# Report on Quadratic Discriminant Analysis (QDA) for MNIST Classification

This report details the implementation and evaluation of a QDA model for classifying handwritten digits in the MNIST dataset.

## Assumptions:

- The features extracted from the images accurately represent the digit information.
- The class labels are accurate and consistent.
- Covariance matrices are positive definite (regularization technique used to address potential issues).
- Classes exhibit Gaussian distributions (although QDA is robust to non-Gaussianity).

## Approach:

1. **Data Loading and Preprocessing:**
   - MNIST dataset loaded using numpy.load.
   - Training and testing data separated (x_train, y_train, x_test, y_test).
   - Pixel values normalized to range between 0 and 1.
   - Training data reshaped to 2D array for efficient calculations.
2. **Prior Probabilities:**
   - Number of samples per class calculated and used to estimate prior probabilities.
3. **QDA Model:**
   - Mean vectors and covariance matrices computed for each class.
   - Covariance matrices regularized with small identity matrix to ensure positive definite.
   - Inverse covariance matrices and determinants calculated.
   - This simplifies the function and reduces computational cost.
4. **Testing and Evaluation:**
   - QDA function applied to test data to produce class scores.
   - Predicted labels assigned based on highest score per sample.
   - Overall accuracy and class-wise accuracy computed.

## Results:

- Overall accuracy: **83.44%**
- Class-wise accuracy is printed in the Jupyter Notebook

## Further Details:

- The report demonstrates a simplified QDA implementation , highlighting its validity in specific cases.
- While overall accuracy is acceptable, some classes show lower performance, indicating potential for improvement using different models or feature engineering techniques.
- The code can be further optimized for efficiency, especially for larger datasets.

Limitations:

- The assumptions listed above might not always hold true, impacting model performance.
- QDA assumes Gaussian distributions, which might not be ideal for all datasets.

Conclusion:

This report demonstrates a basic QDA implementation for MNIST classification, achieving reasonable accuracy. Further exploration and adjustments can enhance performance and applicability to different datasets and classification problems.

Report on Principal Component Analysis (PCA) for MNIST Classification

## Assumptions:

- **Data Distribution:** The code assumes that the data follows a Gaussian distribution, as required for QDA.
- **Representative Sample:** The 100 samples per class are assumed to be representative of the overall dataset.
- **Linear Separability:** The code assumes that the classes are linearly separable in the reduced-dimensional space.

# Approach:

1. **Data Loading and Preprocessing:**
   - Load the MNIST dataset.
   - Select 100 samples per class.
   - Flatten the images into 784-dimensional vectors.
   - Subtract the mean from the data to center it.
2. **Principal Component Analysis (PCA):**
   - Calculate the covariance matrix of the data.
   - Compute the eigenvalues and eigenvectors of the covariance matrix.
   - Sort the eigenvectors by decreasing eigenvalues.
   - Select the top `p` eigenvectors to form the transformation matrix `U_p`.
3. **Data Reconstruction and Visualization:**
   - Project the data onto the reduced-dimensional space using `U_p`.
   - Reconstruct the data from the projected space using `U_p.T`.
   - Calculate the Mean Squared Error (MSE) between the original and reconstructed data.
   - Plot the reconstructed images for different values of `p` to visualize the effect of dimensionality reduction.
4. **Quadratic Discriminant Analysis (QDA):**
   - Define a function `qda` to calculate the QDA score for a given sample.
   - Define a function `applyQda` to apply QDA on the projected data for a given `p`.
   - Calculate prior probabilities, class means, and class covariance matrices in the projected space.
   - Make predictions on the test set using QDA.
   - Calculate the overall accuracy and class-wise accuracies.

5. **Accuracy vs. Dimensionality:**
    - Apply QDA for different values of p (5, 10, 15, 20, ..., 150).
    - Plot the accuracy of the QDA model as a function of p to visualize the trade-off between accuracy and dimensionality.

**Results:**

- The MSE between the original and reconstructed data decreases as p increases, indicating better reconstruction with more dimensions.
- The visual quality of reconstructed images improves with more dimensions.
- The QDA accuracy for different p values showed a pattern that needs further analysis (refer to the specific values and plot).
- Class-wise accuracies provide insights into which digits are easier or harder to classify.

**Further Details:**

- The code uses the full training set for QDA, while earlier calculations used a 1000-sample subset.
- The accuracy values for QDA might differ from those using smaller batches.

**Conclusions:**

- PCA effectively reduces dimensionality while preserving key information.
- QDA can achieve reasonable accuracy on MNIST, but its performance depends on the chosen dimensionality.
- Further analysis is needed to determine the optimal p for QDA and explore potential reasons for the observed accuracy pattern.