Program-1 for Kmeans

Basic Program without using external dataset

```
from sklearn.cluster import KMeans
import numpy as np
```

Data set for height and weight

```
X = np.array([[185,72], [170,56], [168,60],
              [179,68],[182,72], [188,77], [180,71],[180,70],[183,84],[180,88],[180,67],[
```

```
kmeans = KMeans(n_clusters=2, random_state=0).fit(X)  #k=2
```

```
kmeans.labels_
```

```
array([0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
kmeans.predict([[170,56], [125, 30]])  # prediction for new dataset
```

```
array([1, 1])
```

```
kmeans.cluster_centers_   #centroid for clusters (X,Y)
```

```
array([[181.4,  74.5],
       [169. ,  58. ]])
```

Program-2 for Kmeans

K-means Clustering using external Dataset

Import Libraries

```
import pandas as pd
from sklearn.cluster import KMeans
```

```
data_df = pd.read_csv('Cust_Spend_Data_New.csv')
```

```
data_df.head()
```

	Cust_ID	Name	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples
0	1	Abraham	1123	28	1	16	
1	2	Adela	9818	13	5	2	
2	3	Adelina	9824	10	10	2	
3	4	Adrian	3097	23	2	8	
4	5	Adrianna	817	28	1	17	

```
cust_df = data_df.drop(['Name','Cust_ID'], axis=1)
```

```
cust_df.head()
```

	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
0	1123	28	1	16	14
1	9818	13	5	2	5
2	9824	10	10	2	2
3	3097	23	2	8	9
4	817	28	1	17	17

```
from sklearn.preprocessing import StandardScaler
```

```
X = StandardScaler()
```

```
scaled_df = X.fit_transform(cust_df)
```

```
scaled_df
```

```
array([[-1.7896947 ,  2.32407237, -1.44109551,  2.28639713,  1.33272373],
       [ 2.44092704, -1.1127025 ,  0.50541069, -1.53502065, -0.78605994],
       [ 2.44384639, -1.80005748,  2.93854344, -1.53502065, -1.49232116],
       ...,
       [-0.14707492, -0.65446585,  0.01878414,  0.10272982,  0.15562169],
       [-0.35629486,  0.49112577,  0.01878414, -0.17022859,  0.3910421 ],
       [-0.03468002,  0.03288912,  0.50541069,  0.10272982,  0.15562169]])
```

```
k_means = KMeans(n_clusters = 3)
#k_means = KMeans(n_clusters = 4)
```

```
k_means.fit(scaled_df)
```

```
KMeans(n_clusters=3)
```

```
k_means.labels_
```

```
array([2, 1, 1, 0, 2, 0, 0, 1, 0, 0, 1, 0, 1, 0, 2, 2, 2, 0, 0, 2, 0, 1,
       1, 2, 2, 1, 0, 2, 2, 2, 2, 0, 0, 0, 2, 2, 1, 0, 1, 2, 1, 2, 2, 2,
       1, 2, 1, 2, 1, 2, 2, 1, 0, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 1, 0,
       0, 2, 2, 2, 2, 1, 0, 1, 2, 0, 1, 0, 1, 0, 1, 1, 0, 1, 2, 1, 1, 1,
       0, 2, 2, 0, 1, 0, 2, 1, 2, 1, 2, 1, 0, 2, 2, 1, 0, 2, 1, 1, 1, 2,
       1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 2, 2, 1, 2, 1, 1, 2, 1, 0, 1,
       2, 2, 0, 0, 0, 1, 0, 0, 2, 2, 2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1,
       0, 2, 1, 2, 2, 0, 0, 2, 0, 1, 0, 1, 0, 2, 2, 0, 0, 1, 0, 1, 0, 1,
       1, 1, 1, 2, 1, 0, 1, 0, 0, 2, 2, 1, 1, 2, 2, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 2, 1, 0, 2, 0, 1, 1, 0, 2, 2, 0, 2, 2, 2, 0, 1, 2, 0, 2, 1,
       1, 1, 0, 0, 2, 2, 1, 1, 0, 1, 2, 1, 1, 0, 1, 1, 1, 0, 2, 2, 2, 2,
       0, 2, 1, 2, 1, 2, 1, 2, 1, 0, 0, 1, 1, 0, 1, 2, 0, 2, 1, 2, 2, 1,
       2, 2, 0, 1, 1, 2, 1, 1, 1, 1, 0, 1, 0, 1, 2, 2, 1, 0, 1, 1, 1, 2,
       2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0,
       2, 2, 2, 1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 2, 0, 2, 0, 1, 1, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
k_means.inertia_    #The K-means algorithm aims to choose centroids that minimize the ine
```

```
1074.382380298951
```

```
labels = k_means.labels_
```

```
data_df["Clus_kmeans"] = labels
data_df.head(10)
```

	Cust_ID	Name	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staple
0	1	Abraham	1123	28	1	16	
1	2	Adela	9818	13	5	2	
2	3	Adelina	9824	10	10	2	
3	4	Adrian	3097	23	2	8	
4	5	Adrianna	817	28	1	17	
5	6	Aide	3039	21	1	8	
6	7	Alex	4676	22	3	8	
7	8	Alexandria	7869	14	6	3	
8	9	Alline	5585	21	2	10	
9	10	Allyn	4589	23	2	8	

Start coding or generate with AI.