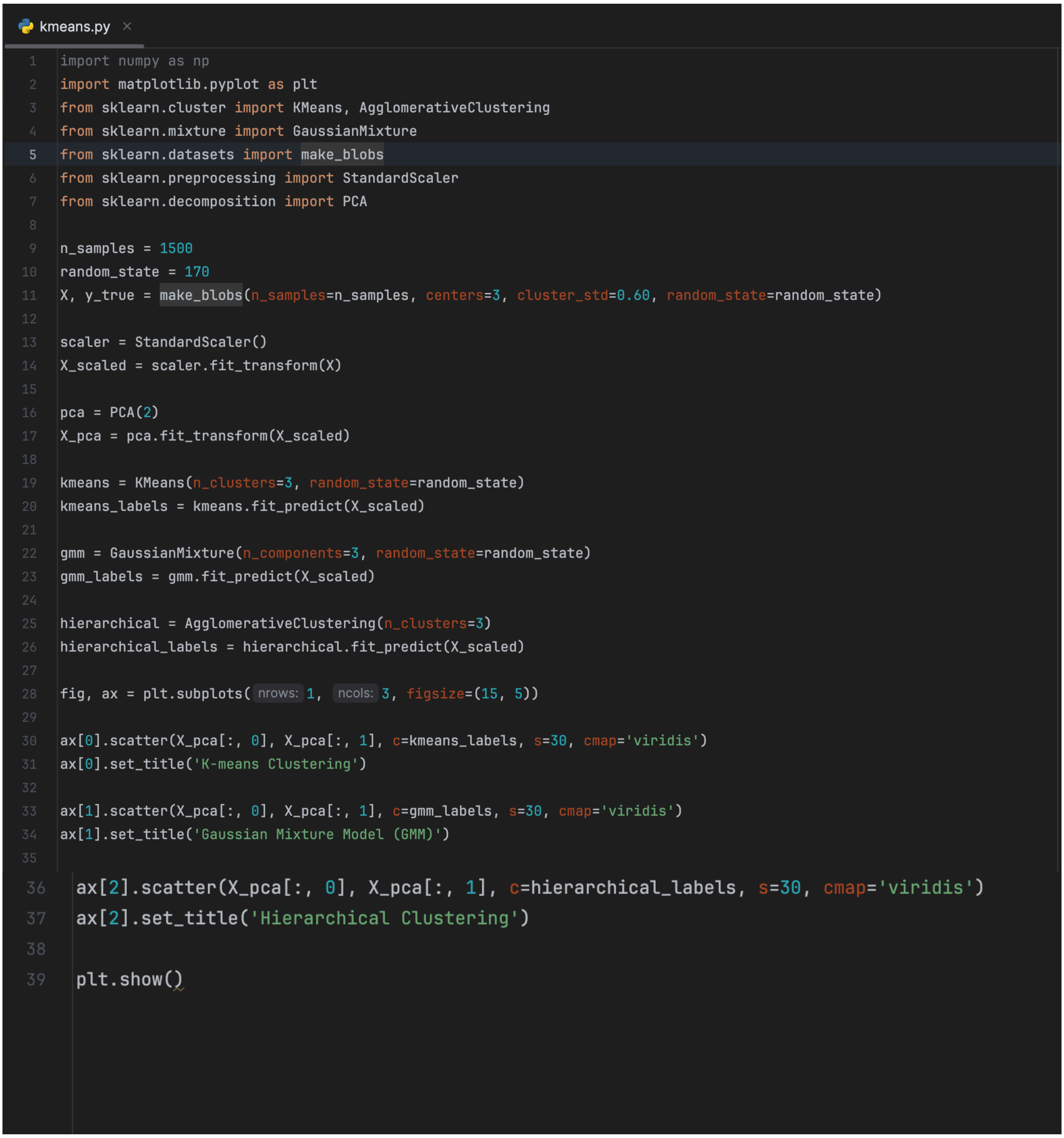
Experiment - 5

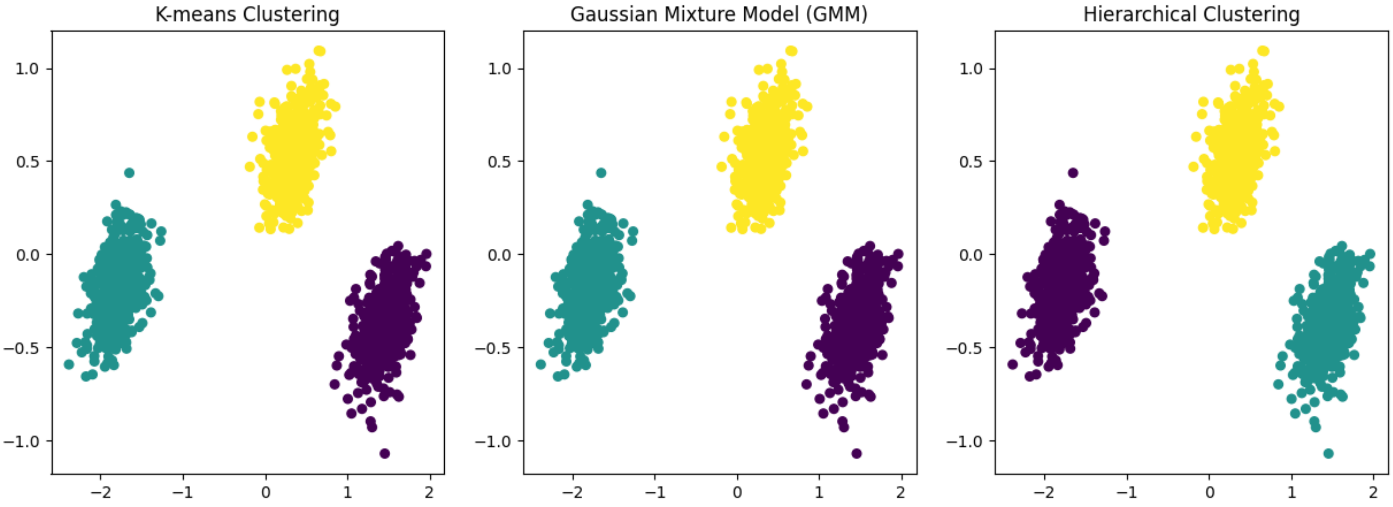
Implement K-means clustering, mixtures of Gaussians, and hierarchial clustering algorithm to categorise data.

<u>CODE</u> :-



```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.mixture import GaussianMixture
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

n_samples = 1500
random_state = 170
X, y_true = make_blobs(n_samples=n_samples, centers=3, cluster_std=0.60, random_state=random_state)

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

pca = PCA(2)
X_pca = pca.fit_transform(X_scaled)

kmeans = KMeans(n_clusters=3, random_state=random_state)
kmeans_labels = kmeans.fit_predict(X_scaled)

gmm = GaussianMixture(n_components=3, random_state=random_state)
gmm_labels = gmm.fit_predict(X_scaled)

hierarchical = AgglomerativeClustering(n_clusters=3)
hierarchical_labels = hierarchical.fit_predict(X_scaled)

fig, ax = plt.subplots( nrows: 1,  ncols: 3, figsize=(15, 5))

ax[0].scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans_labels, s=30, cmap='viridis')
ax[0].set_title('K-means Clustering')

ax[1].scatter(X_pca[:, 0], X_pca[:, 1], c=gmm_labels, s=30, cmap='viridis')
ax[1].set_title('Gaussian Mixture Model (GMM)')

ax[2].scatter(X_pca[:, 0], X_pca[:, 1], c=hierarchical_labels, s=30, cmap='viridis')
ax[2].set_title('Hierarchical Clustering')

plt.show()
```

OUTPUT :-


K-means Clustering
Gaussian Mixture Model (GMM)
Hierarchical Clustering
1.0
0.5
0.0
−0.5
−1.0
−2
−1
0
1
2