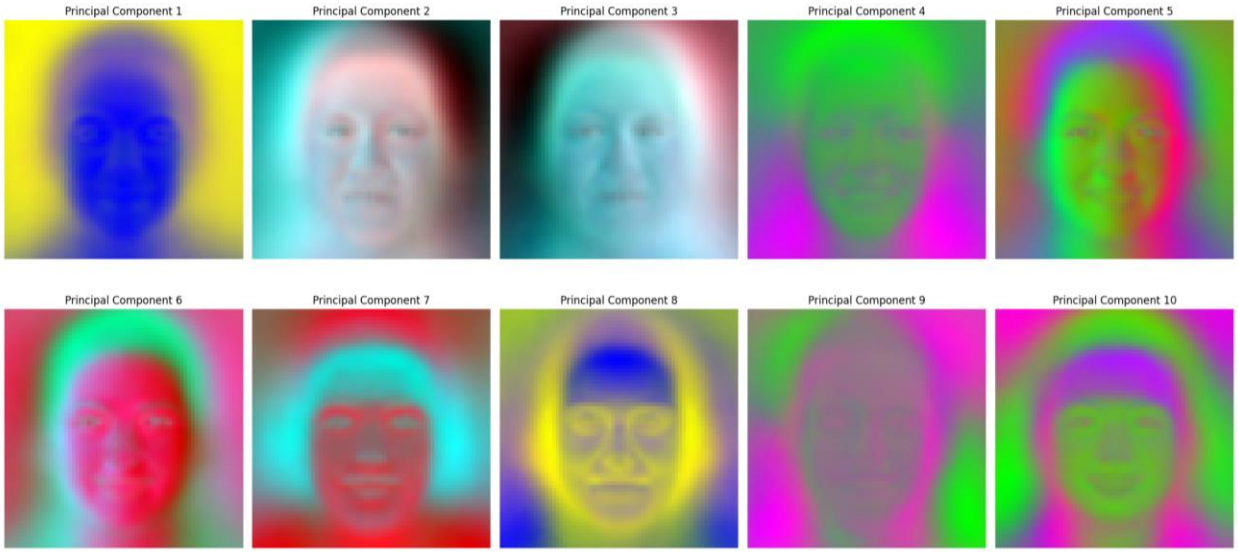
CS 464 Homework 2 - Görkem Kadir Solun – 22003214

PCA Question 1.1

```
Red Channel
For the first 10 components:
PVE values: [0.28929081 0.09373798 0.06788795 0.05859219 0.05419499 0.04384446
 0.02887794 0.02049209 0.01681582 0.01630701]
Sum of PVE values: 0.6900412482040855
Minimum component count to reach the threshold 0.7: 11
Green Channel
For the first 10 components:
PVE values: [0.32041101 0.08559319 0.08177433 0.05796603 0.04110711 0.03958217
 0.02568433 0.01855605 0.01663551 0.01604563]
Sum of PVE values: 0.70335535380908
Minimum component count to reach the threshold 0.7: 10
Blue Channel
For the first 10 components:
PVE values: [0.34357938 0.08854058 0.08477065 0.05722818 0.03805416 0.03354398
 0.02486367 0.01717688 0.01624892 0.01553017]
Sum of PVE values: 0.71953657404184
Minimum component count to reach the threshold 0.7: 9
```

The blue channel, with a cumulative proportion of variance explained (PVE) of 71.9537% achieved by the first 10 components, surpasses the 70% threshold, requiring only 9 components to meet the criterion, making it the most efficient channel due to its higher variance. Similarly, the green channel explains 70.3356% of the variance with 10 components, just exceeding the threshold, indicating that 10 components are sufficient for this channel. In contrast, the red channel achieves a cumulative PVE of 69.0041% with 10 components, falling short of the 70% target, requiring 11 components to exceed the threshold, which makes it the least efficient channel. Thus, the red channel's requirement of 11 components establishes the minimum number needed to ensure that at least 70% variance is explained across all channels.

PCA Question 1.2

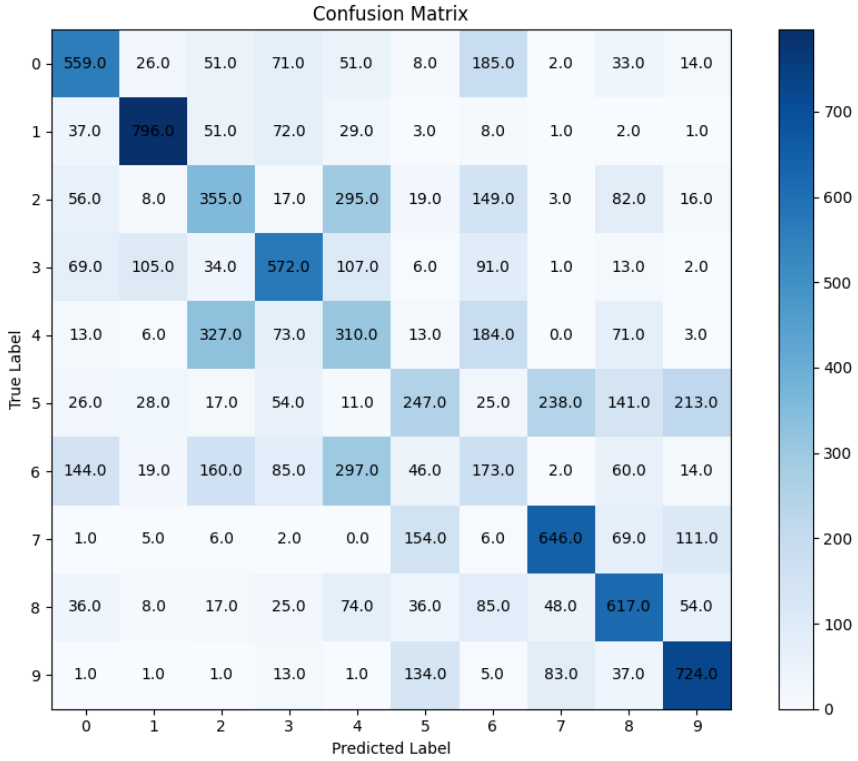


Every principal component (PC) captures a maximum variance direction, emphasizing the dataset's critical features. The predominant color patterns and most notable global variations are represented by PC 1, which mainly shows subtle color intensity changes, such as gradients from yellow to blue. We can conclude that PC 8 is the opposite of PC 1. Sharper contrasts and complex spatial patterns, like variations in particular regions like facial characteristics, are highlighted in PCs 2 and 3. PCs 4, 5, 9, and 10 emphasize intricate color transitions with less emphasis on general structure, concentrating on more delicate and localized patterns, such as nose or hair texture. In addition to representing hair and the bottom portion of the face, PCs 6 and 7 feature intriguing colors. By preventing rendering distortions from raw values exceeding acceptable ranges, min-max scaling normalizes eigenvector values to a range between 0 and 1, guaranteeing proper RGB image display and thereby enhancing the clarity and comparability of color patterns among principal components. The normalized red, green, and blue components are organized to create meaningful RGB graphics, illustrating how variances in the dataset are distributed across the spectrum and visualizing relationships among color channels, with red-green and blue-yellow contrasts highlighting key color changes driving the variability.

PCA Question 1.3

Reconstruction using PCA involves projecting an image onto its principal components, eigenvectors, and taking the first k eigenvectors. These eigenvectors correspond to the directions of maximum variance in the dataset, with earlier components carrying more significant information as they are associated with higher eigenvalues. The choice of k, the number of principal components, controls the level of detail in the reconstructed image. A smaller k captures broader patterns, while a larger k incorporates finer details, achieving complete reconstruction when k equals the total number of original components, illustrating how PCA efficiently reduces dimensionality while preserving essential information.

Logistic Regression Question 2.1




```
Epoch: 95, Validation Accuracy: 0.4953
Epoch: 96, Validation Accuracy: 0.4968
Epoch: 97, Validation Accuracy: 0.4981
Epoch: 98, Validation Accuracy: 0.5002
Epoch: 99, Validation Accuracy: 0.5015
Epoch: 100, Validation Accuracy: 0.5026
Test Accuracy: 0.4999
```

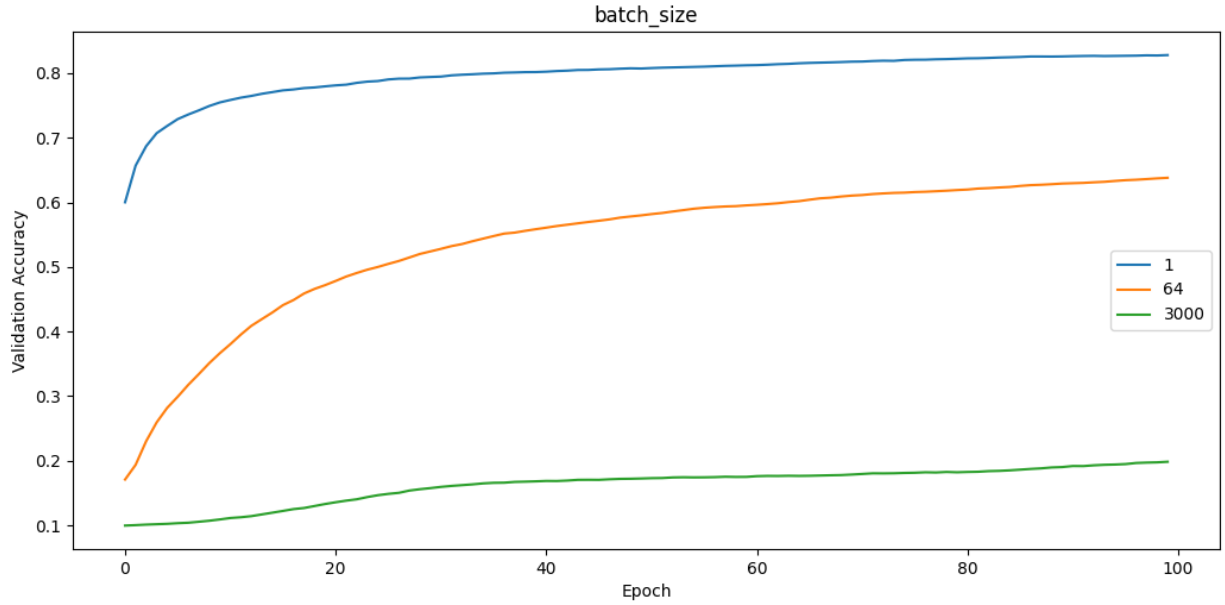
In a classification task with ten classes, the logistic regression model obtained a test accuracy of 49.99%. This suggests a mediocre performance overall, but the model consistently struggles to produce precise predictions in every class. Nevertheless, there are also significant variations in prediction accuracy among various classes.

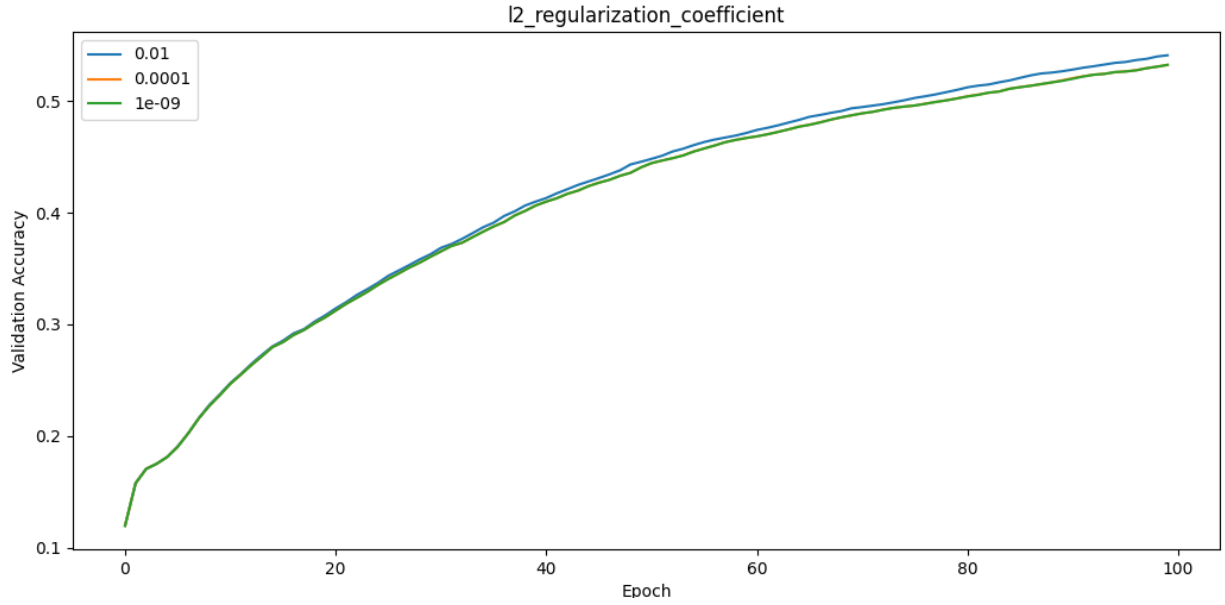
According to the confusion matrix, classes 3, 8, 7, and 9 are predicted more accurately, as evidenced by their diagonal solid values in the matrix. On the other hand, classes 2, 4, 5, and 6 demonstrate a need for greater accuracy, suggesting that the model struggles to differentiate between classes with similar characteristics.

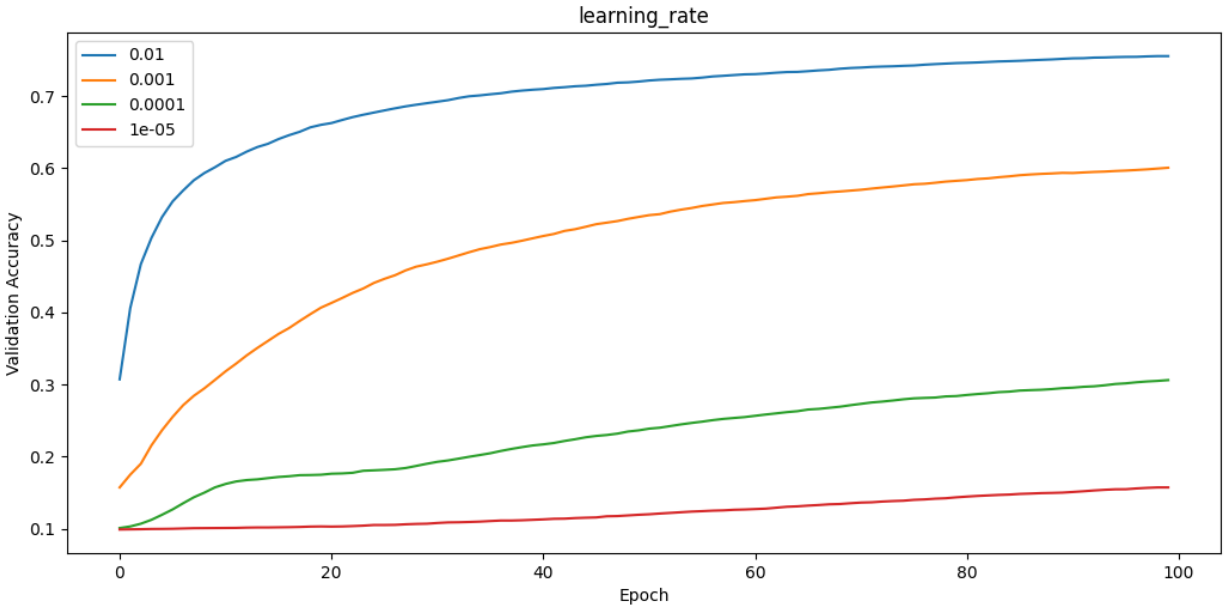
The model's configuration settings, including a batch size of 200, regularization of 1e-4, and a learning rate of 0.0005, enable steady improvement in the validation dataset with each training epoch, ultimately leading to convergence. However, the model's generalization capacity may need to be improved by the low regularization strength, which might not be enough to stop overfitting, especially when working with complicated datasets.

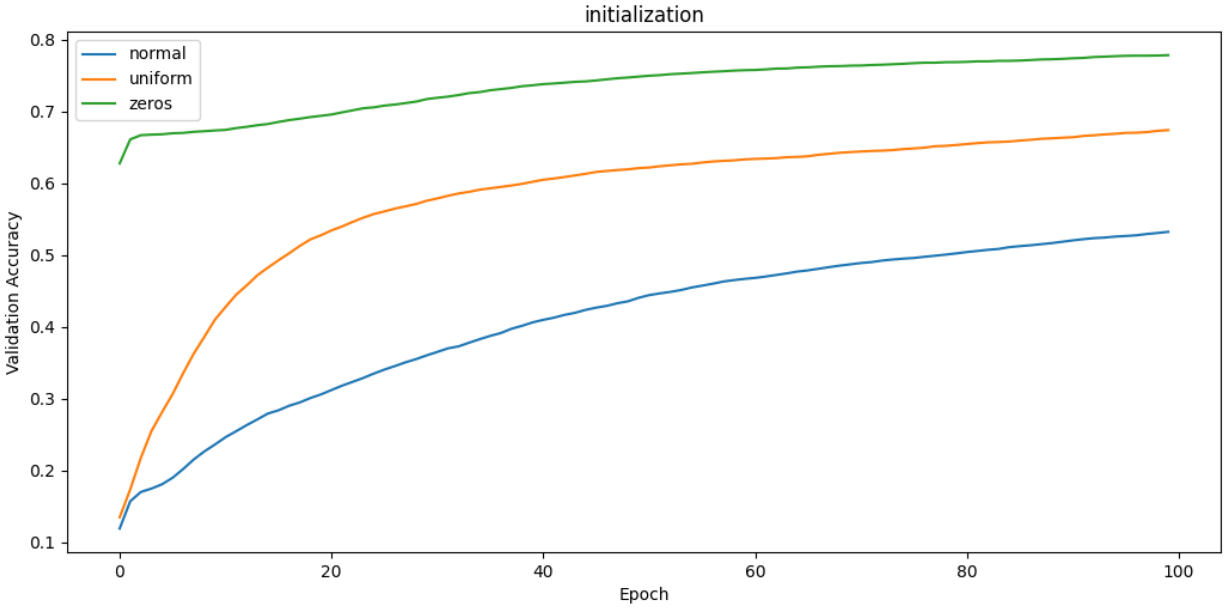
The model uses the conventional method of Gaussian normal initialization for its weights. Alternative weight initialization techniques, however, might produce better optimization results, according to the experiments mentioned in question 2.2. This suggests that model initialization techniques may need to be improved to increase overall performance.

Logistic Regression Question 2.2










```
Epoch: 100, Validation Accuracy: 0.7784
Best Parameters:
batch_size: 1
learning_rate: 0.01
l2_regularization_coefficient: 0.01
initialization: zeros
```

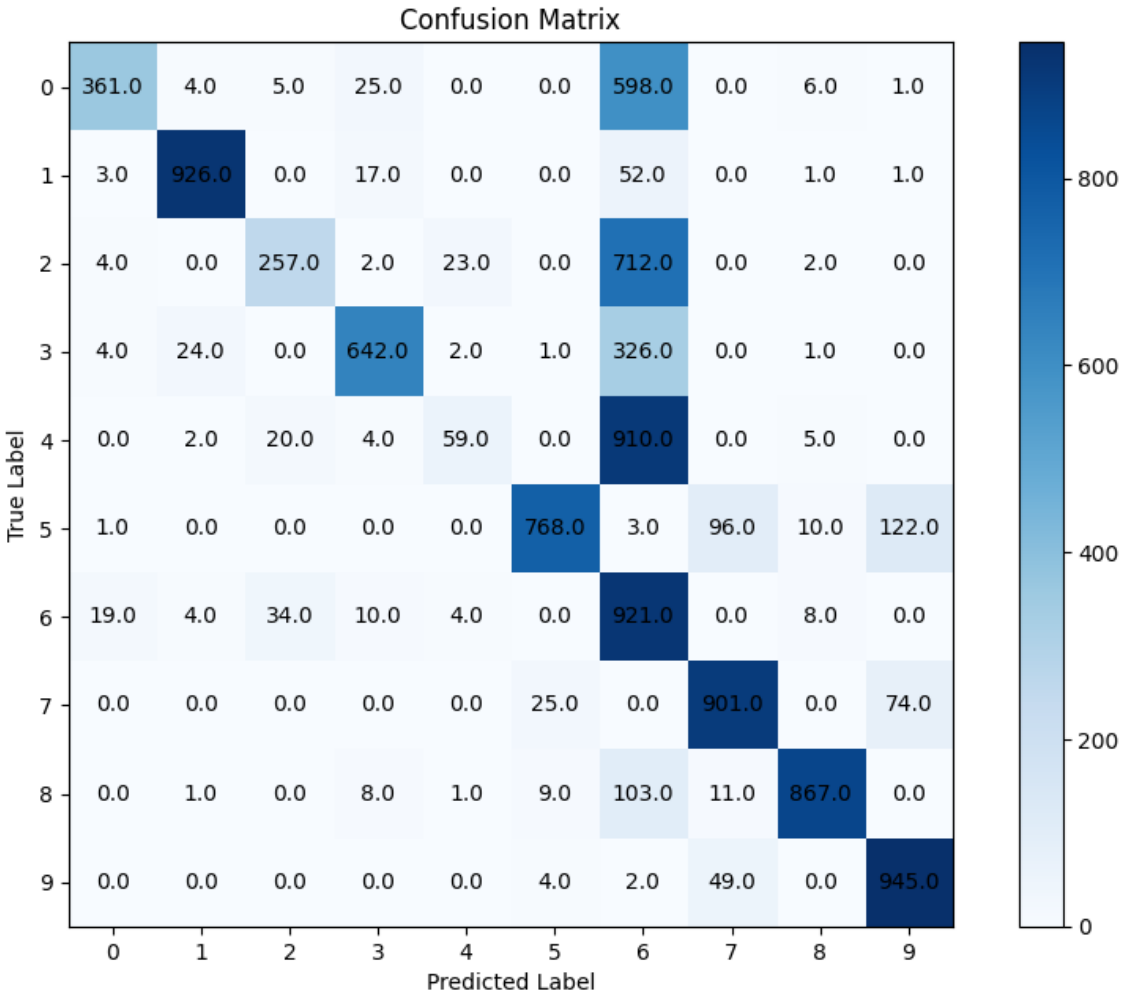
The first is how changes in batch size affect model accuracy. Smaller batch sizes yield higher accuracy (about 80%) and a faster convergence rate, primarily when the batch size is 1. The model can learn more complex patterns given the default hyperparameters since smaller batches enable more frequent weight updates. Nonetheless, batch size 64 shows respectable accuracy and convergence and stabilizes at roughly 65%. The learning is not good, and only 25% accuracy is achieved in a batch of 3000. Thus, huge batch sizes may hinder the model's ability to capture the dataset's features and diminish its generalization capability.

The second point examines the model's accuracy concerning various weight initialization techniques. Zeros, uniform distribution, and Gaussian distribution are examples of initialization techniques. The findings demonstrate that, within the first epochs, the uniform initialization performs better than the others in terms of accuracy. Although it does so gradually, the Gaussian initialization converges more slowly. Notably, zero initiation produces the highest final accuracy after the epochs, at roughly 75%. This implies that zero initialization is the best way to stabilize this strategy. These results demonstrate the importance of using an appropriate starting technique to balance final performance and convergence speed.

The third point shows the impact of various learning rates on the model's accuracy: 0.01, 1e-3, 1e-4, and 1e-5. The findings show that a higher learning rate of 0.01 produces the highest final accuracy of roughly 75% and accelerates convergence. A learning rate of 1e-3 results in poorer ultimate accuracy and slower convergence. The lowest learning rate (1e-5) performs the worst, and smaller learning rates lead to slower learning and poorer end performance. Accordingly, more excellent learning rates must be carefully chosen to prevent problems like overshooting minima or creating instability in the training process, even though they might speed up learning and increase ultimate accuracy.

The fourth point shows how different regularization coefficients influence accuracy. Interestingly, the differences between the tested regularization values—0.01, 1e-04, and 1e-5—are negligible, with all configurations leading to similar accuracy levels of around 50%. This demonstrates that the model's performance in this scenario is relatively insensitive to the regularization parameter within this range. Despite the minimal impact observed, regularization still plays a crucial role in preventing overfitting by penalizing large weights, which helps the model generalize better to unseen data.

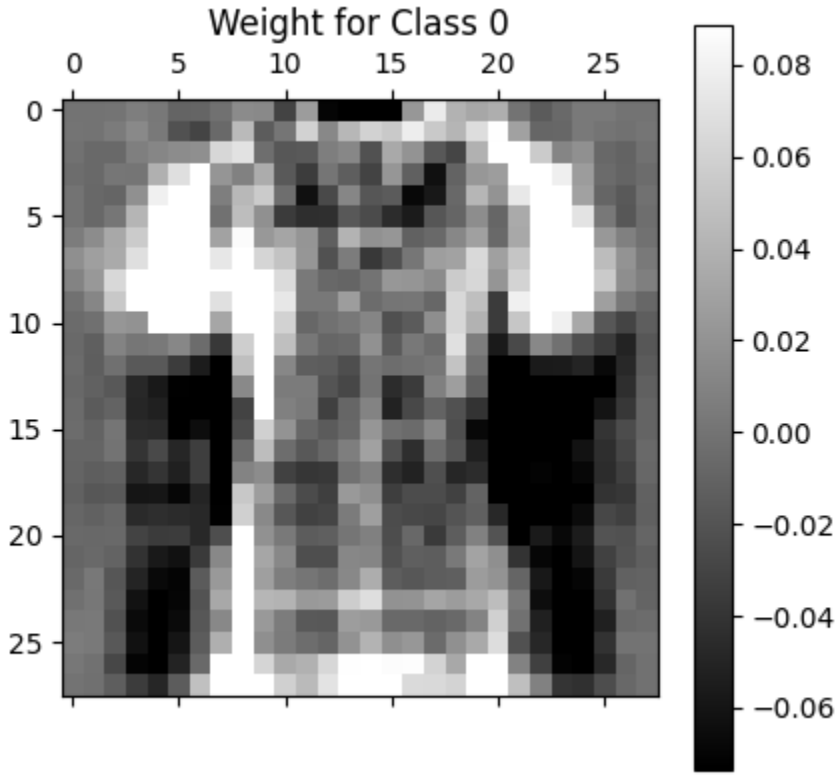
Logistic Regression Question 2.3

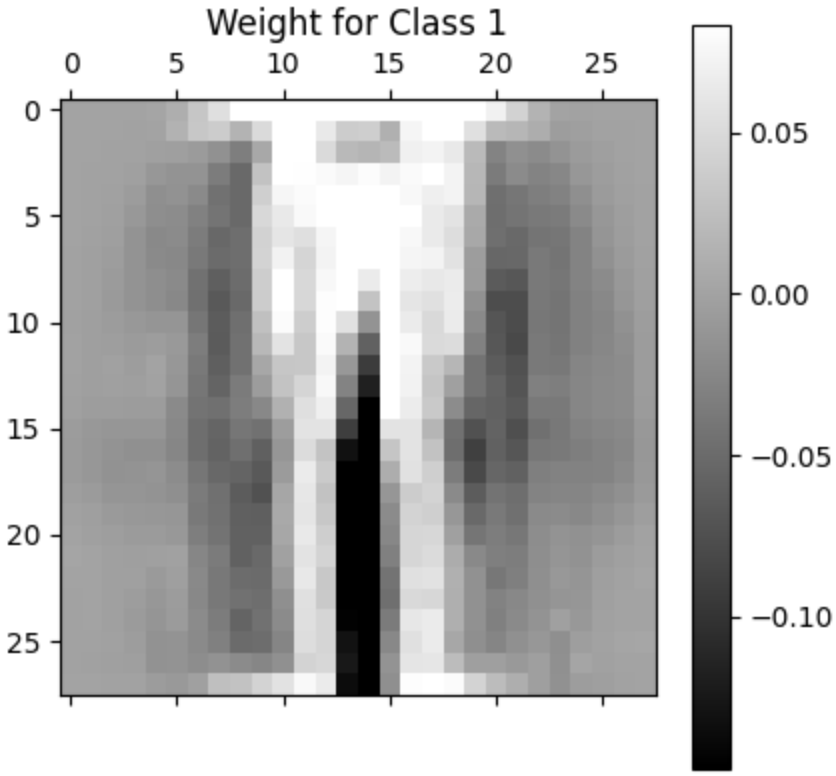


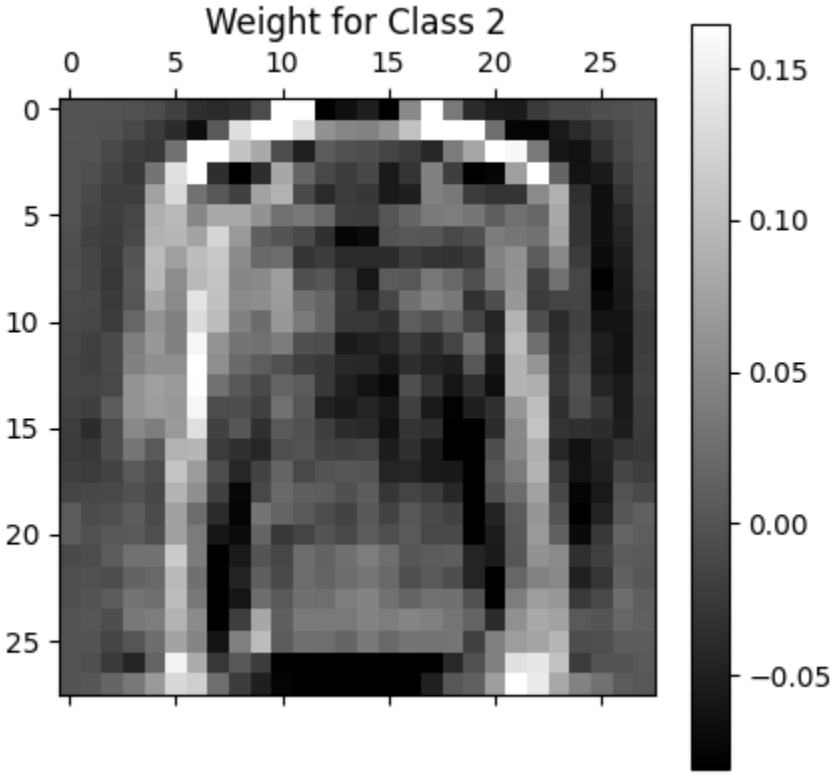

```
Epoch: 96, Validation Accuracy: 0.6739
Epoch: 97, Validation Accuracy: 0.6739
Epoch: 98, Validation Accuracy: 0.6739
Epoch: 99, Validation Accuracy: 0.6739
Epoch: 100, Validation Accuracy: 0.6739
Test Accuracy: 0.6647
```

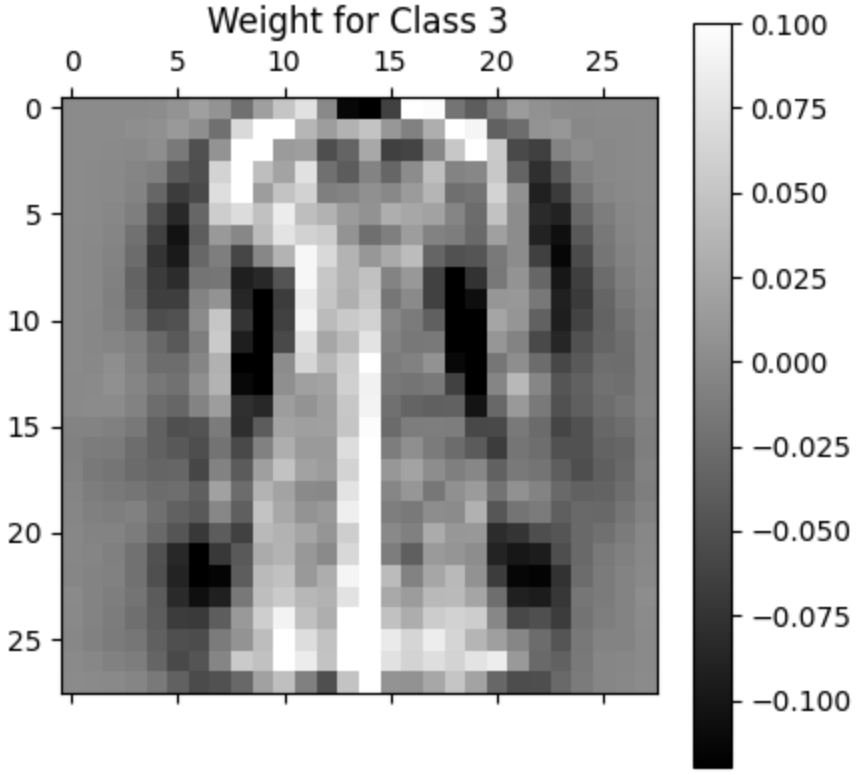
Interestingly, combining the best parameters results in a slightly reduced performance of 66%, while the maximum of the validation accuracies of the parameters tested is around 75%.

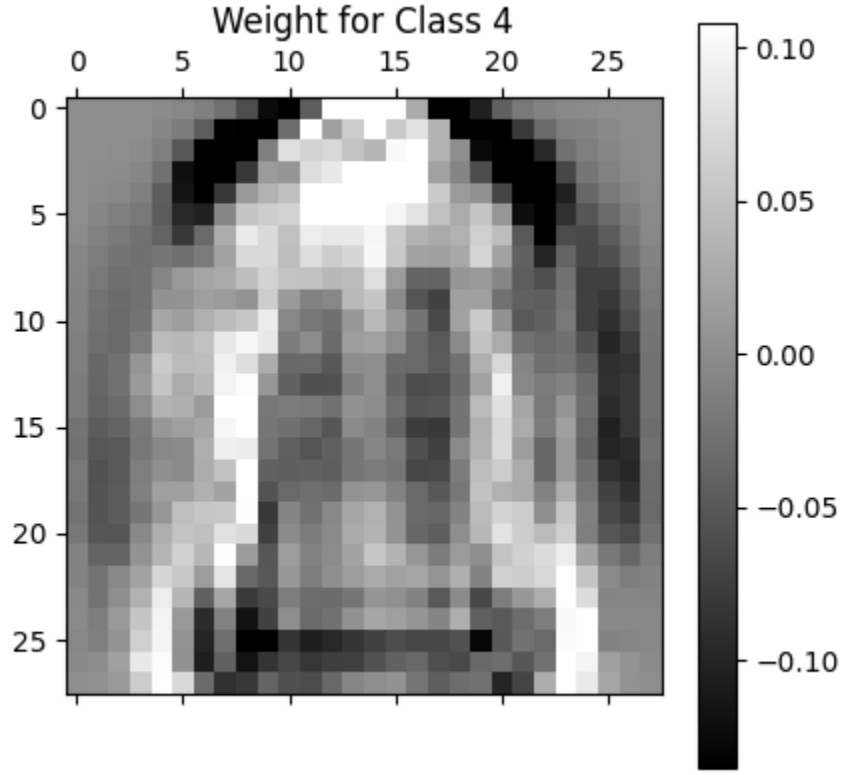
Logistic Regression Question 2.4

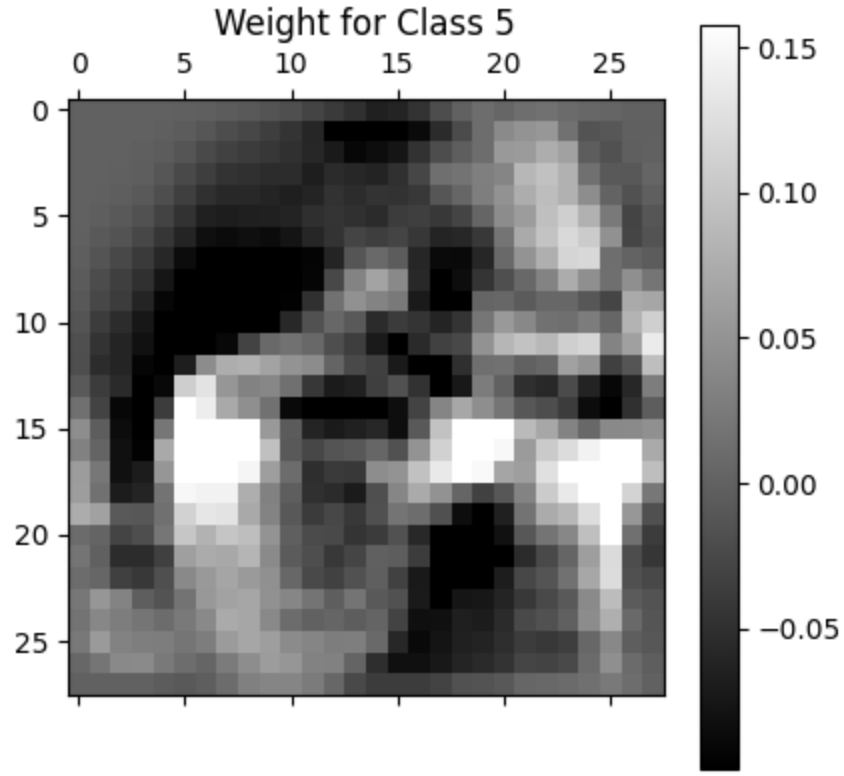


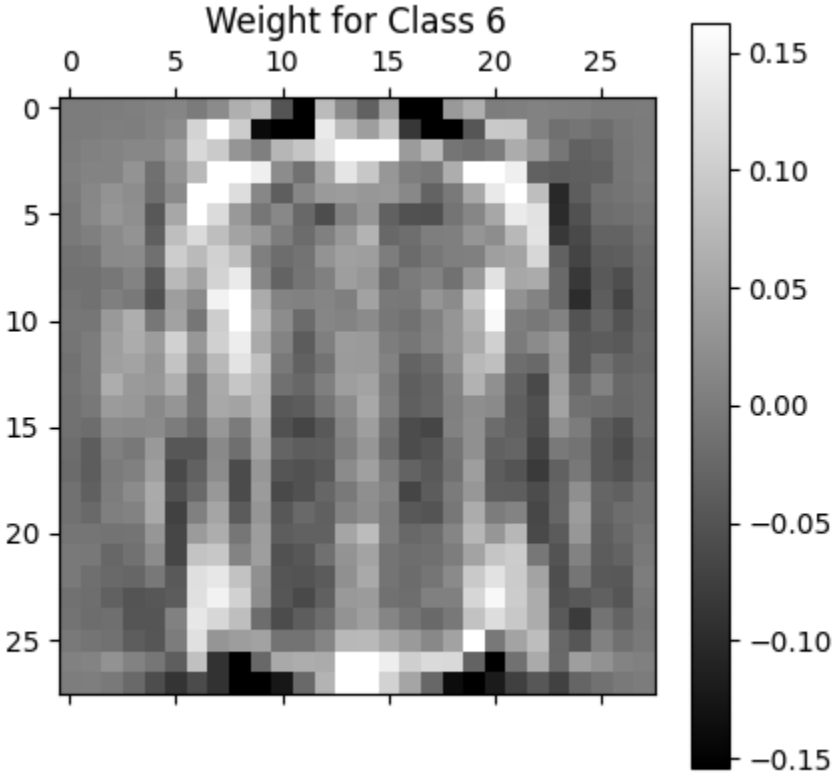

Weight for Class 1
0
5
10
15
20
25
0.05
0.00
−0.05
−0.10

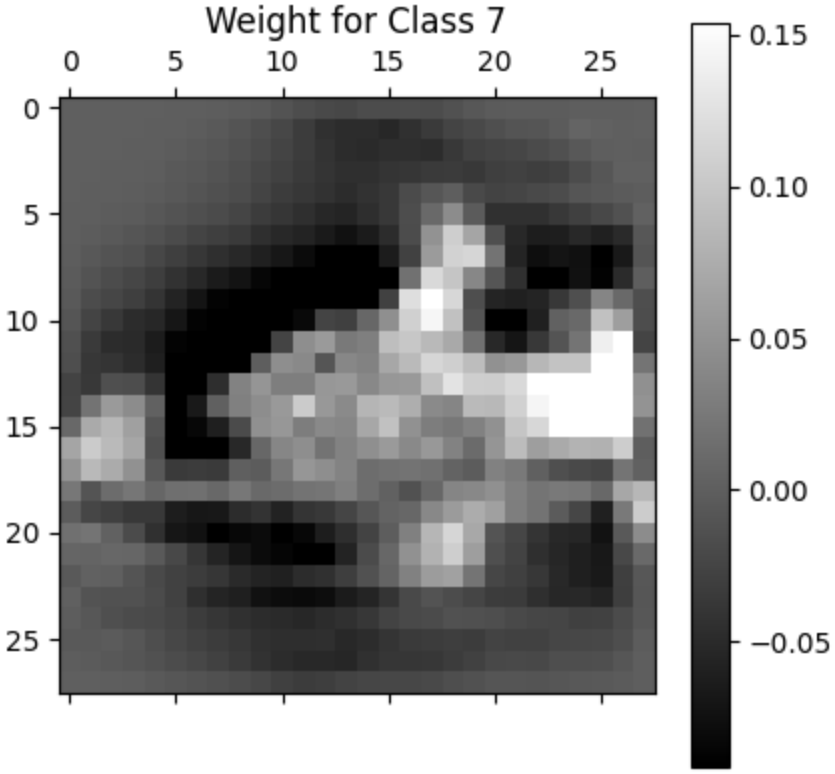

Weight for Class 2
0
5
10
15
20
25
0.15
0.10
0.05
0.00
−0.05

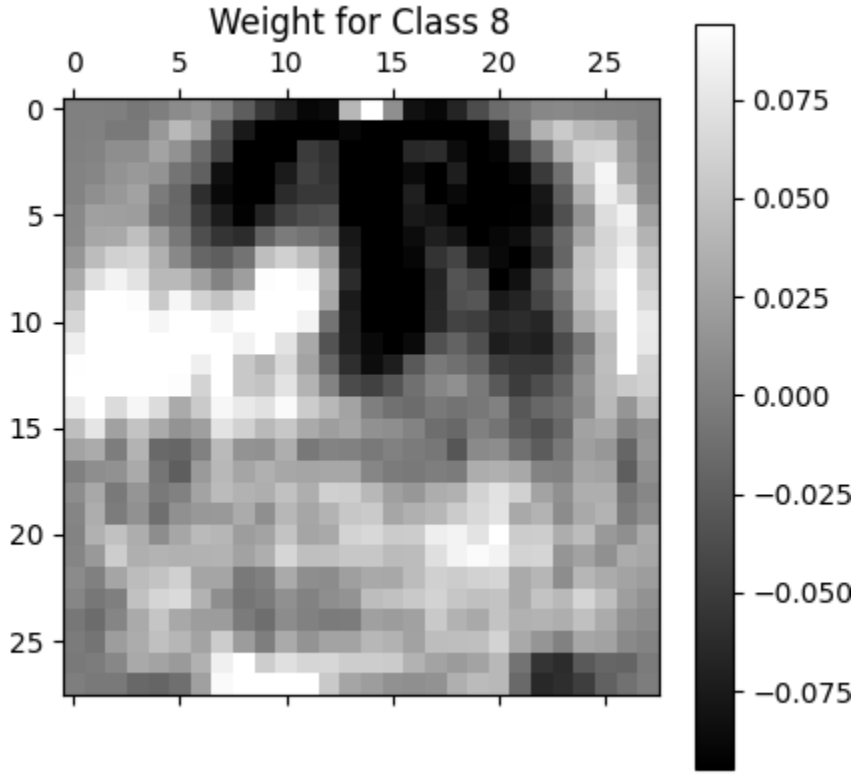

Weight for Class 3
0
5
10
15
20
25
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100

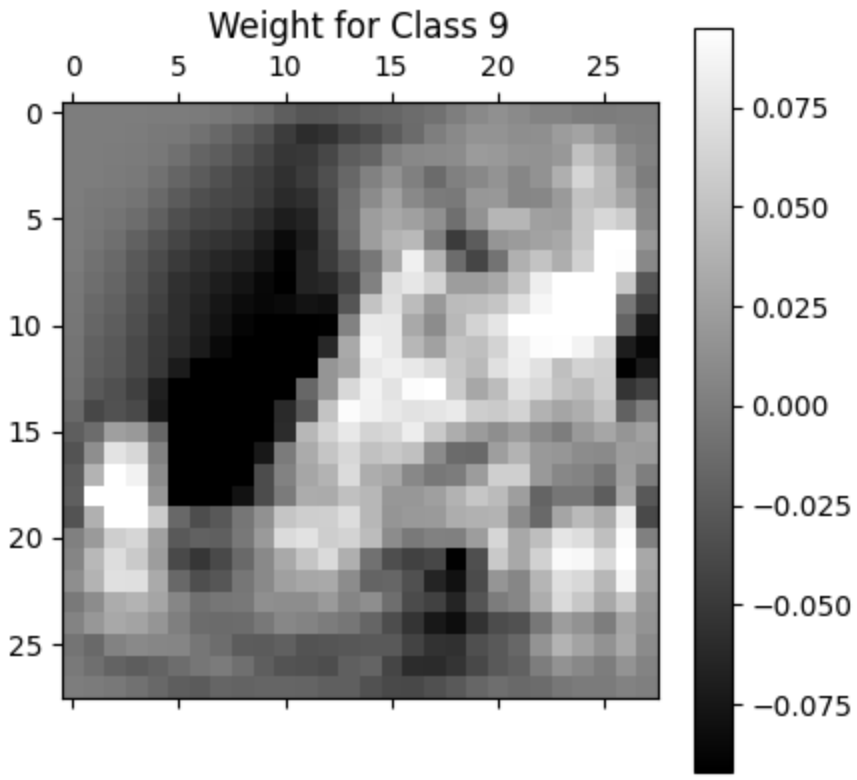

Weight for Class 4
0
5
10
15
20
25
0.10
0.05
0.00
−0.05
−0.10


Weight for Class 5
0
5
10
15
20
25
0.15
0.10
0.05
0.00
−0.05


Weight for Class 6
0
5
10
15
20
25
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15


Weight for Class 7
0
5
10
15
20
25
0.15
0.10
0.05
0.00
−0.05


Weight for Class 8
0
5
10
15
20
25
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075



Each weight map visualization highlights the traits necessary for differentiating one class from another, helping the model make classification judgments by capturing each class's distinctive qualities. The color bars indicate that various classes have different weight values, potentially reflecting the varying complexity or difficulty in categorizing those classes. For instance, certain classes might have more distinctive traits than others, while others might share qualities that make them more challenging to differentiate. Every class is associated with a specific category, such as shoes, t-shirts, or pants. The weight maps highlight the significance of regions for categorization and assist in visualizing the model's emphasis on the salient characteristics that set these categories apart. It is expected that the weight maps may appear hazy. By punishing abnormally high weight values, L2 regularization helps avoid overfitting and improves the model's ability to generalize to new data, which may have caused this blur in the image. The classification of that class is greatly aided by bright regions, which correlate more with input features. Darker areas, on the other hand, indicate negative weights, emphasizing characteristics that actively lower the input's chance of falling into that class. Grey areas indicate little impact on the class's classification. These regions stand for neutrality when the associated pixels don't strongly support or disagree with the classification choice.

Logistic Regression Question 2.5

```
Metrics for the Best Model:
   Precision  Recall  F1 Score  F2 Score
0   0.920918   0.361  0.518678  0.410974
1   0.963580   0.926  0.944416  0.933280
2   0.813291   0.257  0.390578  0.297729
3   0.906780   0.642  0.751756  0.681818
4   0.662921   0.059  0.108356  0.072145
5   0.951673   0.768  0.850028  0.798835
6   0.253929   0.921  0.398098  0.603776
7   0.852412   0.901  0.876033  0.890844
8   0.963333   0.867  0.912632  0.884694
9   0.826772   0.945  0.881941  0.918724
```

	Precision	Recall	F1 Score	F2 Score
0	0.920918	0.361	0.518678	0.410974
1	0.96358	0.926	0.944416	0.93328
2	0.813291	0.257	0.390578	0.297729
3	0.90678	0.642	0.751756	0.681818
4	0.662921	0.059	0.108356	0.072145
5	0.951673	0.768	0.850028	0.798835
6	0.253929	0.921	0.398098	0.603776
7	0.852412	0.901	0.876033	0.890844
8	0.963333	0.867	0.912632	0.884694
9	0.826772	0.945	0.881941	0.918724

After training with the optimal hyperparameters—a batch size of 1, zero-initialized weights, a learning rate of 0.01, and an L2 regularization parameter of 0.01—the model's performance was evaluated using a confusion matrix and achieved a test accuracy of 66.47% under these settings.

The confusion matrix reveals that the model performs exceptionally well in certain classes, such as 1, 7, and 9, which demonstrate high precision and recall, while other classes, like 4, show significant misclassification issues with lower precision and recall. Particularly challenging is class 6, where the model has varying precision (0.25) and recall (0.92) scores. Class 4 shows poor separability, with a precision of 0.66 and recall of 0.05, as many instances are misclassified into class 5, likely due to overlapping features between classes or insufficient representation of critical features in the dataset.

While these hyperparameter values performed best individually with other parameters set to their default values, the overall optimality of the model remains uncertain due to the unknown specifics of those default settings. While combining these hyperparameters resulted in the highest mean validation score, models with alternative hyperparameter settings achieved better accuracy than the initially considered ‘best’ hyperparameters.

Classes 1, 7, and 9 achieved high precision, recall, F1, and F2 scores, whereas the model struggled with classes 0, 2, 4, and 6, which showed low precision and recall, reflecting the difficulty in distinguishing these classes from others, as evidenced by the considerable misclassifications in the confusion matrix.