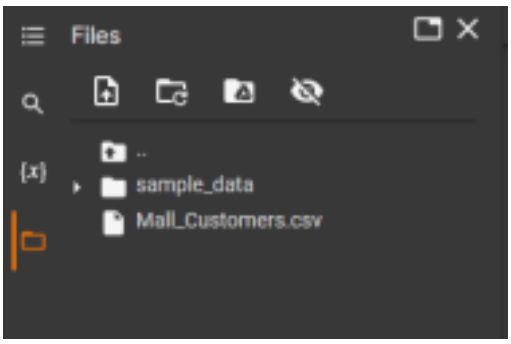
Kawar Nilesh Ramesh Roll no 59: 59

## Practical 5

## Implement K-Means Clustering Algorithm

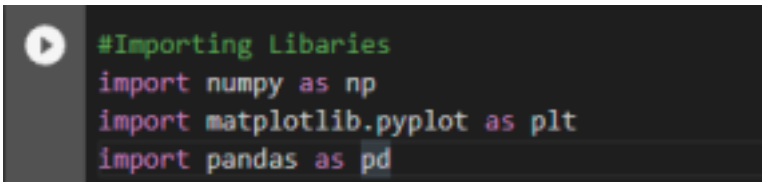
### Program with Output:

Upload Mall_Customers.csv file



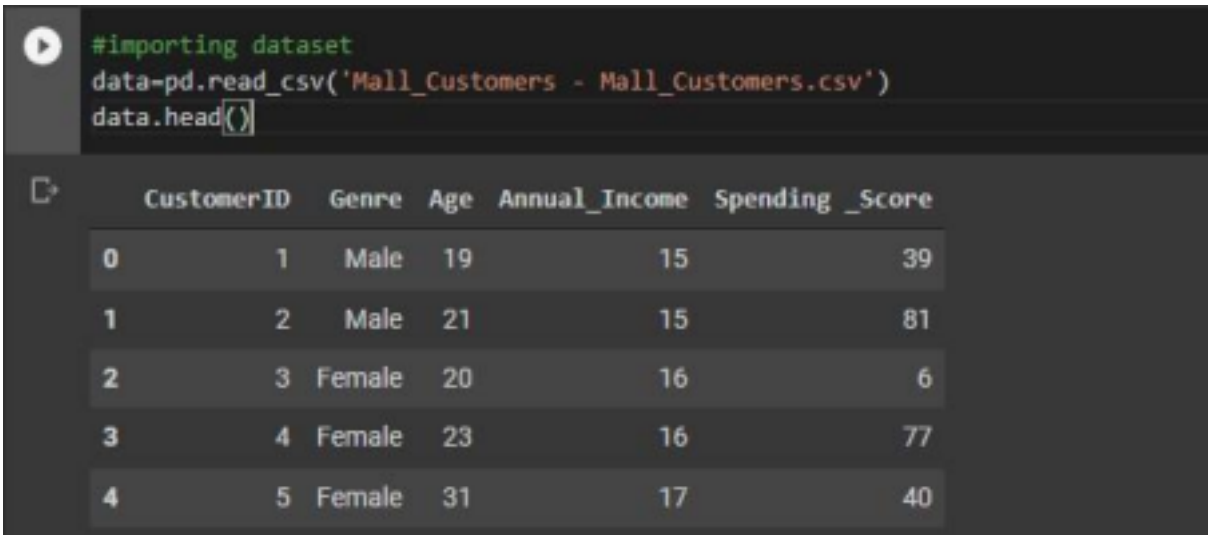

```
#Importing Libaries
```

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




```
#importing dataset
data=pd.read_csv('Mall_Customers - Mall_Customers.csv') data.head()
```

Batch A
Kawar Nilesh Ramesh Roll no 59: 59
Batch A




| | CustomerID | Genre | 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 |

```
x=data.iloc[:,[3,4]].values
x
```

```
x=data.iloc[:,[3,4]].values
x
```

```
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],
```

```
#finding optimal number of cluster using elbow method
from sklearn.cluster import KMeans
wcss_list=[] #initializing the list for wcss value

#putting different values of k ranging from 1-11
for i in range (1,11):
  kmeans=KMeans(n_clusters=i, init='k-means++',
  random_state=42) kmeans.fit(x)
  print(kmeans.inertia_)
  wcss_list.append(kmeans.inertia_)
```

Kawar Nilesh Ramesh Roll no 59: 59

```
plt.xlabel('number of clusters(k)')
plt.ylabel('wcss_list')
plt.show()
```

```
plt.plot(range(1,11),wcss_list)
plt.title('The Elbow method graph')
```

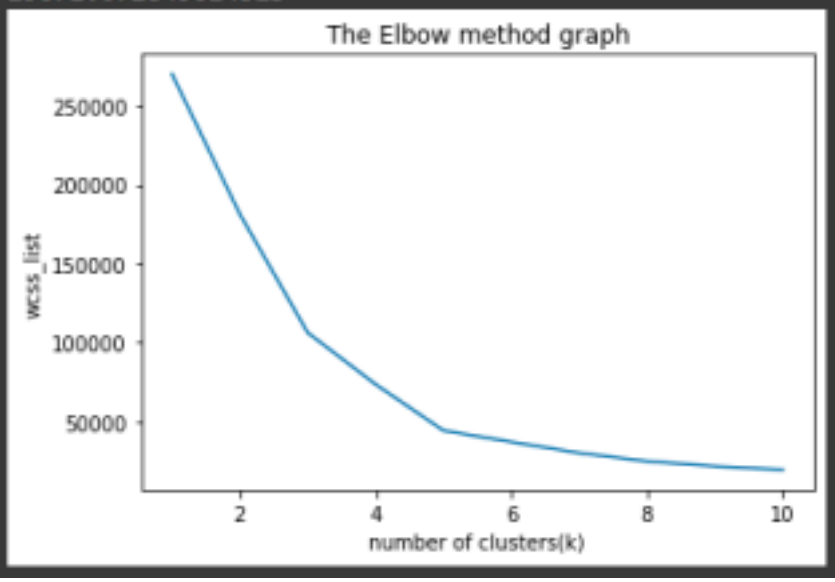
Batch A

```python
#finding optimal number of cluster using elbow method
from sklearn.cluster import KMeans
wcss_list=[]    #initializing the list for wcss value

#putting different values of k ranging from 1-11
for i in range (1,11):
  kmeans=KMeans(n_clusters=i, init='k-means++', random_state=42)
  kmeans.fit(x)
  print(kmeans.inertia_)
  wcss_list.append(kmeans.inertia_)

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

```
269981.28
181363.59595959593
106348.37306211122
73679.78903948836
44448.4554479337
37233.814510710006
30259.65720728547
25011.839349156588
21850.165282585636
19672.072849014323
```



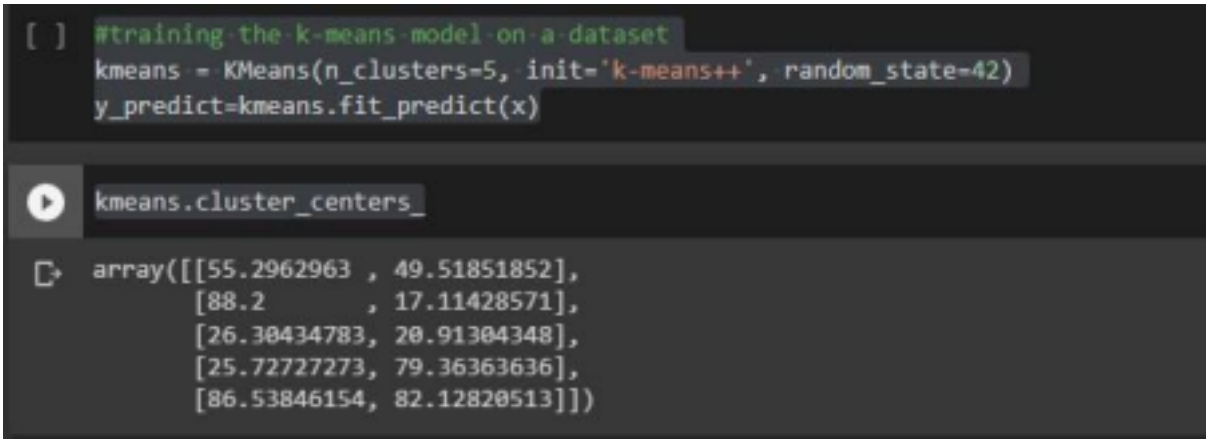

Kawar Nilesh Ramesh Roll no 59:

```
init='k-means++', random_state=42)
y_predict=kmeans.fit_predict(x)
```

```
kmeans.cluster_centers_
```

Batch A

```
#training the k-means model on a
dataset
kmeans = KMeans(n_clusters=5,
```



```
[ ] #training the k-means model on a dataset
kmeans = KMeans(n_clusters=5, init='k-means++', random_state=42)
y_predict=kmeans.fit_predict(x)

kmeans.cluster_centers_

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

```
#visualization of clusters
plt.scatter(x[:,0],x[:,1],c=y_predict, s=50, cmap='viridis')

plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1]
, s=100, c='orange', label='centroid')
plt.title('Clusters of Customers')
plt.xlabel('Annual income of k $')
plt.ylabel('spending Score(1-100)')
plt.legend()
plt.show()
```

```
#visualization of clusters
plt.scatter(x[:,0],x[:,1],c=y_predict, s=50, cmap='viridis')

plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1], s=100, c='orange', label='centroid')
plt.title('Clusters of Customers')
plt.xlabel('Annual Income of k $')
plt.ylabel('spending Score(1-100)')
plt.legend()
plt.show()
```