Sl Assignmnet- 5 -21BPS1630

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

Understanding the dataset

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

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (I-lee)
0	1	Male	19	15	39
	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	74	32	126
198	199	Male	18	32	137
199	200	Male	83	30	137

200 rows x 5 columns

```
data. info()
```

```
(class  pandas.  core.  frame  .  >
RangeIndex: 200 entries, to 199
  Data columns (total 5 columns) :
```

#	Column	Non- Null Count	Dtype
---	------	--------------	-----
e	Cus tomer I	200 non-null	int 64
1	Gender	200 non-null	obj ect
2	Age	200 non-null	int64
3	Annual Income (k$)	200 non-null	int 64
4	Spending Score (1-100)	200 non-null	int 64

```
dtypes: int64(4),
object(l) memory
usage: 7.9+ KB
```

Data Preprocessing

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

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

```
data. describe ( )
```

|	CustomerID	Age	Annual Income (k$)	Spending Score (I-lee)
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

Data Visualization

```
sns.displot(data['Annual Income (k$)'])
```

```
(seaborn.axisgrid. FacetGrid at øx7beec1769a20>
```
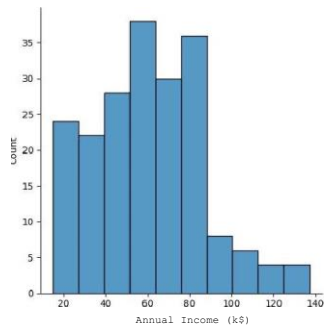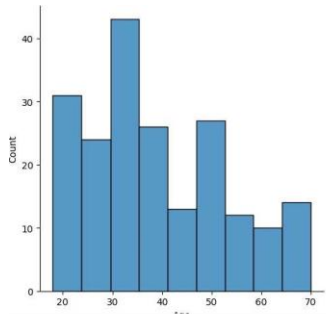
```
sns.displot(data['Spending Score (1-100) ' ])
    (seaborn.axisgrid. FacetGrid at øx7beec1402aae>
```
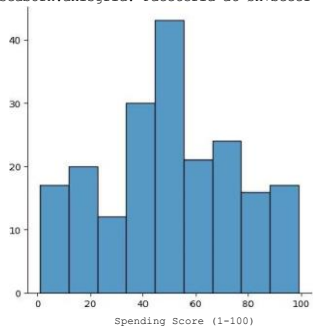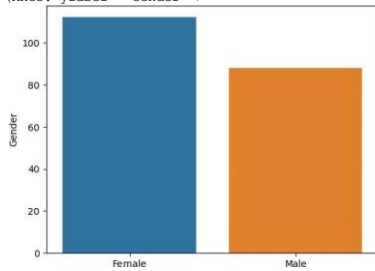


```
sns. isplot(data[' Age ' ] )
```

```
(seaborn. axisgrid.FacetGrid at ex7beebfeeefSØ>
```



```
sns.barplot(x =data.Gender.value_counts().index,y =data.Gender.value_counts() )
    (Axes: ylabel=' Gender' >
```


Data Label Encoding

```
from    sklearn.preprocessing    import
LabelEncoder le =Label1Encoder()
```
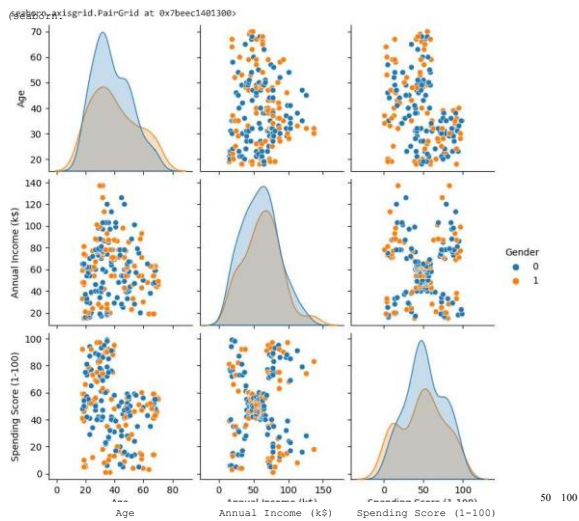
```
data. Gender= le.fit_transform(data.Gender)
```

```
data. head ( )
```

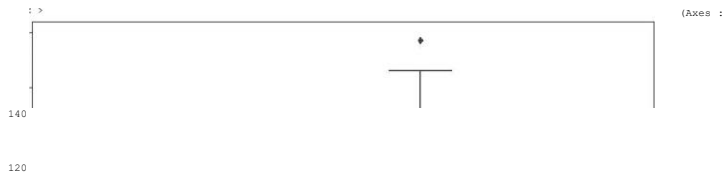|	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (I-lee)
0	1	1	19	15	39
1	2	1	21	15	81
2	3	0	20	16	6
3	4	0	23	16	77
4	5	8 31		17	40

```
#Customer  ID  column  is  not
required  data  =  data  .drop(
'Customer-ID', axis-I)
```

```
sns.palrplot(data, hue. 'Gender' )
```

(seaborn.axisgrid.PairGrid at 0x7beec1401300>


```
plt. igure(figsize=(12,
8)) sns. boxplot
(data)
```

#printMode—true

(Axes :

Machine Learning approach with clustering algorithm

```
Inn 4  _____ from sklearn import cluster
```

```
error—I]  for  i  in range(1,11): kmeans  =  cluster.  KMeans
```

```
n_clusters=i,init = ' k-means++' , random_state=0)kmeans. fit
(data) error. append ( kmeans. inertia_)
```

/us r/local/Iib /python3 .10/ dist -packages/ Skl earn/FutureWarning: The default value of `n_init` will change to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning from 1 @

/us r/10ca1/1ib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:87e: wa rnings. warn (default value of n init-will change f rom 1 to auto in1.4. set the value of `n_init` explicitly to suppress the warning @

/us r/local/Iib /python3 .10/ dist -packages/ Skl earn/FutureWarning: The default value of `n_init` will change to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning from 1 @

/us r/local/ lib /python3 .1B/ dist -packages/ skl earn/FutureWarning•. The default value of n init-will change f rom 1 to 'auto in 1.4. set the value to explicitly suppress the warning @

/us r/local/Iib /python3 .1B/ dist -packages/ earn/Futurewarning: The default value of `n_init` will change to auto in1.4. the value of `n_init` explicitly suppress the warning Set from 1 @

/us r/local/ lib /python3 .10/ dist -packages/ skl earn/FutureWarning•. The default value ' n will change to 'auto 1.4. the value of `n_init` explicitly to suppress the warning init from 1 in Set @

/us r/local/ lib /python3 .10/ dist -packages/ Skl earn/FutureWarning:The default value n init' will change to in1.4. the value `n_init` Explicitly suppress the warning from 1 auto Set of @

/us r/local /lib /python3 .10/ dist -packages/ Skl earn/ Futurewarning:The cl uster/_kmeans.py:87e: wa rnings. (default value n init' will change to' 1.4. the value of `n_init` explicitly suppress the warning from 1 auto in Set @

error

```
C 3e8862.06øøøøøøøø6,
212889.44245524303,
143391.59236035676,
1ø4414.67534220168,
75399.61541401484,
58348.641363315044,
51132.703212576904,
44392.11566567935,
41eee.8742213207,
37649.69225429742]
```

```
plt. plot range(1,11)
,           error)
plt.title( 'Elbow
method')      plt.
xlabel(  'number
of     clusters')
plt.ylabel(
error'   )   plt.
show()
```
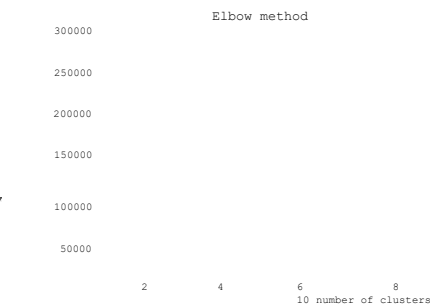
Elbow method

No.of clusters using elbow method=> n=5

```
km model = cluster.KMeans n_clusters=5,init = 'k-means++' , random_state=ø)
```

```
km_model. fit ( data)
```

KMeans(n_clusters=5, random_state=0)

```
pred km_model.
pred predict(data)
```

```
0,   0,   array(Ce, 4,     4,     0, 4, 0, 4,
0,   0,                           0, 4, 0, 4,
                                  2, 2, 2, 2,
               4, 0 4, 0, 4, 0, 4, 0,  2, 2, 2, 2,
  4,                              2, 2, 2, 2,
                                  3, 1, 3, 1,
         1, 3, 1, 3, 1, 3, 1, 2,   3,   3, 1, 3, 1,
         1, 3, 1, 3, 1, 3, 1, 3,   3,   3, 1, 3, 1,
  0,     1, 3, 1, 3, 1, 3, 1, 3,   3,   3, 1, 3, 1,

  4, 0, 4, 0 4,      4,          4, a, 4, 0, 4,
  e, 4, e, 4, e, 4, e, 4,

 0 4, 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, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1,
  3, 1, 2, 1,

         3, 1 3 2, 1,       1, 3, 1, 3, 1,

  3, 1 3,                  1 1, 3, 1, 3, 1,

  3, 1 3,              1, 1, 3, 1, 3, 1,
  3, 1], dtype=int32)
```

Visualizing the customer clusters

4Visualizing all the clusters
datal • Labels'] = km_model. labels_ plt.

figure figsize=(12, 8))
sns. scatterplot(xz' Annual 'Spending Score (I-   hue-   data-
Income (k$) • ,   lee)' ,   Labels • data)
                                             ,
plt.   title(
'Clustering
Customers ') plt.
show()

3/4

Clustering Customers

100

80

Inference :
Type O - Low Income and Low Spending Score Customers

Type 1 - High Income and High Spending Score Customers

Type 2 - Mid Income and Mid Spending
Score Customers Type 3 - Low Income
and High Spending Score Customers

Type 4 - High Income and Low
Spending Score Customers