Devanshu Surana
PC-23, 1032210755
Panel C, Batch C1

ML Lab Assignment 6

FAQ's

1) What is K-means clustering and how does it work?

→ K-means clustering is a popular unsupervised machine learning algorithm used for partioning a dataset into K distinct, non-overlapping clusters. The goal of k-me-ans is to group similar data point together and discover underlying patterns or structures in the data.

1. Choose K: Decide the no. of clusters.
2. Initialize: Randomly pick K data points from dataset as initial centroids.
3. Assign: For each data point, find the nearest centroid and assign it to that clusters.
4. Update: Recalculate the centroid for each cluster by taking the mean of all data points assigned to it.
5. Iterate: Repeat the Assign and update steps until centroids no longer change much.
6. Final Clusters: Once centroids stabilizes the clusters are formed with data points assigned to the closest centroid.

Sundaram
FOR EDUCATIONAL USE

2) What is spectral clustering and how does it work?

→ Spectral clustering is a clustering algorithm that uses the eigen vectors (spectrum) of a similarity matrix to cluster data points. It's possibly particularly useful for clustering data that is not necessarily spherical or equally sized, which are assumptions made by K-means.

Working:

1. Similarity matrix: Create a similarity matrix based on pairwise similarities betⁿ data points.
2. Graph Construction: Interpret the similarity matrix as the adjacency matrix of a graph.
3. Graph Laplacian: Compute the laplacian matrix from the graph.
4. Eigenvalue Decomposition: Find the eigen vectors corresponding to the smallest eigen values of the laplacian matrix.
5. Dimensionality Reduction: Use these eigen vectors to reduce the dimensionality of the data.
6. Clustering: Apply standard clustering algorithms like kmean in reduced-dimensional space to partition the data into clusters.

Sundaram

FOR EDUCATIONAL USE

Page No.
Date

3) What is DBSCAN clustering and how does it work?

→ DBSCAN (Density Based Spatial Clustering of Application with noise) is a density based clustering algorithm that can identify clusters of varying shapes and sizes in a dataset. Unlike K-means or spectral clustering, DBSCAN ~~clean~~ doesn't require specifying the no. of clusters beforehead and can find ~~algorithm~~ arbitrarily shaped classes.

1. Parameter Selection: Choose 2 params: epsilon (ε), the radius within which to search for neighbouring points and minpts, the minimum no. of points required to form a dense region.

2. Core point Identification: For each data point, count the no. of pts within ε distance. If a point has atleast MinPts neighbours, it's marked as core point.

3. Density Connected Points: Form clusters by connecting core points to their density-reachable neighbours. Points within ε distance of a core point are considered density-connected.

4. Cluster formation: Assign each core point and its density-reachable neighbors to the same cluster. Points that are not core points or part of any cluster are marked as noise.

5. Output: Return the clusters formed and the noise points identified.

4) How do you choose the optimal number of clusters in K-means clustering?

→ Cross-Validation:

1. Split the dataset into training and validation sets
2. Fit K-means with different values of K on the training set and evaluate performance on the validation set using a metric like WCSS.

Hierarchical Clustering Dendogram:

1. Perform hierarchical clustering and visualise the resulting dendogram.
2. Look for a level in the dendogram where clusters merge, indicating the optimal number of clusters.

5) Can DBSCAN handle datasets with different densities?

→ Yes, DBSCAN clustering is well suited for datasets with different densities. It can adapt to varying densities within the data by adjusting it params: epsilon (ε), and MinPts.

In Regions of high density, where there are many neighboring points within the specified epsilon radius, DBSCAN will identify these points are core points and form clusters around them.

In regions of low density, where there are fewer neighboring points, DBSCAN will not form clusters but instead mark these points as noise.

Assignment - 6 K Means Clustering

In [31]:

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_wine
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
```

In [32]:

```
wine = load_wine()
wine_data = pd.DataFrame(wine.data, columns=wine.feature_names)
```

In [33]:

```
print(wine_data)
```

```
     alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
0      14.23        1.71  2.43               15.6      127.0           2.80
1      13.20        1.78  2.14               11.2      100.0           2.65
2      13.16        2.36  2.67               18.6      101.0           2.80
3      14.37        1.95  2.50               16.8      113.0           3.85
4      13.24        2.59  2.87               21.0      118.0           2.80
..       ...         ...   ...                ...        ...            ...
173    13.71        5.65  2.45               20.5       95.0           1.68
174    13.40        3.91  2.48               23.0      102.0           1.80
175    13.27        4.28  2.26               20.0      120.0           1.59
176    13.17        2.59  2.37               20.0      120.0           1.65
177    14.13        4.10  2.74               24.5       96.0           2.05

     flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
0          3.06                  0.28             2.29             5.64  1.04
1          2.76                  0.26             1.28             4.38  1.05
2          3.24                  0.30             2.81             5.68  1.03
3          3.49                  0.24             2.18             7.80  0.86
4          2.69                  0.39             1.82             4.32  1.04
..          ...                   ...              ...              ...   ...
173        0.61                  0.52             1.06             7.70  0.64
174        0.75                  0.43             1.41             7.30  0.70
175        0.69                  0.43             1.35            10.20  0.59
176        0.68                  0.53             1.46             9.30  0.60
177        0.76                  0.56             1.35             9.20  0.61

     od280/od315_of_diluted_wines  proline
0                            3.92   1065.0
1                            3.40   1050.0
2                            3.17   1185.0
3                            3.45   1480.0
4                            2.93    735.0
..                            ...      ...
173                          1.74    740.0
174                          1.56    750.0
175                          1.56    835.0
176                          1.62    840.0
177                          1.60    560.0

[178 rows x 13 columns]
```

In [34]:

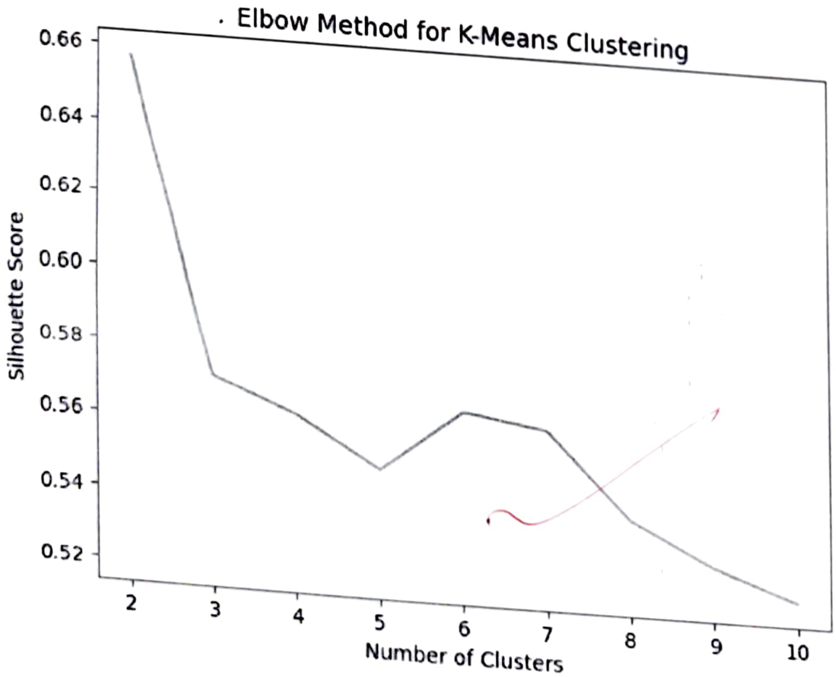
```
from sklearn.metrics import silhouette_score

silhouette_scores = []
for k in range(2, 11):
        kmeans = KMeans(n_clusters=k)
        kmeans.fit(wine_data)
        silhouette_scores.append(silhouette_score(wine_data, kmeans.labels_))
```

```
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  super()._check_params_vs_input(X, default_n_init=10)
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  super()._check_params_vs_input(X, default_n_init=10)
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
```

```
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to supp
he warning
  super()._check_params_vs_input(X, default_n_init=10)
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The defaul
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress
he warning
  super()._check_params_vs_input(X, default_n_init=10)
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  super()._check_params_vs_input(X, default_n_init=10)
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  super()._check_params_vs_input(X, default_n_init=10)
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  super()._check_params_vs_input(X, default_n_init=10)
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  super()._check_params_vs_input(X, default_n_init=10)
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  super()._check_params_vs_input(X, default_n_init=10)
```

In [35]:

```
plt.plot(range(2, 11), silhouette_scores)
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Score")
plt.title("Elbow Method for K-Means Clustering")
plt.show()
```



In [36]:

```
# Perform K-Means clustering with 7 clusters
kmeans = KMeans(n_clusters=7, random_state=42)
kmeans.fit(wine_data)
```

```
l:\Prodigy Internship\.venv\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default
value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  super()._check_params_vs_input(X, default_n_init=10)
```

Out[36]:

```
▾            KMeans
KMeans(n_clusters=7, random_state=42)
```

In [37]:

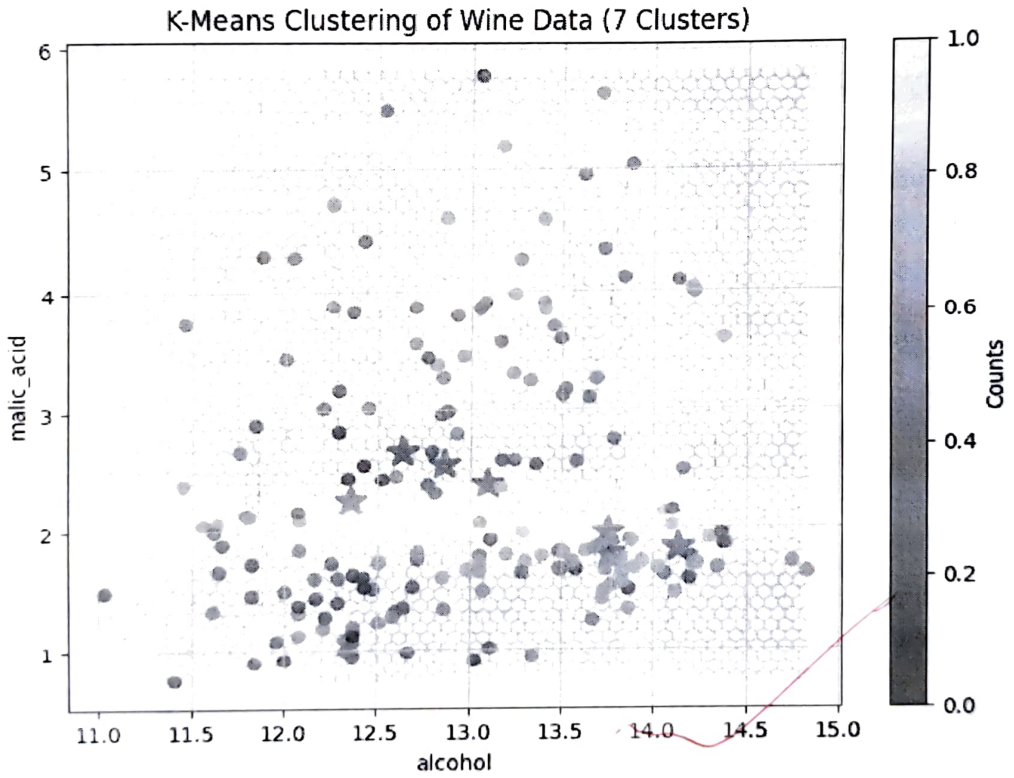
```
cluster_labels = kmeans.labels_
cluster_centers = kmeans.cluster_centers_
```

In [38]:

```
# Select the first two features for visualization (assuming informative)
features_to_plot = [0, 1]
```

In [41]:

```
plt.figure(figsize=(8, 6))
plt.hexbin(wine_data.iloc[:, features_to_plot[0]], wine_data.iloc[:, features_to_plot[1]],
           gridsize=50, cmap='plasma', alpha=0.8)

cluster_centers = kmeans.cluster_centers_
plt.scatter(cluster_centers[:, features_to_plot[0]], cluster_centers[:, features_to_plot[1]],
           marker='*', c='red', s=200, linewidths=2)

for cluster_label in range(7):
    cluster_data = wine_data[cluster_labels == cluster_label]
    plt.scatter(cluster_data.iloc[:, features_to_plot[0]], cluster_data.iloc[:, features_to_plot[1]],
                label=f'Cluster {cluster_label}', alpha=0.2)

plt.xlabel(wine.feature_names[features_to_plot[0]])
plt.ylabel(wine.feature_names[features_to_plot[1]])
plt.title('K-Means Clustering of Wine Data (7 Clusters)')
plt.colorbar(label='Counts')
plt.grid(True)
plt.show()
```



Lavkal
25/4/24

In [ ]: