ML_HW5

Mohammad Javad Pesarakloo
810100103

July 17, 2024

Question 1

A

We denote mean of class one with m_1 and mean of class two with m_2.First we need to find m_1 and m_2:

$$m_1 = \begin{pmatrix} 3 \\ 6 \end{pmatrix}, m_2 = \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}$$

Now we subtract means from their corresponding samples:

$$X_1 - m_1 = \{\begin{pmatrix} 1 \\ -2.6 \end{pmatrix}, \begin{pmatrix} -1 \\ 0.4 \end{pmatrix}, \begin{pmatrix} -1 \\ -0.6 \end{pmatrix}, \begin{pmatrix} 0 \\ 2.4 \end{pmatrix}, \begin{pmatrix} 1 \\ 0.4 \end{pmatrix}\}$$

$$X_2 - m_2 = \{\begin{pmatrix} 0.6 \\ 2.4 \end{pmatrix}, \begin{pmatrix} -2.4 \\ 0.4 \end{pmatrix}, \begin{pmatrix} 0.6 \\ -2.6 \end{pmatrix}, \begin{pmatrix} -0.4 \\ -0.6 \end{pmatrix}, \begin{pmatrix} 1.6 \\ 0.4 \end{pmatrix}\}$$

Now we can calculate S_1 and S_2:

$$S_1^2 = \begin{pmatrix} 1 & -2.6 \\ -2.6 & 6.76 \end{pmatrix} + \begin{pmatrix} 1 & -0.4 \\ -0.4 & 0.16 \end{pmatrix} + \begin{pmatrix} 1 & 0.6 \\ 0.6 & 0.36 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 5.76 \end{pmatrix} + \begin{pmatrix} 1 & 0.4 \\ 0.4 & 0.16 \end{pmatrix}$$

$$= \begin{pmatrix} 4 & -2 \\ -2 & 13.2 \end{pmatrix}$$

$$S_2^2 = \begin{pmatrix} 0.36 & 1.44 \\ 1.44 & 5.76 \end{pmatrix} + \begin{pmatrix} 5.76 & -0.96 \\ -0.96 & 0.16 \end{pmatrix} + \begin{pmatrix} 0.36 & -1.56 \\ -1.56 & 6.76 \end{pmatrix} + \begin{pmatrix} 0.16 & 0.24 \\ 0.24 & 0.36 \end{pmatrix} + \begin{pmatrix} 2.56 & 0.64 \\ 0.64 & 0.16 \end{pmatrix}$$

$$= \begin{pmatrix} 9.2 & -0.2 \\ -0.2 & 13.2 \end{pmatrix}$$

And now S_w can be calculated:

$$S_w = \begin{pmatrix} 4 & -2 \\ -2 & 13.2 \end{pmatrix} + \begin{pmatrix} 9.2 & -0.2 \\ -0.2 & 13.2 \end{pmatrix} = \begin{pmatrix} 13.2 & -2.2 \\ -2.2 & 26.4 \end{pmatrix}$$

B

$$m_1 - m_2 = \begin{pmatrix} 3 & 3.6 \end{pmatrix} - \begin{pmatrix} 8.4 & 7.6 \end{pmatrix} = \begin{pmatrix} -5.4 & -4 \end{pmatrix}$$

$$S_B = \begin{pmatrix} -5 & -4 \end{pmatrix} \begin{pmatrix} -5.4 \\ -4 \end{pmatrix} = \begin{pmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{pmatrix}$$

C

$$A = S_w^{-1} S_B$$

$$S_w^{-1} = \frac{1}{343.64} \begin{pmatrix} 26.4 & 2.2 \\ 2.2 & 13.2 \end{pmatrix} = \begin{pmatrix} 0.076 & 0.006 \\ 0.006 & 0.038 \end{pmatrix}$$

$$\Rightarrow A = \begin{pmatrix} 2.34 & 1.73 \\ 0.99 & 0.73 \end{pmatrix}$$

$$\Rightarrow A - \lambda I \begin{pmatrix} 2.34 - \lambda & 1.73 \\ 0.93 & 0.73 - \lambda \end{pmatrix}$$

$$|A - \lambda I| = (\lambda - 2.34) \times (\lambda - 0.73) = 0$$

$$\Rightarrow \lambda^2 - 3.07\lambda - 0.004 = 0$$

$$\lambda_1 = -0.001, \lambda_2 = 3.07$$

Thus the greatest eigen value is **3.07**.

Question 2

A

Model selection is the process of selecting among many candidate models for a modeling problem while model assessment is the process of estimating the model's prediction error (generalization error) over an independent new data sample.The reason for which we use model selection is that each model have multiple hyper parameters that can affect the performance of the model;in this process we find the best hyper parameters to get the best modeling result.Also this process can be accross different types of machine learning models such as SVM, Logistic Regression, etc.

B

In process of model selection, we need sufficient data while in most cases, we don't have this amount of data. Instead, there are two main classes of techniques to approximate the ideal case of model selection:

1. **Probabilistic Measures** : Choose a model via in-sample error and complexity.There are four major probabilistic measures:
   - Akaike Information Criterion (AIC)
   - Bayesian Information Criterion (BIC)
   - Minimum Description Length (MDL)
   - Structural Risk Minimization (SRM)
2. **Resampling Methods** : Choose a model via estimated out-of-sample error.In this mothod, we seek to estimate the performance of the model by splitting the training dataset into sub train and test sets.This process may be repeated multiple times and the mean performance accross each trail is reported.There exists three common resampling methods:
   - Random train-test splits.
   - Cross-Validation (k-fold, LOOCV, etc.)
   - Bootstrap

C

When number of data samples is not sufficient, it is known that training error is optimistically biased and therefore is not a good basis for choosing a model.The performance can be penalized based on how optimistic the training error is believed to be. This is typically achieved using algorithm-specific methods, often linear, that penalize the score based on the complexity of the model explained in the previous subsections.

Question 3

$$\theta = \{\lambda_1, \lambda_2, \alpha\}$$

First we define hidden variable z such that:

$$z_i = \begin{cases} 0 & x_i comes from the first component \\ 1 & otherwise \end{cases}$$

$$p(x_i, z_i|\theta) = (\alpha\lambda_1 e^{-\lambda_1 x_i})^{1-z_i}((1-\alpha)\lambda_2 e^{-\lambda_2 x_i})^{z_i}$$

$$\xrightarrow{\log} \log p(x_i, z_i|\theta) = (1-z_i)(\log\alpha + \log\lambda_1 - \lambda_1 x_i) + z_i(\log(1-\alpha) + \log\lambda_2 - \lambda_2 x_i)$$

$$\log\_complete\_likelihood = \sum_{i=1}^{n}(1-z_i)(\log\alpha+\log\lambda_1-\lambda_1 x_i)+z_i(\log(1-\alpha)+\log\lambda_2-\lambda_2 x_i)$$

$$Q(\theta,\theta^t) = E[log\_complete\_likelihood|\theta]$$

$$= \sum_{i=1}^{n}(1-E[z_i])(\log\alpha+\log\lambda_1-\lambda_1 x_i)+E[z_i](\log(1-\alpha)+\log\lambda_2-\lambda_2 x_i)$$

$$\Rightarrow E[z_i] = p(z_i=1|x_i,\theta^t) = \frac{p(x_i|z_i=1,\theta^t)p(z_i=1|\theta^t)}{p(x_i|\theta^t)}$$

$$= \frac{\lambda_2^t e^{-\lambda_2^t x_i}(1-\alpha^t)}{\alpha^t\lambda_1^t e^{-\lambda_1^t x_i}+(1-\alpha^t)\lambda_2^t e^{-\lambda_2^t x_i}}$$

$$= \gamma_i^t$$

Now that we have the expected likelihood, we can find the updating rule for parameters:

$$\frac{\partial Q}{\partial\alpha} = \sum_{i=1}^{n}(1-\gamma_i^t)\frac{1}{\alpha} - \frac{\gamma_i^t}{1-\alpha} = 0$$

$$\Rightarrow (1-\alpha)\sum_{i=1}^{n}(1-\gamma_i^t) = \alpha\sum_{i=1}^{n}\gamma_i^t$$

$$\Rightarrow \alpha = 1-\frac{\sum_{i=1}^{n}\gamma_i^t}{n}$$

$$\frac{\partial Q}{\partial\lambda_1} = \sum_{i=1}^{n}(1-\gamma_i^t)(\frac{1}{\lambda_1}-x_i) = 0$$

$$\Rightarrow \lambda_1 = \frac{n-\sum_{i=1}^{n}\gamma_i^t}{\sum_{i=1}^{n}(1-\gamma_i^t)x_i}$$

And with the same methodology, we have:

$$\lambda_2 = \frac{\sum_{i=1}^{n}\gamma_i^t}{\sum_{i=1}^{n}\gamma_i^t x_i}$$

Question 4

A

The output layer contains three parts, each of them responsible for one set of GMM parameters:mean, convariance and weights.

B

The activation function should be selected according to some features that each parameter have:

1. Means : Linear(no constraint exists)
2. Covariances : ReLU, Exponential or SoftPlus(to ensure positive values)
3. Weights : SoftMax(to ensure the weights sum up to 1 and are non-negative)

C

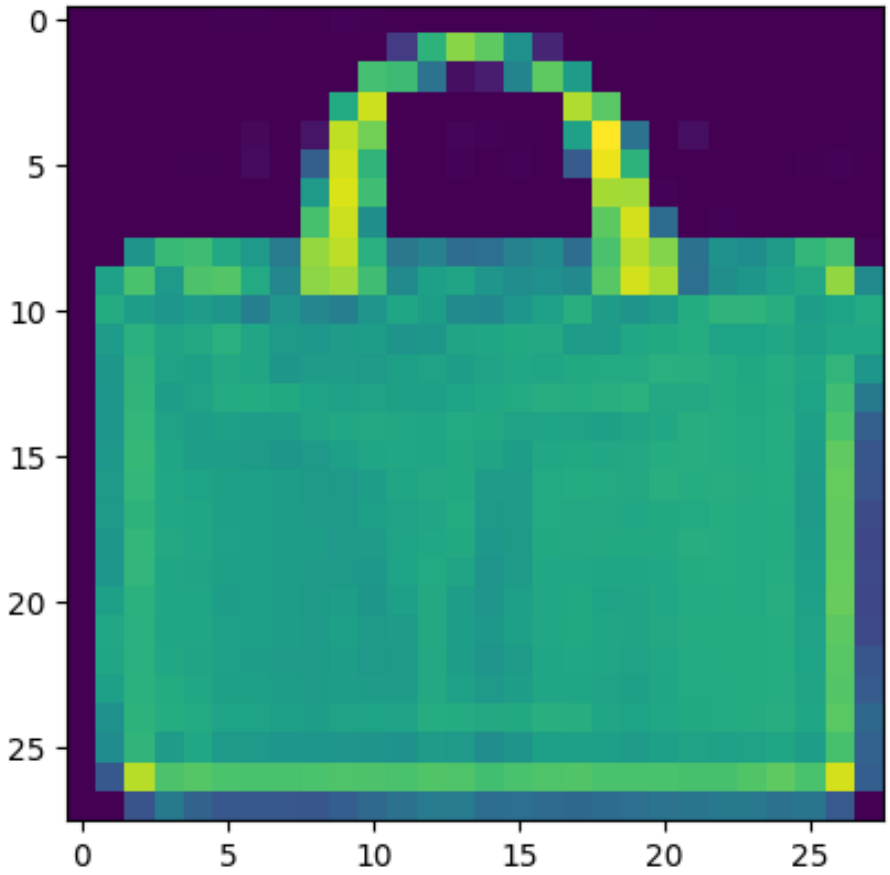
The loss function for training this neural network should be based on the log-likelihood of the data under the estimated GMM:

$$\log p(x) = \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

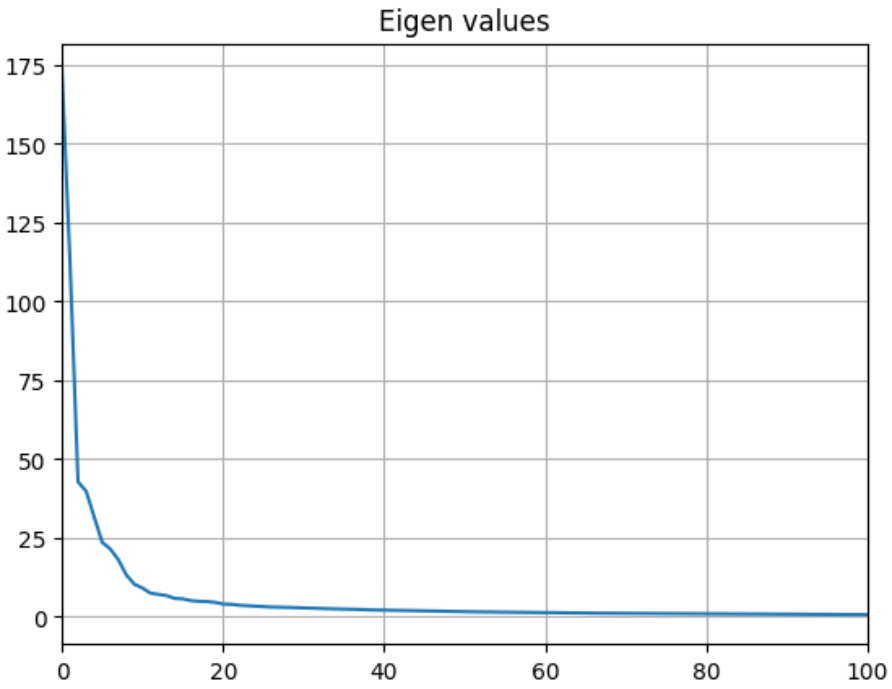
where π_k are the mixture weights and $\mathcal{N}(x|\mu_k, \Sigma_k)$ is the probability density function of a Gaussian distribution with mean μ_k and covariance Σ_k.

Question 5

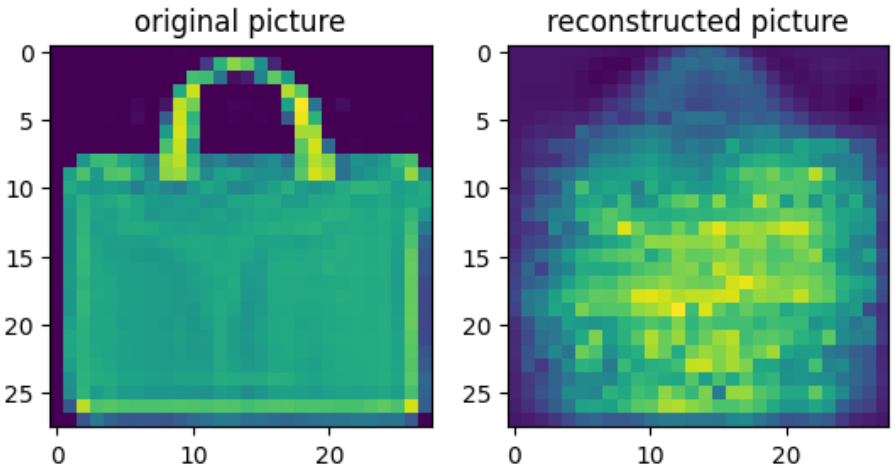
Codes related to this question are stored in Q5.ipynb notebook file.As suggested in the description, selecting a random index and visualizing its picture, we have:



Now in order to standardize the images to make them feedable to PCA models, we can use **StandardScalar**.The covariance matrix can also be calculated using the **np.cov** built-in function of numpy.In the next step, eigen values and their corresponding eigen vectors can also be calculated using the **np.linalg.eig** built-in function of numpy.Visualizing the eigen values, we get the following figure:

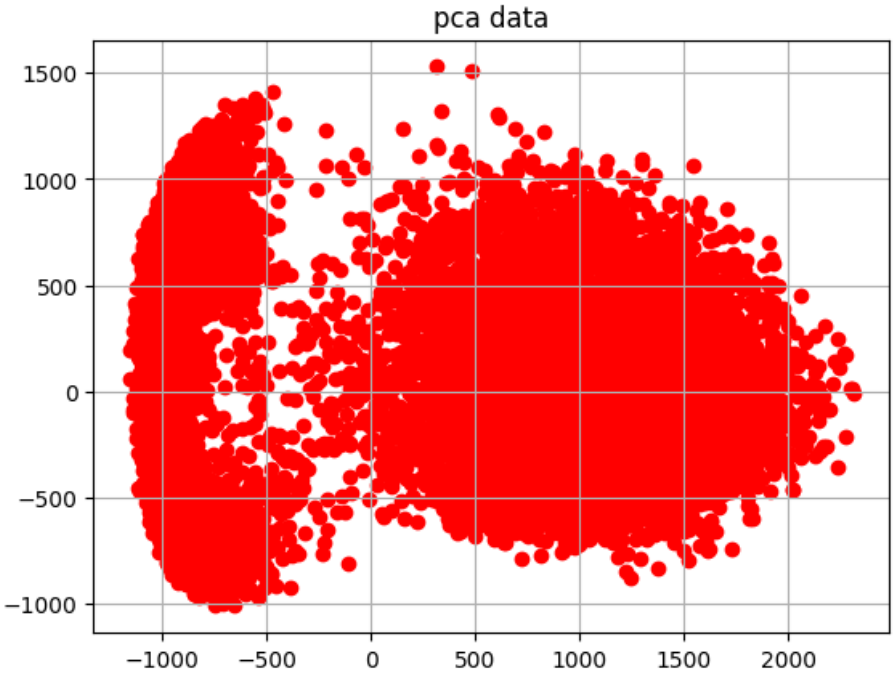


As we see in the above figure, most of the variance is held in the first 25 components;thus we can compress the pictures using these components. Then by reconstructing the picture and comparing it with the original one, we have the following figure:

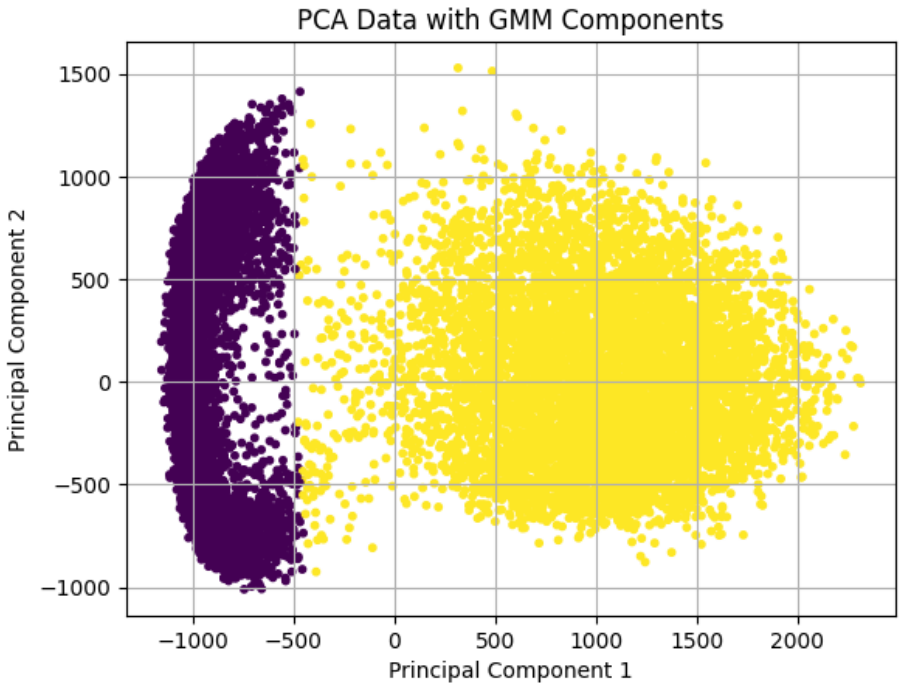


Question 6

Codes related to this question are stored in Q6.ipynb notebook file.At first, we can load data and extract the samples related to 0 and 1.Then we can concatenate these two classes and apply a principal component analysis on them to be able to visualze them;we have the following figure:

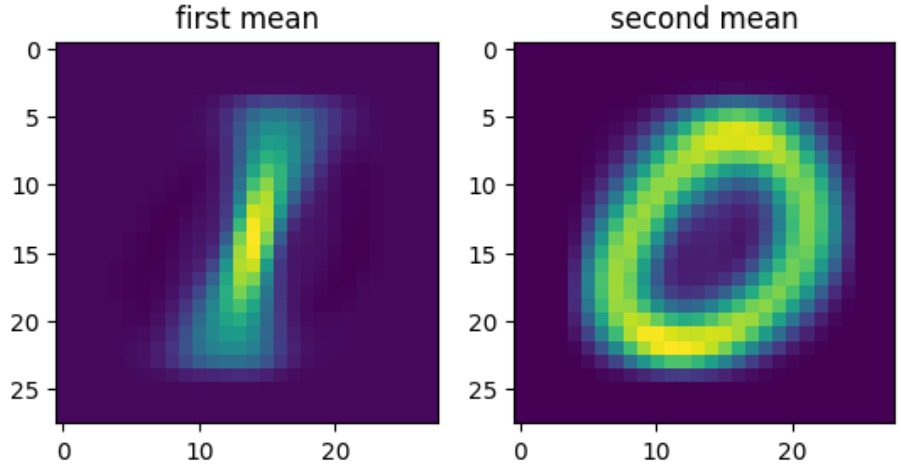


Now we can fit a gaussian mixture model with 2 components which gives us the following figure:

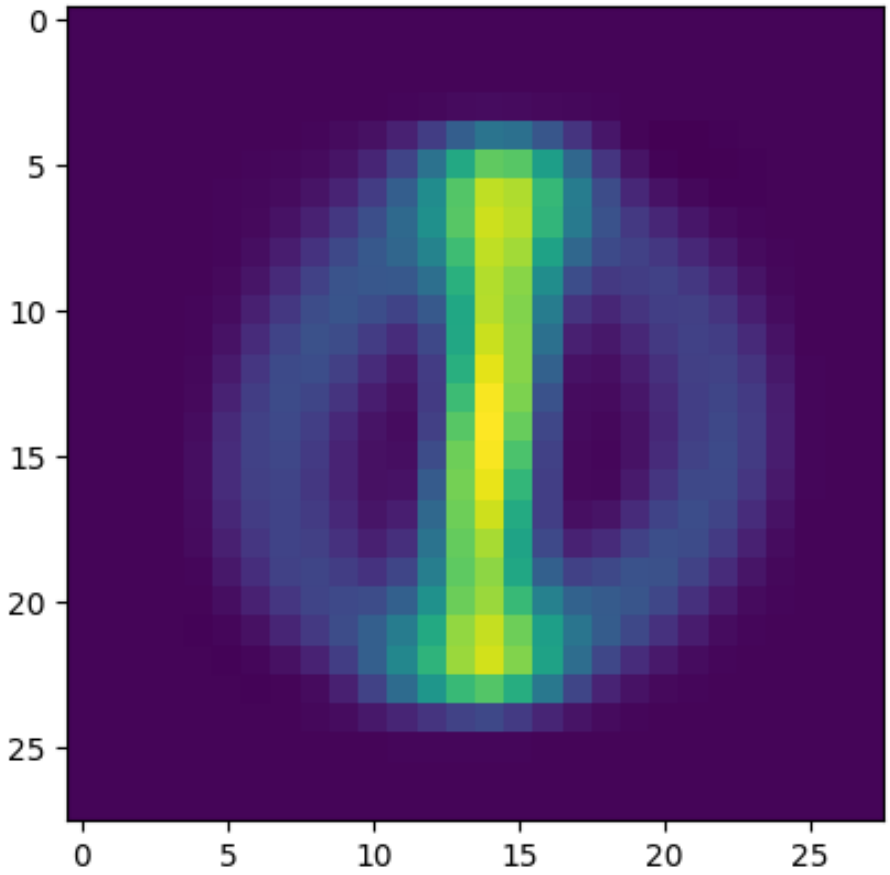


As asked in the description, difference between mean of two components is : **1915.67**.

Now inverse transforming the data to the original dimensions, we expect that means have high resemblance to components 0 and 1 which we see in the following figure:



Now if we find the samples whose probability of belonging to each component have low differnce, we expect that points to be similar to both classes.Doing this analysis, we get the following figure:



Now we are asked to analyze the differnce between means of components for all different pairs of classes.using the following code, we found the highest and the shortest differnce possbile:

```
max_distance = 0
min_distance = float('inf')
max_pair = None
min_pair = None

for class1, class2 in combinations(range(10), 2):
    data_class1 = X[Y == class1]
    data_class2 = X[Y == class2]
    data = np.concatenate([data_class1, data_class2])

    pca = PCA(n_components=2)
    pca_data = pca.fit_transform(data)

    gmm = GaussianMixture(n_components=2)
    gmm.fit(pca_data)
```

```
            distance = np.linalg.norm(gmm.means_[0] - gmm.means_
                [1])

            if distance > max_distance:
                max_distance = distance
                max_pair = (class1, class2)

            if distance < min_distance:
                min_distance = distance
                min_pair = (class1, class2)

    print(f'Pair with highest distance: {max_pair}, Distance
        : {max_distance}')
    print(f'Pair with shortest distance: {min_pair},
        Distance: {min_distance}')
```

which gives the following output:

1. Pair with highest distance: (0, 1), Distance: 1915.66
2. Pair with shortest distance: (4, 9), Distance: 979.84

The above results make perfect sense because we expect the distance to be maximized when two classes do not have high resemblance like classes 0 and 1.On the other hand, 4 and 9 look so much like each other which is why we see the shortest distance between these two.

Question 7

Codes related to this question are stored in Q7.ipynb notebook file.First, lets explore some methods of finding the optimal number of clusters in a clustering problem:

- k-means distortion and elbow analysis

  K-means distortion is calculated as the sum of squared distances between each data point and its assigned cluster centroid. The goal of K-means clustering is to minimize this distortion, as a lower distortion indicates better clustering performance. Elbow analysis is a method used to determine the optimal number of clusters (k) in a K-means clustering algorithm. The idea is to plot the distortion values for different values of k and look for an elbow point on the graph where the rate of decrease in distortion starts to slow down significantly. This point represents the optimal number of clusters to use for the dataset.

- Silhouette Score

  The silhouette score is a metric used to evaluate the quality of clusters produced by a clustering algorithm, such as K-means. It calculates the

average similarity of each data point with its own cluster compared to neighboring clusters.A score close to 1 indicates that the data point is well matched to its own cluster and poorly matched to neighboring clusters. A score close to 0 indicates that the data point is on or very close to the decision boundary between two clusters. A score close to -1 indicates that the data point is likely assigned to the wrong cluster.

- Davies-Bouldin Index

  The Davies-Bouldin Index is calculated by considering the average similarity between each cluster and its most similar cluster, while also taking into account the average distance between the centroids of the clusters.

- Calinski-Harabasz Index

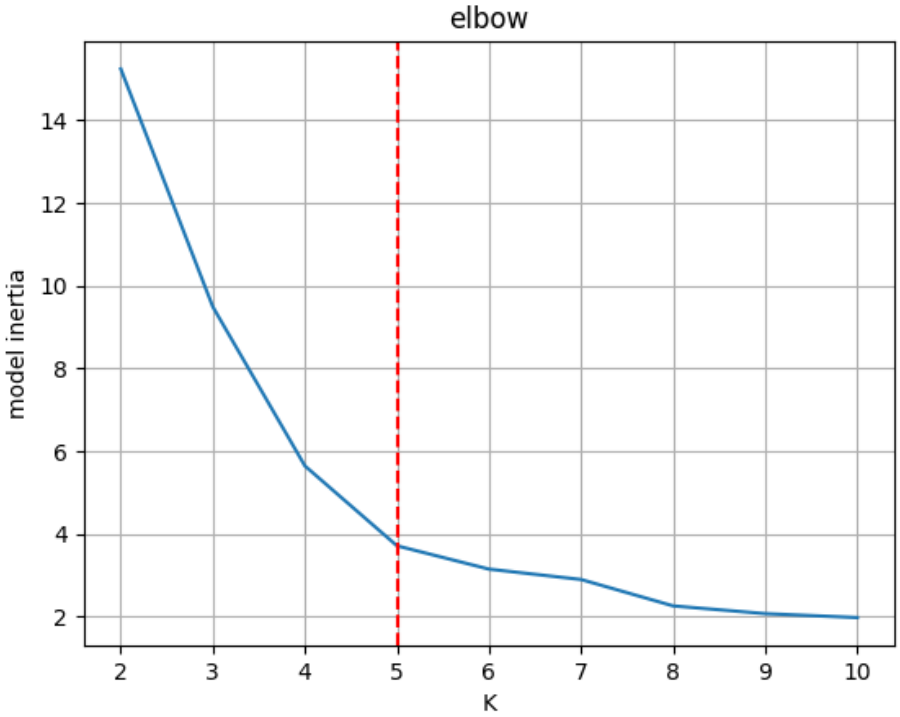
  The Calinski-Harabasz Index (also known as the Variance Ratio Criterion) is a metric which calculates a ratio that measures the separation between clusters and the compactness within clusters. The Calinski-Harabasz Index is calculated as the ratio of the sum of between-cluster dispersion to the sum of within-cluster dispersion.

- Dunn Index

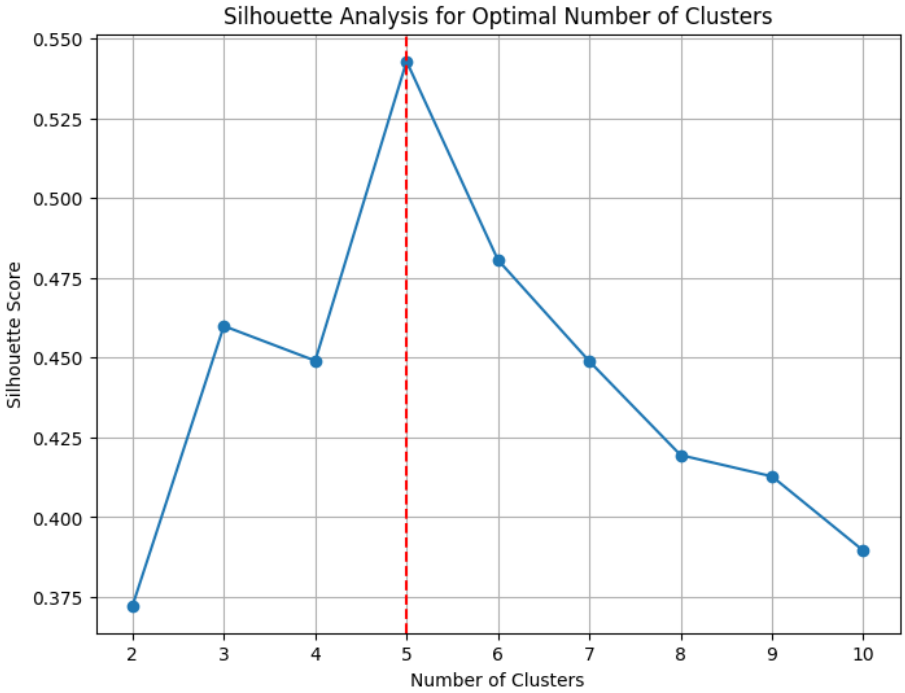
  The Dunn Index measures the clustering quality based on the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance. The Dunn Index is calculated as the ratio of the minimum inter-cluster distance (the distance between the nearest points of different clusters) to the maximum intra-cluster distance (the distance between the farthest points within the same cluster).

Now lets use the explained methods to find the optimal number of clusters:

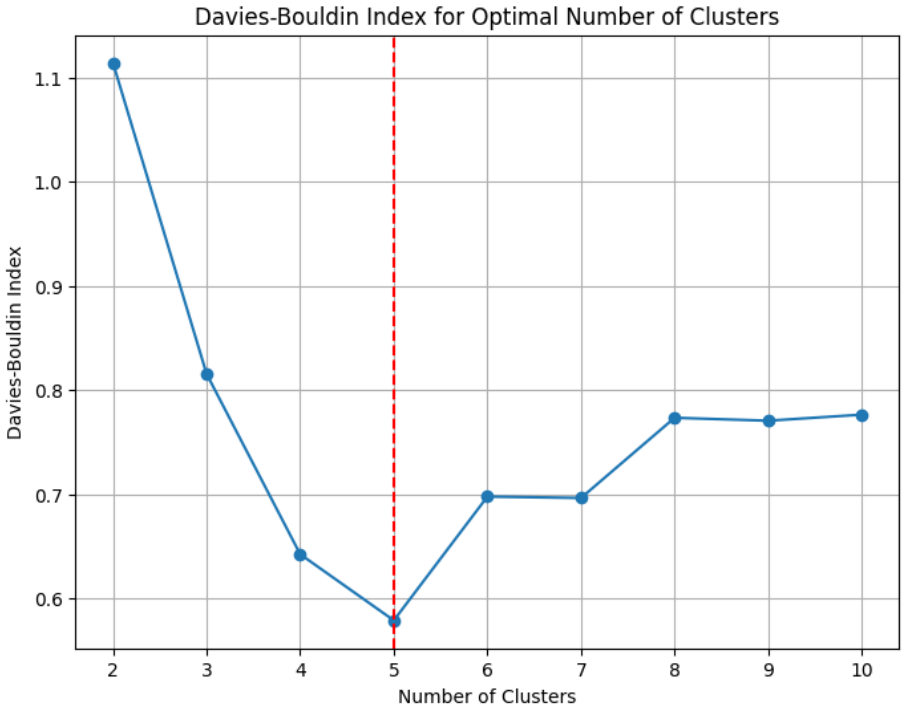
Elbow



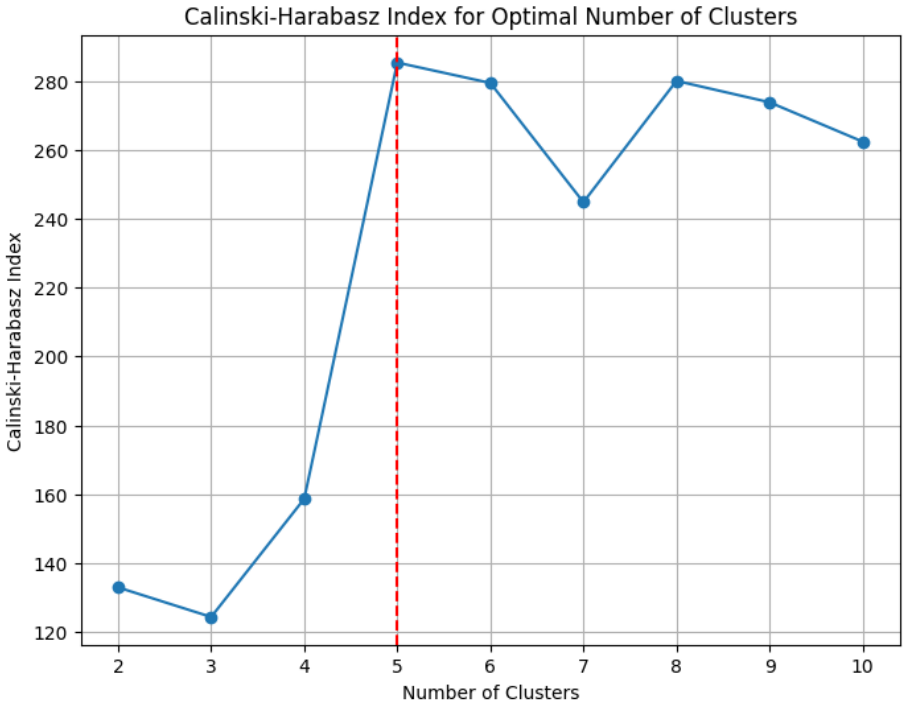
Silhouette Score



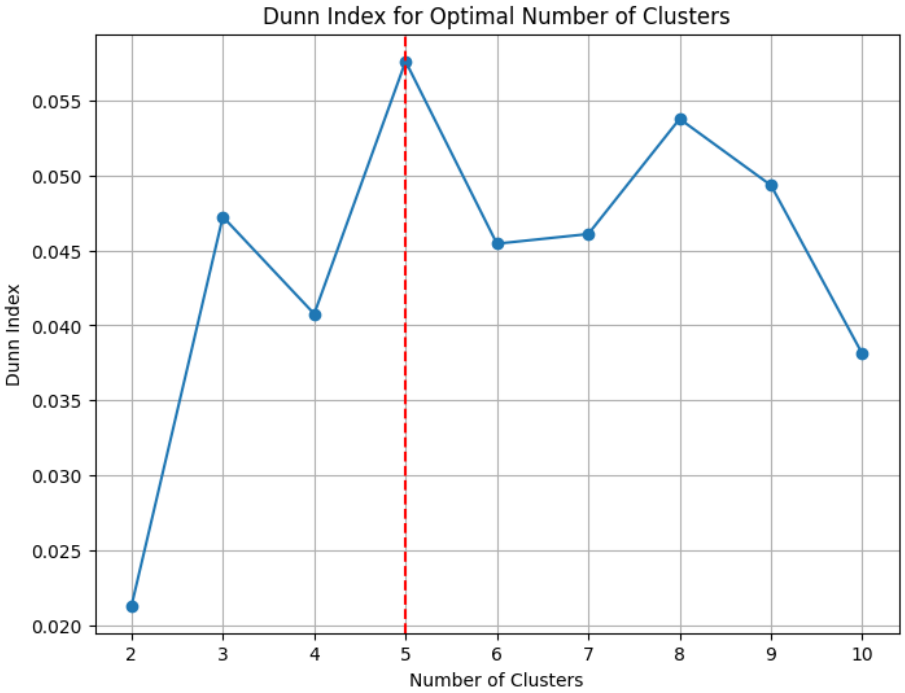
Davies-Bouldin Index



Calinski-Harabasz Index



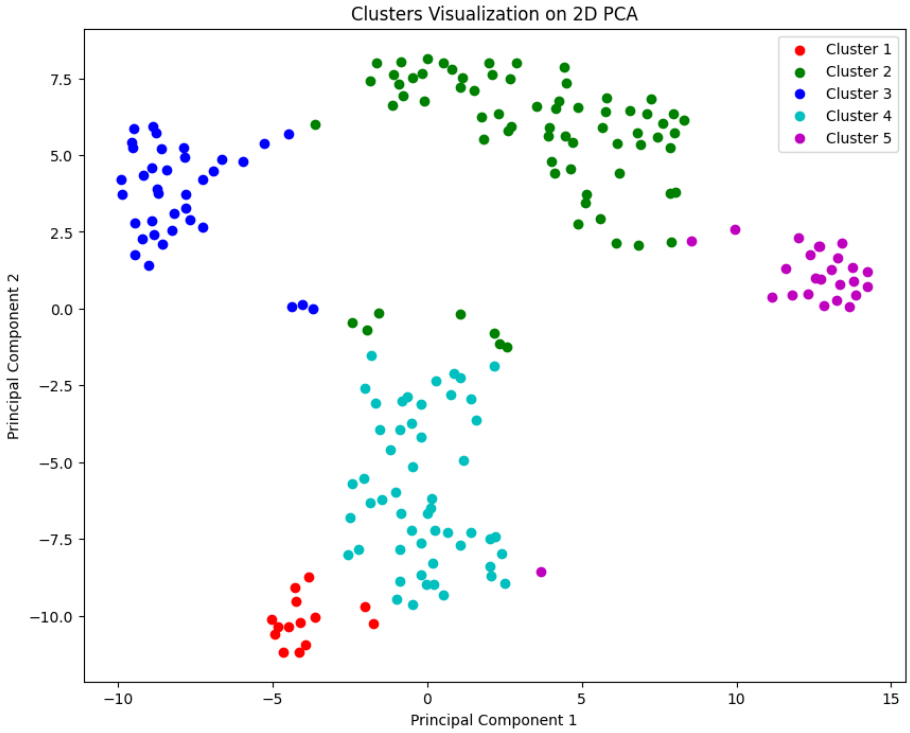
Dunn Index



As we see in all of the above figures, the optimal number of clusters is k=5.Now we can fit the optimal model to the data. But our data vector has 4 dimensions and can not be visualized on a 2D grid.There exists differnet methods for reducing the number of dimensions to make them visualizable.Here we explore PCA nd TNSE:

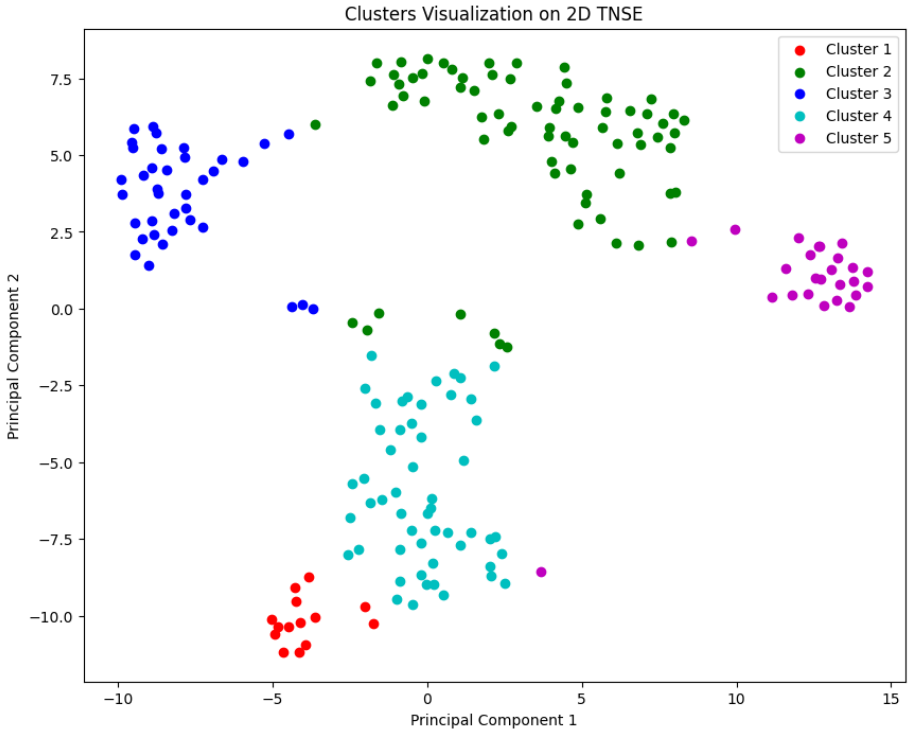
PCA

As we know, PCA makes components which maximaizes the variance.Fitting a PCA model with 2 components to the data and plotting the clusters, we get the following figure:



TNSE

T-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction technique used for visualizing high-dimensional data in a lower-dimensional space. t-SNE aims to preserve the local structure of the data by mapping similar data points in the high-dimensional space to nearby points in the low-dimensional space. Fitting a TNSE model with 2 components and plotting the clusters on a 2D grid, we get the following figure:


Clusters Visualization on 2D TNSE
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Principal Component 2
Principal Component 1
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10
−5
0
5
10
15