from sklearn import metrics
```

[27]:
```
sample_cores=np.zeros_like(labels,dtype=bool)
sample_cores[dbscan.core_sample_indices_]=True
```

[28]:
```
#Calculating the number of clusters
n_clusters=len(set(labels))- (1 if -1 in labels else 0)
```

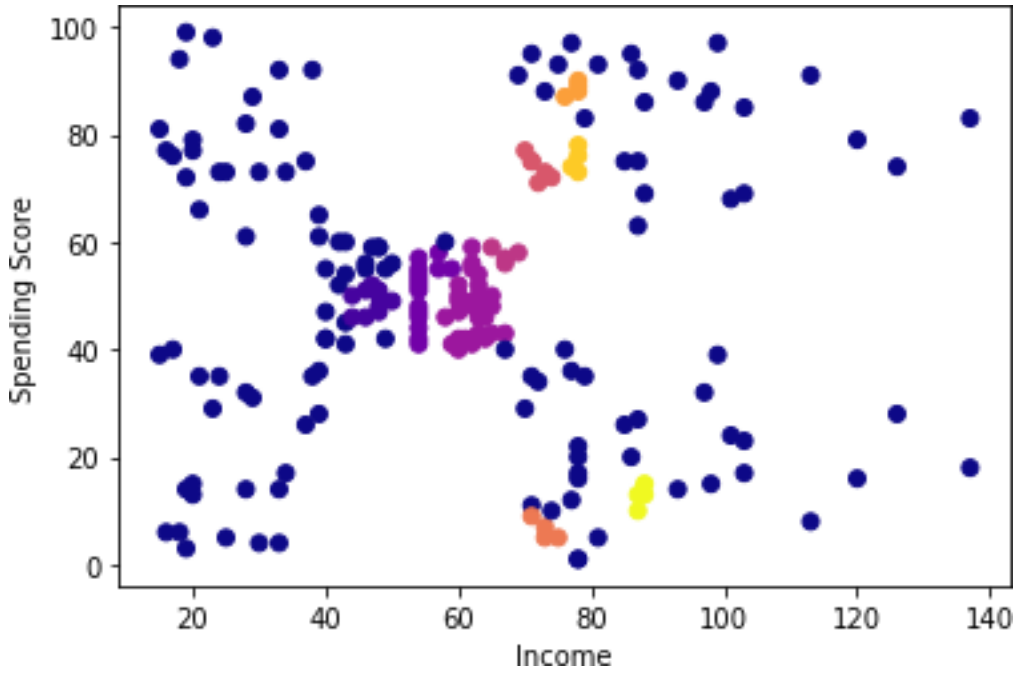
[29]:
```
n_clusters
```

[29]: 9

[30]:
```
print(metrics.silhouette_score(X,labels))
```

```
-0.1908319132560097
```

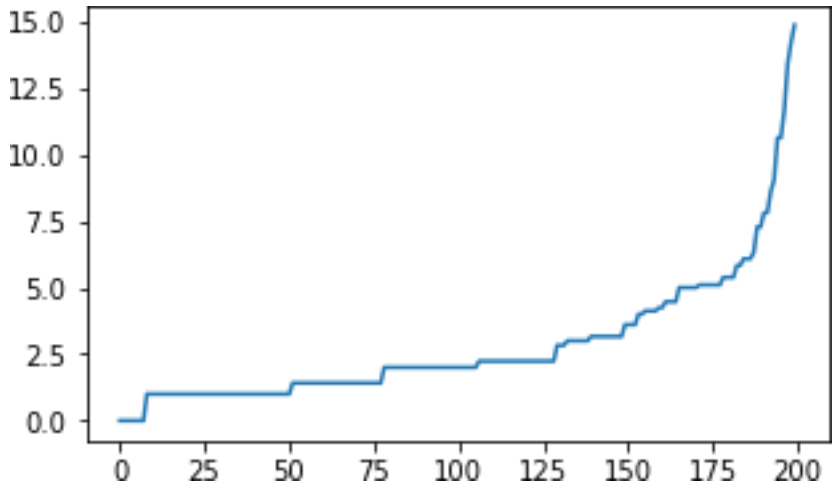
[31]:
```
# Plot the clusters
plt.scatter(X[:, 0], X[:,1], c = labels, cmap= "plasma") # plotting the clusters
plt.xlabel("Income") # X-axis label
plt.ylabel("Spending Score") # Y-axis label
plt.show() # showing the plot
```



KNN Algorithm

[32]:
```
from sklearn.neighbors import NearestNeighbors # importing the library
neighb = NearestNeighbors(n_neighbors=2) # creating an object of the␣
↪NearestNeighbors class
nbrs=neighb.fit(X) # fitting the data to the object
distances,indices=nbrs.kneighbors(X) # finding the nearest neighbours
```

[33]:
```
# Sort and plot the distances results
distances = np.sort(distances, axis = 0) # sorting the distances
distances = distances[:, 1] # taking the second column of the sorted distances
plt.rcParams['figure.figsize'] = (5,3) # setting the figure size
plt.plot(distances) # plotting the distances
plt.show() # showing the plot
```


15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
25
50
75
100
125
150
175
200

[ ]:

[ ]: