system-using-k-means-on-mall-data

October 6, 2023

1 Customer Segmentation System Using K-Means

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
df = pd.read_csv(r"C:\Ganesh\pandas datafile\mall_kmeans.csv")
```

[3]:
```
df
```

[3]:

	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72
10	11	Male	67	19	14
11	12	Female	35	19	99
12	13	Female	58	20	15
13	14	Female	24	20	77
14	15	Male	37	20	13
15	16	Male	22	20	79
16	17	Female	35	21	35
17	18	Male	20	21	66
18	19	Male	52	23	29
19	20	Female	35	23	98
20	21	Male	35	24	35

21	22	Male	25	24	73
22	23	Female	46	25	5
23	24	Male	31	25	73
24	25	Female	54	28	14
25	26	Male	29	28	82
26	27	Female	45	28	32
27	28	Male	35	28	61
28	29	Female	40	29	31
29	30	Female	23	29	87
30	31	Male	60	30	4
31	32	Female	21	30	73
32	33	Male	53	33	4
33	34	Male	18	33	92
34	35	Female	49	33	14
35	36	Female	21	33	81
36	37	Female	42	34	17
37	38	Female	30	34	73
38	39	Female	36	37	26
39	40	Female	20	37	75
40	41	Female	65	38	35
41	42	Male	24	38	92
42	43	Male	48	39	36
43	44	Female	31	39	61
44	45	Female	49	39	28
45	46	Female	24	39	65
46	47	Female	50	40	55
47	48	Female	27	40	47
48	49	Female	29	40	42
49	50	Female	31	40	42
50	51	Female	49	42	52
51	52	Male	33	42	60
52	53	Female	31	43	54
53	54	Male	59	43	60
54	55	Female	50	43	45
55	56	Male	47	43	41
56	57	Female	51	44	50
57	58	Male	69	44	46
58	59	Female	27	46	51
59	60	Male	53	46	46
60	61	Male	70	46	56
61	62	Male	19	46	55
62	63	Female	67	47	52
63	64	Female	54	47	59
64	65	Male	63	48	51
65	66	Male	18	48	59
66	67	Female	43	48	50
67	68	Female	68	48	48

68	69	Male	19	48	59
69	70	Female	32	48	47
70	71	Male	70	49	55
71	72	Female	47	49	42
72	73	Female	60	50	49
73	74	Female	60	50	56
74	75	Male	59	54	47
75	76	Male	26	54	54
76	77	Female	45	54	53
77	78	Male	40	54	48
78	79	Female	23	54	52
79	80	Female	49	54	42
80	81	Male	57	54	51
81	82	Male	38	54	55
82	83	Male	67	54	41
83	84	Female	46	54	44
84	85	Female	21	54	57
85	86	Male	48	54	46
86	87	Female	55	57	58
87	88	Female	22	57	55
88	89	Female	34	58	60
89	90	Female	50	58	46
90	91	Female	68	59	55
91	92	Male	18	59	41
92	93	Male	48	60	49
93	94	Female	40	60	40
94	95	Female	32	60	42
95	96	Male	24	60	52
96	97	Female	47	60	47
97	98	Female	27	60	50
98	99	Male	48	61	42
99	100	Male	20	61	49
100	101	Female	23	62	41
101	102	Female	49	62	48
102	103	Male	67	62	59
103	104	Male	26	62	55
104	105	Male	49	62	56
105	106	Female	21	62	42
106	107	Female	66	63	50
107	108	Male	54	63	46
108	109	Male	68	63	43
109	110	Male	66	63	48
110	111	Male	65	63	52
111	112	Female	19	63	54
112	113	Female	38	64	42
113	114	Male	19	64	46
114	115	Female	18	65	48

115	116	Female	19	65	50
116	117	Female	63	65	43
117	118	Female	49	65	59
118	119	Female	51	67	43
119	120	Female	50	67	57
120	121	Male	27	67	56
121	122	Female	38	67	40
122	123	Female	40	69	58
123	124	Male	39	69	91
124	125	Female	23	70	29
125	126	Female	31	70	77
126	127	Male	43	71	35
127	128	Male	40	71	95
128	129	Male	59	71	11
129	130	Male	38	71	75
130	131	Male	47	71	9
131	132	Male	39	71	75
132	133	Female	25	72	34
133	134	Female	31	72	71
134	135	Male	20	73	5
135	136	Female	29	73	88
136	137	Female	44	73	7
137	138	Male	32	73	73
138	139	Male	19	74	10
139	140	Female	35	74	72
140	141	Female	57	75	5
141	142	Male	32	75	93
142	143	Female	28	76	40
143	144	Female	32	76	87
144	145	Male	25	77	12
145	146	Male	28	77	97
146	147	Male	48	77	36
147	148	Female	32	77	74
148	149	Female	34	78	22
149	150	Male	34	78	90
150	151	Male	43	78	17
151	152	Male	39	78	88
152	153	Female	44	78	20
153	154	Female	38	78	76
154	155	Female	47	78	16
155	156	Female	27	78	89
156	157	Male	37	78	1
157	158	Female	30	78	78
158	159	Male	34	78	1
159	160	Female	30	78	73
160	161	Female	56	79	35
161	162	Female	29	79	83

162	163	Male	19	81	5
163	164	Female	31	81	93
164	165	Male	50	85	26
165	166	Female	36	85	75
166	167	Male	42	86	20
167	168	Female	33	86	95
168	169	Female	36	87	27
169	170	Male	32	87	63
170	171	Male	40	87	13
171	172	Male	28	87	75
172	173	Male	36	87	10
173	174	Male	36	87	92
174	175	Female	52	88	13
175	176	Female	30	88	86
176	177	Male	58	88	15
177	178	Male	27	88	69
178	179	Male	59	93	14
179	180	Male	35	93	90
180	181	Female	37	97	32
181	182	Female	32	97	86
182	183	Male	46	98	15
183	184	Female	29	98	88
184	185	Female	41	99	39
185	186	Male	30	99	97
186	187	Female	54	101	24
187	188	Male	28	101	68
188	189	Female	41	103	17
189	190	Female	36	103	85
190	191	Female	34	103	23
191	192	Female	32	103	69
192	193	Male	33	113	8
193	194	Female	38	113	91
194	195	Female	47	120	16
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

[4]:
```
df.head()
```

[4]:

	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

[5]:
```
df.tail()
```

[5]:

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

[6]:
```
df.shape
```

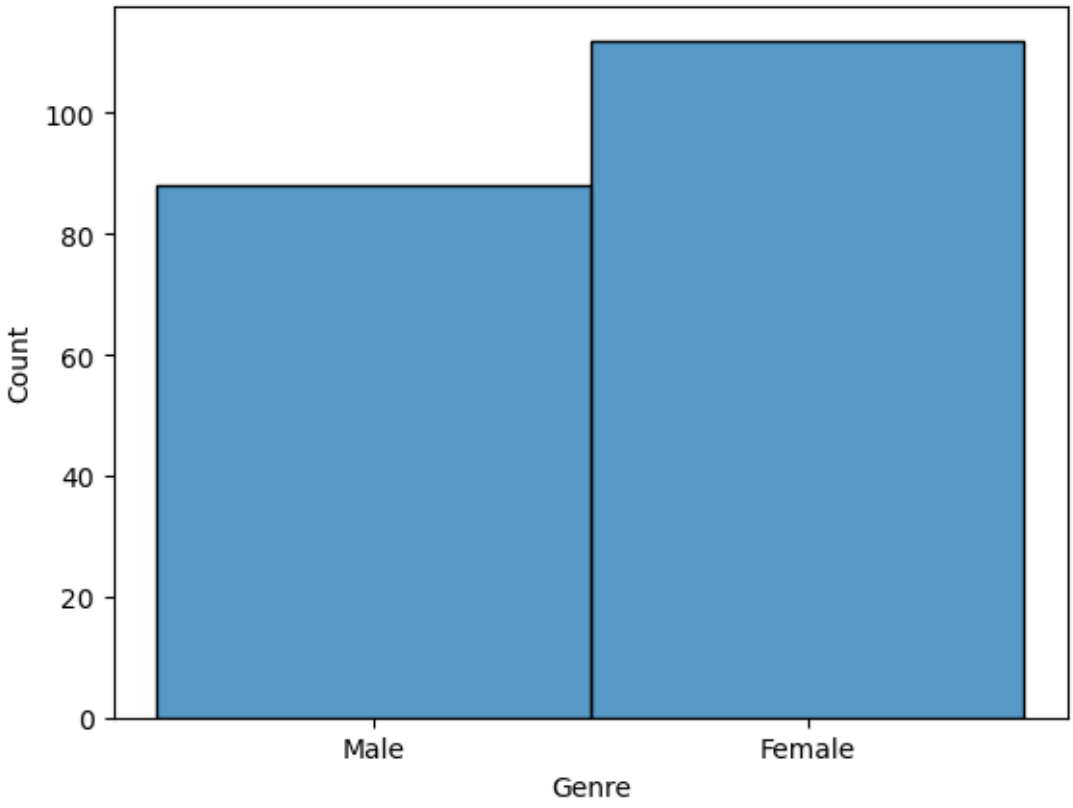
[6]:
```
(200, 5)
```

[7]:
```
df.describe()
```

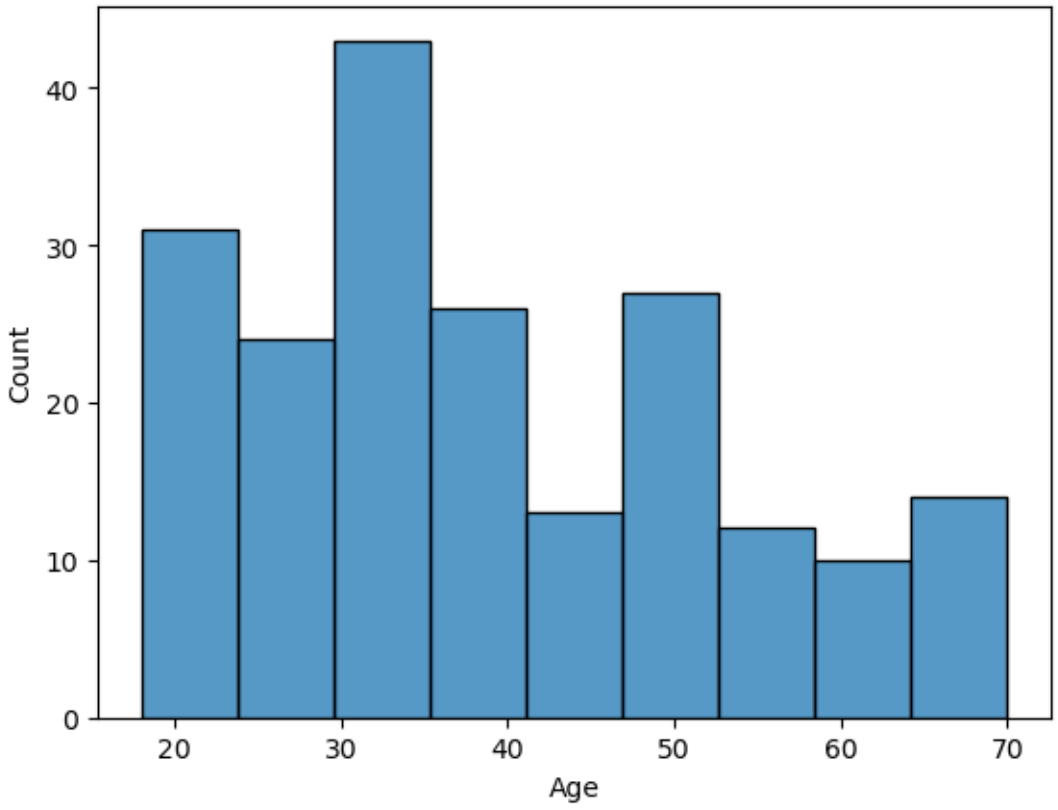
[7]:

	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

[8]:
```
sns.histplot(x='Genre' ,data = df) ;
```



[9]:
```
sns.histplot(x='Age' ,data = df) ;
```



[10]:
```
df.isnull().sum()
```

[10]:
```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

[11]:
```
df=df.drop(['CustomerID'],axis=1)
```

[12]:
```
df.columns
```

[12]:
```
Index(['Genre', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)'],
dtype='object')
```

[13]:
```
df.rename(columns={'Annual Income (k$)':'Annual_Income'},inplace=True)
```

[14]:
```
df.rename(columns={'Spending Score (1-100)':'Spending_Score'},inplace=True)
```

[15]:
```
df.Genre.replace({'Male':1 ,'Female':0},inplace=True)
```

[16]:
```
df
```

[16]:

	Genre	Age	Annual_Income	Spending_Score
0	1	19	15	39
1	1	21	15	81
2	0	20	16	6
3	0	23	16	77
4	0	31	17	40
5	0	22	17	76
6	0	35	18	6
7	0	23	18	94
8	1	64	19	3
9	0	30	19	72
10	1	67	19	14
11	0	35	19	99
12	0	58	20	15
13	0	24	20	77
14	1	37	20	13
15	1	22	20	79
16	0	35	21	35
17	1	20	21	66
18	1	52	23	29
19	0	35	23	98
20	1	35	24	35
21	1	25	24	73
22	0	46	25	5
23	1	31	25	73
24	0	54	28	14
25	1	29	28	82
26	0	45	28	32
27	1	35	28	61
28	0	40	29	31
29	0	23	29	87
30	1	60	30	4
31	0	21	30	73
32	1	53	33	4
33	1	18	33	92
34	0	49	33	14
35	0	21	33	81
36	0	42	34	17
37	0	30	34	73
38	0	36	37	26
39	0	20	37	75
40	0	65	38	35
41	1	24	38	92
42	1	48	39	36
43	0	31	39	61

44	0	49	39	28
45	0	24	39	65
46	0	50	40	55
47	0	27	40	47
48	0	29	40	42
49	0	31	40	42
50	0	49	42	52
51	1	33	42	60
52	0	31	43	54
53	1	59	43	60
54	0	50	43	45
55	1	47	43	41
56	0	51	44	50
57	1	69	44	46
58	0	27	46	51
59	1	53	46	46
60	1	70	46	56
61	1	19	46	55
62	0	67	47	52
63	0	54	47	59
64	1	63	48	51
65	1	18	48	59
66	0	43	48	50
67	0	68	48	48
68	1	19	48	59
69	0	32	48	47
70	1	70	49	55
71	0	47	49	42
72	0	60	50	49
73	0	60	50	56
74	1	59	54	47
75	1	26	54	54
76	0	45	54	53
77	1	40	54	48
78	0	23	54	52
79	0	49	54	42
80	1	57	54	51
81	1	38	54	55
82	1	67	54	41
83	0	46	54	44
84	0	21	54	57
85	1	48	54	46
86	0	55	57	58
87	0	22	57	55
88	0	34	58	60
89	0	50	58	46
90	0	68	59	55

91	1	18	59	41
92	1	48	60	49
93	0	40	60	40
94	0	32	60	42
95	1	24	60	52
96	0	47	60	47
97	0	27	60	50
98	1	48	61	42
99	1	20	61	49
100	0	23	62	41
101	0	49	62	48
102	1	67	62	59
103	1	26	62	55
104	1	49	62	56
105	0	21	62	42
106	0	66	63	50
107	1	54	63	46
108	1	68	63	43
109	1	66	63	48
110	1	65	63	52
111	0	19	63	54
112	0	38	64	42
113	1	19	64	46
114	0	18	65	48
115	0	19	65	50
116	0	63	65	43
117	0	49	65	59
118	0	51	67	43
119	0	50	67	57
120	1	27	67	56
121	0	38	67	40
122	0	40	69	58
123	1	39	69	91
124	0	23	70	29
125	0	31	70	77
126	1	43	71	35
127	1	40	71	95
128	1	59	71	11
129	1	38	71	75
130	1	47	71	9
131	1	39	71	75
132	0	25	72	34
133	0	31	72	71
134	1	20	73	5
135	0	29	73	88
136	0	44	73	7
137	1	32	73	73

138	1	19	74	10
139	0	35	74	72
140	0	57	75	5
141	1	32	75	93
142	0	28	76	40
143	0	32	76	87
144	1	25	77	12
145	1	28	77	97
146	1	48	77	36
147	0	32	77	74
148	0	34	78	22
149	1	34	78	90
150	1	43	78	17
151	1	39	78	88
152	0	44	78	20
153	0	38	78	76
154	0	47	78	16
155	0	27	78	89
156	1	37	78	1
157	0	30	78	78
158	1	34	78	1
159	0	30	78	73
160	0	56	79	35
161	0	29	79	83
162	1	19	81	5
163	0	31	81	93
164	1	50	85	26
165	0	36	85	75
166	1	42	86	20
167	0	33	86	95
168	0	36	87	27
169	1	32	87	63
170	1	40	87	13
171	1	28	87	75
172	1	36	87	10
173	1	36	87	92
174	0	52	88	13
175	0	30	88	86
176	1	58	88	15
177	1	27	88	69
178	1	59	93	14
179	1	35	93	90
180	0	37	97	32
181	0	32	97	86
182	1	46	98	15
183	0	29	98	88
184	0	41	99	39

```
185      1   30             99              97
186      0   54            101              24
187      1   28            101              68
188      0   41            103              17
189      0   36            103              85
190      0   34            103              23
191      0   32            103              69
192      1   33            113               8
193      0   38            113              91
194      0   47            120              16
195      0   35            120              79
196      0   45            126              28
197      1   32            126              74
198      1   32            137              18
199      1   30            137              83
```

[17]:
```
X= df.iloc[:, [2, 3]].values
```

[18]:
```
X
```

[18]:
```
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],
```

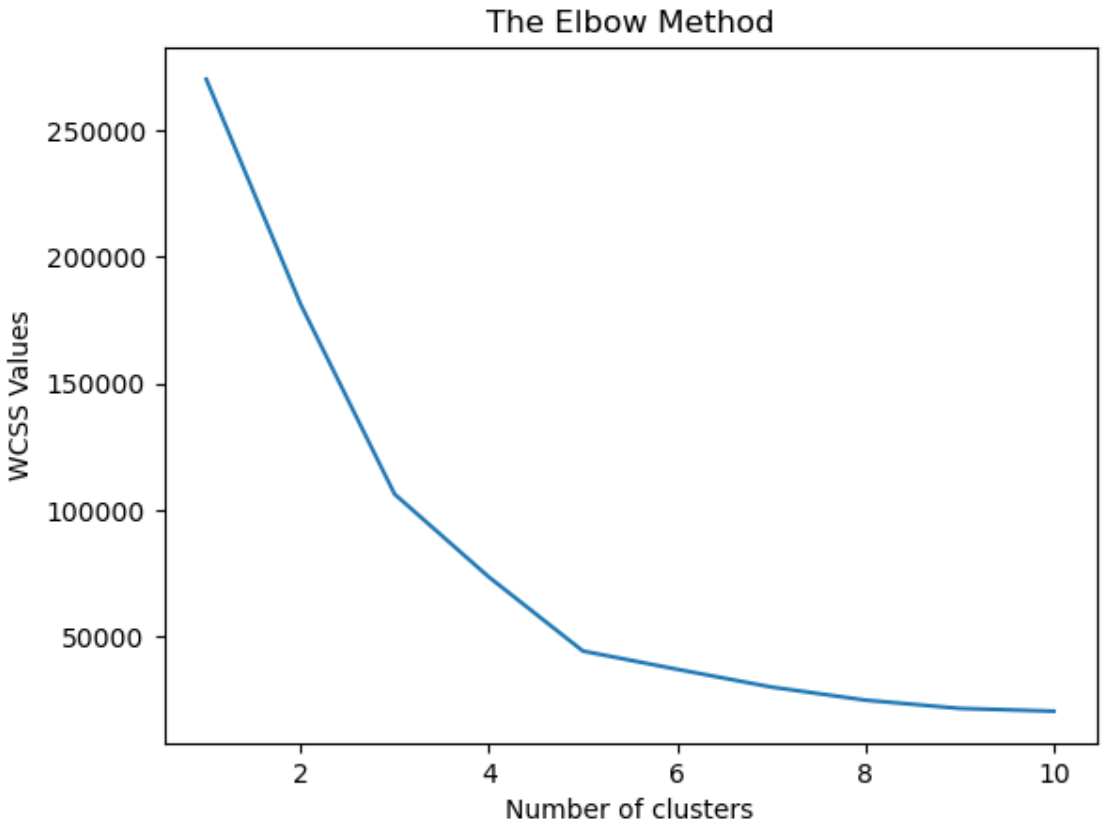
```
       [ 87,  27],
       [ 87,  63],
       [ 87,  13],
       [ 87,  75],
       [ 87,  10],
       [ 87,  92],
       [ 88,  13],
       [ 88,  86],
       [ 88,  15],
       [ 88,  69],
       [ 93,  14],
       [ 93,  90],
       [ 97,  32],
       [ 97,  86],
       [ 98,  15],
       [ 98,  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)
```

[19]:
```
from sklearn.cluster import KMeans
wcss = []
```

[20]:
```
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
```

[21]:
```
plt.plot(range(1, 11), wcss)
plt.title("The Elbow Method")
plt.xlabel('Number of clusters')
plt.ylabel('WCSS Values')
```

[21]:
```
Text(0, 0.5, 'WCSS Values')
```



2 Training a model using Unsupervised Learning Algorithm

[22]:
```
kmeans_model = KMeans(n_clusters =5 , init='k-means++',random_state=0)
y_kmeans = kmeans_model.fit_predict(X)  # model
```

[23]:
```
kmeans_model.cluster_centers_
```

[23]:
```
array([[88.2       , 17.11428571],
       [55.2962963 , 49.51851852],
       [86.53846154, 82.12820513],
       [25.72727273, 79.36363636],
       [26.30434783, 20.91304348]])
```

[24]:
```
kmeans_model.score(X)  # this is the value of SSD given number of cluster
```

[24]:
```
-44448.45544793371
```

[25]:
```
kmeans_model.cluster_centers_
```

[25]:
```
array([[88.2       , 17.11428571],
       [55.2962963 , 49.51851852],
       [86.53846154, 82.12820513],
       [25.72727273, 79.36363636],
       [26.30434783, 20.91304348]])
```

[26]:
```
plt.scatter(X[y_kmeans == 0,0], X[y_kmeans == 0,1], s= 80, c = "red",␣
 ↪label='Customer 1')
plt.scatter(X[y_kmeans == 1,0], X[y_kmeans == 1,1], s= 80, c = "blue",␣
 ↪label='Customer 2')
plt.scatter(X[y_kmeans == 2,0], X[y_kmeans == 2,1], s= 80, c = "yellow",␣
 ↪label='Customer 3')
plt.scatter(X[y_kmeans == 3,0], X[y_kmeans == 3,1], s= 80, c = "cyan",␣
 ↪label='Customer 4')
plt.scatter(X[y_kmeans == 4,0], X[y_kmeans == 4,1], s= 80, c = "black",␣
 ↪label='Customer 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=␣
 ↪100, c= 'magenta', label='Centroids')
plt.title('Clusters of Customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```


Clusters of Customers
Spending Score (1-100)
Annual Income (k$)
Customer 1
Customer 2
Customer 3
Customer 4
Customer 5
Centroids
0
20
40
60
80
100
20
40
60
80
100
120
140

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: