CS464 Introduction to Machine Learning
Fall 2023
Homework 2

Bahadır Yüzlü

21903504

1) PCA Analysis

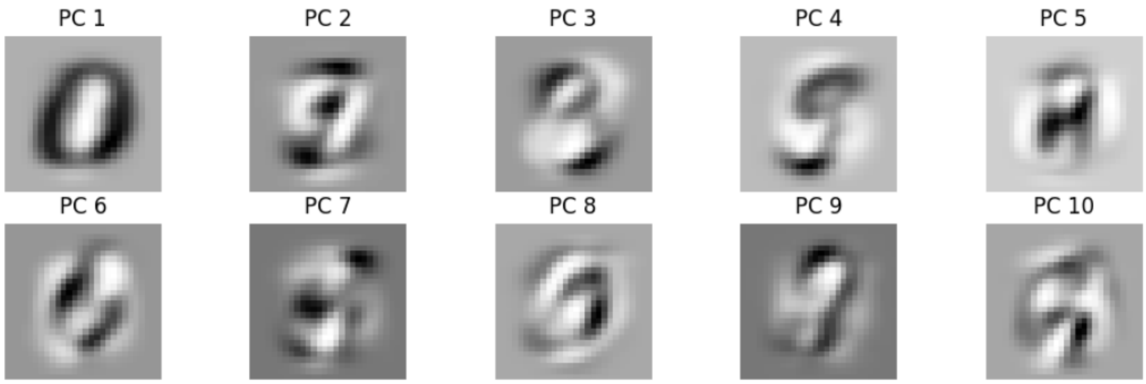
Q-1.1, 1.2)

```
indexes of the largest eigenvalues:  [0 1 2 3 4 5 6 7 8 9]
PVE for each principal component:
 [0.09704664 0.07095924 0.06169089 0.05389419 0.04868797 0.04312231
 0.0327193  0.02883895 0.02762029 0.02357001]
total variance explained by top 10 principal components:  0.4881498038548512
min number of principal components needed for >= %70 PVE:  26
```

The cumulative Proportion of Variance Explained (PVE) for the first 10 principal components amounted to approximately 48.8%. This finding emphesizes the significance of these initial components in capturing nearly half of the dataset's overall variance. Furthermore, a more comprehensive understanding of the dataset, encompassing 70% of its variance, was achieved by considering a larger set of principal components. Specifically, it was determined that a set of 26 principal components is required to explain 70% of the data. This insight highlights the trade-off between dimensionality reduction and information gain.

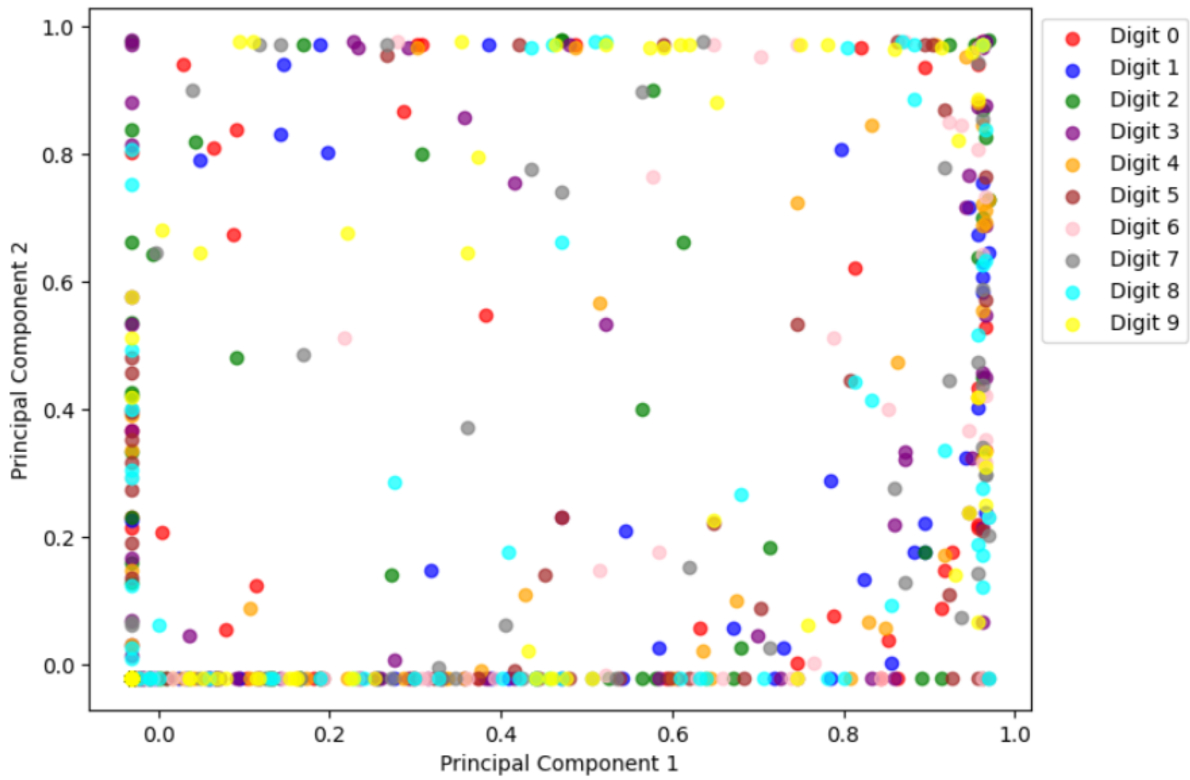
Q-1.3)

The visualization of the first 10 principal components resulting from PCA offers insights into the essential features and structures within the dataset. After reshaping and min-max scaling to ensure values fall within the [0, 1] range, the grayscale images reveal distinct patterns. The first component likely captures the primary source of variance, with subsequent components capturing diminishing amounts. These patterns may represent specific features or textures in the original data. The visualization serves as a qualitative assessment of PCA's effectiveness in summarizing dataset variability and provides a foundation for further analysis, facilitating dimensionality reduction and the identification of key data features.



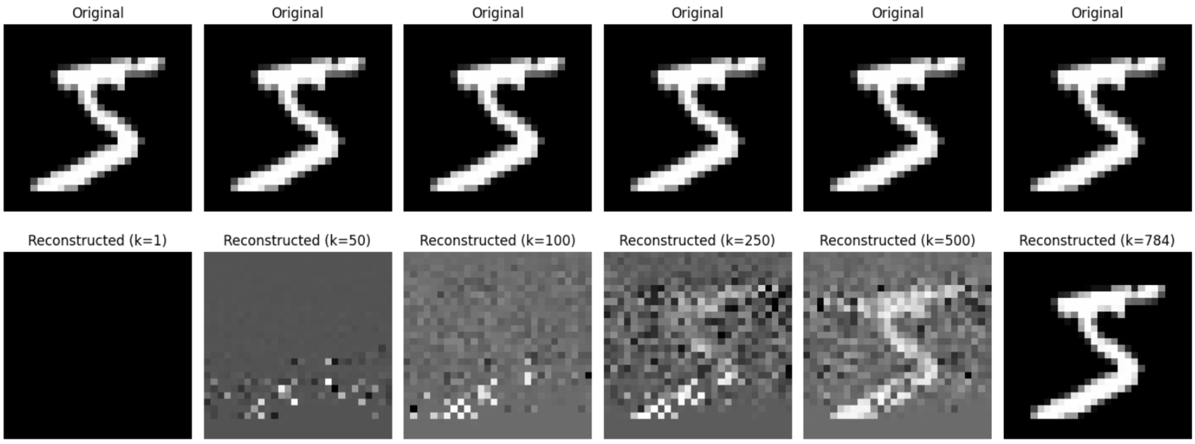
Q-1.4)

The plot is following:



Q-1.5)

Exploring different values of "k" in PCA-based image reconstruction highlights the trade-off between compression and precision. Smaller values of "k," like 1, result in highly compressed but somewhat blurred reconstructions. As "k" increases, fidelity improves, with a balanced representation achieved around "k = 250," offering substantial detail while maintaining computational efficiency. However, further increases in "k" lead to nearly perfect reconstructions, approaching the total number of components (k = 784), at the expense of losing the benefits of dimensionality reduction. The choice of the optimal "k" depends on careful consideration of application requirements, balancing storage and processing efficiency against the desire for high-fidelity reconstructions.



2) Logistic Regression

Q-2.1)

The initial experiment involves training the Multinomial Logistic Regression Classifier with default hyperparameters, as outlined in the prompt. The model is trained on the new training set of 50,000 images for 100 epochs, utilizing Softmax activation function, and updating weights based on the cross-entropy loss with L2 regularization. An epsilon constant(1e-10) has been utilized in the log function, to avoid log(0)'s. Important thing is in that stage the test data is used for the calculation of accuracy and performance because, in this stage, instead of fine tuning, a model is trained and tested.

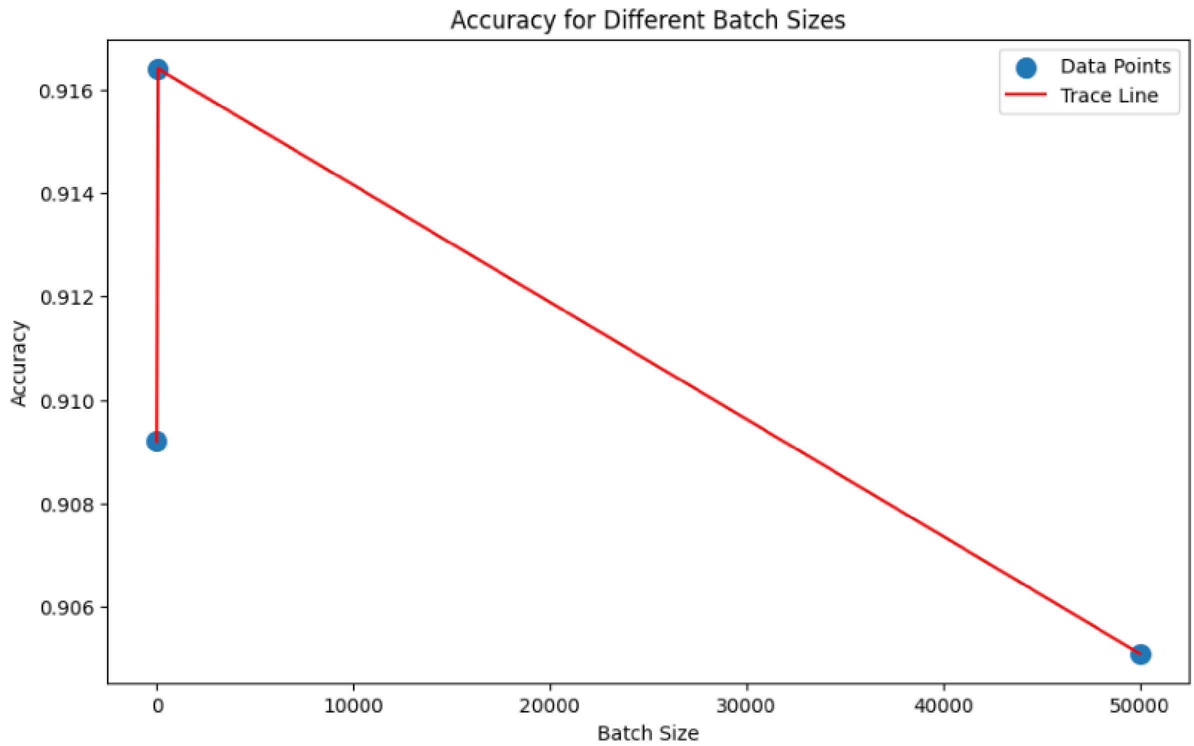
To assess the Multinomial Logistic Regression Classifier's performance, a confusion matrix is generated by comparing model predictions with true labels on the test dataset. The matrix reveals correct and misclassified instances for each class. Simultaneously, the accuracy test score is computed as the ratio of correct predictions to the total number of samples, offering a concise measure of overall model performance.

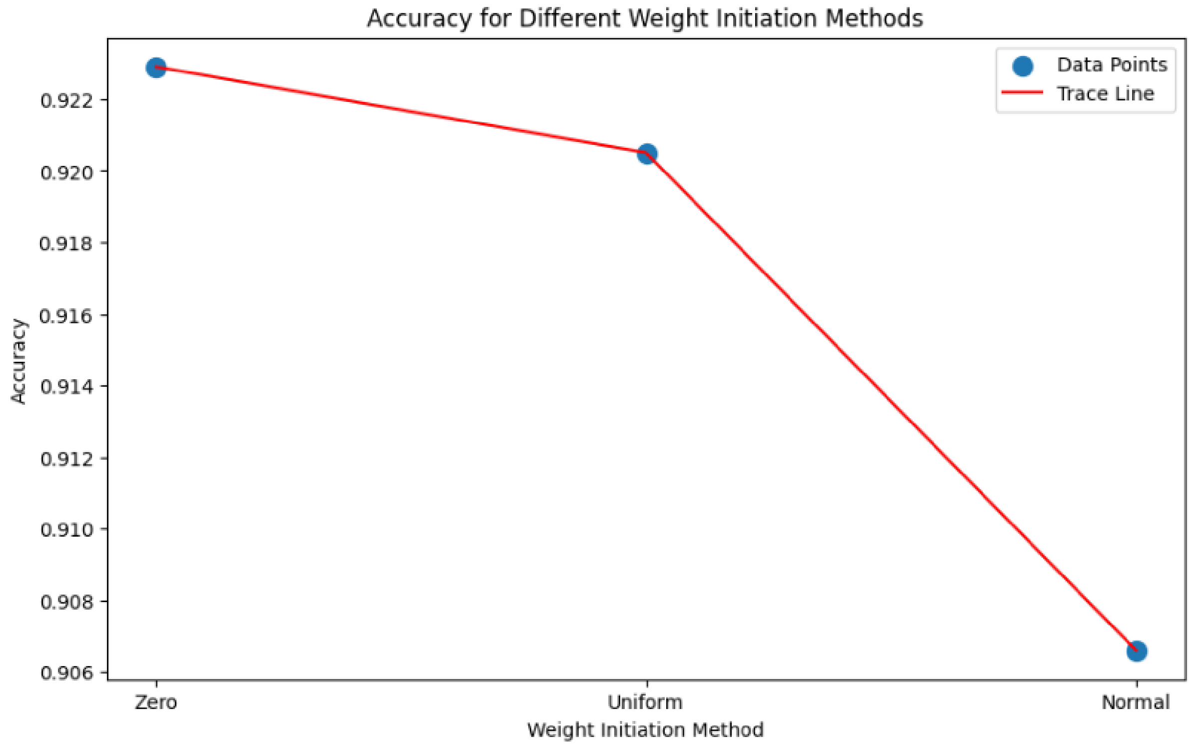
```
Confusion Matrix:
[952, 0, 4, 3, 0, 12, 19, 1, 7, 9]
[0, 1100, 12, 0, 2, 2, 3, 8, 10, 8]
[2, 7, 904, 22, 4, 3, 8, 23, 6, 1]
[2, 4, 21, 897, 6, 40, 1, 8, 21, 7]
[2, 0, 12, 1, 885, 12, 13, 7, 11, 31]
[7, 1, 7, 35, 3, 756, 22, 1, 34, 11]
[6, 4, 11, 3, 14, 18, 887, 0, 11, 0]
[3, 3, 15, 11, 4, 8, 0, 933, 10, 22]
[3, 16, 40, 29, 8, 33, 5, 10, 848, 13]
[3, 0, 6, 9, 56, 8, 0, 37, 16, 907]
Accuracy: 0.9069
```

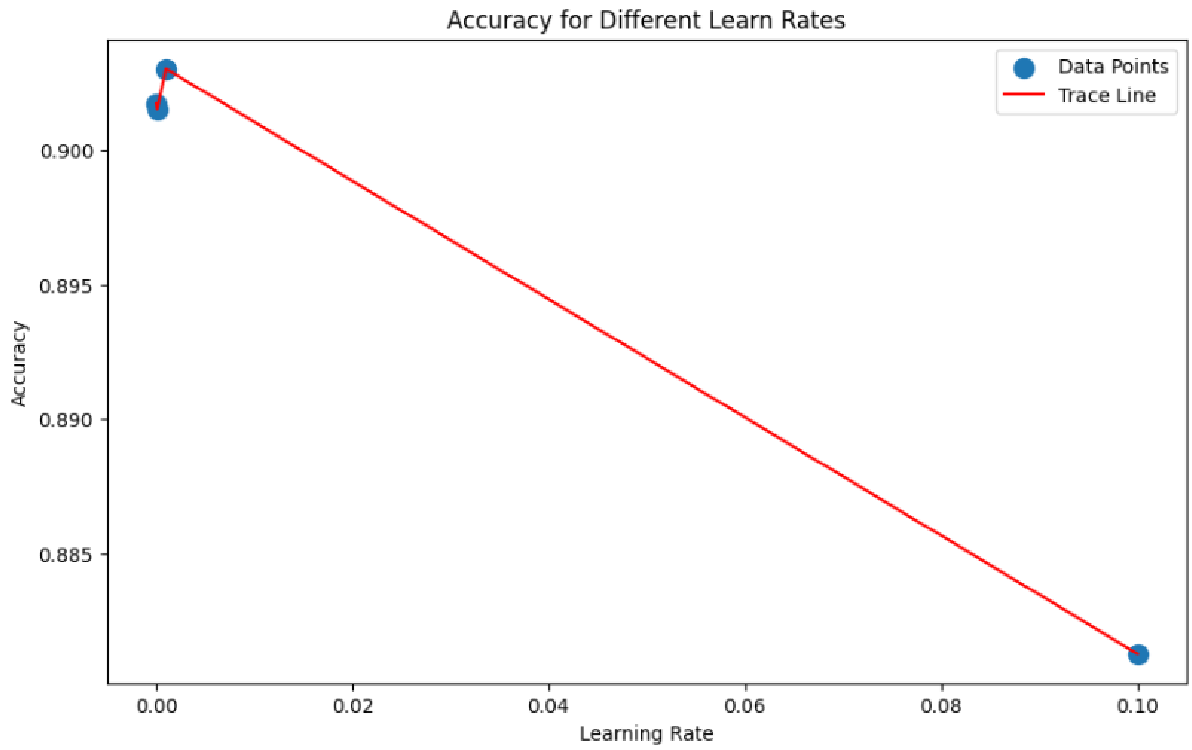
Q-2.2)

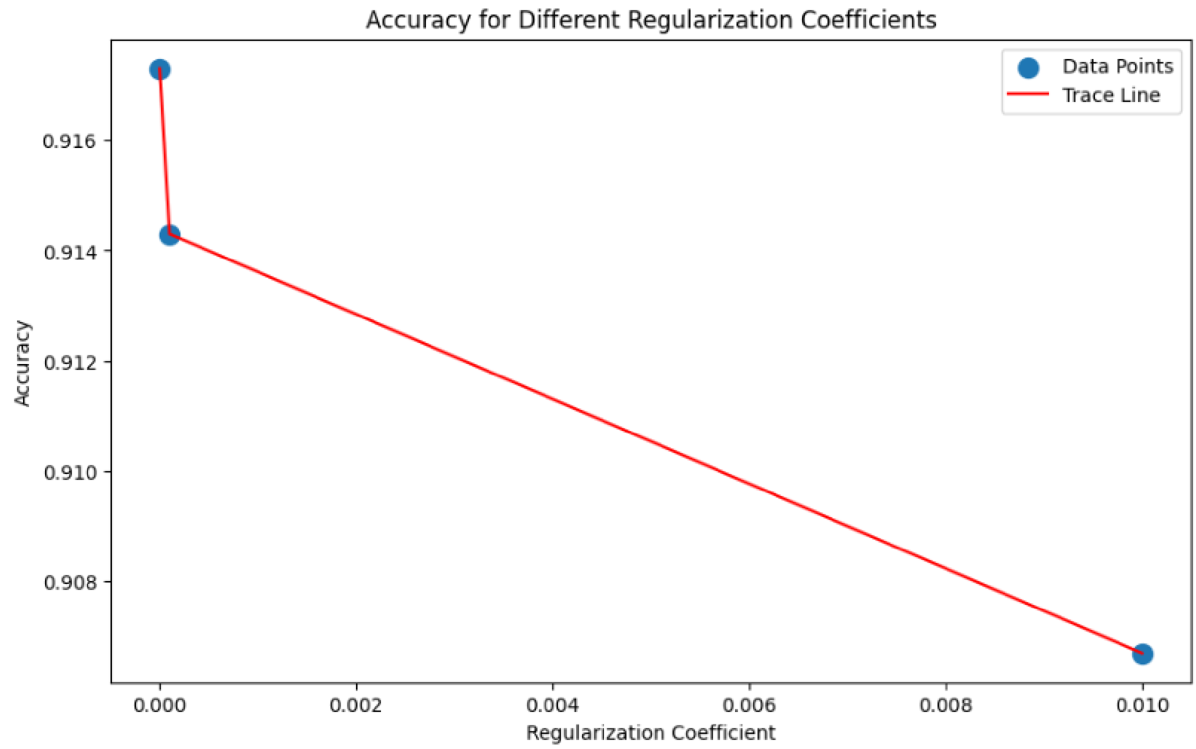
In this section of the experiment, a thorough exploration of the Multinomial Logistic Regression Classifier's sensitivity to key hyperparameters is conducted. The specified hyperparameters, namely batch size, weight initialization technique, learning rate, and regularization coefficient (λ), are individually varied while keeping the others at their default values. The batch sizes of 1, 64, and

50000 are assessed for their impact on model performance, visually represented through accuracy graphs across epochs. Similarly, different weight initialization techniques, including zero initialization, uniform distribution, and normal distribution, are tested. The learning rates of 0.1, 10^(-3), 10^(-4), and 10^(-5) are individually applied to analyze their influence on convergence and accuracy. Furthermore, regularization coefficients of 10^(-2), 10^(-4), and 10^(-9) are explored to understand their role in eliminating overfitting. With the usage of validation data set, each different combination of hyperparameters are evaluated. Enabling the identification of optimal hyperparameter configurations for the Multinomial Logistic Regression Classifier. The accuracy graphs with legends provide a visual representation of model performance across different hyperparameter values. Then, the model which has the best accuracy is determined as the main model.




Accuracy for Different Weight Initiation Methods
Data Points
Trace Line
0.922
0.920
0.918
0.916
0.914
0.912
0.910
0.908
0.906
Accuracy
Zero
Uniform
Normal
Weight Initiation Method


Accuracy for Different Learn Rates
Data Points
Trace Line
0.900
0.895
0.890
0.885
Accuracy
0.00
0.02
0.04
0.06
0.08
0.10
Learning Rate



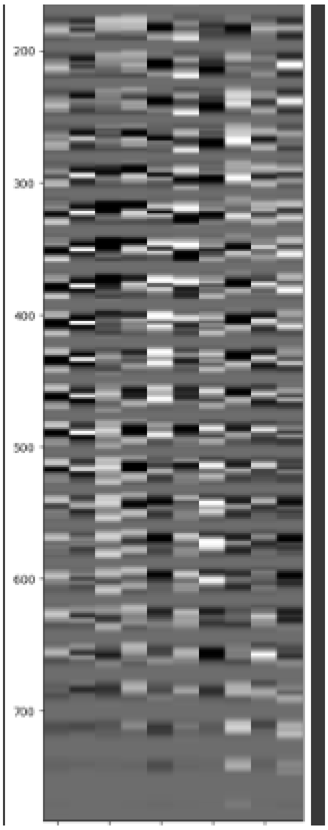
Q-2.3)

Following the optimization of hyperparameters through 13 combinations, the best Multinomial Logistic Regression model is tested using the test dataset. The accuracy score demonstrates overall correctness, while the confusion matrix provides a detailed portrait of predictions. This final evaluation highlights the model's effectiveness in real-world scenarios and identifies areas for potential improvement.

```
Confusion Matrix for best performing model:
[955, 0, 7, 3, 0, 11, 14, 1, 6, 9]
[0, 1105, 11, 0, 1, 1, 3, 7, 10, 8]
[2, 4, 915, 21, 7, 3, 4, 21, 6, 1]
[1, 3, 17, 919, 3, 36, 2, 7, 22, 8]
[3, 1, 9, 0, 908, 13, 8, 6, 8, 24]
[8, 1, 5, 27, 1, 771, 23, 0, 32, 7]
[5, 4, 12, 3, 12, 13, 899, 0, 11, 0]
[4, 2, 11, 10, 3, 6, 2, 947, 8, 16]
[1, 15, 41, 21, 8, 32, 3, 6, 858, 10]
[1, 0, 4, 6, 39, 6, 0, 33, 13, 926]
Accuracy for best performing model: 0.9203
```

Q-2.4)

After training the best Multinomial Logistic Regression model, the ten weight vectors corresponding to different digit labels are displayed as images using a provided script. It's noted that the images might appear a bit blurry. This blurriness could suggest that the model has learned common features shared among different digit classes during training. The visual representation of weight vectors offers insights into how the model interprets and extracts features, providing a simpler way to understand the learned patterns.



Q-2.5)

Using the best model, we calculate precision, recall, F1 score, and F2 score for each digit class. These metrics help us evaluate how well the model is making accurate predictions and capturing all relevant instances. The results are then analyzed alongside the confusion matrix and weight images, giving insights into the model's performance on different digits and the features it finds important for classification. This allows us to understand where the model excels and areas where it could be improved.

```
Precision: [0.9744898  0.97356828 0.88662791 0.90990099 0.92464358 0.86434978
 0.93841336 0.92120623 0.88090349 0.91774034]
Recall: [0.94930417 0.96422339 0.92987805 0.90275049 0.92653061 0.88114286
 0.93743483 0.93855302 0.86231156 0.90077821]
F1 Score: [0.96173212 0.9688733  0.90773809 0.90631164 0.92558614 0.87266553
 0.93792384 0.92979872 0.87150838 0.90918017]
F2 Score: [0.95423661 0.96607799 0.92089372 0.90417159 0.92615259 0.87773224
 0.93763037 0.9350316  0.8659669  0.90412029]
Accuracy: 0.9203
```