ASSIGNMENT 2 : MACHINE LEARNING

Task 1(A).

https://colab.research.google.com/drive/1MLHnyY-fqhHyzvecO1izBTSALp9ql7aF?usp=sharing

1. The aim is to apply K-means clustering on the MNIST handwritten digit dataset using cosine similariy instead of the traditional Euclidean distance metric.

2. The code fetches the MNIST dataset from OpenML. It comprises of grayscale images of handwritten digits (28x28 pixels). Each image is reshaped into a 1D vector of length 784, representing the pixel values.The pixel values are normalized to a range between 0 and 1. A subset of 50,000 images is used for clustering.

3. Instead of the conventional Euclidean distance, Cosine Similarity is used which is particularly useful while processing high dimensional data. It measures the cosine of the angle between two data vectors.

4. K-Means is performed with 10, 7, 4 clusters. For each cluster configuration, the following steps are taken:

- Initialize cluster centroids randomly.
- Datta points are assigned to the nearest cluster based on Cosine Similarity.
- Then the cluster centroids are updated as per how far the cluster is with respect to the centroid.
- This process is repeated and steps are updated until convergence

5. Post clustering, a sample of images from each cluster is visualized to inspect the kind of digits grouped into each cluster. For K=4, broad patterns/shape are represented in each cluster where as with increasing K value, the clusters are formed more distinctly.

Task 1(c).

In K-Means clustering utilizing Cosine Similarity:

1. When segmented into 10 clusters, each cluster corresponds to a unique digit from 0 through 9. Owing to their high specificity, these clusters exhibit minimal variations within themselves, offering a detailed differentiation of each digit.

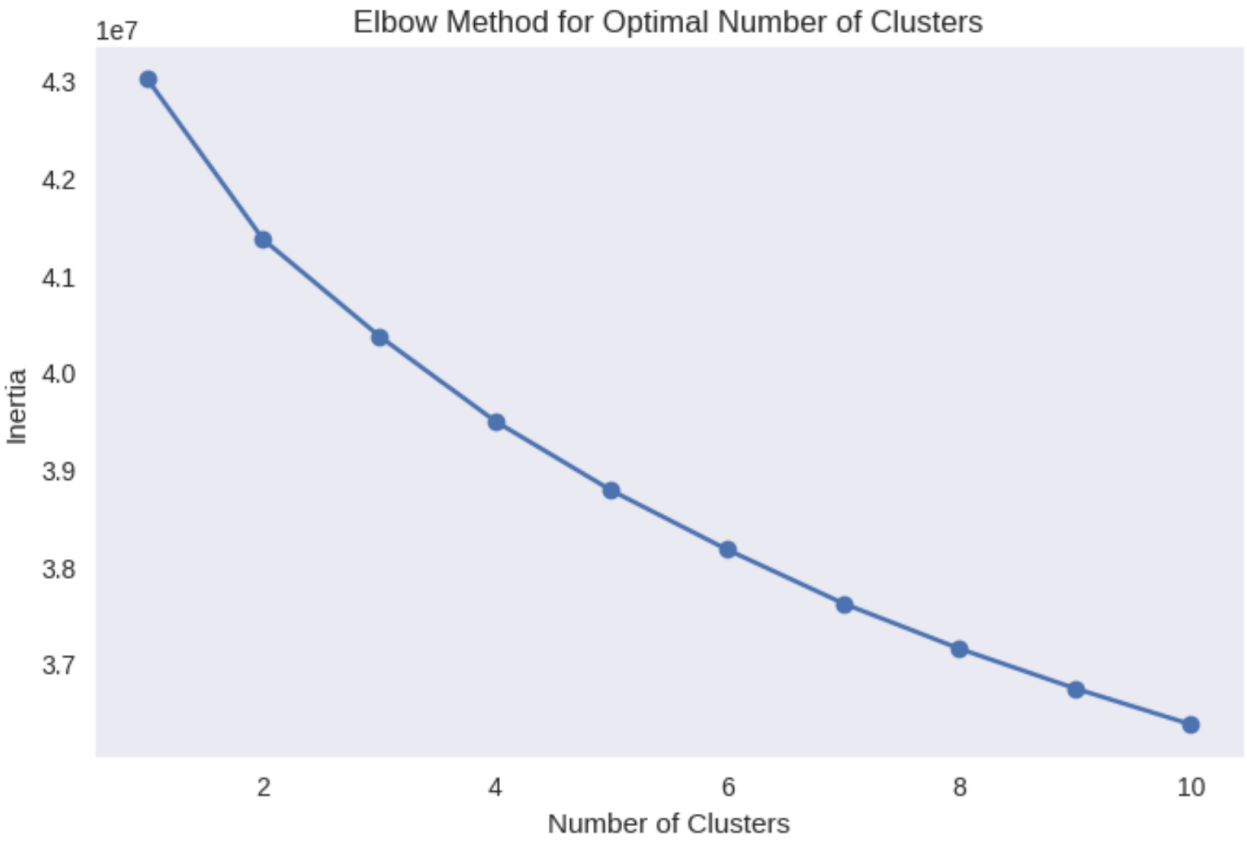
2. At a reduced cluster count of 7, certain digits with resemblances might be grouped under a single cluster. For instance, due to their shape similarities, 6 and 9 could land in one cluster. This leads to some clusters having a mix of several digit types, and the distinctions within each cluster get somewhat blurred, making them broader in representation.

3. As the cluster count shrinks further to 4, these groupings become even more generalized, encompassing multiple digit classes within single clusters. This amalgamation makes it harder to pinpoint individual digits within these groups, as they become more of an overarching depiction of the data.

4. In summary, the granularity of the clusters in K-Means is directly influenced by the chosen cluster count. More clusters lead to finer, distinct classifications, while fewer clusters result in more generalized groupings. To determine an optimal cluster count, techniques like the elbow method can be instrumental.

Task 1(D)

1. To find the optimal number of clusters for a dataset, a common method called "elbow method" can be used. This method involves running the K-Means algorithm for a range of cluster numbers and measuring the sum of squared distances (inertia) between data points and their respective cluster centers. The point where the inertia starts to level off (resembling an "elbow" in the plot) indicates the optimal number of clusters. Here's a Python function to find the optimal number of clusters using the elbow



Task 2(a).

https://colab.research.google.com/drive/1lO4OlBJtcrTMW1XF5j_6ew_sRp0dS8rD?usp=sharing

1. Principal Component Analysis (PCA) is used to decrease the dimensions of the original MNIST dataset through several stages:

a. Determine the mean to center the data.
b. Adjust the data by deducting the mean.
c. Compute the dataset's covariance matrix.
d. Use Singular Value Decomposition (SVD) to extract eigenvectors.
e. Choose the top n components for dimensionality reduction.
f. Map the data onto these principal eigenvectors.

2. Subsequently, clustering on the PCA-altered dataset is conducted using the Gaussian Mixture Model (GMM). Analysis is executed with clusters of 10, 7, and 4 to investigate various clustering possibilities. The standard procedure for each scenario is:
   a. Apply the GMM model for the given cluster count.
   b. Assign cluster labels based on data points.

3. When using 32 PCA components alongside 10 GMM clusters, the clusters tend to be detailed, grouping digits of similar appearance. However, there's some inter-cluster overlap, potentially because of reduced information due to limited components.

4. Opting for 64 PCA components with 7 GMM clusters leads to clearer cluster differentiation. The enhancement in PCA components translates to well-defined, less overlapping clusters. Digits with mutual characteristics are typically grouped.

5. With 128 PCA components and 4 GMM clusters, clustering appears broader. Digits are classified in more generalized clusters. Although some clusters have diminished separations, this solution provides consistent outcomes with a smaller number of clusters.

Task 2(c).

1. In the K-means clustering approach, the orientation or direction of data points plays a significant role in shaping the clusters. This is particularly evident when employing Cosine Similarity, resulting in cohesive groupings. Understanding cluster centers can be intricate due to the inherent uncertainty of the similarity measure.

2. On the other hand, during the PCA and GMM clustering activity, the groupings are molded by the statistical distribution of data points within the compressed feature space. Consequently, clusters might either be dispersed or more generalized based on

the selected components and cluster numbers. GMM's probabilistic aspect provides added adaptability in capturing intricate data distributions.