LAB 6: Clustering

Clustering is an unsupervised machine learning technique that involves grouping a set of data points into clusters based on their similarity. The goal is to ensure that points within the same cluster are more similar to each other than to those in other clusters. Clustering is widely used in pattern recognition, data mining, and image analysis.

Types of Clustering

1. **Agglomerative Clustering**
    - Agglomerative clustering is a type of hierarchical clustering that builds clusters from the bottom up. Initially, each data point is considered as a separate cluster, and then pairs of clusters are merged iteratively based on some similarity criterion until all points are grouped into a single large cluster or the desired number of clusters is reached.
    - It follows a **"bottom-up" approach** and uses methods like "single-linkage," "complete-linkage," or "average-linkage" to decide which clusters to merge.

2. **Hierarchical Clustering**
    - Hierarchical clustering builds a hierarchy of clusters by either:
        - Agglomerative (bottom-up): Starting with individual data points and merging them into larger clusters, or
        - Divisive (top-down): Starting with one large cluster and dividing it into smaller clusters.
    - The result is often visualized using a dendrogram, which shows the hierarchy of clusters and allows the user to choose the level at which to cut the tree to form clusters.

3. **K-Means Clustering**
    - K-means clustering is a centroid-based algorithm that divides data into **K** clusters. The algorithm starts by randomly selecting K points as centroids and assigning each data point to the nearest centroid.

    - Then, it recalculates the centroid of each cluster based on the current members and reassigns points until the clusters stabilize.
    - K-means is efficient and widely used, but it requires the user to specify the number of clusters **K** in advance and may not perform well with irregularly shaped clusters.

About the dataset:

Tasks:

You will be provided with a python notebook in which you are required to complete the functions under the “Only Edit this” section. In the next section, change the dataset file path according to the path on your machine.

Function Descriptions:

Load_data:

- Load data from csv using `\t` as the separator.
- Drop irrelevant columns
- Label encode the Non numerical columns
- Select the main features and make an array of values (an array of arrays) and form a features vector.
    - A random code example:

        x = data_main[['Education', 'Kidhome', 'Teenhome' …..
        'Z_CostContact', 'Z_Revenue']].values
- Scalarization of the features
- Return `scaled_x` variable which is a variable containing the scaled values of x

Apply_pca:

- Due to the size of the dataset you are required to employ PCA for dimensionality reduction.
- For ideal results, use n_components as 2. Feel free to experiment with this value but n_components = 2 is an ideal case.
- Return `x_pca` variable which is the set of reduced features.

Find_optimal_clusters:

- This is to find the optimal number of clusters using the elbow method.
- Find the values of inertia till max_clusters+1 and append the inertia values of each run to an array. Then, plot the graph to find the breakpoints (Incase of

multiple breakpoints, consider the latest breakpoint). This gives the number of clusters (n_clusters).
- You are also requested to write the code to plot the inertia vs n_cluster graph here.
- Returns the inertia array.

Perform_kmeans_clustering:
- Used to perform k-means clustering.
- Some algorithm parameters:
    - init = 'k-means++'
    - max_iter = 300
    - n_init = 10
    - n_clusters as obtained from find_optimal_clusters function.
- Random state value is your choice but maintain the same standard throughout the code.
- Returns the k-means fit predict values.

Perform Agglomerative clustering:
- Used to perform Agglomerative clustering.
- Some Algorithm parameters:
    - Linkage method is "ward".
    - n_clusters as obtained from find_optimal_clusters function.

Get_linkages:
- This is for the dendrogram graph under Hierarchical Clustering.
- It is used to form the linkages between 'x' features using the 'ward' linkage method.
- Return a single variable, containing the linkages that are obtained from the linkage method from scipy.cluster.heirarchy.

Plot_dendrogram:
- This is for the dendrogram graph under Hierarchical Clustering.
- It is used to plot the dendrogram.
- Uses the linkages obtained from get_linkages method to plot the graph of index vs distance.

Important Points to note:

- You won't be able to resubmit, so kindly check your files. Remove all print statements. Resubmitting from different mail will lead to zero marks. Failing some hidden cases will lead to partial marks. Test cases provided to you are for reference only, hidden test cases will be similar.
- There may also be some commented lines of code in the notebook. You can use these to find the scores of your model.
- Do not make changes to the function definitions that are provided to you. As specified earlier, you will be provided with a notebook containing two sections and please follow the guidelines. We will be reading your scripts in detail and any breach of guidelines will impact your results for this lab.
- You must use PyTorch only.
- You may write additional helper functions.
- You can use any built-in functions (if needed)

Submission Guidelines:

You are expected to submit two files:

- The python notebook. Naming format: **<Campus>_<Section>_<SRN>_Lab6.ipynb**
- A PDF File containing the screenshots of the **Inertia value plot, Dendrogram graph, Testcase1 passed, Testcase2 passed**. These test cases will pass once you complete the K-Means Clustering and Agglomerative Clustering related functions.