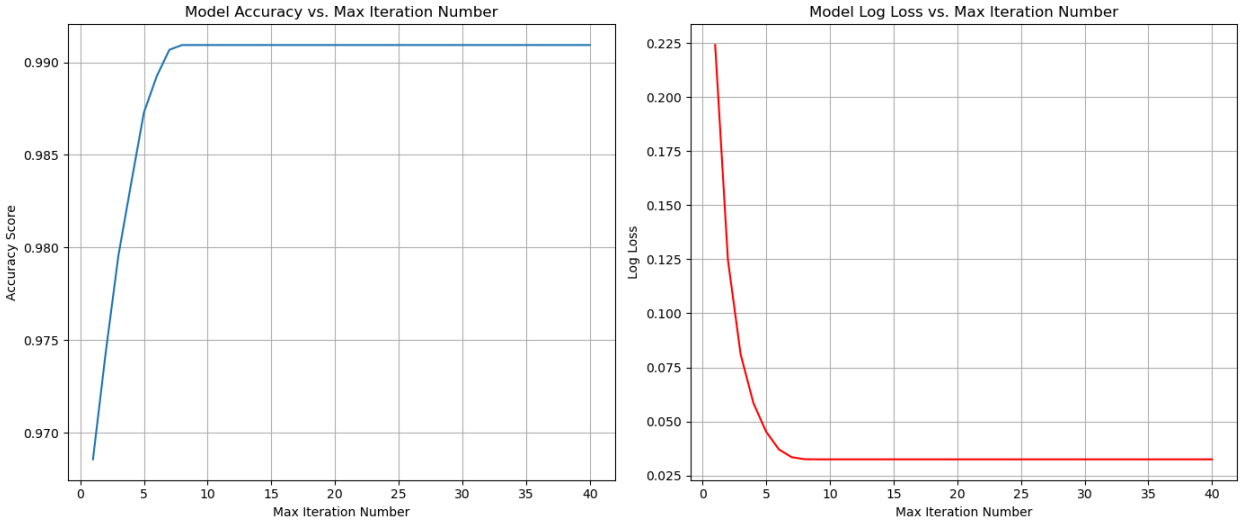
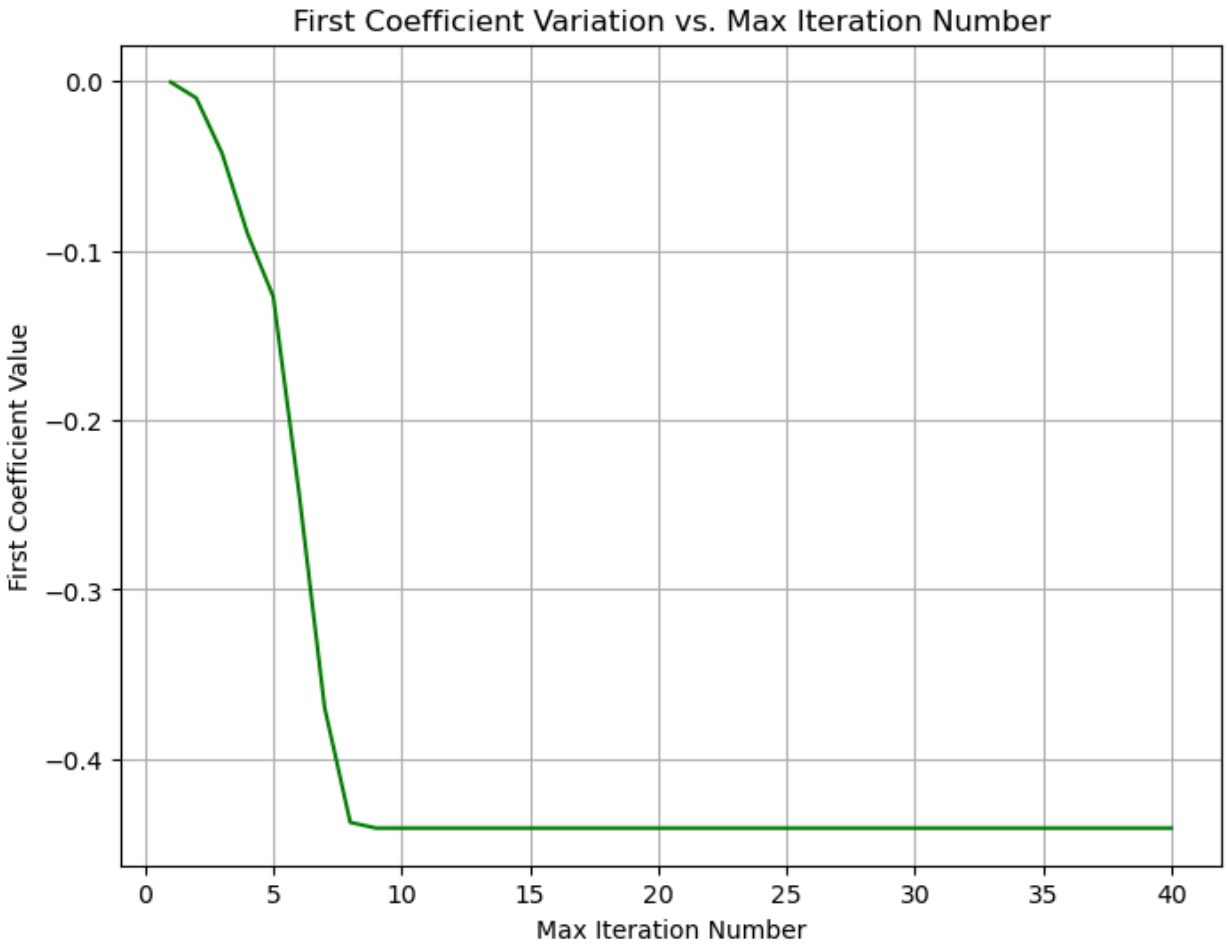
Part I: Logistics Regression for Digit Classification



1)

The graphs illustrate the relationship between the maximum number of iterations and two metrics, accuracy and log loss, for a predictive model. Initially, the model's accuracy is at its lowest with just a single iteration, but it improves significantly as the number of iterations increases. The accuracy stabilizes at approximately 99% once the model exceeds 6 iterations, suggesting that further iterations do not significantly enhance performance, indicating model convergence.

The log loss, which quantifies the model's predictive error, exhibits an inverse relationship to accuracy. A high log loss indicates poor model performance, while a log loss approaching zero suggests high accuracy. The log loss is at its peak with only one iteration, but it decreases sharply and plateaus after 6 iterations, aligning with the stabilization of accuracy. This plateau suggests that the model has reached its optimal predictive capability, and additional iterations do not meaningfully decrease error.



2)

The graph depicts how the initial weight assigned to pixel000 by the model evolves. At the start, with only one iteration, pixel000 carries a weight of zero, suggesting it has no influence on the model's determination of the input being an 8 or a 9. As the model undergoes more iterations and starts to stabilize, the weight assigned to pixel000 stabilizes at approximately -0.47. Such a negative weight suggests that the presence of this feature—pixel000—is associated with the input representing the number 8.

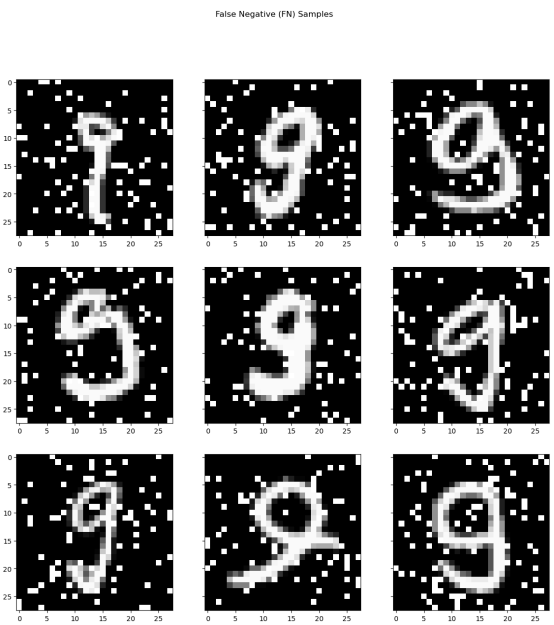
3)
Model with least loss on data:
Regularization Parameter C: 0.03162277660168379
Accuracy: 0.9672213817448311

Predicted True	0	1
0	942	32
1	33	976

C serves as the reciprocal of the regularization parameter, meaning that a lower value of C corresponds to more intense regularization. The optimal C value for the best-performing model appears to be around 0.03. This indicates a preference for the model to prioritize regularization overfitting perfectly to the training set, suggesting the possibility that the training data might not be a perfect representation of data in the real world. Consequently, this approach results in the model achieving the lowest log loss and a high level of accuracy, at 96.6%, when applied to test data. Examining the confusion matrix, it's evident that the model correctly predicted a large portion of the test set, achieving an accuracy of approximately 96.7% for the negative class (represented by the number 8) and 96.5% for the positive class (represented by the number 9).

4)

9 Samples of False Negatives
Predicted: 0, Actual: 1

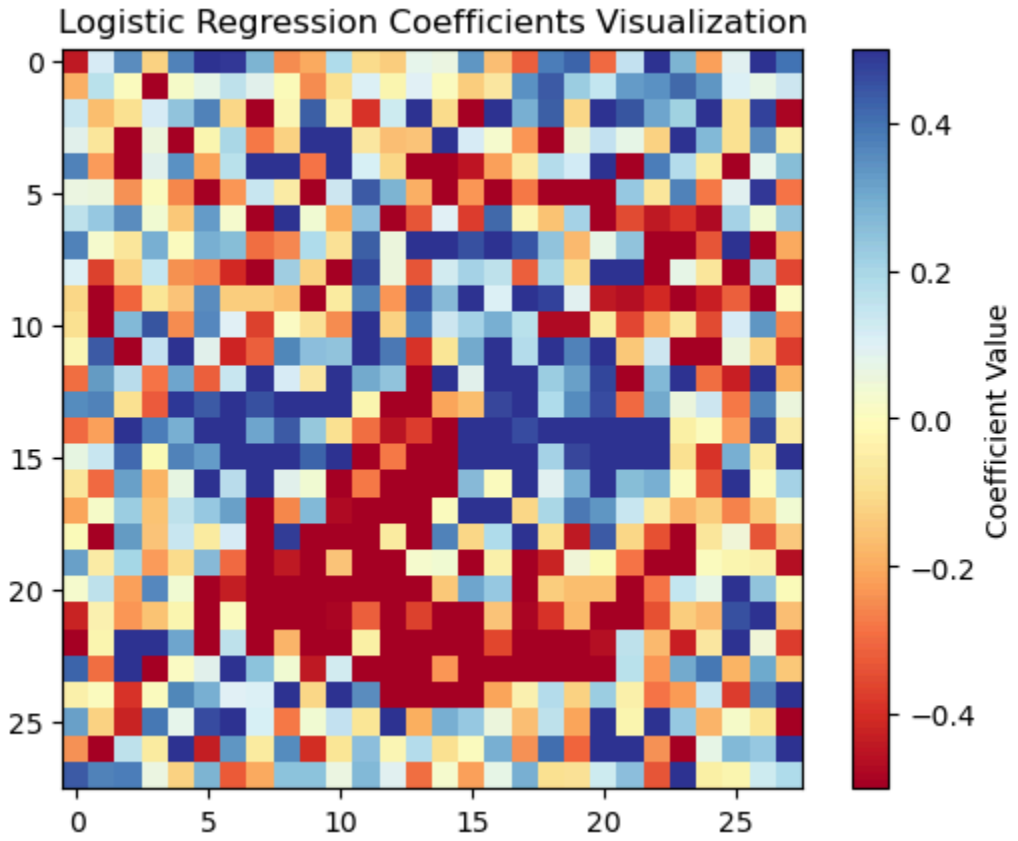


9 Samples of False Positive
Predicted: 1, Actual: 0

Each of the nine test inputs, which truly represent the negative class, signifying the digit 8, has been incorrectly categorized by the classifier as the positive class, implying the digit 9. An observation of the eights reveals that their lower loops tend to drift slightly to the right rather than being centered, potentially contributing to this misclassification.

Conversely, every one of the nine test inputs that actually belongs to the positive class, representing the digit 9, was wrongly identified by the classifier as the negative class, suggesting the digit 8. Upon examining the nines, it's noted that their tails are unusually curved and tilt at an angle, which might suggest that the model disproportionately associates these pixel characteristics with the digit 8.

5)



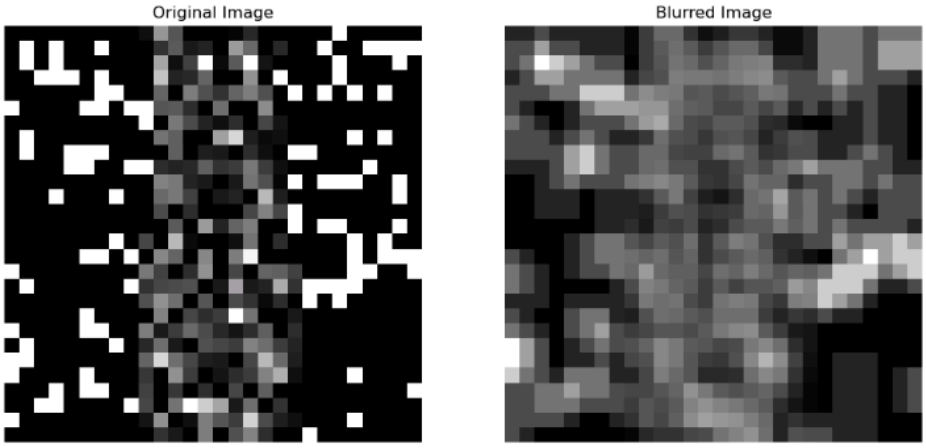
The graph displays the weight coefficients for the pixels from the original digit images, employing a 'RdYlBl' (Red, Yellow, Blue) divergent color map for visual representation. In this color scheme, yellow signifies a neutral coefficient value of zero, with the intensity of blue indicating positive coefficients increasing to a maximum of 0.5, and red denoting negative coefficients down to -0.5, as depicted by the color scale on the right.

Pixels that appear in shades of orange to red carry negative weight and are associated with the digit 8, while those in blue shades carry positive weight and are linked to the digit 9. The many light yellow areas around the edges reflect weights of zero or near zero, suggesting no association with either digit. This observation aligns with earlier samples where the digits 8 and 9 typically appear in the center of the images, leaving the periphery irrelevant. The presence of distinct red regions on the lower left could be indicative of the unique lower left curve in the digit 8.

It is important to recognize that areas shaded in red influence the model's decision towards a prediction value characteristic of the negative class, while blue-shaded regions sway the prediction towards the positive class. These weights are not direct classifications of 'negative' or 'positive' but rather they influence the model's classification decision based on the intensity and location of these weighted pixels.

Part II Trousers vs Dresses

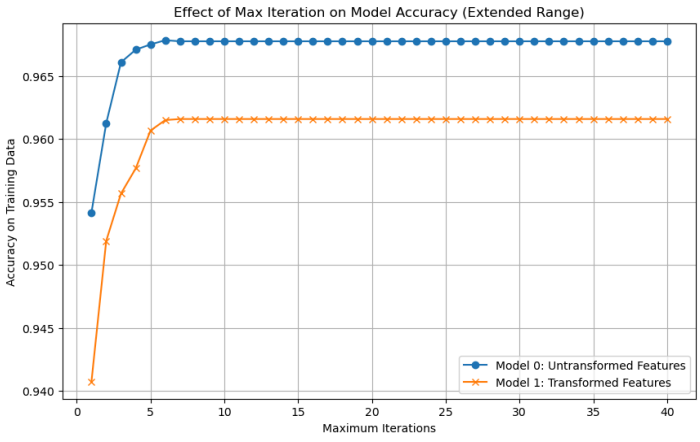
Feature Transformation: Noise Reduction



Blurring was applied to the training and test images to reduce noise, averaging each pixel with its surrounding 3x3 pixel area. The transformation lightens the peripheral black pixels (considered noise) to a grey shade, while intensifying the central lighter pixels to darker tones, making the trousers appear more defined. However, this can cause some originally dark pixels to lighten.

This blurring blends irrelevant side pixels, reducing their influence on the model and enhancing the central image clarity, which could potentially improve model performance. But since the outcome varies across the 12,000 images, these observations aren't conclusive about the overall effectiveness of noise reduction or its impact on model accuracy.

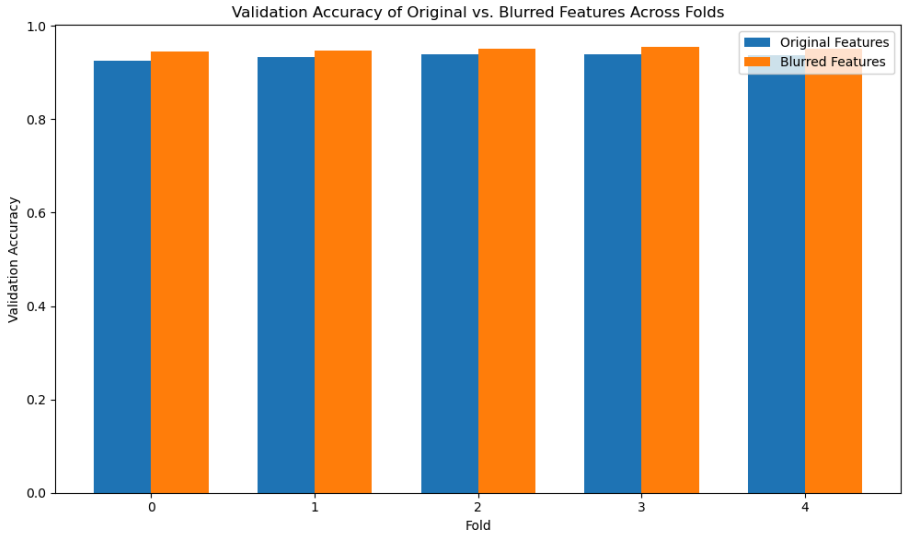
Logistic regression models were trained using sklearn's liblinear solver on the entire dataset, varying only the max_iter parameter to investigate the impact of iterations on convergence. The models compared include Model 0 with original features and Model 1 with transformed features.



The graph, plotting maximum iterations against accuracy, shows that Model 0 outperforms Model 1 on the training data, with both models converging around the ninth iteration.

Accuracy is measured on the training set, indicating the fit of the model; hence, Model 0's higher accuracy suggests a better or even overfitted model. Further analyses will delve into models with transformed features, but an optimal max_iter of 7 was determined for both models and will be adopted in subsequent training.

Max Iterations



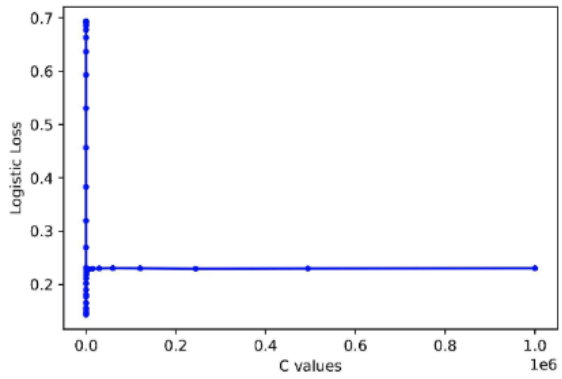
	Fold	Val Accuracy Original	Val Accuracy Blurred
0	0	0.925833	0.944583
1	1	0.932917	0.946667
2	2	0.938750	0.950417
3	3	0.939583	0.954167
4	4	0.937917	0.951667

The models trained on blurred features consistently show higher validation accuracies across all folds compared to those trained on the original features. This suggests that blurring, as a preprocessing step, effectively enhances the model's ability to generalize by smoothing out noise and irrelevant details in the images.

The consistent improvement in accuracy for blurred features across different folds indicates that the benefit of this preprocessing technique is not dependent on a specific subset of the data. This consistency is crucial for validating the robustness of the blurring transformation across varying training and validation sets.

Since blurring helps in smoothing out the noise, it can potentially reduce overfitting to the training data. Models trained on less noisy data might generalize better to unseen data (though testing on a separate test set is necessary to confirm this).

Preprocessing as a Critical Step: The analysis highlights the importance of data preprocessing in machine learning. Simple transformations like blurring can significantly impact model performance, underscoring the need to explore and apply appropriate preprocessing techniques tailored to the specific characteristics of the data and the task at hand.



Regularization Using C Parameter

A range of C values was investigated, from fractional numbers to one million, to gauge their effect on mitigating overfitting. Since C represents the inverse of regularization strength, a smaller C intensifies regularization, thereby reducing the risk of overfitting, while a larger C lessens regularization and may lead to an overfit model. Conversely, an excessively

small C value can cause too much regularization, potentially leading to an underfit model. The optimal C value, which minimizes logistic loss on the test data, was **found to be 0.7543.**

L1 v. L2 Regularization:

	L1	L2
Accuracy	0.954	0.953
Logistic Loss	0.137	0.143

Investigations into L1 and L2 regularization revealed variations in model efficacy. L1 regularization, also known as Lasso regression, resulted in better accuracy and a decreased logistic loss when compared to L2 regularization, or Ridge regression.

The rationale behind this is that Lasso regression employs L1 penalty to reduce the coefficients of less critical features to zero, effectively eliminating their impact on the model's decision-making process. In the context of the given scenario, where image noise has been reduced, this could mean that peripheral pixels that do not vary much across images are assigned zero weight by the model, recognizing that these features do not play a significant role in distinguishing between trousers and dresses.

Summary

	1st Round	2nd Round	Last Round
Parameters	Max-iter = 7 C = default 1.0 Penalty = default L2	Max-iter = 7 C = 0.7543120063354607 Penalty = default L2	Max-iter = 7 C = 0.7543120063354607 Penalty = L1
Accuracy	0.983546	0.983537	0.982977
Logistic Loss	0.048333	0.0.048333	0.047917

The model's error rate decreased and AUROC increased slightly over time, indicating improvements from regularizations, with a final error rate of 0.054 and an AUROC of 0.981, suggesting a good model. The approach to enhancing model performance involved experimenting with data transformations and parameter regularizations. Noise reduction, particularly through averaging pixel values in a 3x3 area, was a straightforward method tried, though it risked blending the subject with the background in small 28x28 images. The possibility of label inaccuracies in the 12,000 image dataset also raises questions about data reliability. The project effectively used cross-validation and regularization to minimize overfitting and identify an optimal model for hypothetical testing scenarios.