ML Lab 11: Clustering

Name: Soumyadeep Ganguly

Reg No: 24MDT0082

In [18]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
```

In [19]:
```
X, _ = make_blobs(n_samples = 300, centers=4, cluster_std = 2, random_state= 42)
X = StandardScaler().fit_transform(X)
```

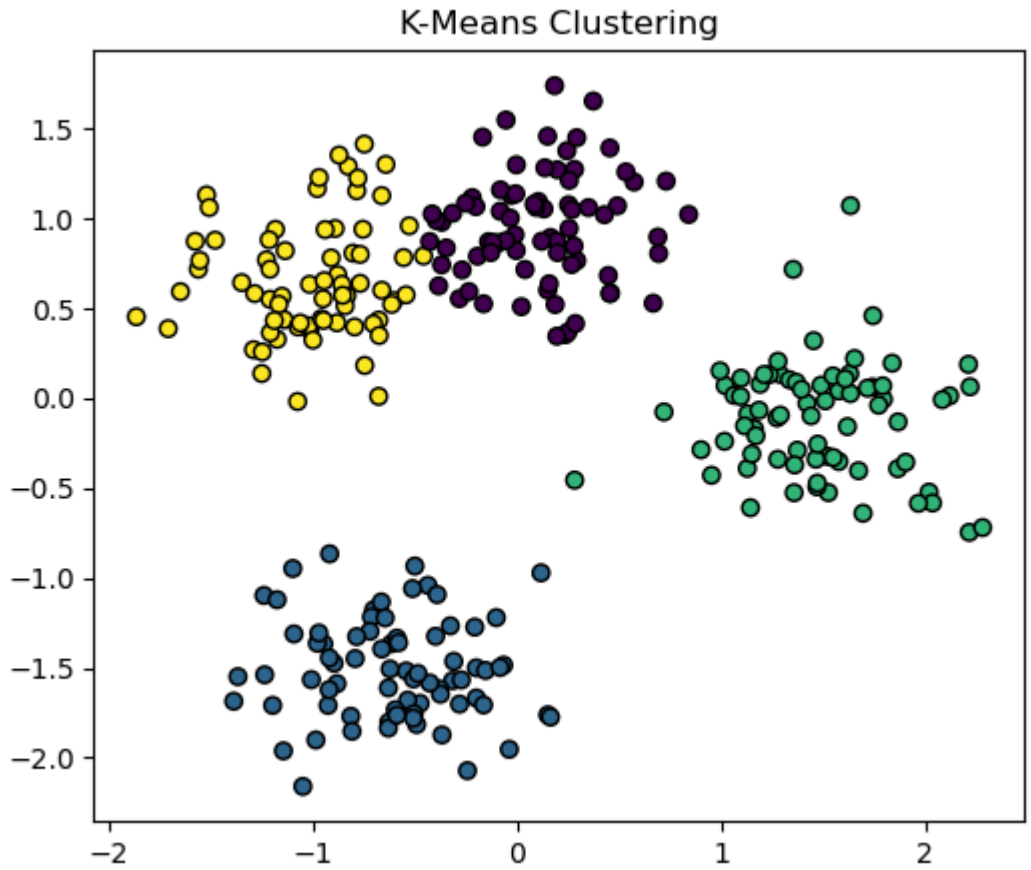
K-Means Clustering

In [7]:
```
k = 4
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans_labels = kmeans.fit_predict(X)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will ch
ange from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak o
n Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_
NUM_THREADS=2.
  warnings.warn(
```

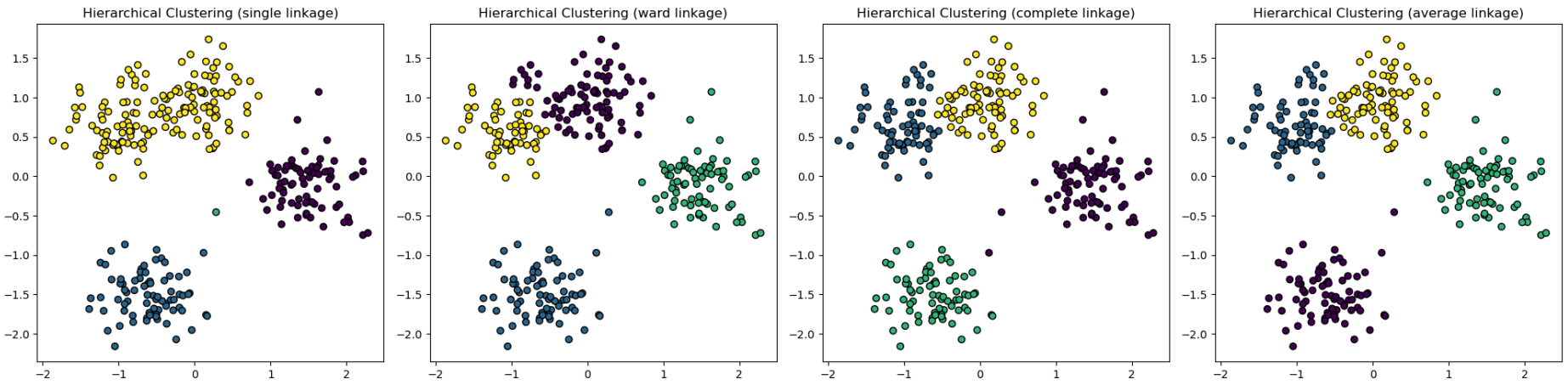
In [8]:
```
plt.figure(figsize=(6,5))
plt.scatter(X[:, 0], X[:, 1], c = kmeans_labels, cmap="viridis", edgecolors= 'k')
```

```
plt.title("K-Means Clustering")
plt.show()
```

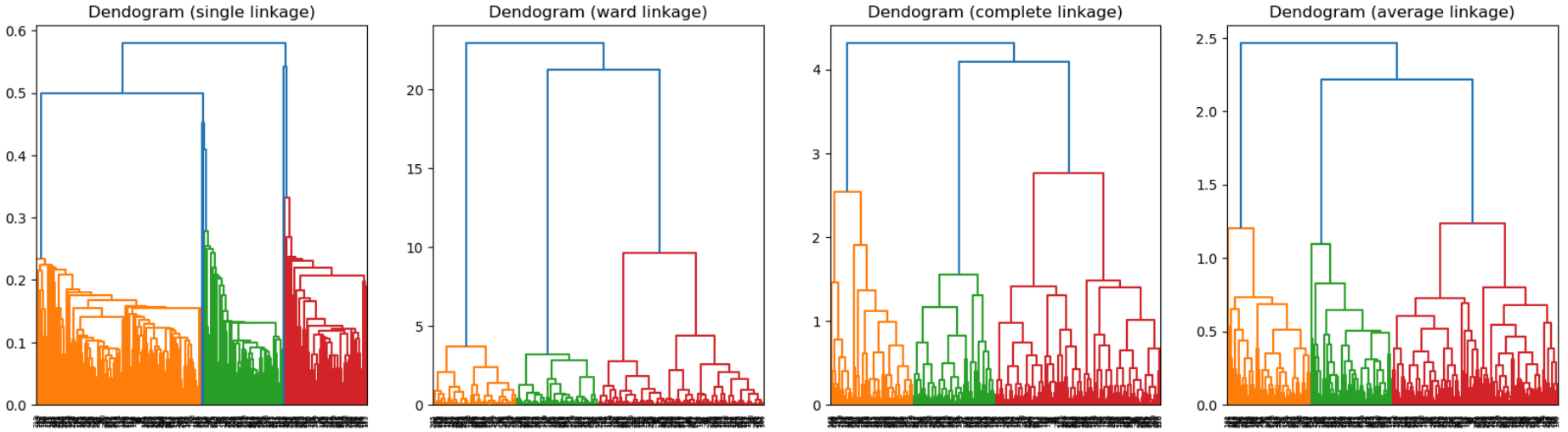


Agglomeratie Clustering

In [14]:
```
linkage_methods = ['single', 'ward', 'complete', 'average']
fig,ax = plt.subplots(1,4, figsize=(20,5))
for i, method in enumerate(linkage_methods):
    agm = AgglomerativeClustering(n_clusters=4, linkage=method)
    labels = agm.fit_predict(X)
    ax[i].scatter(X[:, 0], X[:, 1], c = labels, cmap="viridis", edgecolors= 'k')
    ax[i].set_title(f"Hierarchical Clustering ({method} linkage)")
plt.tight_layout()
plt.show()
```



In [23]:
```
fig,ax = plt.subplots(1,4, figsize=(20,5))
for i, method in enumerate(linkage_methods):
    linkage_matrix = linkage(X, method=method)
    ax[i].set_title(f"Dendogram ({method} linkage)")
    dendrogram(linkage_matrix, ax = ax[i])
plt.show()
```



In [ ]: