Unsupervised Clustering Algorithms

1. KMeans

```
# Import necessary libraries
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
# Generate synthetic data using make_blobs
X, y = make_blobs(n_samples=300, centers=4, random_state=42)
```

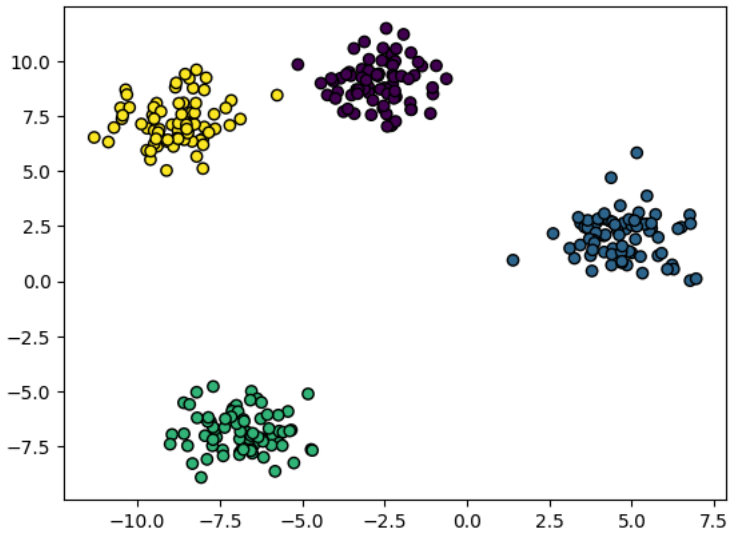
```
# Apply K-means clustering
kmeans = KMeans(n_clusters=4, random_state=42)
kmeans.fit(X)
```

```
KMeans
KMeans(n_clusters=4, random_state=42)
```

```
# Get cluster labels and centroids
kmeans_labels = kmeans.labels_
centroids = kmeans.cluster_centers_
```

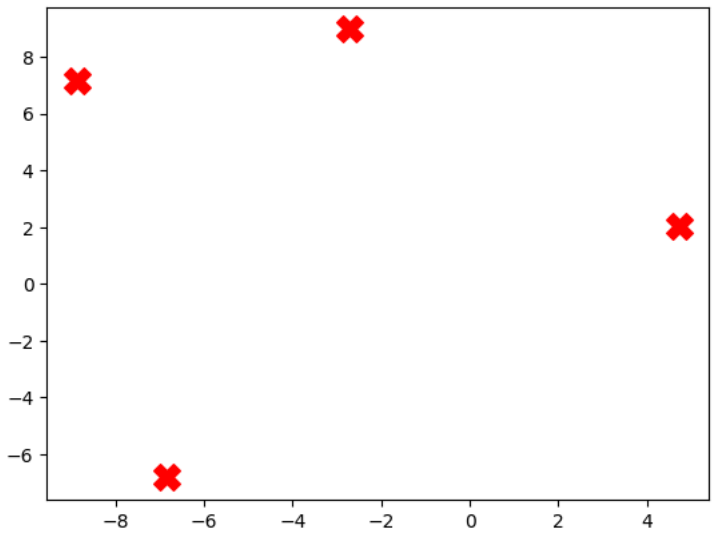
```
# Plot the original data points with colors representing the ground truth clusters
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='viridis', marker='o', edgecolors='k', label='Ground Truth')
```

```
<matplotlib.collections.PathCollection at 0x7af1721d6c50>
```

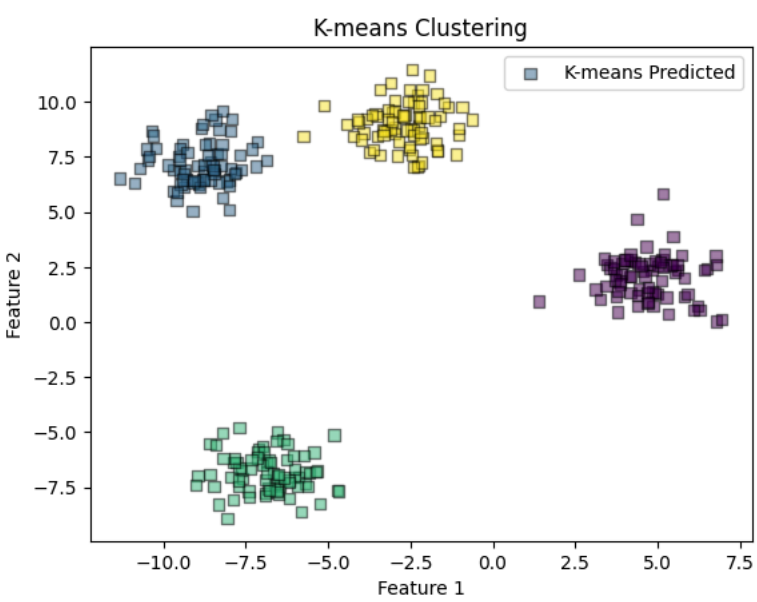



```
# Plot the cluster centroids
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='X', s=200, label='Centroids')
```

```
<matplotlib.collections.PathCollection at 0x7af17210e0b0>
```



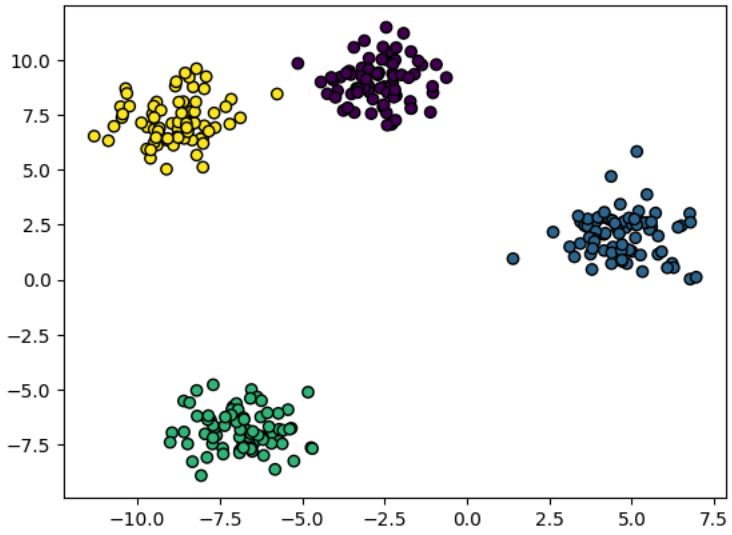
```
# Plot the data points with colors representing the predicted clusters by K-means
plt.scatter(X[:, 0], X[:, 1], c=kmeans_labels, cmap='viridis', marker='s', edgecolors='k', alpha=0.5, label='K-means Predicted')
plt.title('K-means Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.show()
```



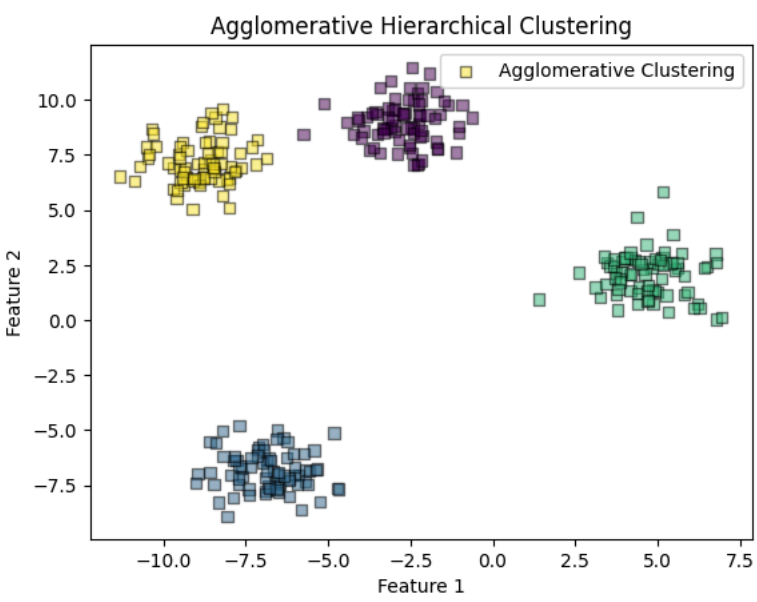
2. Agglomerative Clustering

```
# Import necessary libraries
from sklearn.datasets import make_blobs
from sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt


# Generate synthetic data using make_blobs
X, y = make_blobs(n_samples=300, centers=4, random_state=42)


# Apply Agglomerative Hierarchical Clustering
agg_clustering = AgglomerativeClustering(n_clusters=4)
agg_labels = agg_clustering.fit_predict(X)


# Plot the original data points with colors representing the ground truth clusters
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='viridis', marker='o', edgecolors='k', label='Ground Truth')
```

```
<matplotlib.collections.PathCollection at 0x7af170835600>
```



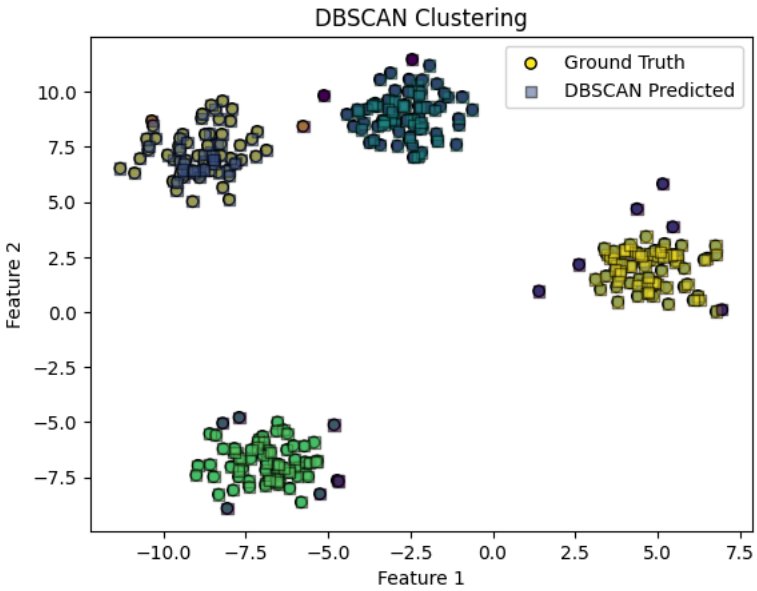
```
# Plot the data points with colors representing the predicted clusters by Agglomerative Clustering
plt.scatter(X[:, 0], X[:, 1], c=agg_labels, cmap='viridis', marker='s', edgecolors='k', alpha=0.5, label='Agglomerative Clustering')
plt.title('Agglomerative Hierarchical Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.show()
```



3. DBSCAN

```
# Import necessary libraries
from sklearn.datasets import make_blobs
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt

# Generate synthetic data using make_blobs
X, y = make_blobs(n_samples=300, centers=4, random_state=42)

# Apply DBSCAN clustering
dbscan_clustering = DBSCAN(eps=0.8, min_samples=5)
dbscan_labels = dbscan_clustering.fit_predict(X)

# Plot the original data points with colors representing the ground truth clusters
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='viridis', marker='o', edgecolors='k', label='Ground Truth')
# Plot the data points with colors representing the predicted clusters by DBSCAN
plt.scatter(X[:, 0], X[:, 1], c=dbscan_labels, cmap='viridis', marker='s', edgecolors='k', alpha=0.5, label='DBSCAN Predicted')
plt.title('DBSCAN Clustering')
plt.xlabel('Feature 1')
```

```
plt.ylabel('Feature 2')
plt.legend()
plt.show()
```



4. Hierarchical Clustering

```
# Import necessary libraries
from sklearn.datasets import make_blobs
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt
# Generate synthetic data using make_blobs
X, y = make_blobs(n_samples=300, centers=4, random_state=42)
# Apply Hierarchical Clustering
linkage_matrix = linkage(X, method='ward')
# Plot the dendrogram
plt.figure(figsize=(7.5,7.5))
dendrogram(linkage_matrix)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()
```

Hierarchical Clustering Dendrogram