9. IMPLEMENTATION OF VARIOUS CLUSTERING TECHNIQUES

Contents

Introduction:

Certainly! Clustering is an unsupervised machine learning technique that involves grouping similar data points together. Here are examples of implementing three popular clustering techniques using Python and scikit-learn:

1. K-Means Clustering:

```python
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

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

# Create a KMeans clusterer
kmeans = KMeans(n_clusters=4, random_state=42)

# Fit the model to the data
kmeans.fit(X)

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

# Plot the clusters and centroids
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis', alpha=0.7)
plt.scatter(centroids[:, 0], centroids[:, 1], marker='X', s=200, linewidths=2, color='red', label='Centroids')
plt.title('K-Means Clustering')
plt.legend()
plt.show()
```

2. Hierarchical Clustering (Agglomerative):

```python
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

# Create an Agglomerative clustering model
agg_clustering = AgglomerativeClustering(n_clusters=4)

# Fit the model to the data
agg_labels = agg_clustering.fit_predict(X)

# Plot the dendrogram
linkage_matrix = linkage(X, 'ward')
dendrogram(linkage_matrix)
plt.title('Hierarchical Clustering Dendrogram')
plt.show()

# Plot the clustered data points
plt.scatter(X[:, 0], X[:, 1], c=agg_labels, cmap='viridis', alpha=0.7)
plt.title('Hierarchical Clustering')
plt.show()
```

3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

```python
from sklearn.cluster import DBSCAN
```

```
import matplotlib.pyplot as plt


# Create a DBSCAN clustering model
dbscan = DBSCAN(eps=0.5, min_samples=5)


# Fit the model to the data
dbscan_labels = dbscan.fit_predict(X)


# Plot the clustered data points
plt.scatter(X[:, 0], X[:, 1], c=dbscan_labels, cmap='viridis', alpha=0.7)
plt.title('DBSCAN Clustering')
plt.show()
```

These examples cover three different clustering algorithms: K-Means, Hierarchical (Agglomerative), and DBSCAN. Each algorithm has its own parameters and assumptions, so it's important to choose the one that fits your data characteristics. Additionally, you might need to perform data preprocessing and parameter tuning for optimal results in a real-world scenario.