

MoP – K-Means Clustering / segmentation

1. Environment Setup

Tools needed:

- Anaconda (recommended) or any Python 3 installation
- Launch Anaconda Navigator → click Jupyter Notebook

Use same folder: "machine_learning"

Use same notebook in jupyter browse in above folder: "machinelearning"

2. Create or Download the Sample Dataset

You can create a small file manually in Jupyter to simulate a telecom dataset.

```
import pandas as pd
```

```
data = """customer_id,data_gb,call_mins  
1001,5.2,230  
1002,12.5,410  
1003,7.8,320  
1004,15.6,580  
1005,3.4,120  
1006,9.0,360  
1007,2.5,80  
1008,11.0,400  
1009,18.3,600  
1010,4.0,200  
"""
```

```
open("telecom_usage.csv","w").write(data)  
print("✓ Sample telecom_usage.csv created.")
```

This file contains 10 customers' data usage (in GB) and call minutes.

3. Load the Dataset

```
import pandas as pd
```

```
df = pd.read_csv("telecom_usage.csv")  
print("✓ Data loaded successfully!")  
print(df.head())
```

 Output:

```
customer_id  data_gb      call_mins
1001    5.2    230
1002   12.5    410
1003    7.8    320
```

4. Import Required Libraries

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
```

5. Apply K-Means Clustering

We'll create 3 clusters — light, moderate, and heavy users.

```
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(df[['data_gb', 'call_mins']])
df['cluster'] = kmeans.labels_
print(df)
```

Explanation:

- n_clusters=3 → we want 3 customer groups.
- fit() → trains the algorithm to find clusters.
- labels_ → assigns each customer to a cluster.

Example output:

```
customer_id  data_gb      call_mins      cluster
1001    5.2    230      1
1002   12.5    410      2
1003    7.8    320      1
1004   15.6    580      2
1007    2.5     80      0
```

Interpretation:

- Cluster 0 -> Mod users
- Cluster 1 -> Low users
- Cluster 2 -> Heavy users