Statistical Machine Learning 2024

ASSIGNMENT 2

Quadratic Discriminant Analysis (QDA) on the MNIST Dataset and Principal Component Analysis (PCA) and QDA on the MNIST Dataset.

ASSUMPTIONS

lamda=1e-6 for making the determinant non-singular, and for determinant == 0 or infinite, I have put determinant=1e-6, and according to that, I have calculated the pseudo inverse of the covariance matrix**.**

APPROACH

Question 1 Approach :

The provided code defines a Quadratic Discriminant Analysis (QDA) classifier and applies it to the MNIST dataset for handwritten digit recognition. Here's a step-by-step overview of the code:

1. **Imports:**
    - TensorFlow and its components (for machine learning).
    - NumPy for numerical operations.
    - Pandas for data manipulation.
    - Seaborn and Matplotlib for data visualization.
    - MNIST dataset from TensorFlow.
2. **QDA Class Definition:**
    - `QDA` class is defined for Quadratic Discriminant Analysis.
    - `__init__`: Initializes the `estimates` attribute.
    - `train`: Trains the QDA model using the provided training data (`X_train`, `y_train`).

   - `compute_estimates`: Computes mean vectors, covariance matrices, determinant, and inverse covariance matrices for each class.
   - `predict`: Predicts the labels for given test data using the trained QDA model.
3. **Utility Functions:**
   - `class_idx`: Creates a dictionary of class indices from the provided labels.
   - `image_plot`: Plots a grid of sample images for each digit class.
4. **Data Loading and Preprocessing:**
   - MNIST dataset is loaded and normalized.
   - The dataset is split into training and testing sets.
   - Images are flattened to vector form for both training and testing datasets.
5. **Model Training and Prediction:**
   - An instance of the QDA class (`qda_clf`) is created.
   - The QDA model is trained using the training data.
   - Predictions (`y_pred`) are made on the test data.
6. **Evaluation:**
   - The accuracy of the QDA model is calculated on the test set.
   - Class-wise accuracy is also calculated and printed.
7. **Visualization:**
   - The function `image_plot` is used to visualize a grid of sample images for each digit class.
8. **Output:**
   - The accuracy of the QDA model on the entire test set and for each digit class is printed.

Note: There is a small typo in the `__init__` method of the `QDA` class. The correct method name should be `__init__` instead of `_init_`.

Question 2 Approach :

The provided code implements a Quadratic Discriminant Analysis (QDA) classifier on the MNIST dataset for handwritten digit recognition. Below is a step-by-step overview of the code:

1. **QDA_clf Class Definition:**

- `QDA_clf` class is defined for Quadratic Discriminant Analysis.
- `__init__`: Initializes the `estimates` attribute.
- `train`: Trains the QDA model using the provided training data (`X_train`, `y_train`).
- `compute_estimates`: Computes mean vectors, covariance matrices, determinant, and inverse covariance matrices for each class.
- `predict`: Predicts the labels for given test data using the trained QDA model.

2. **Utility Functions:**
    - `image_plot`: Plots a grid of sample images for each digit class.
    - `class_idx`: Creates a dictionary of class indices from the provided labels.
3. **Data Loading and Preprocessing:**
    - MNIST dataset is loaded and normalized.
    - The dataset is split into training and testing sets.
    - Images are flattened to vector form for both training and testing datasets.
4. **Model Training and Prediction:**
    - An instance of the QDA_clf class (`qda_`) is created.
    - The QDA model is trained using the training data.
    - Predictions (`y_pred_pca`) are made on the test data for different values of principal components (p).
5. **Principal Component Analysis (PCA):**
    - Principal Component Analysis is applied to the training and test datasets to reduce dimensionality.
    - Reconstruction and visualization of images are performed for different values of principal components.
6. **Evaluation:**
    - The accuracy of the QDA model is calculated on the test set for each value of p.
    - The predicted labels are printed for each value of p.
7. **Output:**
    - The reconstructed images and accuracy of the QDA model for different values of principal components are printed.
8. **Note:**

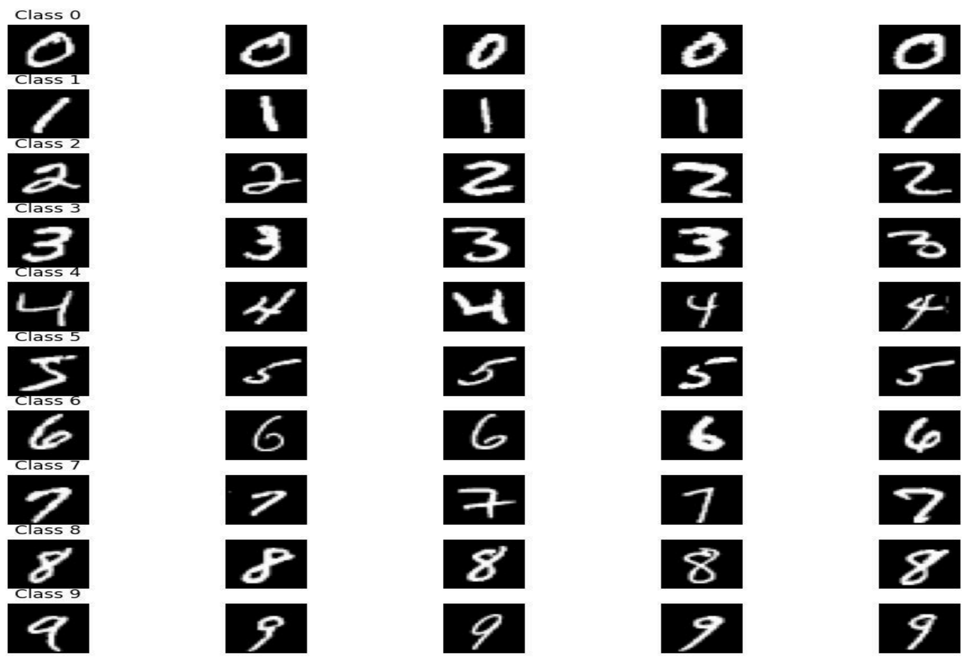
- There is a typo in the `__init__` method of the `QDA_clf` class. The correct method name should be `__init__` instead of `_init_`.

It's worth noting that the code demonstrates the application of QDA after reducing the dimensionality of the dataset using PCA, and it evaluates the performance of the QDA classifier for different numbers of principal components. Additionally, the code includes visualization of the reconstructed images after PCA.

RESULTS

Question 1 Result :

In the below image, I have shown 5 Samples from each class in the training dataset.



The accuracy of my QDA model is as follows:

Accuracy Of QDA: 0.858

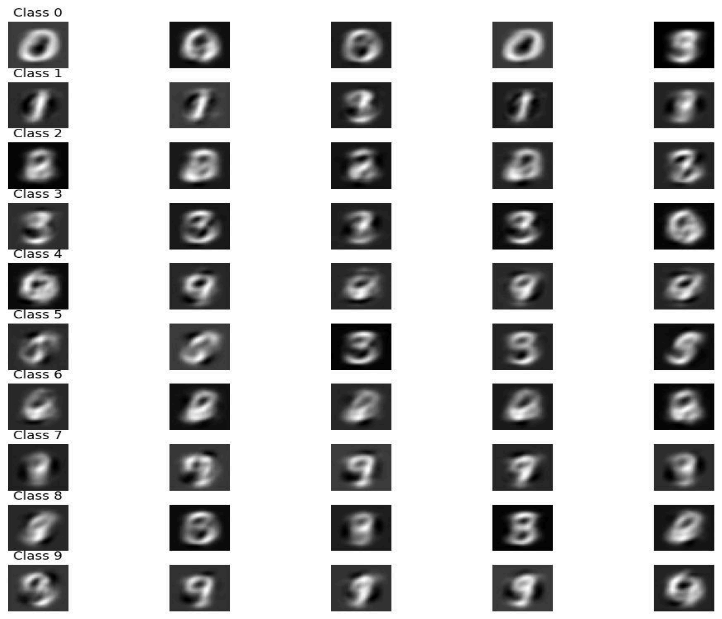
The classwise accuracy for my QDA model is as follows:

```
Accuracy for class 7: 0.8628404669260701

Accuracy for class 2: 0.936046511627907

Accuracy for class 1: 0.6740088105726872

Accuracy for class 0: 0.9346938775510204

Accuracy for class 4: 0.9083503054989817

Accuracy for class 9: 0.8295341922695738

Accuracy for class 5: 0.797085201793722

Accuracy for class 6: 0.8914405010438413

Accuracy for class 3: 0.8782178217821782

Accuracy for class 8: 0.8880903490759754
```

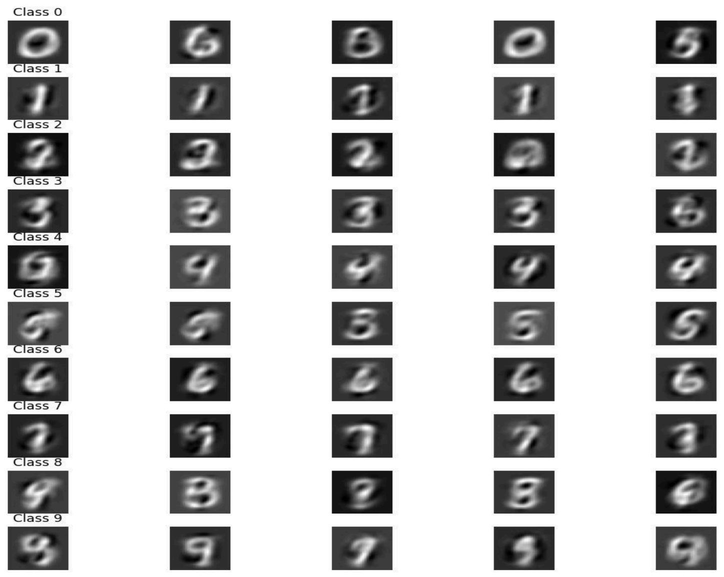
Question 2 Result :

<u>The output for the different inputs for my code</u>

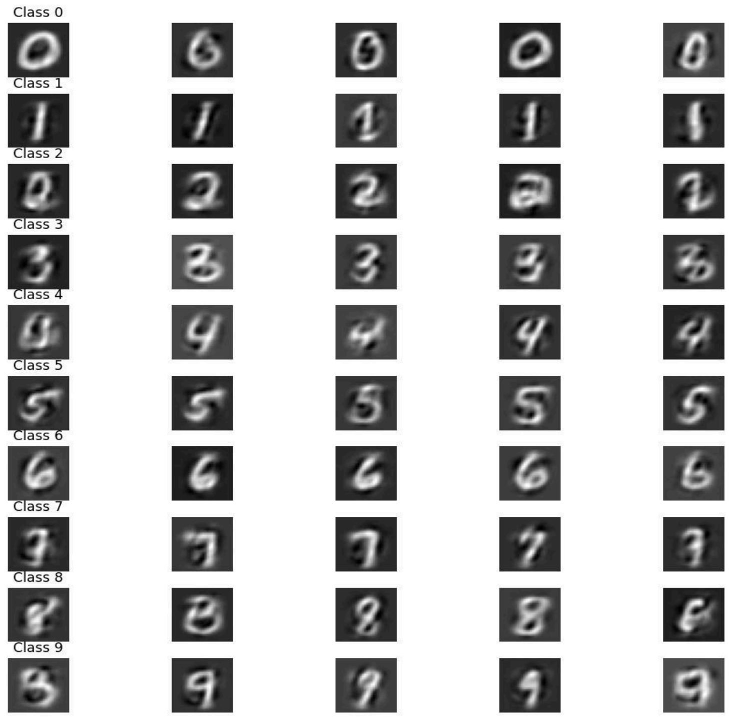
P = 5


Class 0
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9

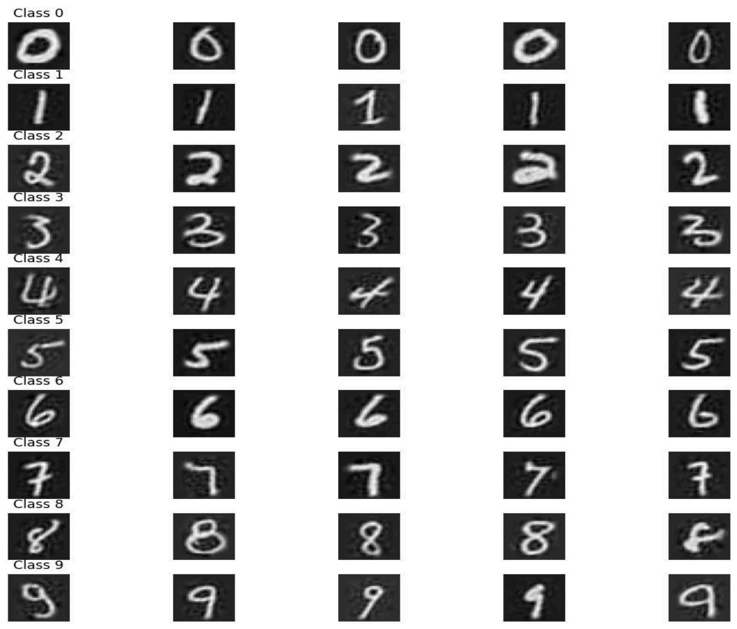
P = 10


Class 0
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9

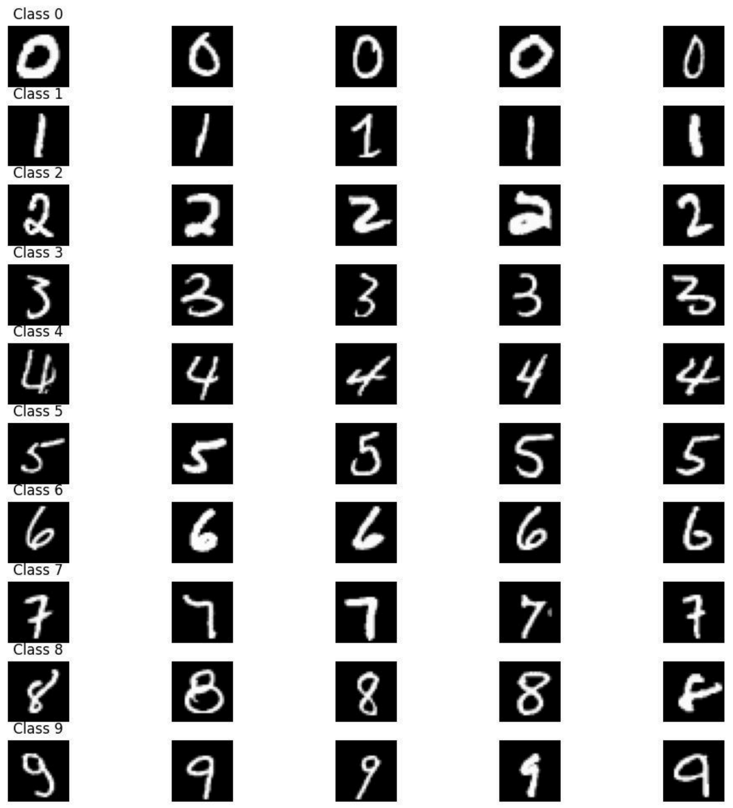
P = 20


Class 0
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9

P = 100


Class 0
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9

P = 783



The Mean Square Error is 2.286914586611201e-31

Below are the predicted labels for different p-values

8, 3, 8, 3, 5, 5, 3, 8, 3, 5, 3, 8, 3, 8, 8, 3, 2, 8, 8, 5, 3, 5, 5, 3, 3, 3, 5, 3, 3, 3, 3, 3, 5, 9, 3, 8, 8, 8, 3,
8, 8, 3, 8, 5, 3, 5, 8, 3, 3, 8, 3, 9, 8, 3, 2, 6, 8, 3, 2, 8, 8, 8, 5, 8, 3, 3, 3, 3, 0, 2, 8, 3, 8, 5, 2, 8, 0, 3,
3, 8, 2, 5, 3, 8, 3, 3, 8, 2, 8, 0, 3, 3, 3, 2, 3, 5, 5, 8, 2, 0, 3, 3, 8, 3, 2, 3, 3, 0, 3, 3, 8, 8, 8, 2, 8, 8, 3,
2, 3, 5, 8, 8, 8, 0, 8, 3, 3, 3, 5, 2, 8, 8, 0, 5, 3, 8, 3, 2, 3, 5, 5, 8, 8, 8, 0, 8, 2, 8, 2, 5, 8, 3, 8, 8, 5, 4,
3, 8, 0, 3, 3, 3, 0, 3, 0, 2, 8, 5, 4, 8, 5, 2, 2, 0, 3, 5, 3, 4, 0, 0, 2, 3, 3, 0, 5, 5, 3, 8, 8, 3, 8, 5, 2, 3, 8,
3, 3, 5, 2, 5, 0, 8, 4, 3, 3, 3, 3, 8, 3, 3, 0, 3, 8, 5, 0, 3, 3, 8, 3, 8, 8, 3, 8, 0, 8, 8, 0, 8, 3, 5, 3, 3, 2, 3,
3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 8, 0, 5, 0, 8, 8, 3, 3, 4, 4, 2, 8, 3, 3, 0, 3, 0, 2, 2, 0, 8, 5, 8, 4, 3, 0, 2, 5, 3,
5, 3, 3, 5, 3, 4, 8, 2, 0, 4, 3, 8, 0, 8, 2, 8, 5, 2, 2, 2, 3, 9, 5, 3, 5, 3, 2, 5, 0, 0, 9, 6, 8, 3, 3, 8, 5, 3, 2,
9, 0, 5, 2, 2, 9, 8, 0, 3, 3, 5, 5, 3, 3, 5, 5, 3, 2, 2, 5, 0, 8, 2, 7, 3, 5, 8, 3, 2, 3, 8, 5, 0, 8, 3, 4, 2, 8, 5,
8, 2, 2, 5, 8, 3, 5, 3, 3, 0, 3, 8, 5, 2, 2, 8, 0, 3, 6, 2, 8, 8, 3, 3, 0, 2, 2, 5, 3, 3, 3, 2, 2, 5, 2, 3, 3, 3, 5,
3, 5, 0, 2, 3, 2, 2, 5, 3, 8, 2, 4, 8, 3, 3, 3, 2, 8, 8, 3, 8, 8, 0, 0, 2, 8, 3, 4, 8, 2, 8, 5, 3, 8, 2, 3, 2, 0, 3,
2, 8, 4, 2, 5, 8, 3, 8, 5, 0, 0, 4, 8, 3, 8, 2, 3, 2, 3, 3, 3, 2, 5, 8, 3, 8, 3, 5, 3, 3, 3, 9, 5, 8, 3, 3, 3, 9, 0,

```
0, 3, 9, 3, 5, 8, 5, 5, 2, 8, 0, 8, 5, 3, 0, 8, 3, 0, 3, 8, 3, 3, 8, 3, 8, 4, 8, 8, 2, 8, 2, 3, 3, 8, 8, 9, 9, 3, 3,
3, 8, 3, 3, 9, 9, 3, 8, 8, 8, 3, 3, 6, 4, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 3, 3, 8, 9, 3, 3, 8, 8, 8, 3, 3, 3, 3, 5,
0, 0, 3, 8, 3, 3, 3, 3, 3, 3, 3, 5, 2, 3, 0, 0, 8, 8, 3, 3, 5, 3, 3, 8, 2, 3, 3, 3, 3, 8, 3, 8, 2, 8, 5, 3, 8, 3, 8,
2, 3, 0, 3, 8, 2, 0, 5, 8, 3, 3, 3, 5, 3, 8, 0, 3, 3, 5, 3, 3, 8, 0, 5, 2, 3, 2, 3, 0, 3, 8, 3, 2, 8, 8, 3, 2, 8, 3,
5, 8, 8, 3, 3, 3, 2, 5, 8, 0, 8, 3, 3, 8, 8, 8, 8, 3, 0, 3, 3, 2, 3, 5, 3, 5, 3, 3
```

Accuracy for different p values given to the code

Accuracy Of QDA for p=5: 0.8899999999999999

Accuracy Of QDA for p=10: 1.19

Accuracy Of QDA for p=20: 1.7100000000000002

Accuracy Of QDA for p=100: 1.04

Accuracy Of QDA for p=783: 0.95