21BDS0340 - Abhinav Dinesh Srivatsa

In [1]:
```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn_extra.cluster import KMedoids
from sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.decomposition import PCA
```

In [2]:
```
data = pd.read_csv("CC GENERAL.csv")
data = data.drop('CUST_ID', axis=1)
data
```

Out[2]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	IN
0	40.900749	0.818182	95.40	0.00	
1	3202.467416	0.909091	0.00	0.00	
2	2495.148862	1.000000	773.17	773.17	
3	1666.670542	0.636364	1499.00	1499.00	
4	817.714335	1.000000	16.00	16.00	
...	...	...	...	...	
8945	28.493517	1.000000	291.12	0.00	
8946	19.183215	1.000000	300.00	0.00	
8947	23.398673	0.833333	144.40	0.00	
8948	13.457564	0.833333	0.00	0.00	
8949	372.708075	0.666667	1093.25	1093.25	

8950 rows × 17 columns

In [3]:
```
data = data.fillna(data.median())
data
```

Out[3]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	IN
0	40.900749	0.818182	95.40	0.00	
1	3202.467416	0.909091	0.00	0.00	
2	2495.148862	1.000000	773.17	773.17	
3	1666.670542	0.636364	1499.00	1499.00	
4	817.714335	1.000000	16.00	16.00	
...	...	...	...	...	
8945	28.493517	1.000000	291.12	0.00	
8946	19.183215	1.000000	300.00	0.00	
8947	23.398673	0.833333	144.40	0.00	
8948	13.457564	0.833333	0.00	0.00	
8949	372.708075	0.666667	1093.25	1093.25	

8950 rows × 17 columns

In [4]:
```
data = StandardScaler().fit_transform(data)
data
```

Out[4]:
```
array([[-0.73198937, -0.24943448, -0.42489974, ..., -0.3024    ,
        -0.52555097,  0.36067954],
       [ 0.78696085,  0.13432467, -0.46955188, ...,  0.09749953,
         0.2342269 ,  0.36067954],
       [ 0.44713513,  0.51808382, -0.10766823, ..., -0.0932934 ,
        -0.52555097,  0.36067954],
       ...,
       [-0.7403981 , -0.18547673, -0.40196519, ..., -0.32687479,
         0.32919999, -4.12276757],
       [-0.74517423, -0.18547673, -0.46955188, ..., -0.33830497,
         0.32919999, -4.12276757],
       [-0.57257511, -0.88903307,  0.04214581, ..., -0.3243581 ,
        -0.52555097, -4.12276757]])
```

K Mediods clustering

In [5]:
```
k_mediods_clusters = {}

for k in [2, 5]:
    km = KMedoids(n_clusters=k, random_state=42)
    clusters = km.fit_predict(data)
    k_mediods_clusters[k] = clusters

k_mediods_clusters
```

Out[5]:
```
{2: array([0, 0, 1, ..., 1, 0, 0]), 5: array([1, 3, 2, ..., 4, 1, 1])}
```

Agglomerative clustering

In [6]:
```
agglomerative_clusters = {}

for k in [2, 5]:
```

```
    a = AgglomerativeClustering(n_clusters=k)
    clusters = a.fit_predict(data)
    agglomerative_clusters[k] = clusters

agglomerative_clusters
```

Out[6]:
```
{2: array([0, 0, 0, ..., 0, 0, 0]), 5: array([0, 2, 0, ..., 4, 4, 4])}
```

Clustering visualisation

In [7]:
```
data = PCA(n_components=2).fit_transform(data)
data
```

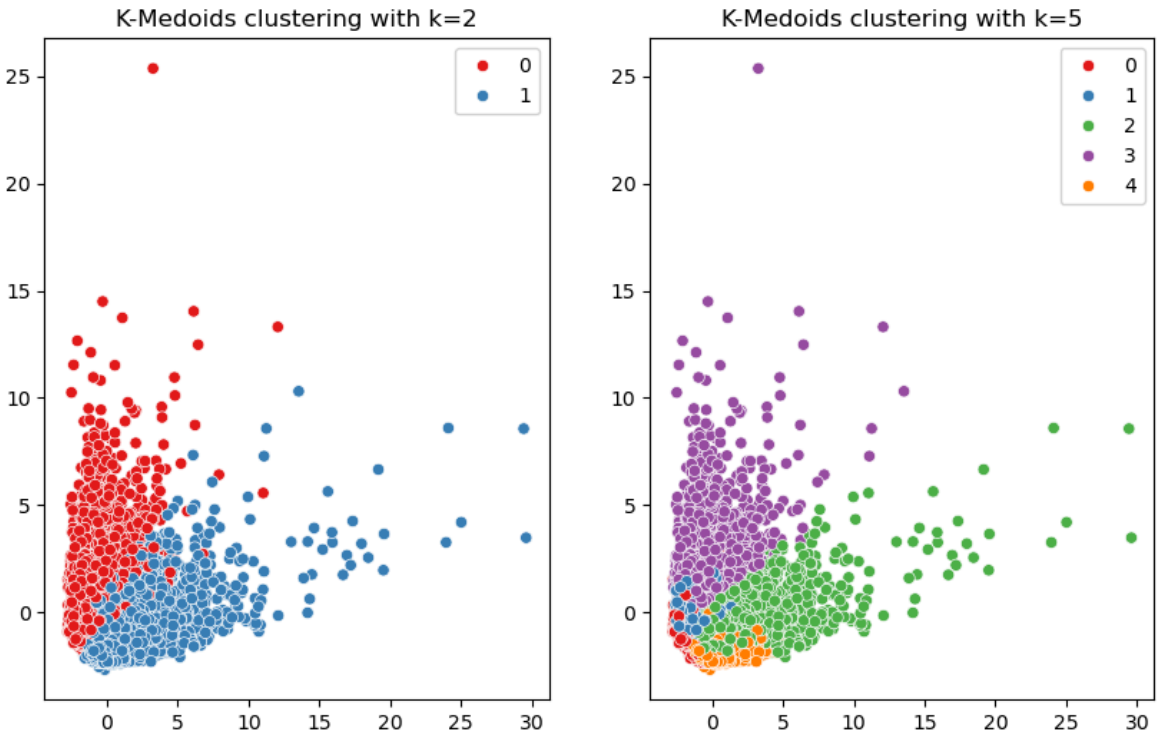
Out[7]:
```
array([[-1.68364879, -1.07224148],
       [-1.13408493,  2.50914981],
       [ 0.96939499, -0.3835769 ],
       ...,
       [-0.92898512, -1.80804835],
       [-2.33784475, -0.65361133],
       [-0.55802653, -0.4006461 ]])
```

In [8]:
```
plt.figure(figsize=(10, 6))

plt.subplot(1, 2, 1)
sns.scatterplot(x=data[:, 0], y=data[:, 1], hue=k_mediods_clusters[2], pa
plt.title(f'K-Medoids clustering with k=2')

plt.subplot(1, 2, 2)
sns.scatterplot(x=data[:, 0], y=data[:, 1], hue=k_mediods_clusters[5], pa
plt.title(f'K-Medoids clustering with k=5')
```

Out[8]:
```
Text(0.5, 1.0, 'K-Medoids clustering with k=5')
```



In [9]:
```
plt.figure(figsize=(10, 6))

plt.subplot(1, 2, 1)
sns.scatterplot(x=data[:, 0], y=data[:, 1], hue=agglomerative_clusters[2]
```

```
plt.title(f'Agglomerative clustering with k=2')

plt.subplot(1, 2, 2)
sns.scatterplot(x=data[:, 0], y=data[:, 1], hue=agglomerative_clusters[5]
plt.title(f'Agglomerative clustering with k=5')
```

Out[9]: Text(0.5, 1.0, 'Agglomerative clustering with k=5')