GE 461 - INTRODUCTION TO DATA SCIENCE

PROJECT 2

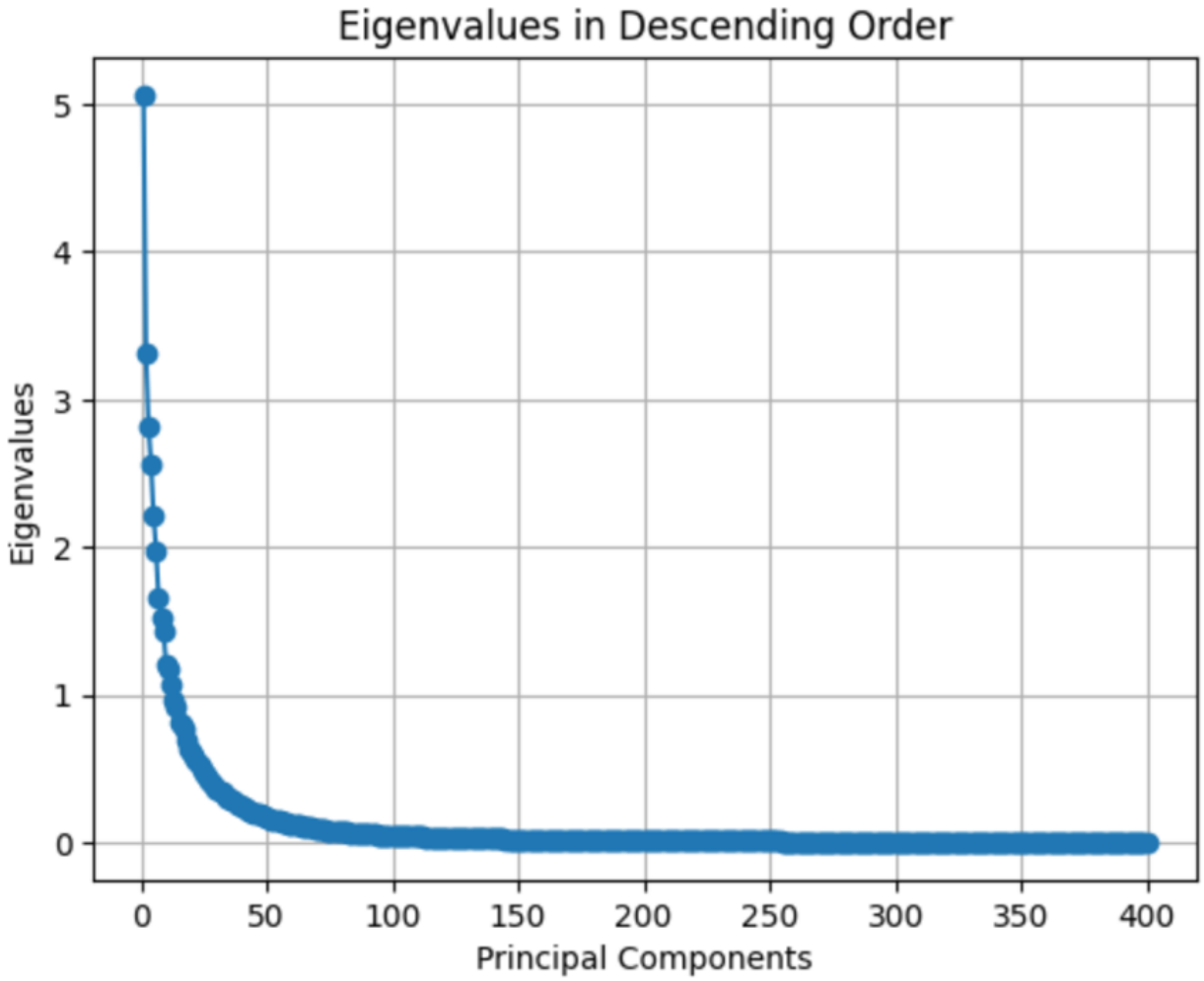
Necati Furkan Çolak / 21803512

Abstract:

This study explores Quadratic Discriminant Analysis (QDA) combined with Principal Component Analysis (PCA) and Isomap for classifying handwritten digits, utilizing the MNIST dataset. We compare classification accuracy, stability, and computational efficiency of both methods. Additionally, we employ t-Distributed Stochastic Neighbor Embedding (t-SNE) for data visualization. Our results show that QDA with Isomap outperforms QDA with PCA in classification performance and stability. T-SNE visualizations effectively reveal distinct digit clusters, providing insights into the dataset's structure and dimensionality reduction efficacy.

- I don’t include question 1.1

Question-1.2:

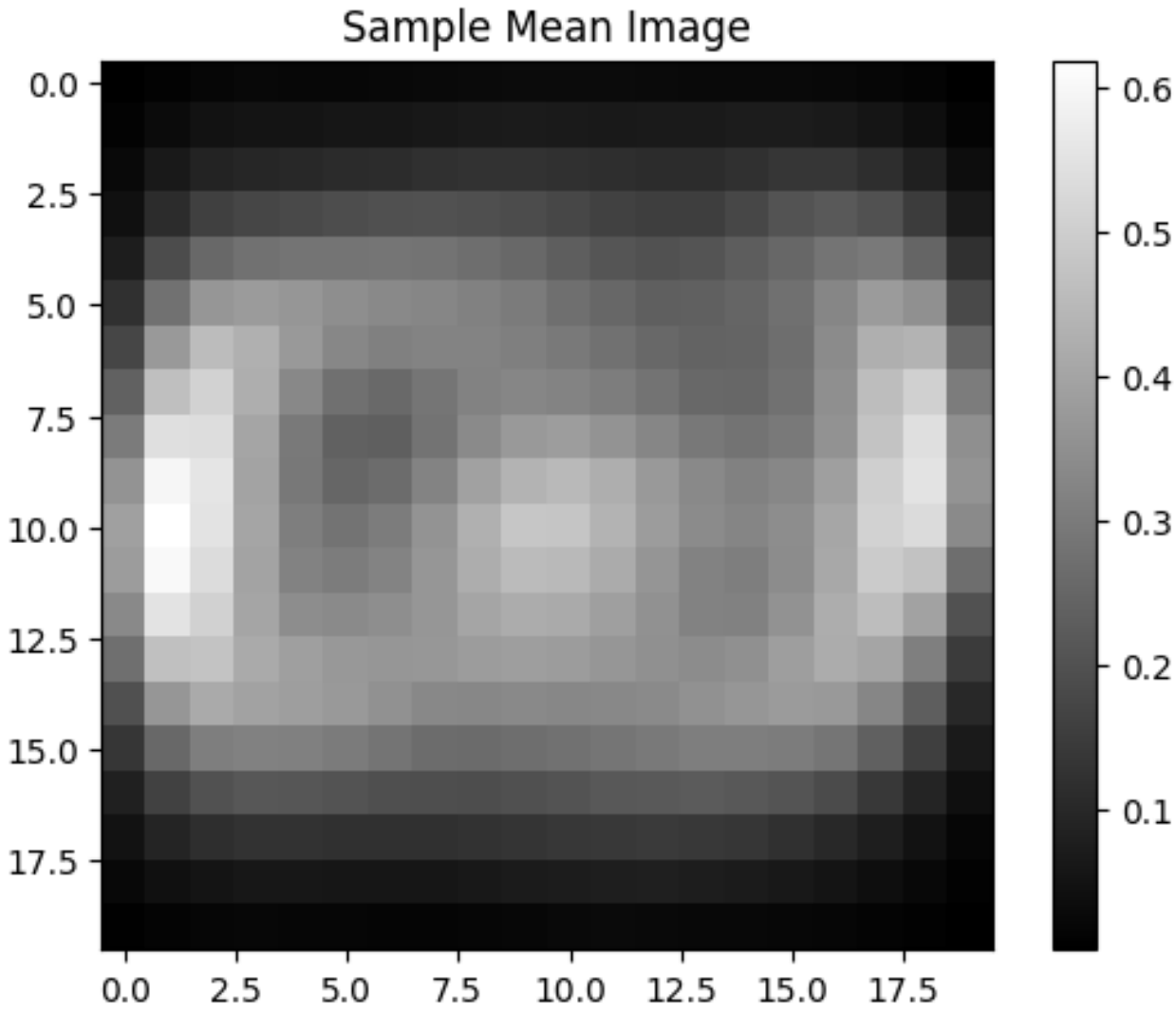


This chart shows the eigenvalues in descending order obtained from applying PCA to the train data. The eigenvalues represent the explained variance of each principal component, while the eigenvectors correspond to the principal components themselves [1].

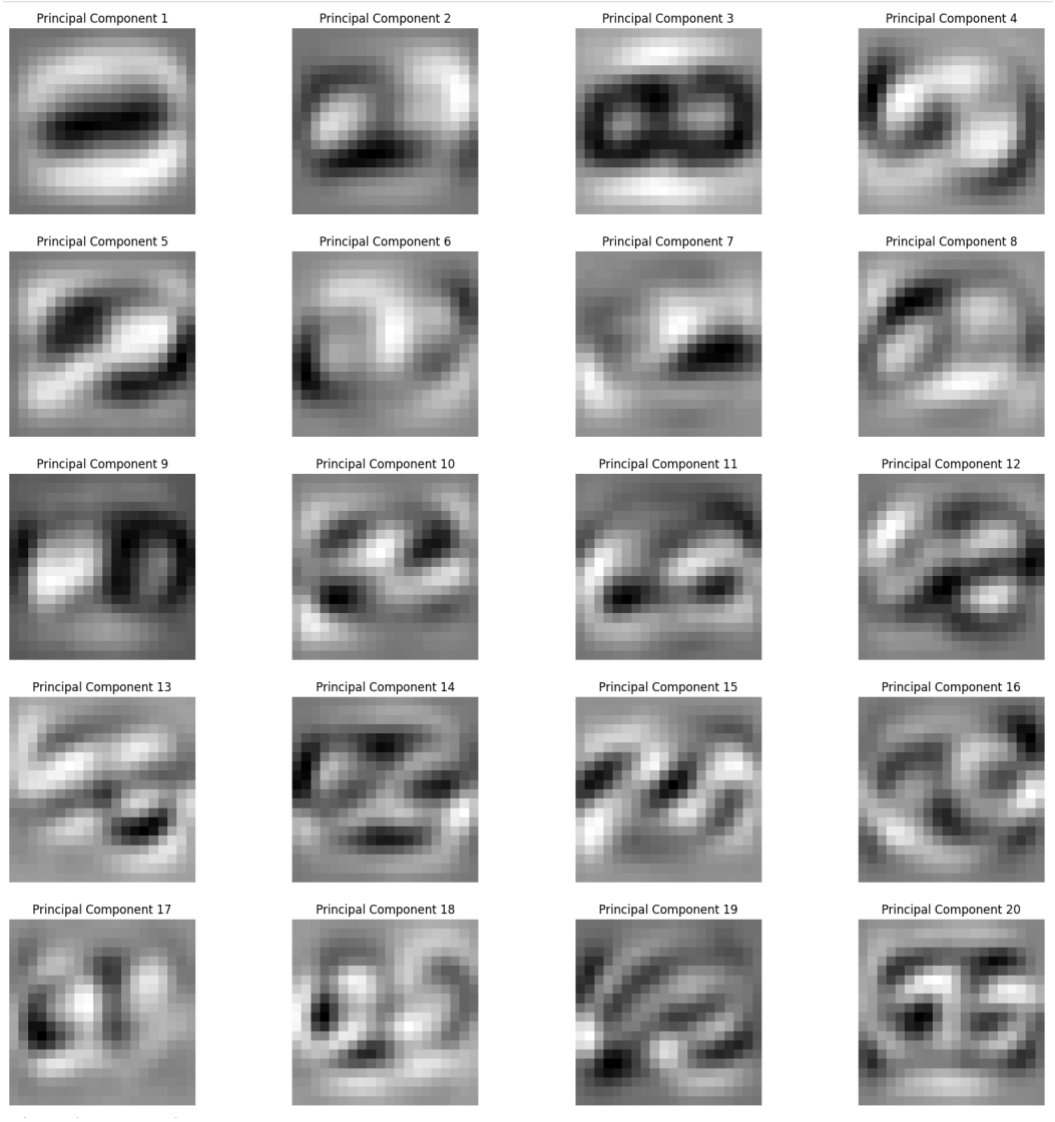
Looking at the plot, we can observe that the eigenvalues rapidly decrease in magnitude after the first few components, and the decline becomes more gradual as we move towards the end of the spectrum. This suggests that a small number of principal components can capture most of the variance in the data, and that additional components contribute less and less to the overall explanation of the variation.

Based on this observation, it may be sufficient to retain only the first 30 principal components to explain most of the variation in the data, as the eigenvalues start to level off after this point. This can help us reduce the dimensionality of the data while preserving most of the important information. If we choose more component then we can cause overfitting, and this may lead increasement at test error.

Question-1.3:



Looking at the image, we can see that the mean of the training data resembles the number 9 or 8 in horizontal way. This can be explained by the fact that the number 8 or the number 9 have similar structures with other digits like loops and curves. This explains why the mean image has approached the number 8 or number 9.



When examining the principal components, we observe that eigenvalues are sorted in decreasing order from component 1 to 20, as component 1 resembles digit 0 and component 3 resembles digit 8. Consequently, it can be inferred that the initial components possess larger eigenvalues than the latter ones, enabling us to derive meaningful insights by examining these eigenvectors. This observation also relates to question 1.2, as in both cases, the earlier eigenvectors carry more information, represented by larger eigenvalues, compared to the later ones. As we analyze further eigenvectors, the amount of meaningful information they convey decreases, leading to less insightful interpretations of the data.

Question 1.4:

In this question, we train our model with dimensionality reduction technique PCA and the Quadratic Discriminant Analysis in the scikit learn library. The reason why we choose Quadratic Discriminant Analysis since it calculates mean and covariance matrix for every distinct class unlike Linear Discriminant Analysis [2]. The content of the formulas is also checked from the website of scikit learn. [3]

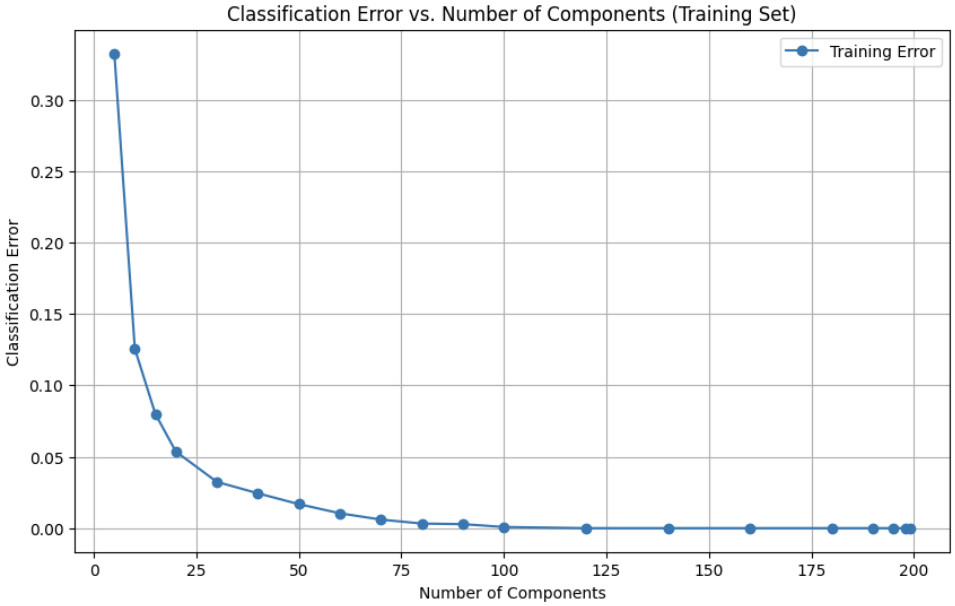
Subspace Dimension:	Train Error:	Test Error:
5	**0.3324**	**0.3584**
10	**0.1256**	**0.1428**
15	**0.0800**	**0.1008**
20	**0.0536**	**0.0772**
30	**0.0324**	**0.0668**
40	**0.0244**	**0.0688**
50	**0.0168**	**0.0752**
60	**0.0104**	**0.0784**
70	**0.0060**	**0.0876**
80	**0.0032**	**0.0916**
90	**0.0028**	**0.0976**
100	**0.0008**	**0.1024**
120	**0.0000**	**0.1240**
140	**0.0000**	**0.1468**
160	**0.0000**	**0. 1844**
180	**0.0000**	**0. 2352**
190	**0.0000**	**0. 2796**
195	**0.0000**	**0. 3124**
198	**0.0000**	**0. 3272**
199	**0.0000**	**0.3292**

In the table, subspace dimensions range from 5 to 199. Upon examining the values, it becomes evident that the model achieves its best performance with approximately 30 components, as the test error is at its lowest at this point. Beyond 30 components, each additional component increases the test error, indicating that they do not enhance the model's performance. Furthermore, the model exhibits overfitting, as the train error reaches 0.000 after 120 components, while the test error continues to rise. This situation is a clear indication of overfitting. The table also underscores the importance of subspace selection, as choosing too many

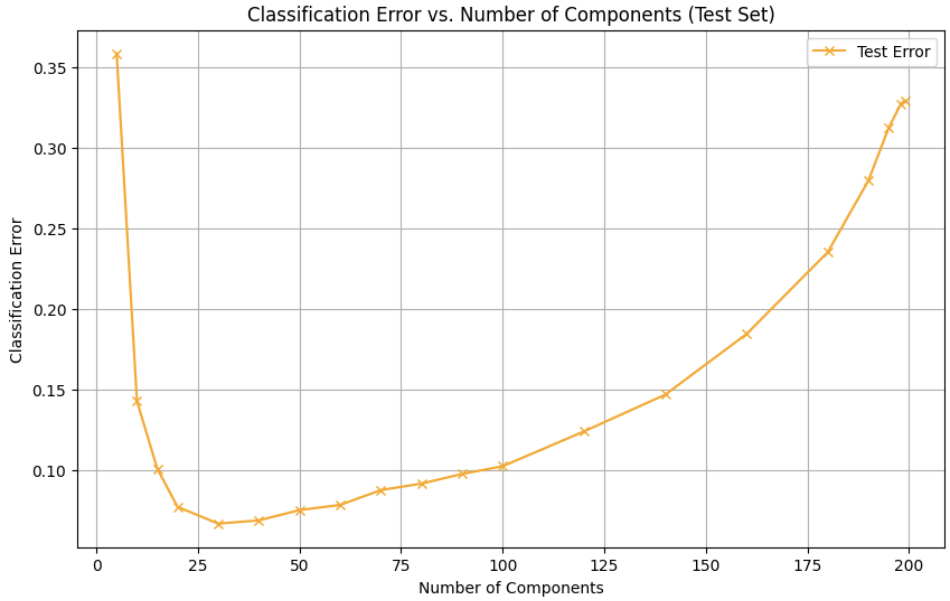
dimensions can lead to overfitting, while selecting too few dimensions may not yield an effective classifier.

Question 1.5:

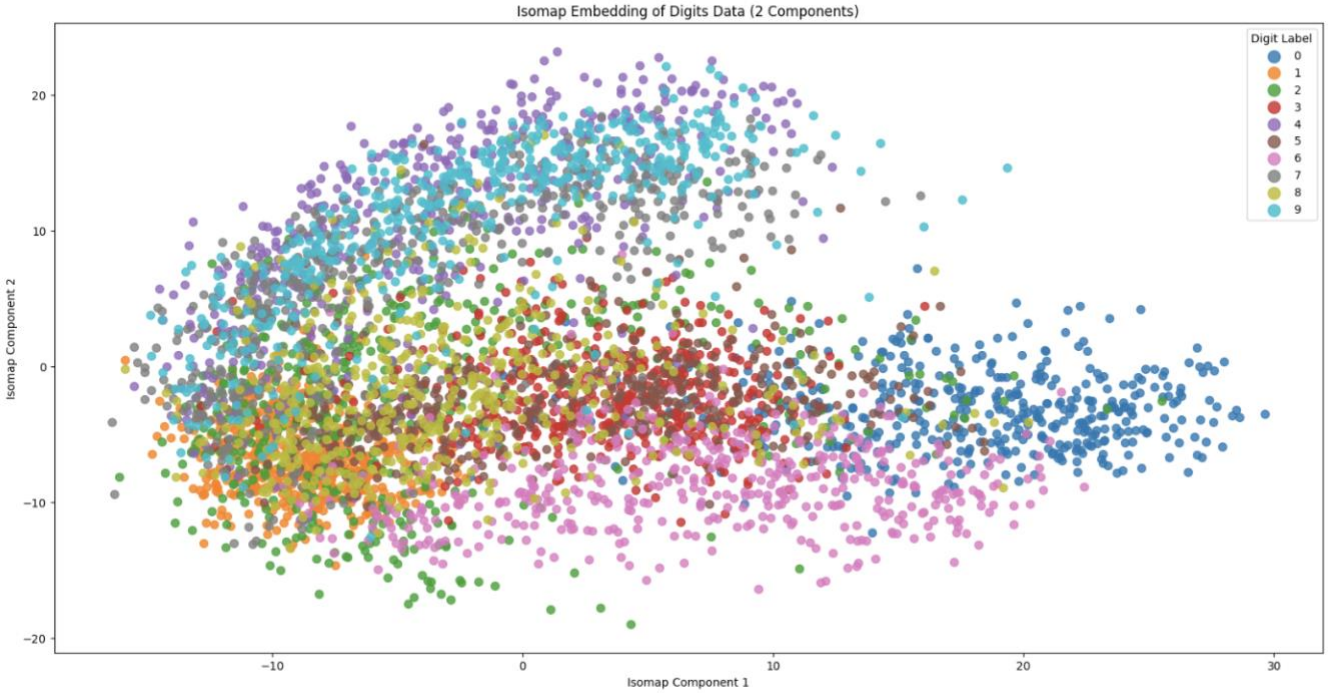
1-) Train Error for Quadratic Discriminant Analysis with PCA:



2-) Test Error for Quadratic Discriminant Analysis with PCA:



Question- 2.1:



- It resembles to swiss-roll a bit [6].

Question- 2.2:

In this question, we train our model with dimensionality reduction technique Isomap and the Quadratic Discriminant Analysis in the scikit learn library.

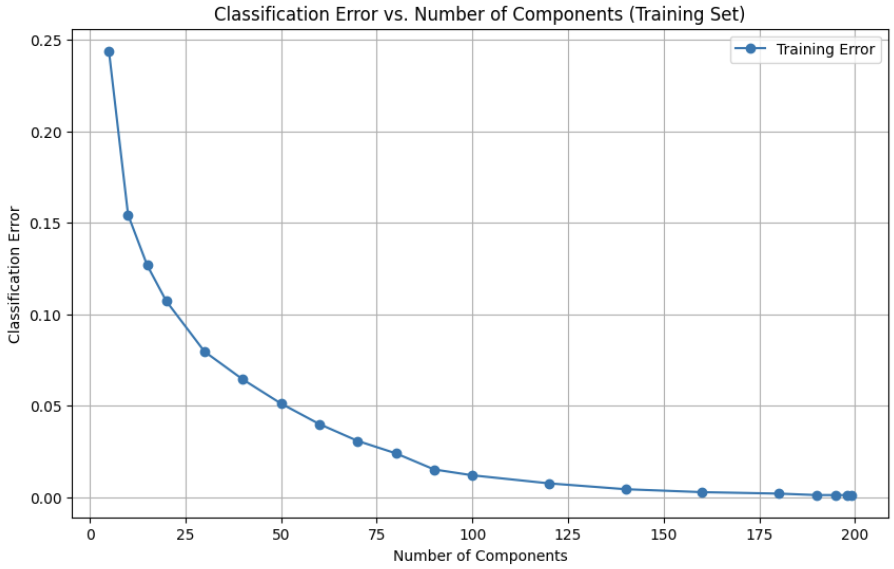
Subspace Dimension:	Train Error:	Test Error:
5	**0.2440**	**0.2608**
10	**0.1544**	**0.1848**
15	**0.1268**	**0.1772**
20	**0.1072**	**0.1636**
30	**0.0796**	**0.1488**
40	**0.0644**	**0.1488**
50	**0.0512**	**0.1508**
60	**0.0400**	**0. 1516**
70	**0.0308**	**0.1512**
80	**0.0240**	**0.1556**
90	**0.0152**	**0.1604**
100	**0.0120**	**0.1672**

120	**0.0076**	**0.1772**
140	**0.0044**	**0.1848**
160	**0.0028**	**0.1988**
180	**0.0020**	**0. 2300**
190	**0.0012**	**0. 2540**
195	**0.0012**	**0. 2724**
198	**0.0012**	**0. 2868**
199	**0.0012**	**0. 2880**

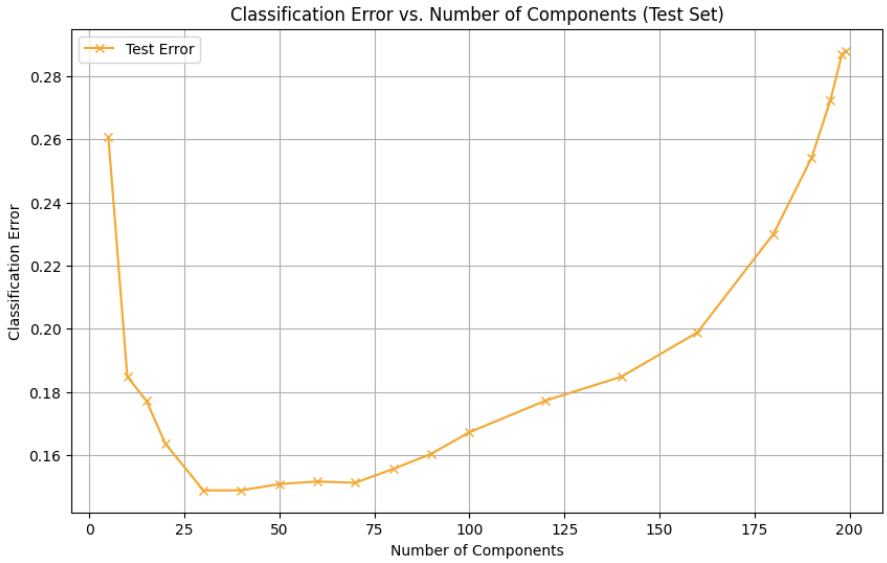
Analyzing the table, it's clear that the model performs best with around 30 and 40 components, as the test error reaches its lowest value of 0.1488. Beyond 30 components, the test error increase significantly. Indicating that components after 40 affects model's performance in negative way. The model exhibits overfitting, as train error drops to a minimal value (0.0012) after 190 components, while test error continues to rise. This discrepancy highlights the importance of proper subspace selection. Selecting too many dimensions can lead to overfitting and reduced performance on new data, while too few dimensions might result in an ineffective classifier as in PCA method. The table underlines the significance of carefully determining the optimal number of components for subspace selection.

Question- 2.3:

1-) Train Error for Quadratic Discriminant Analysis with Isomap:



2-) Test Error for Quadratic Discriminant Analysis with Isomap:



Discussion of the Results of PCA and Isomap:

When we compare both models, it can be said that although the model with PCA reaches lower test error scores, the model constructed with Isomap constructs more stable model. Therefore, I can say that the model constructed with Isomap shows better performance than model constructed with PCA.

First let we discuss the differences between the methods,

PCA (Principal Component Analysis) is a linear dimensionality reduction technique that finds a set of orthogonal axes (principal components) that capture the maximum variance in the data. It assumes that the data lies in a linear subspace and may not perform well if the underlying data structure is nonlinear.

Isomap (Isometric Mapping) is a nonlinear dimensionality reduction technique that aims to preserve the geodesic distances (shortest path) between data points. It does this by constructing a nearest-neighbor graph and then computing the shortest paths between all pairs of points. Finally, it applies classical multidimensional scaling to embed the data points in a lower-dimensional space while preserving the pairwise distances.

- Therefore, we can say that the main difference between PCA and Isomap can be said that while PCA works better with the linearly dependent datas, Isomap is more successful in non-linear datasets.

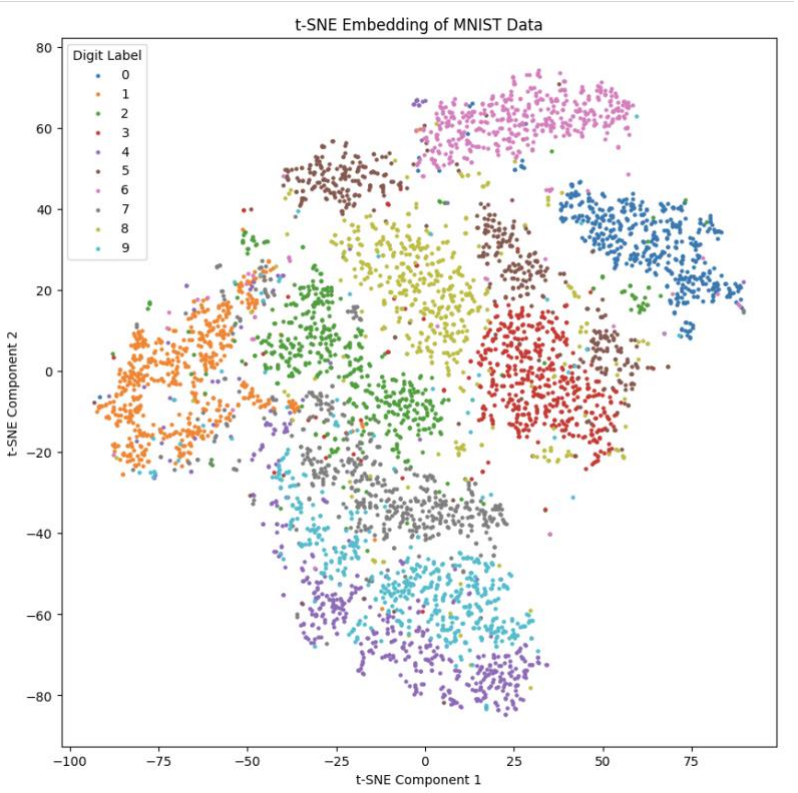
- Since, the dataset consists of handwritten digits, the structure of dataset is non-linear.

By looking the information below, I consider two main principles about why Isomap works better in PCA,

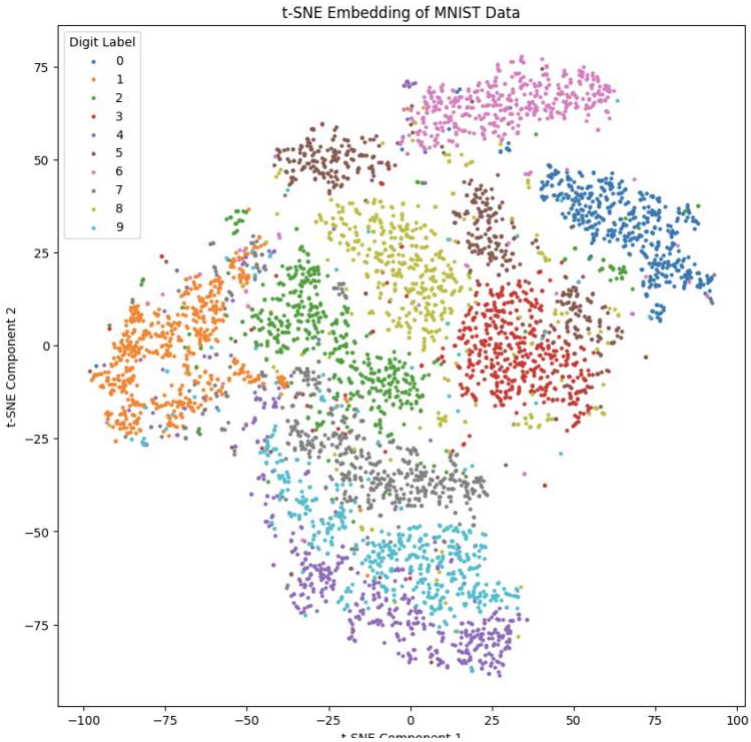
1.Nonlinearity: The digits dataset, which consists of handwritten digits, may have nonlinear structures that Isomap can capture more effectively than PCA. Since Isomap is designed to handle nonlinear data, it can better represent the underlying structure, leading to improved classification performance.

2.Stability: The second model demonstrates more stability in its test error, with a smaller difference between the maximum and minimum test errors compared to the first model. This suggests that the second model generalizes better to unseen data. One possible explanation is that Isomap's ability to preserve geodesic distances results in a more robust representation of the data, making the classifier more resistant to variations in the test set [4].

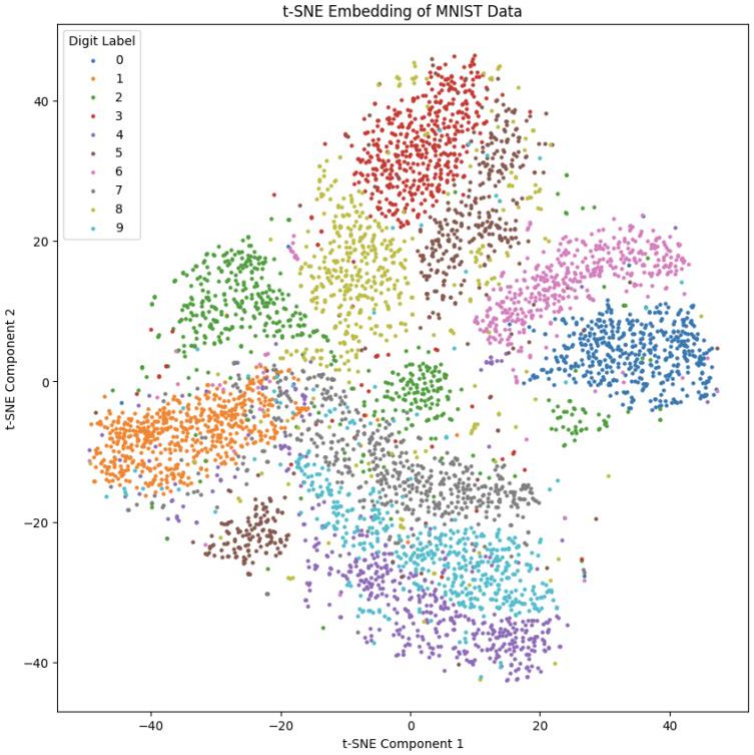
Question-3):



- **n_components=2,n_iter=1000, random_state=42, perplexity=100,learning_rate=200**



- **n_components=2,n_iter=1000, random_state=42,perplexity=25,learning_rate=400**



- **n_components=2,n_iter=1000, random_state=42, perplexity=100,learning_rate=200**

Discussion for t-SNE:
In the setup I used, the main purpose is to visualize digits data in a 2D space. t-SNE also one of the non-linear dimensionality reduction algorithms. The structure behind t-SNE is minimizing divergence between two probability distributions [5].

Parameters in my code:

- **`n_components`:** The number of dimensions for the reduced data. In this case, I chose to reduce the data to 2 dimensions for visualization purposes.

- **`n_iter`:** The number of iterations for the optimization. I set this to 1000, which is a reasonable number of iterations for the algorithm to converge.

- **`random_state`:** The seed used by the random number generator. I set this to 42, which ensures that the results are reproducible.

  I also change these two parameters constantly:

- **"perplexity":** Perplexity balances local and global structure, influencing neighborhood size.

- **"learning_rate":** Learning rate controls gradient descent step size, affecting convergence speed and stability in t-SNE optimization.

  - Changing perplexity in t-SNE affects the balance between local and global structure preservation. Higher perplexity focuses more on global structure, while lower perplexity emphasizes local structure, potentially altering cluster separation and data representation in the resulting visualization [7].

  - Changing the learning rate in t-SNE visuals affects the optimization process. A higher learning rate may lead to faster convergence but can cause instability or overshooting the optimal solution. A lower learning rate results in slower convergence but may provide a more stable and accurate representation of the data [8].

After we fit t-SNE to "digitss" data, which is our original dataset, the resulting visualization is scatter plot. Each point and color represent a digit it can also be seen in legend in the plot.

From the plot provided, it can be understood that visuals are clearer rather than Isomap. The reasons can be explained below.

1. Clustering: The digits dataset consists of handwritten digits, which often form clusters in the high-dimensional space. t-SNE is designed to preserve local structures and reveal clusters in the data, making it easier to visually assess the presence of distinct digit groups. Isomap, which focuses on preserving geodesic distances, might not generate visualizations with such clearly separated clusters.

2. Local structure preservation: The digits dataset has a complex structure with variations of writing styles, stroke thickness, and digit orientation. t-SNE is particularly effective at capturing local similarities in the data, enabling it to visualize these variations in a 2D space effectively. Isomap, which focuses on the global structure, may not emphasize these local structures as much.

3. Manifold structure: The digits dataset does not have an clear, easily discernible manifold structure that Isomap can uncover. In contrast, t-SNE is more flexible in modeling complex, nonlinear relationships and works well even when there isn't a clear underlying manifold.

4. Noise handling: Handwritten digits can be affected by noise in the form of imperfections, smudges, or other irregularities. t-SNE is more robust to noise and less sensitive to outliers, as it prioritizes preserving local structure. Isomap, which aims to preserve global structure, might be more affected by noise and outliers in the digits dataset.

References:

[1] Clairvoyantsoft. (n.d.). Eigen decomposition and PCA. Clairvoyant Blog. Retrieved from https://blog.clairvoyantsoft.com/eigen-decomposition-and-pca-c50f4ca15501

[2] Chandra, S. (2019, January 2). Differences of LDA, QDA and Gaussian Naive Bayes classifiers. Towards Data Science. Retrieved from https://towardsdatascience.com/differences-of-lda-qda-and-gaussian-naive-bayes-classifiers-eaa4d1e999f6

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830. Retrieved from https://scikit-learn.org/stable/modules/lda_qda.html

[4] Kumar, V. (2018, October 10). Dimension reduction using Isomap. Medium. Retrieved from https://medium.com/data-science-in-your-pocket/dimension-reduction-using-isomap-72ead0411dec

[5] Wattenberg, M., Viégas, F. B., & Johnson, I. (2016). How to use t-SNE effectively. Distill. Retrieved from https://medium.com/towards-data-science/an-introduction-to-t-sne-with-python-example-5a3a293108d1

[6] Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500), 2319-2323.

[7] Sharma, A. (2018, April 15). An introduction to t-SNE with Python example. Towards Data Science. Retrieved from https://medium.com/towards-data-science/an-introduction-to-t-sne-with-python-example-5a3a293108d1.

[8] van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579-2605.

References for Codes:

- `numpy`: used for scientific computing
  - Source: https://numpy.org/doc/stable/index.html

- `math`: used for mathematical functions
  - Source: https://docs.python.org/3/library/math.html

- `sklearn.decomposition`: used for matrix factorization and dimensionality reduction
  - Source: https://scikit-learn.org/stable/modules/decomposition.html

- `sklearn.discriminant_analysis`: used for linear and quadratic discriminant analysis
  - Source: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.discriminant_analysis

- `matplotlib.pyplot`: used for data visualization
  - Source: https://matplotlib.org/stable/contents.html

- `sklearn.datasets.fetch_openml`: used for fetching datasets from openml.org
  - Source: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_openml.html

- `scipy.linalg.eigh`: used for computing eigenvalues and eigenvectors of symmetric matrices
  - Source: https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.eigh.html

- `sklearn.manifold.Isomap`: used for isometric mapping
  - Source: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html

- `pandas`: used for data manipulation and analysis
  - Source: https://pandas.pydata.org/docs/

- `sklearn.model_selection.train_test_split`: used for splitting datasets into training and testing subsets
  - Source: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

- `sklearn.metrics.accuracy_score`: used for computing classification accuracy.
  - Source: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html