ASSIGNMENT 5

Harsh Kumar

21BDS0391

harsh.kumar2021@vitstudent.ac.in

```
Assignment 5:
    Take all the columns in mall_customers.csv
    gender age annual income spending score
    perform label encoding on gender
    train your data
```

1. import necessary libraries.

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

2. import dataset.

```
df = pd.read_csv("Mall_Customers.csv")
df.head()
```

	CustomerID	Genre	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.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   Genre                   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
```

4. Label Encoding

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

```
df["Genre"] = le.fit_transform(df["Genre"])
```

3. Select the feature to cluster.

```
# Selecting annual income and spending as features for clustering.
```

```
X = df.iloc[:, 1:].values
X
```

```
       [  0,  28,  76,  40],
       [  0,  32,  76,  87],
       [  1,  25,  77,  12],
       [  1,  28,  77,  97],
       [  1,  48,  77,  36],
       [  0,  32,  77,  74],
       [  0,  34,  78,  22],
       [  1,  34,  78,  90],
       [  1,  43,  78,  17],
       [  1,  39,  78,  88],
       [  0,  44,  78,  20],
       [  0,  38,  78,  76],
       [  0,  47,  78,  16],
       [  0,  27,  78,  89],
       [  1,  37,  78,   1],
       [  0,  30,  78,  78],
       [  1,  34,  78,   1],
       [  0,  30,  78,  73],
       [  0,  56,  79,  35],
       [  0,  29,  79,  83],
       [  1,  19,  81,   5],
       [  0,  31,  81,  93],
       [  1,  50,  85,  26],
       [  0,  36,  85,  75],
       [  1,  42,  86,  20],
       [  0,  33,  86,  95],
       [  0,  36,  87,  27],
       [  1,  32,  87,  63],
       [  1,  40,  87,  13],
       [  1,  28,  87,  75],
       [  1,  36,  87,  10],
       [  1,  36,  87,  92],
       [  0,  52,  88,  13],
       [  0,  30,  88,  86],
       [  1,  58,  88,  15],
       [  1,  27,  88,  69],
       [  1,  59,  93,  14],
       [  1,  35,  93,  90],
       [  0,  37,  97,  32],
       [  0,  32,  97,  86],
       [  1,  46,  98,  15],
       [  0,  29,  98,  88],
       [  0,  41,  99,  39],
       [  1,  30,  99,  97],
       [  0,  54, 101,  24],
       [  1,  28, 101,  68],
       [  0,  41, 103,  17],
       [  0,  36, 103,  85],
       [  0,  34, 103,  23],
       [  0,  32, 103,  69],
       [  1,  33, 113,   8],
       [  0,  38, 113,  91],
       [  0,  47, 120,  16],
       [  0,  35, 120,  79],
       [  0,  45, 126,  28],
       [  1,  32, 126,  74],
       [  1,  32, 137,  18],
       [  1,  30, 137,  83]])
```

```
type(X)
```

```
numpy.ndarray
```

5. Find the optimal number of clusters -- elbow method.

```
from sklearn.cluster import KMeans

# Trying different values of k and calculating WCSS for each value of k
wcss = []
for k in range(1,16):
  kmeans = KMeans(n_clusters = k, init="k-means++", random_state=0)
  kmeans.fit(X)
  wcss.append(kmeans.inertia_) #savind wcss value in a list
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
```

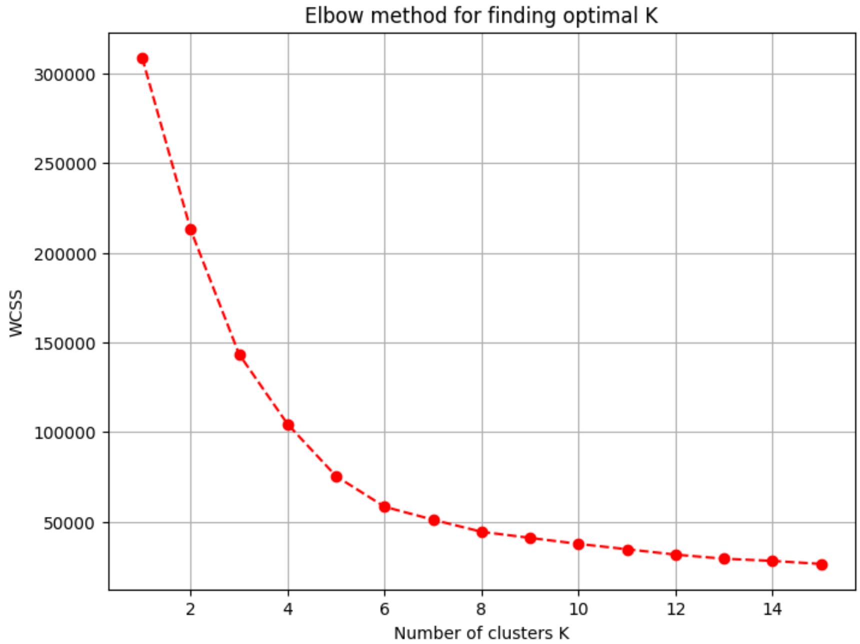
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
```

```
wcss
```

```
[308862.06000000006,
 212889.44245524303,
 143391.59236035676,
 104414.67534220168,
 75399.61541401484,
 58348.641363315044,
 51132.703212576904,
 44392.11566567935,
 41000.8742213207,
 37649.69225429742,
 34665.087277598795,
 31659.187454375693,
 29388.61288883975,
 28170.631036266237,
 26470.47953378321]
```

```
# PLOTTING ELBOW METHOD GRAPH

plt.figure(figsize = (8,6))
plt.plot(range(1,16), wcss, "o--", color="red")
plt.title("Elbow method for finding optimal K")
plt.xlabel("Number of clusters K")
plt.ylabel("WCSS")
plt.grid()
plt.show()
```




```
# Taking K=6 instead of becausa WCSS value is low
```

6. Train the model on dataset using optimal cluster value k=6.

```
kmeans = KMeans(n_clusters = 6, init="k-means++", random_state=0) #"k-means++" means initializing the random clusters
#return a label for data based on their cluster.
Y_kmeans = kmeans.fit_predict(X)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
```

```
Y_kmeans
```

```
array([5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4,
       5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 1, 4, 1, 0,
       5, 4, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0,
       1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 3, 0, 3, 2, 3, 2, 3, 2, 3,
       0, 3, 2, 3, 2, 3, 2, 3, 2, 3, 0, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3], dtype=int32)
```

```
# Age that belongs to cluster 0
X[Y_kmeans == 0,1]
```

```
array([31, 27, 29, 31, 33, 31, 27, 19, 18, 19, 32, 26, 40, 23, 38, 21, 22,
       34, 18, 40, 32, 24, 27, 20, 23, 26, 21, 19, 38, 19, 18, 19, 27, 38,
       40, 23, 25, 28])
```

```
# Age that belongs to cluster 1
X[Y_kmeans == 1,1]
```

```
array([65, 48, 50, 49, 59, 50, 47, 51, 69, 53, 70, 67, 54, 63, 43, 68, 70,
       47, 60, 60, 59, 45, 49, 57, 67, 46, 48, 55, 50, 68, 48, 47, 48, 49,
       67, 49, 66, 54, 68, 66, 65, 63, 49, 51, 50])
```

```
# Age that belongs to cluster 2
X[Y_kmeans == 2,1]
```

```
array([43, 59, 47, 20, 44, 19, 57, 25, 48, 34, 43, 44, 47, 37, 34, 56, 19,
       50, 42, 36, 40, 36, 52, 58, 59, 37, 46, 41, 54, 41, 34, 33, 47, 45,
       32])
```

```
# Data points that belong to cluster 0
X[Y_kmeans == 0]
```

```
array([[ 0, 31, 39, 61],
       [ 0, 27, 40, 47],
       [ 0, 29, 40, 42],
       [ 0, 31, 40, 42],
       [ 1, 33, 42, 60],
       [ 0, 31, 43, 54],
       [ 0, 27, 46, 51],
       [ 1, 19, 46, 55],
       [ 1, 18, 48, 59],
       [ 1, 19, 48, 59],
       [ 0, 32, 48, 47],
       [ 1, 26, 54, 54],
       [ 1, 40, 54, 48],
       [ 0, 23, 54, 52],
       [ 1, 38, 54, 55],
       [ 0, 21, 54, 57],
       [ 0, 22, 57, 55],
       [ 0, 34, 58, 60],
       [ 1, 18, 59, 41],
       [ 0, 40, 60, 40],
       [ 0, 32, 60, 42],
       [ 1, 24, 60, 52],
       [ 0, 27, 60, 50],
       [ 1, 20, 61, 49],
       [ 0, 23, 62, 41],
       [ 1, 26, 62, 55],
       [ 0, 21, 62, 42],
       [ 0, 19, 63, 54],
       [ 0, 38, 64, 42],
       [ 1, 19, 64, 46],
       [ 0, 18, 65, 48],
       [ 0, 19, 65, 50],
       [ 1, 27, 67, 56],
       [ 0, 38, 67, 40],
```

```
       [ 0, 40, 69, 58],
       [ 0, 23, 70, 29],
       [ 0, 25, 72, 34],
       [ 0, 28, 76, 40]])
```