MACHINE LEARNING - Lab Assignment 7A

Title: K-Means Clustering with scikit-learn.

By: Madhumithaa RP | 20BCE1648

Concept:

Clustering (or cluster analysis) is a technique that allows us to find groups of similar objects, objects that are more related to each other than to objects in other groups. Examples of business-oriented applications of clustering include the grouping of documents, music, and movies by different topics, or finding customers that share similar interests based on common purchase behaviors as a basis for recommendation engines.

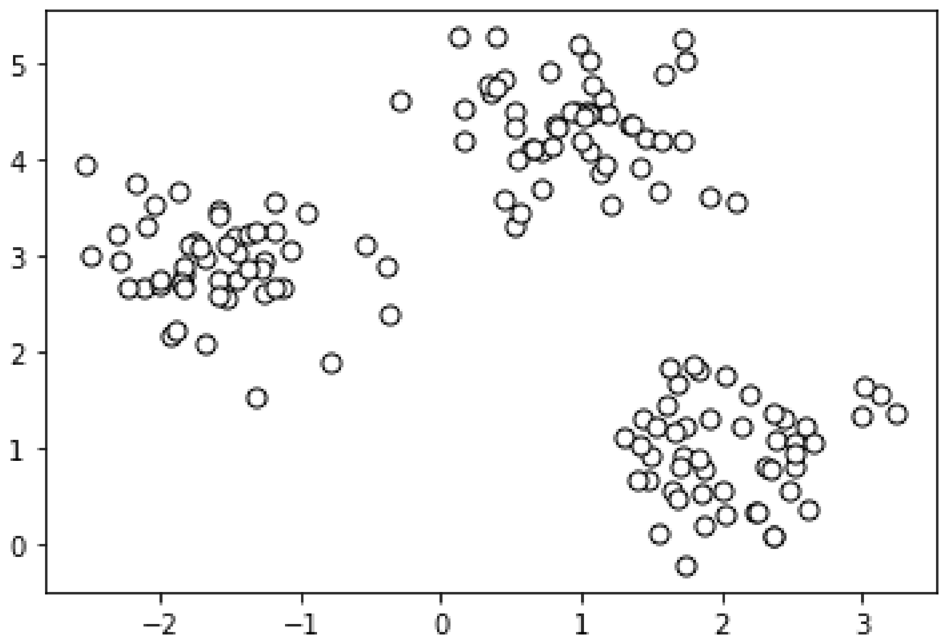
Step 1: Importing Libraries:

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

Step 2: Create the dataset and visualize the same

```
X, y = make_blobs(
   n_samples=150, n_features=2,
   centers=3, cluster_std=0.5,
   shuffle=True, random_state=0
)

# plot
plt.scatter(
   X[:, 0], X[:, 1],
   c='white', marker='o',
   edgecolor='black', s=50
)
plt.show()
```



ALGORITHM:

1. Randomly pick k centroids from the sample points as initial cluster centers.
2. Assign each sample to the nearest centroid μ^(j), j ∈ {1, ..., k}.
3. Move the centroids to the center of the samples that were assigned to it.
4. Repeat steps 2 and 3 until the cluster assignments do not change or a user-defined tolerance or maximum number of iterations is reached.

Squared Eucidean Distance: d(x,y)2 - sum(xi - yj)2

Sum of Squared Errors: sum sum w(i,j)||x(i) - m(j)||2

where, m(j) is the centroid for cluster j

w(i,j) = 1 if the sample x(i) is in cluster j, = 0 otherwise.

Step 3: Train the KMeans module

```
km = KMeans(
    n_clusters=3, init='random',
    n_init=10, max_iter=300,
    random_state=0
)
y_km = km.fit_predict(X)
```

k = 3 -> No of clusters.

n_init = 10 -> No of iterations.

max_iter = 300 -> maximum iterations to try finding the best cluster center

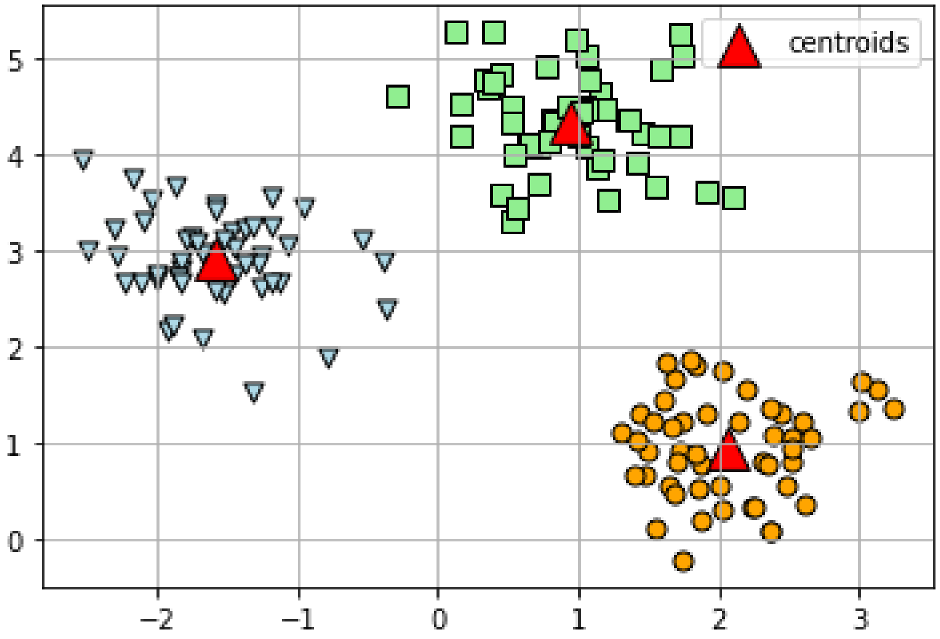
Step 4: Visualize the output of the algorithm

```
# plot the 3 clusters
plt.scatter(
    X[y_km == 0, 0], X[y_km == 0, 1],
    s=50, c='lightgreen',
    marker='s', edgecolor='black',
    #label='cluster 1'
)

plt.scatter(
    X[y_km == 1, 0], X[y_km == 1, 1],
    s=50, c='orange',
    marker='o', edgecolor='black',
    #label='cluster 2'
)

plt.scatter(
    X[y_km == 2, 0], X[y_km == 2, 1],
    s=50, c='lightblue',
    marker='v', edgecolor='black',
    #label='cluster 3'
)

# plot the centroids
plt.scatter(
    km.cluster_centers_[:, 0], km.cluster_centers_[:, 1],
    s=250, marker='^',
    c='red', edgecolor='black',
    label='centroids'
)
plt.legend(scatterpoints=1)
plt.grid()
plt.show()
```



Thank You.