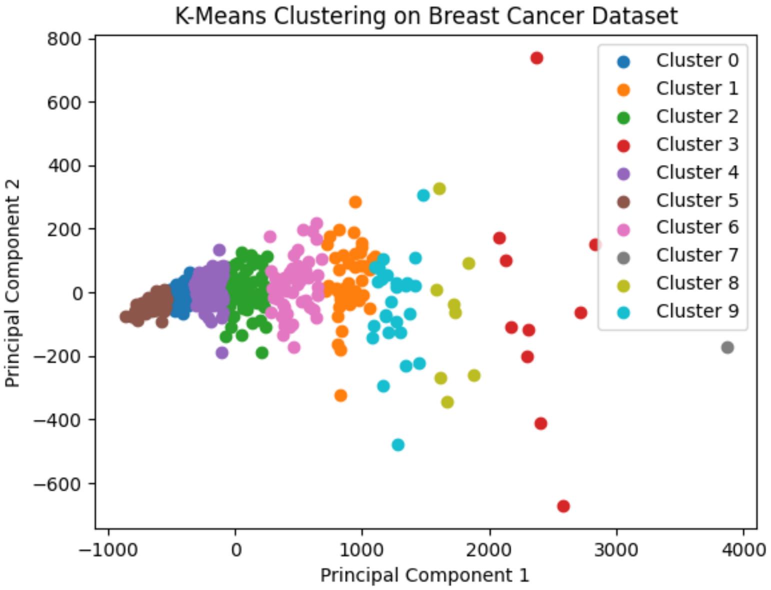
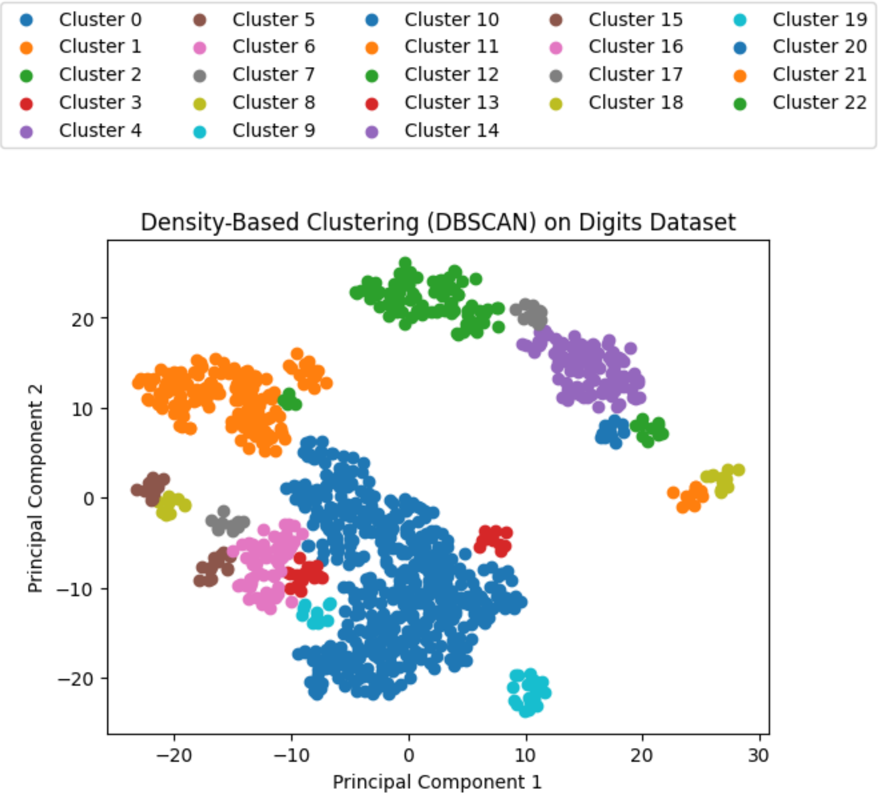
```
# Part 1: k-means clustering on the breast cancer dataset

# Import required libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

# Load the breast cancer dataset
data = load_breast_cancer().data

# Reduce the dimensions to 2 using PCA
pca = PCA(2)
df = pca.fit_transform(data)

# Create an object of KMeans with n_clusters = 10 and n_init = 10
kmeans = KMeans(n_clusters=10, n_init=10)

# Fit and predict the transformed dataset df
label = kmeans.fit_predict(df)

# Plot all the observations on a 2D coordinate system
for i in range(10):
    plt.scatter(df[label == i, 0], df[label == i, 1], label=f'Cluster {i}')

plt.legend()
plt.title("K-Means Clustering on Breast Cancer Dataset")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()
```

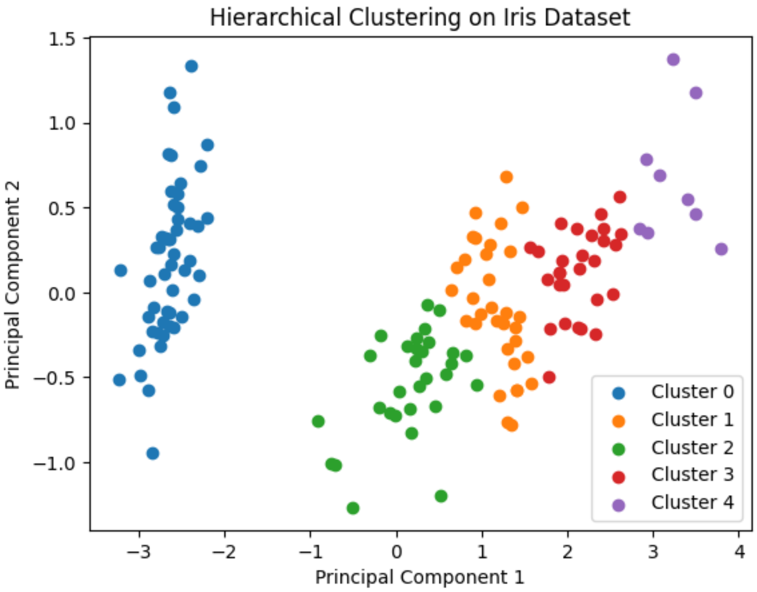


```
# Part 2: density-based clustering (DBSCAN) on the digits dataset

# Import required libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits
from sklearn.decomposition import PCA
from sklearn.cluster import DBSCAN

# Load the digits dataset
data = load_digits().data

# Reduce the dimensions to 2 using PCA
pca = PCA(2)
df = pca.fit_transform(data)

# Create an object of DBSCAN with min_samples = 10 and eps = 1.5
dbscan = DBSCAN(min_samples=10, eps=1.5)
```

```
# Fit and predict the transformed dataset df
label = dbscan.fit_predict(df)

# Plot all the observations on a 2D coordinate system
unique_labels = np.unique(label)
for lbl in unique_labels:
    if lbl != -1:  # Exclude outliers (-1 label)
        plt.scatter(df[label == lbl, 0], df[label == lbl, 1], label=f'Cluster {lbl}')

plt.legend(loc='upper center', bbox_to_anchor=(0.5, 1.5), ncol=5)
plt.title("Density-Based Clustering (DBSCAN) on Digits Dataset")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()
```

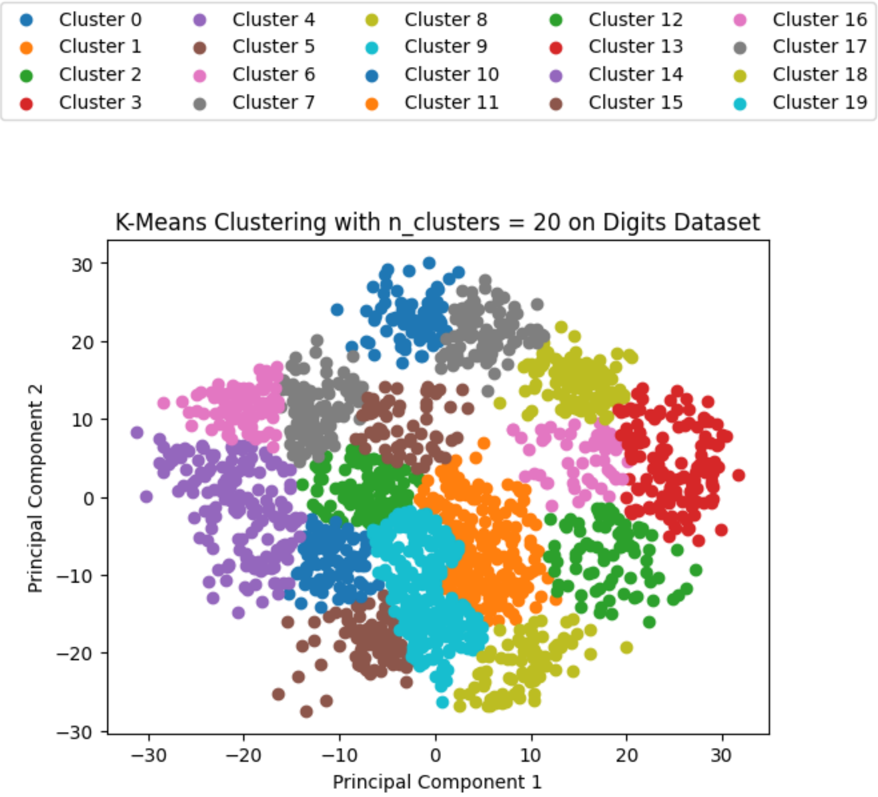


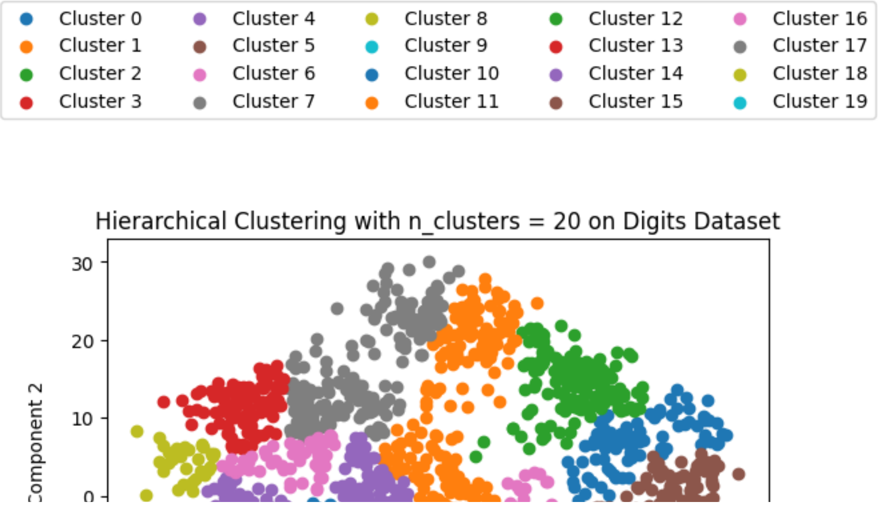
```
# Part 3: hierarchical clustering (Agglomerative Clustering) on the iris dataset

# Import required libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
from sklearn.cluster import AgglomerativeClustering

# Load the iris dataset
data = load_iris().data

# Reduce the dimensions to 2 using PCA
pca = PCA(2)
df = pca.fit_transform(data)

# Create an object of AgglomerativeClustering with n_clusters = 5
agg_clustering = AgglomerativeClustering(n_clusters=5)

# Fit and predict the transformed dataset df
label = agg_clustering.fit_predict(df)

# Plot all the observations on a 2D coordinate system
for i in range(5):
    plt.scatter(df[label == i, 0], df[label == i, 1], label=f'Cluster {i}')

plt.legend()
plt.title("Hierarchical Clustering on Iris Dataset")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()
```



```
# Part 4: K-means clustering with n_clusters = 20 on digits dataset

# Import required libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans, AgglomerativeClustering

# Load the digits dataset
data = load_digits().data

# Reduce the dimensions to 2 using PCA
pca = PCA(2)
df = pca.fit_transform(data)

# Create an object of KMeans with n_clusters = 20 and n_init = 10
kmeans = KMeans(n_clusters=20, n_init=10)

# Fit and predict the transformed dataset df
label = kmeans.fit_predict(df)

# Plot all the observations on a 2D coordinate system
for i in range(20):
    plt.scatter(df[label == i, 0], df[label == i, 1], label=f'Cluster {i}')

plt.legend(loc='upper center', bbox_to_anchor=(0.5, 1.5), ncol=5)
plt.title("K-Means Clustering with n_clusters = 20 on Digits Dataset")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()

# Part 4: Hierarchical clustering with n_clusters = 20 on digits dataset

# Load the digits dataset
data = load_digits().data

# Reduce the dimensions to 2 using PCA
pca = PCA(2)
df = pca.fit_transform(data)

# Create an object of AgglomerativeClustering with n_clusters = 20
agg_clustering = AgglomerativeClustering(n_clusters=20)

# Fit and predict the transformed dataset df
label = agg_clustering.fit_predict(df)

# Plot all the observations on a 2D coordinate system
```

```
for i in range(20):
    plt.scatter(df[label == i, 0], df[label == i, 1], label=f'Cluster {i}')

plt.legend(loc='upper center', bbox_to_anchor=(0.5, 1.5), ncol=5)
plt.title("Hierarchical Clustering with n_clusters = 20 on Digits Dataset")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()
```





✓ 3s completed at 12:53 PM ● ×