ASSIGNMENT – 4

PROBLEM STATEMENT: CUSTOMER SEGMENTATION ANALYSIS

ASSIGNMENT DATE	17 OCTOBER 2022
STUDENT NAME	S.HARSHINI
STUDENT ROLL NUMBER	CS19012
MAXIMUM MARKS	2 MARKS

import pandas as pd import numpy as np import

matplotlib.pyplot as plt import seaborn as sns from google.colab

import files data_to_load = files.upload()

< IPython.core.display.HTML object >

Saving Mall_Customers.csv to Mall_Customers.csv import io df =

pd.read_csv(io.BytesIO(data_to_load['Mall_Customers.csv'])) df.head()

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 df.tail()

CustomerID Gender Age Annual Income (k$) Spending Score (1100)

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 df.shape

(200 , 5) df.info()

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

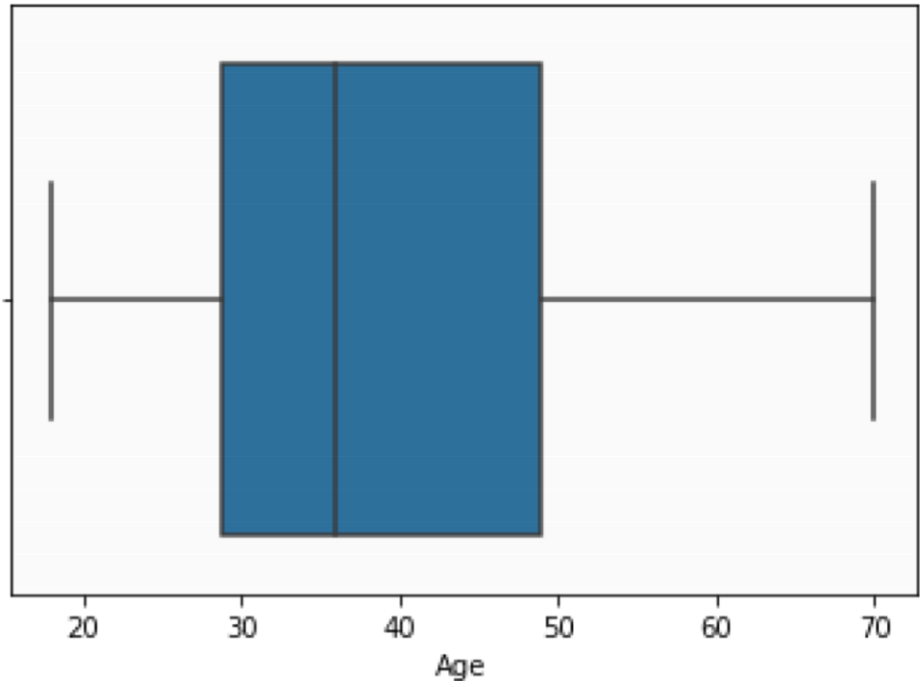
Data columns (total 5 columns):

Column Non-Null Count Dtype

--- ------ -------------- -----

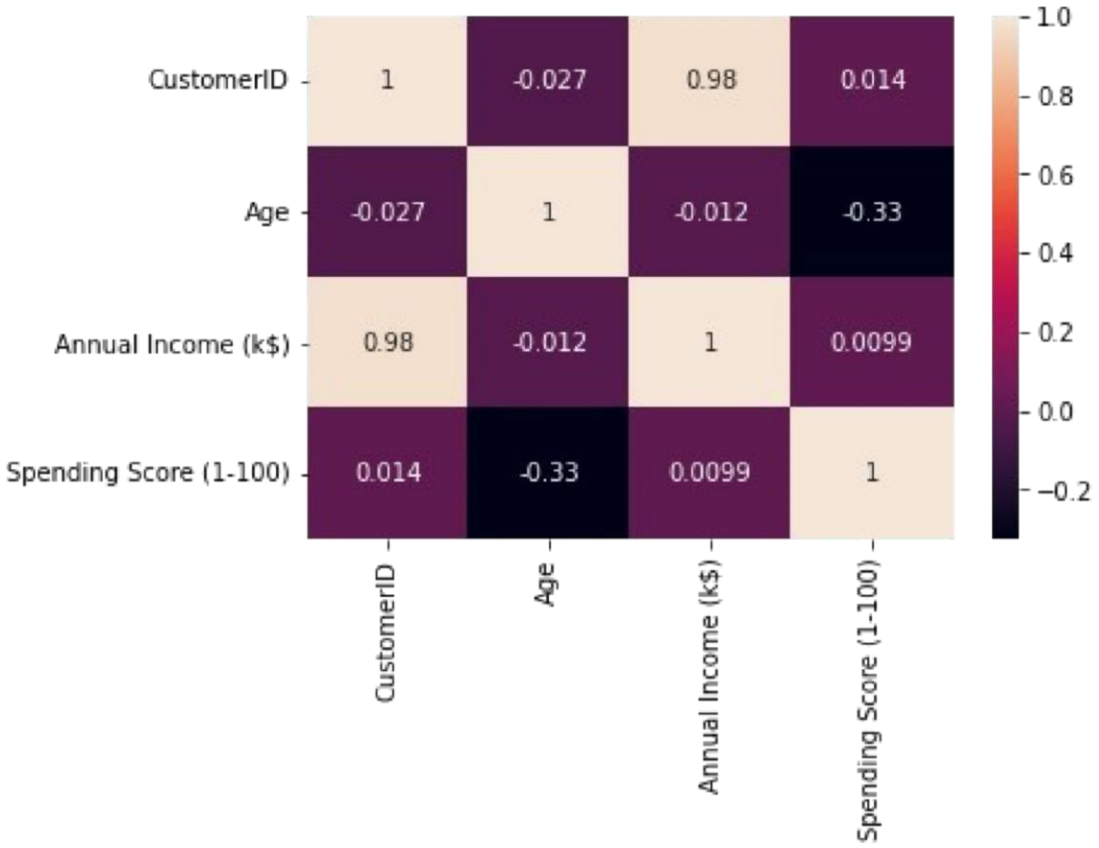
0 CustomerID 200 non-null int64 1 Gender 200 nonnull object 2 Age 200 non-null int64 3 Annual Income (k$) 200 non-null int64 4 Spending Score (1-100) 200 non-null int64 dtypes: int64(4), object(1) memory usage: 7.9+ KB sns.boxplot(x=df['Age'])

< matplotlib.axes._subplots.AxesSubplot at 0x7ff8695244d 0>



sns.heatmap(df.corr(),annot=True)

< matplotlib.axes._subplots.AxesSubplot at 0x7ff869516990>



df.describe()

CustomerID Age Annual Income (k$) Spending Score (1-100)

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 df.describe().T

	count	mean	std	min	25%	50%	75 %
CustomerID	200.0	100.50	57.879185	1.0	50.75	100.5	
150.25							
Age	200.0	38.85	13.969007	18.0	28.75	36.0	
49.00							
Annual Income (k$)	200.0	60.56	26.264721	15.0	41.50	61.5	
78.00							
Spending Score (1-100)	200.0	50.20	25.823522	1.0	34.75	50.0	
73.00							

	max
CustomerID	200.0
Age	70.0
Annual Income (k$)	137.0
Spending Score (1-100)	99.0

df.isna().sum()

CustomerID 0

Gender 0 Age 0
Annual Income (k$) 0 Spending Score (1- 100)
0 dtype: int64 df['Gender'].replace({'Male':1,

'Female':0},inplace=True) df.head()

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

df.Gender.unique() array([1, 0]) fig,ax = plt.subplots(figsize=(25,5)) plt.subplot(1,5,1)

sns.boxplot(x=df["CustomerID"]) plt.subplot(1,5,2) sns.boxplot(x=df["Gender"])

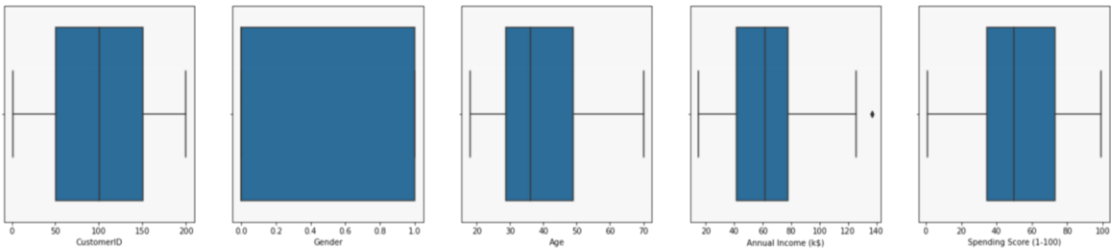
plt.subplot(1,5,3) sns.boxplot(x=df["Age"]) plt.subplot(1,5,4)

sns.boxplot(x=df["Annual Income (k$)"])

plt.subplot(1,5,5) sns.boxplot(x=df["Spending

Score (1-100)"])

< matplotlib.axes._subplots.AxesSubplot at 0x 7ff866592c10>



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

CustomerID Gender Age Annual Income (k$) Spending Score

(1-100)

0.25 50.75 0.0 28.75 41.5

34.75

0.75 150.25 1.0 49.00 78.0 73.00 iqr=qnt.loc[0.75]-qnt.loc[0.25] iqr

CustomerID 99.50

Gender 1.00 Age 20.25
Annual Income (k$) 36.50 Spending Score (1- 100)

38.25 dtype: float64 lower=qnt.loc[0.25]-(1.5*iqr) lower

CustomerID -98.500

Gender -1.500 Age -1.625

Annual Income (k$) -13.250 Spending Score (1- 100) -

22.625 dtype: float64 upper=qnt.loc[0.75]+(1.5*iqr) upper

CustomerID 299.500
Gender 2.500 Age 79.375 Annual
Income (k$) 132.750 Spending Score (1100)

130.375 dtype: float64 df.mean()

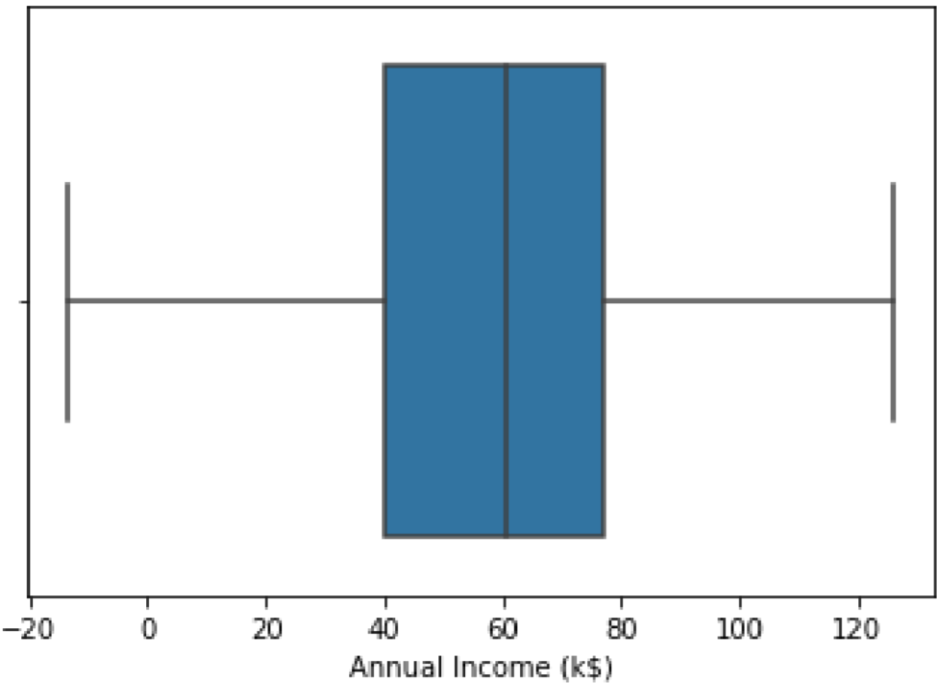
CustomerID 100.50

Gender 0.44 Age 38.85
Annual Income (k$) 60.56 Spending Score (1-
100) 50.20 dtype: float64 df['Annual Income

(k$)']=np.where(df['Annual Income (k$)']>132.750,- 13.250,df['Annual

Income (k$)']) sns.boxplot(x=df['Annual Income (k$)'])

< matplotlib.axes._subplots.AxesSubplot at 0x7ff8664e7ed 0>



df.head()

CustomerID Gender Age Annual Income (k$) Spending Score (1-100) 0 1 1 19
15.0 39

1 2 1 21 15.0 81 2 3 0 20 16.0 6 3 4

0 23 16.0 77 4 5 0 31 17.0 40 df['Gender'] .unique

() array([1, 0]) from sklearn.preprocessing import MinMaxScaler sc=MinMaxScaler()

data=sc.fit_transform(df.iloc[:,1:]) data

array([[1. , 0.01923077, 0.20287253, 0.3877551], [1. , 0.05769231,
0.20287253, 0.81632653],

[0. , 0.03846154, 0.21005386, 0.05102041], [0. , 0.09615385, 0.21005386, 0.7755102], [0. , 0.25 , 0.21723519, 0.39795918],

[0. , 0.07692308, 0.21723519, 0.76530612],

[0. , 0.32692308, 0.22441652, 0.05102041],

[0. , 0.09615385, 0.22441652, 0.94897959],

[1. , 0.88461538, 0.23159785, 0.02040816], [0. , 0.23076923, 0.23159785, 0.7244898], [1. , 0.94230769, 0.23159785, 0.13265306], [0. , 0.32692308, 0.23159785, 1.], [0. , 0.76923077, 0.23877917, 0.14285714], [0. , 0.11538462, 0.23877917, 0.7755102], [1. , 0.36538462, 0.23877917, 0.12244898],

[1. , 0.07692308, 0.23877917, 0.79591837], [0. , 0.32692308, 0.2459605 , 0.34693878],

[1. , 0.03846154, 0.2459605 , 0.66326531], [1. , 0.65384615, 0.26032316, 0.28571429],

[0. , 0.32692308, 0.26032316, 0.98979592],

[1. , 0.32692308, 0.26750449, 0.34693878],

[1. , 0.13461538, 0.26750449, 0.73469388],

[0. , 0.53846154, 0.27468582, 0.04081633],

[1. , 0.25 , 0.27468582, 0.73469388], [0. , 0.69230769, 0.2962298 , 0.13265306],

[1. , 0.21153846, 0.2962298 , 0.82653061],

[0. , 0.51923077, 0.2962298 , 0.31632653], [1. , 0.32692308, 0.2962298 , 0.6122449], [0. , 0.42307692, 0.30341113, 0.30612245],

[0. , 0.09615385, 0.30341113, 0.87755102],

[1. , 0.80769231, 0.31059246, 0.03061224],

[0. , 0.05769231, 0.31059246, 0.73469388],

[1. , 0.67307692, 0.33213645, 0.03061224],

[1. , 0. , 0.33213645, 0.92857143],

[0. , 0.59615385, 0.33213645, 0.13265306],

[0. , 0.05769231, 0.33213645, 0.81632653],

[0. , 0.46153846, 0.33931777, 0.16326531],
[0. , 0.23076923, 0.33931777, 0.73469388],

[0. , 0.34615385, 0.36086176, 0.25510204],

[0. , 0.03846154, 0.36086176, 0.75510204],

[0. , 0.90384615, 0.36804309, 0.34693878],

[1. , 0.11538462, 0.36804309, 0.92857143],

[1. , 0.57692308, 0.37522442, 0.35714286], [0. , 0.25 , 0.37522442, 0.6122449
],

[0. , 0.59615385, 0.37522442, 0.2755102], [0. , 0.11538462, 0.37522442, 0.65306122],

[0. , 0.61538462, 0.38240575, 0.55102041],

[0. , 0.17307692, 0.38240575, 0.46938776],

[0. , 0.21153846, 0.38240575, 0.41836735],

[0. , 0.25 , 0.38240575, 0.41836735], [0. , 0.59615385, 0.3967684 ,
0.52040816],

[1. , 0.28846154, 0.3967684 , 0.60204082], [0. , 0.25 , 0.40394973,
0.54081633],

[1. , 0.78846154, 0.40394973, 0.60204082],

[0. , 0.61538462, 0.40394973, 0.44897959],

[1. , 0.55769231, 0.40394973, 0.40816327], [0. , 0.63461538,
0.41113106, 0.5], [1. , 0.98076923, 0.41113106,
0.45918367],

[0. , 0.17307692, 0.42549372, 0.51020408],

[1. , 0.67307692, 0.42549372, 0.45918367],

[1. , 1. , 0.42549372, 0.56122449],

[1. , 0.01923077, 0.42549372, 0.55102041],

[0. , 0.94230769, 0.43267504, 0.52040816],

[0. , 0.69230769, 0.43267504, 0.59183673],

[1. , 0.86538462, 0.43985637, 0.51020408],

[1. , 0. , 0.43985637, 0.59183673], [0. , 0.48076923, 0.43985637, 0.5], [0. , 0.96153846, 0.43985637, 0.47959184],

[1. , 0.01923077, 0.43985637, 0.59183673],

[0. , 0.26923077, 0.43985637, 0.46938776], [1. , 1. , 0.4470377 , 0.55102041],

[0. , 0.55769231, 0.4470377 , 0.41836735], [0. , 0.80769231, 0.45421903, 0.48979592],

[0. , 0.80769231, 0.45421903, 0.56122449],

[1. , 0.78846154, 0.48294434, 0.46938776],

[1. , 0.15384615, 0.48294434, 0.54081633],

[0. , 0.51923077, 0.48294434, 0.53061224],

[1. , 0.42307692, 0.48294434, 0.47959184],

[0. , 0.09615385, 0.48294434, 0.52040816],

[0. , 0.59615385, 0.48294434, 0.41836735],

[1. , 0.75 , 0.48294434, 0.51020408],

[1. , 0.38461538, 0.48294434, 0.55102041],

[1. , 0.94230769, 0.48294434, 0.40816327],

[0. , 0.53846154, 0.48294434, 0.43877551],

[0. , 0.05769231, 0.48294434, 0.57142857],

[1. , 0.57692308, 0.48294434, 0.45918367],

[0. , 0.71153846, 0.50448833, 0.58163265],

[0. , 0.07692308, 0.50448833, 0.55102041],

[0. , 0.30769231, 0.51166966, 0.60204082],

[0. , 0.61538462, 0.51166966, 0.45918367],

[0. , 0.96153846, 0.51885099, 0.55102041],

[1. , 0. , 0.51885099, 0.40816327],

[1. , 0.57692308, 0.52603232, 0.48979592],

[0. , 0.42307692, 0.52603232, 0.39795918],

[0. , 0.26923077, 0.52603232, 0.41836735], [1. , 0.11538462, 0.52603232, 0.52040816],

[0. , 0.55769231, 0.52603232, 0.46938776], [0. , 0.17307692, 0.52603232, 0.5], [1. , 0.57692308, 0.53321364, 0.41836735],

[1. , 0.03846154, 0.53321364, 0.48979592],

[0. , 0.09615385, 0.54039497, 0.40816327],

[0. , 0.59615385, 0.54039497, 0.47959184],
[1. , 0.94230769, 0.54039497, 0.59183673],

[1. , 0.15384615, 0.54039497, 0.55102041],

[1. , 0.59615385, 0.54039497, 0.56122449],

[0. , 0.05769231, 0.54039497, 0.41836735], [0. , 0.92307692, 0.5475763 , 0.5], [1. , 0.69230769, 0.5475763 , 0.45918367],

[1. , 0.96153846, 0.5475763 , 0.42857143],

[1. , 0.92307692, 0.5475763 , 0.47959184],

[1. , 0.90384615, 0.5475763 , 0.52040816],

[0. , 0.01923077, 0.5475763 , 0.54081633], [0. , 0.38461538, 0.55475763, 0.41836735],

[1. , 0.01923077, 0.55475763, 0.45918367],

[0. , 0. , 0.56193896, 0.47959184], [0. , 0.01923077, 0.56193896, 0.5], [0. , 0.86538462, 0.56193896, 0.42857143],

[0. , 0.59615385, 0.56193896, 0.59183673],

[0. , 0.63461538, 0.57630162, 0.42857143],

[0. , 0.61538462, 0.57630162, 0.57142857],

[1. , 0.17307692, 0.57630162, 0.56122449],

[0. , 0.38461538, 0.57630162, 0.39795918],

[0. , 0.42307692, 0.59066427, 0.58163265],

[1. , 0.40384615, 0.59066427, 0.91836735], [0. , 0.09615385,
0.5978456 , 0.28571429], [0. , 0.25 , 0.5978456 ,
0.7755102], [1. , 0.48076923, 0.60502693, 0.34693878],

[1. , 0.42307692, 0.60502693, 0.95918367],

[1. , 0.78846154, 0.60502693, 0.10204082],

[1. , 0.38461538, 0.60502693, 0.75510204],

[1. , 0.55769231, 0.60502693, 0.08163265],

[1. , 0.40384615, 0.60502693, 0.75510204],

[0. , 0.13461538, 0.61220826, 0.33673469],

[0. , 0.25 , 0.61220826, 0.71428571],
[1. , 0.03846154, 0.61938959, 0.04081633], [0. , 0.21153846,
0.61938959, 0.8877551], [0. , 0.5 , 0.61938959,
0.06122449],

[1. , 0.26923077, 0.61938959, 0.73469388],

[1. , 0.01923077, 0.62657092, 0.09183673], [0. , 0.32692308,
0.62657092, 0.7244898], [0. , 0.75 , 0.63375224, 0.04081633],

[1. , 0.26923077, 0.63375224, 0.93877551],

[0. , 0.19230769, 0.64093357, 0.39795918],

[0. , 0.26923077, 0.64093357, 0.87755102], [1. , 0.13461538, 0.6481149
, 0.1122449], [1. , 0.19230769, 0.6481149 ,
0.97959184],

[1. , 0.57692308, 0.6481149 , 0.35714286],

[0. , 0.26923077, 0.6481149 , 0.74489796], [0. , 0.30769231,
0.65529623, 0.21428571],

[1. , 0.30769231, 0.65529623, 0.90816327],

[1. , 0.48076923, 0.65529623, 0.16326531], [1. , 0.40384615,
0.65529623, 0.8877551], [0. , 0.5 , 0.65529623, 0.19387755],

[0. , 0.38461538, 0.65529623, 0.76530612],

[0. , 0.55769231, 0.65529623, 0.15306122],

[0. , 0.17307692, 0.65529623, 0.89795918], [1. , 0.36538462,

0.65529623, 0.], [0. , 0.23076923, 0.65529623, 0.78571429], [1.
, 0.30769231, 0.65529623, 0.], [0. , 0.23076923, 0.65529623,
0.73469388],

[0. , 0.73076923, 0.66247756, 0.34693878],

[0. , 0.21153846, 0.66247756, 0.83673469],

[1. , 0.01923077, 0.67684022, 0.04081633],

[0. , 0.25 , 0.67684022, 0.93877551],

[1. , 0.61538462, 0.70556553, 0.25510204],

[0. , 0.34615385, 0.70556553, 0.75510204],

[1. , 0.46153846, 0.71274686, 0.19387755],

[0. , 0.28846154, 0.71274686, 0.95918367],

[0. , 0.34615385, 0.71992819, 0.26530612],

[1. , 0.26923077, 0.71992819, 0.63265306],
[1. , 0.42307692, 0.71992819, 0.12244898],

[1. , 0.19230769, 0.71992819, 0.75510204],

[1. , 0.34615385, 0.71992819, 0.09183673],

[1. , 0.34615385, 0.71992819, 0.92857143],

[0. , 0.65384615, 0.72710952, 0.12244898],

[0. , 0.23076923, 0.72710952, 0.86734694],

[1. , 0.76923077, 0.72710952, 0.14285714],

[1. , 0.17307692, 0.72710952, 0.69387755],

[1. , 0.78846154, 0.76301616, 0.13265306],

[1. , 0.32692308, 0.76301616, 0.90816327],

[0. , 0.36538462, 0.79174147, 0.31632653],

[0. , 0.26923077, 0.79174147, 0.86734694], [1. , 0.53846154,
0.7989228 , 0.14285714], [0. , 0.21153846, 0.7989228 , 0.8877551],
[0. , 0.44230769, 0.80610413, 0.3877551], [1. , 0.23076923,
0.80610413, 0.97959184],

[0. , 0.69230769, 0.82046679, 0.23469388],

[1. , 0.19230769, 0.82046679, 0.68367347],

[0. , 0.44230769, 0.83482944, 0.16326531],

[0. , 0.34615385, 0.83482944, 0.85714286], [0. , 0.30769231, 0.83482944, 0.2244898], [0. , 0.26923077, 0.83482944, 0.69387755],

[1. , 0.28846154, 0.90664273, 0.07142857],

[0. , 0.38461538, 0.90664273, 0.91836735],

[0. , 0.55769231, 0.95691203, 0.15306122], [0. , 0.32692308, 0.95691203, 0.79591837], [0. , 0.51923077, 1. , 0.2755102], [1. , 0.26923077, 1. , 0.74489796],

[1. , 0.26923077, 0. , 0.17346939], [1. , 0.23076923, 0. , 0.83673469]]) from sklearn.cluster import KMeans TWSS=[] k=list(range(2,9))

```
for i in k:

kmeans=KMeans(n_clusters=i,init='k-means++')     kmeans.fit(data)

   TWSS.append(kmeans.inertia_)

TWSS
```

[35.09354046290808 ,

27.37315679730296,

20.211573858371988,

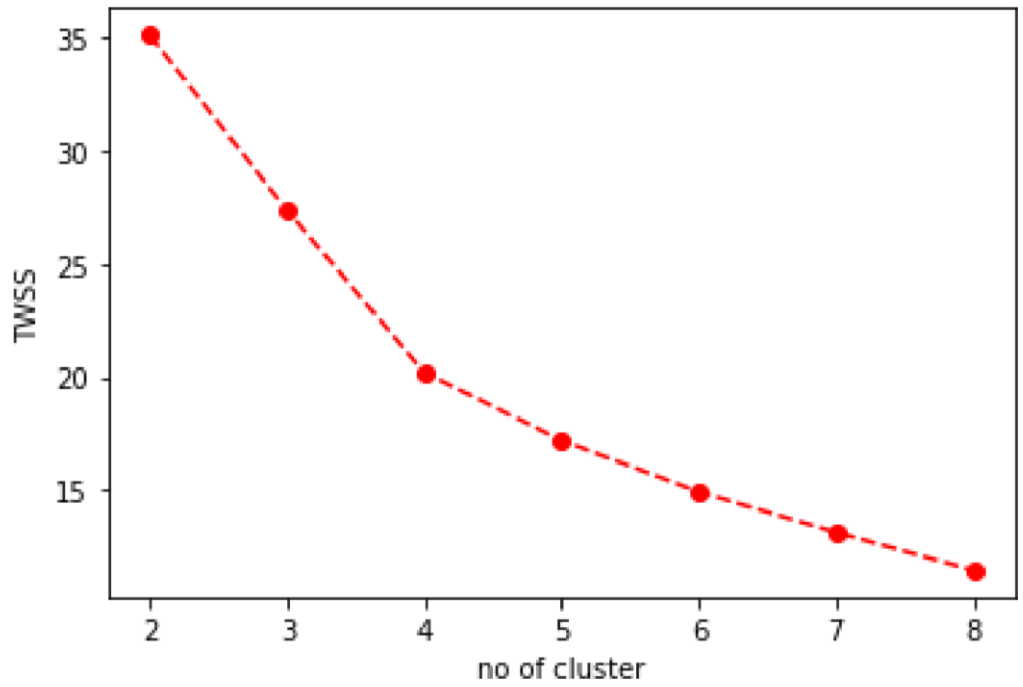
17.210964888908972,

14.941607138943485,

13.153866803186235, 11.485368243450253]

```
plt.plot(k,TWSS,'ro--') plt.xlabel('no of cluster') plt.ylabel('TWSS')

Text(0, 0.5, 'TWSS')
```



model=KMeans(n_clusters=4) model.fit(data) KMeans(n_clusters=4) mb=pd.Series(model.labels_)

df.head(3)

CustomerID Gender Age Annual Income (k$) Spending Score (1-100) 0 1 1 19
15.0 39
1 2 1 21 15.0 81 2 3 0 20 16.0 6 df.tail()

CustomerID Gender Age Annual Income (k$) Spending Score (1100)

195 196 0 35 120.00
79

196 197 0 45 126.00
28

197 198 1 32 126.00
74

198 199 1 32 -13.25
18

199 200 1 30 -13.25 83 x=df.drop(columns=['CustomerID',
'Gender'],axis=1) x.head()

Age Annual Income (k$) Spending Score (1-100) 0 19 15.0 39

1 21 15.0 81 2 20 16.0 6 3 23
16.0 77 4 31 17.0 40

y=df['Gender'] y

```
0       1
1       1
2       0
3       0
4       0    ..
195     0
196     0
197     1
198     1
199     1
```

Name: Gender, Length: 200, dtype: int64 from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2) from sklearn.ensemble import

RandomForestClassifier

rf=RandomForestClassifier() rf.fit(x_train,y_train)

RandomForestClassifier() pred=rf.predict(x_test) from sklearn.metrics import

accuracy_score accuracy_score(y_test,pred)

0.575 from sklearn import metrics metrics.confusion_matrix(y_test,pred)

array([[17, 8], [9, 6]])