2024 Sem 1

CITS5508 Machine Learning

Assignment 1

Mila Zhang (22756463)

Due date: 12 April, 8pm

Table of Contents

1. Summarising the datasets *3*

1.1 Task D1 3

1.2 Task D2 3

1.3 Task D3 4

2. Logistic regression classifier without regularisation *5*

2.1 Task D4 5

2.2 Task D5 6

2.3 Task D6 6

3. Logistic regression classifier with L2 regularisation *7*

3.1 Task D7 7

3.2 Task D8 7

3.3 Task D9 8

3.4 Task D10 8

3.5 Task D11 8

3.6 Task D12 9

3.7 Task D13 9

4. Analysing performance *10*

4.1 Task D14 10

4.2 Task D15 11

4.3 Task D16 12

4.4 Task D17 12

4.5 Task D18 13

4.6 Task D19 13

5. Comparing models *14*

5.1 Task D20 14

5.2 Task D21 14

5.3 Task D22 15

5.4 Task D23 15

6. Limitations and further improvements *15*

1. Summarising the datasets

1.1 Task D1

The number of intances in the training set	11988
The number of intances in the test set	2000
The total number of instances	13988

Figure 1. Number of Instances

1.2 Task D2

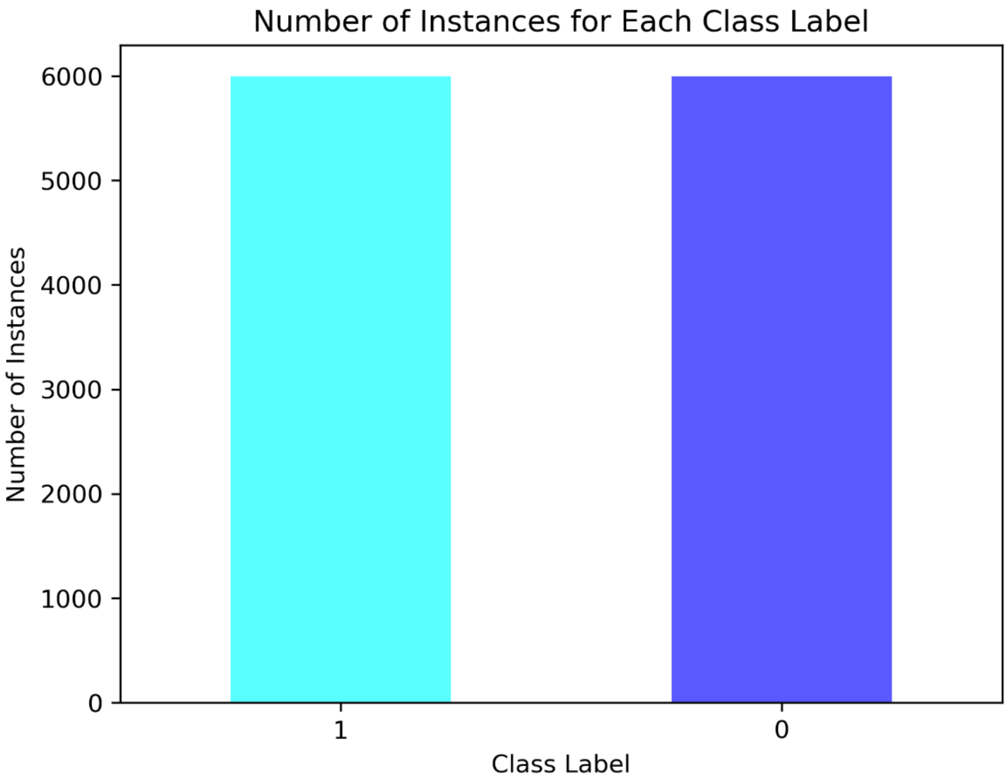


Figure 2. Number of Instances for Each Class Label

According to the bar plot, we do not have an imbalanced training set. Each class label has the same number of instances.

1.3 Task D3

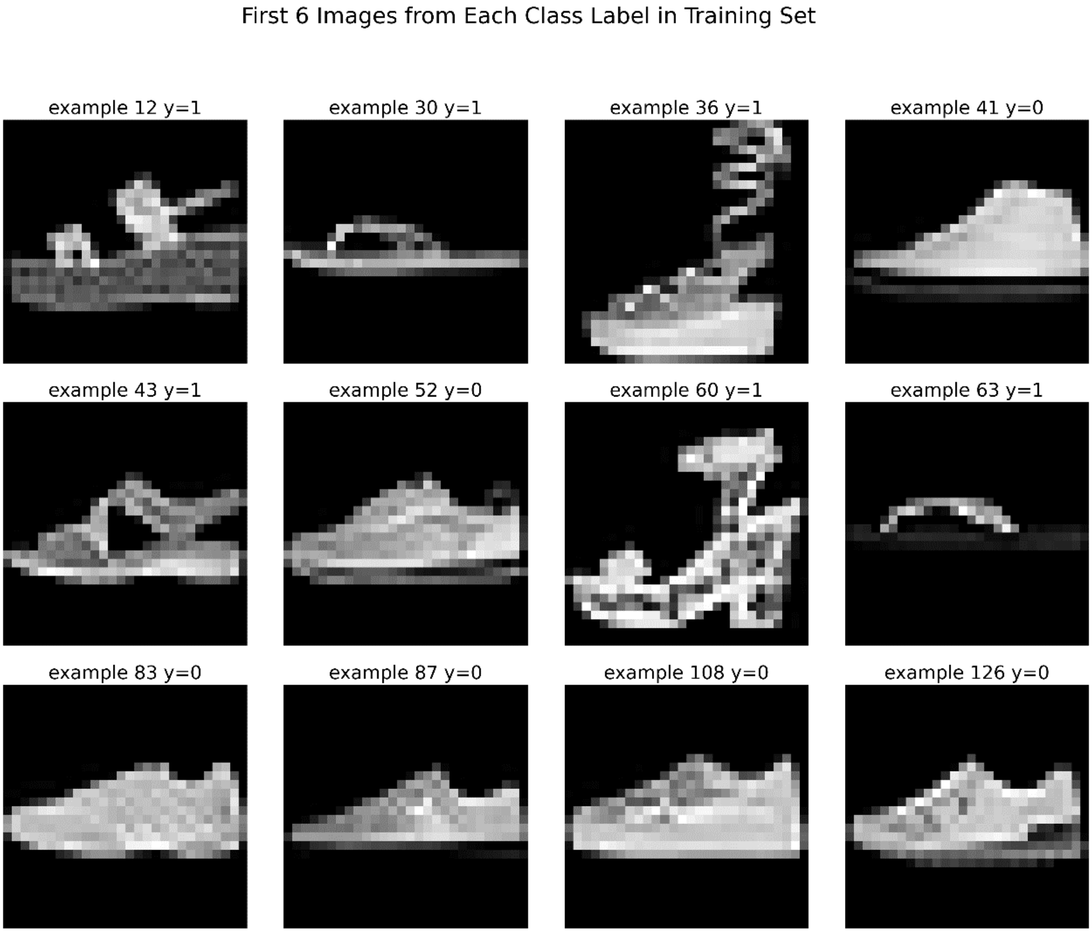


Figure 3. First Six Images from Each Class Label in Training Set

2. Logistic regression classifier without regularisation

2.1 Task D4

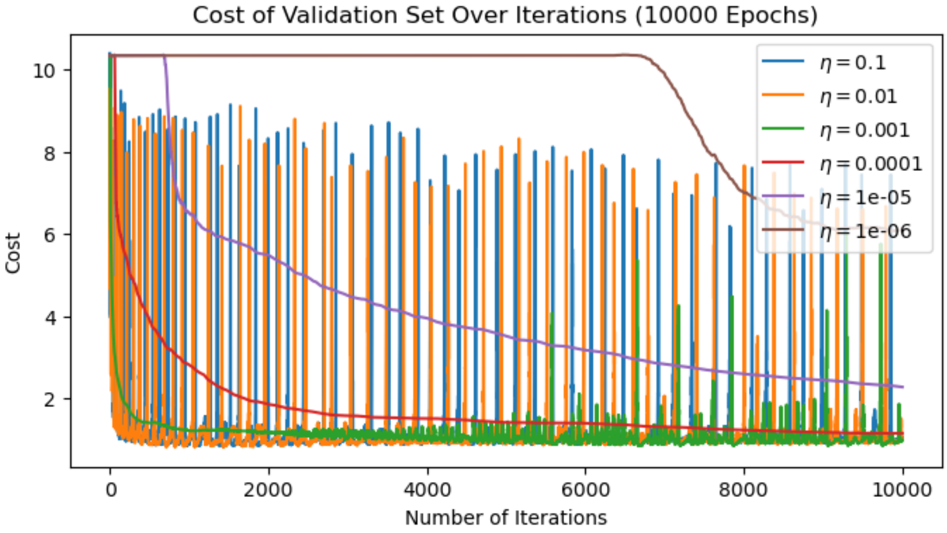


Figure 4

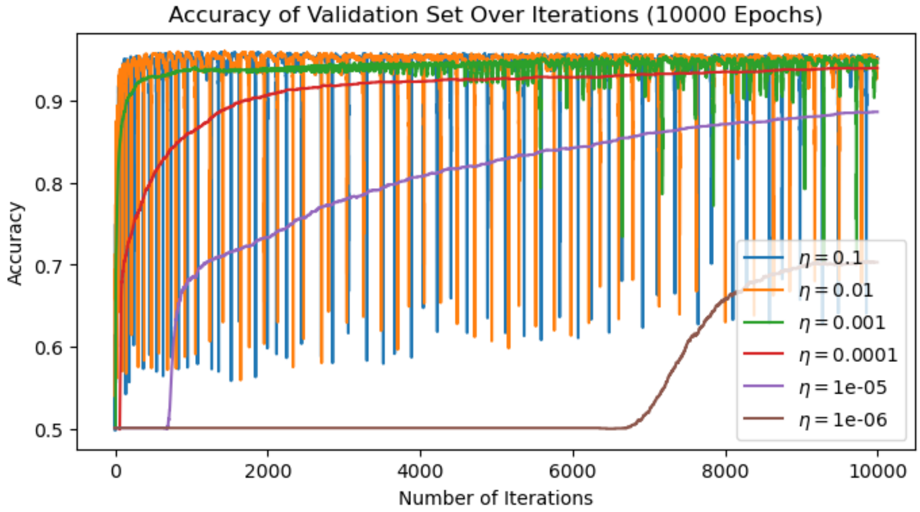


Figure 5

Given our balanced training set, accuracy to reflect the performance of the model under different learning rates. After extensive experimentation with rates ranging from 0.0 to 1.0, including values such as 1, 0.1, 0.01, 0.001, 0.0001, 0.00001, and 0.000001, a learning rate of 0.0001 emerged as the optimal choice.

This decision was informed by a comprehensive analysis of validation set scores depicted in Figures 4 and 5. Here, we observed that the learning rate of 0.0001 exhibits rapid convergence to minimal training loss and achieves commendable accuracy rates early in the training process, with minimal fluctuations.

Conversely, for rates exceeding this threshold, we encountered instances where the cost and misclassification rates diverged after an initial rapid descent. On the other hand, rates below 0.0001 failed to converge within the allotted 10,000 iterations, indicative of insufficient learning.

In summary, the learning rate of 0.0001 emerged as the optimal choice, striking a balance between rapid convergence and stability, thereby facilitating effective model training and performance evaluation.

2.2 Task D5

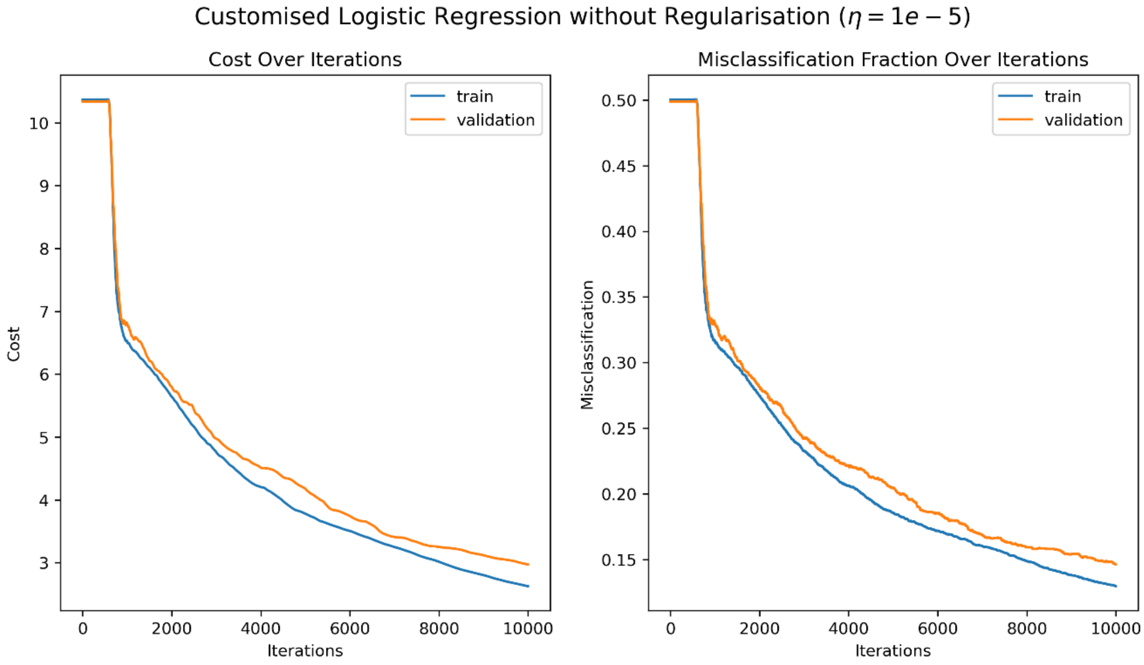


Figure 6

2.3 Task D6

The current model with a learning rate of 10-5 and 10,000 iterations, has yet to achieve convergence. To rectify this, there are two options: increasing the number of maximum iterations until the cost and misclassification curves plateau, or alternatively, adjusting the learning rate to accelerate the learning process and facilitate quicker convergence.

Upon analysis, it's evident that with a learning rate of 10-5, the model initially experiences a period of plateauing, which persists until approximately the 500th iteration. Subsequently, the cost and misclassification rate descend rapidly at around 1000th iteration. Beyond this point, the decline in these values becomes more gradual, culminating in a cost value of approximately 2 and a misclassification rate of roughly 0.13.

3. Logistic regression classifier with L2 regularisation

Assume C is the hyperparameter Alpha (regularisation strength), not the inverse.

3.1 Task D7

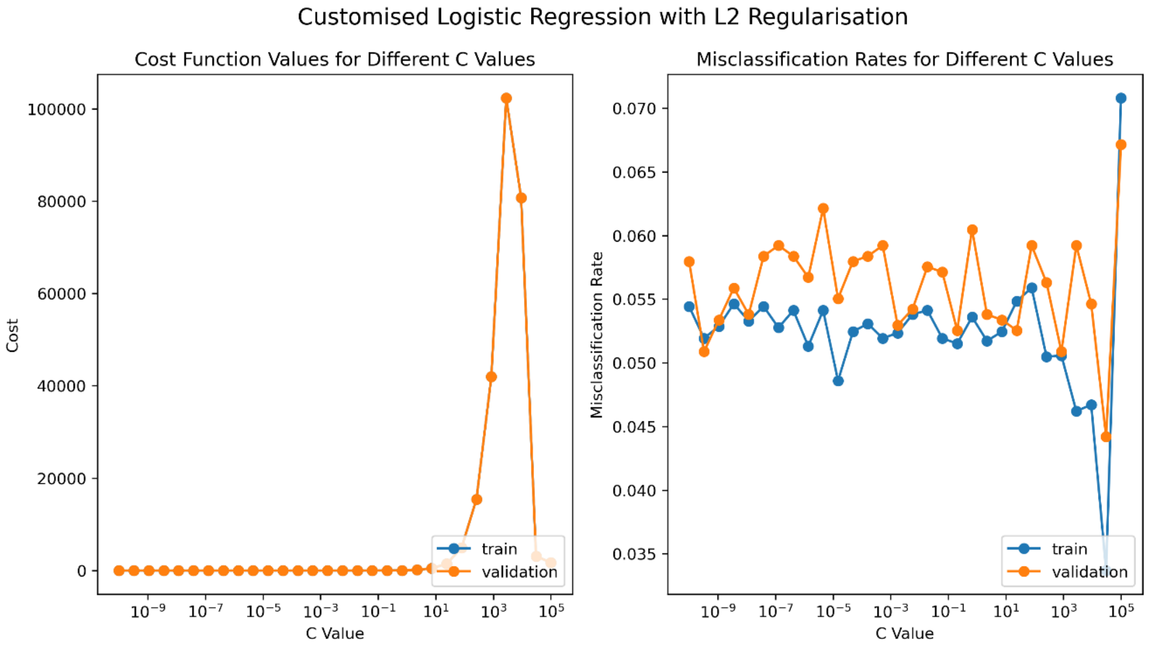


Figure 7

3.2 Task D8

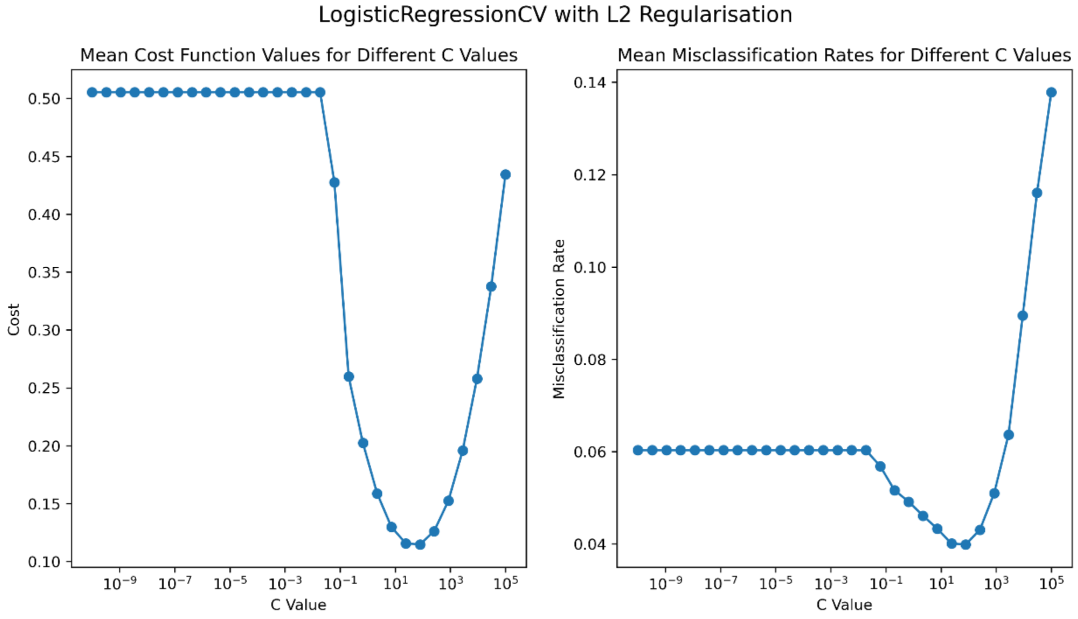


Figure 8

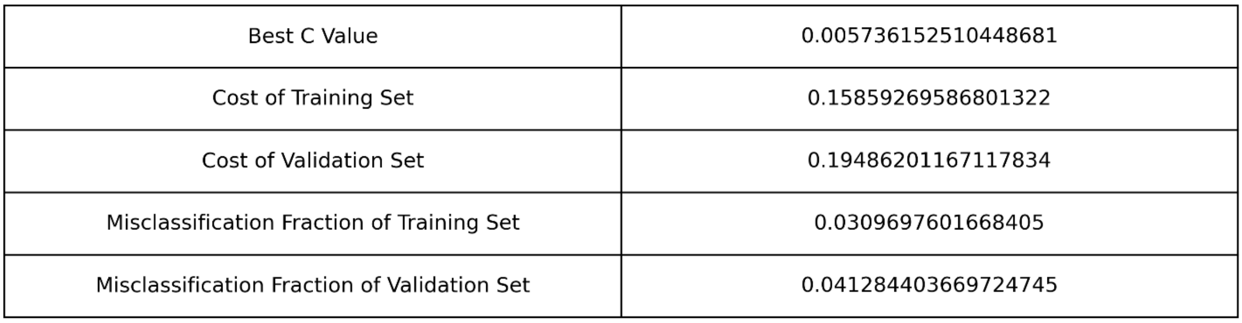
3.3 Task D9

Initially, a plateau in the curves is observed before reaching a C value of approximately 0.1. Subsequently, both the cost and misclassification rates experience a rapid descent, reaching their respective minima. However, beyond this point, a notable increase is observed in both metrics as the C value continues to rise.

The optimal regularization hyperparameter, as determined by the point of lowest mean cost and mean misclassification rate, is identified at a C value of approximately 78.8046, corresponding to an inverse value of 0.0127.

Using a 10-fold cross-validation approach yields a more consistent and reliable curve for the mean misclassification rate. In contrast, the fixed validation method employed in D7 exhibits a more zigzag curve and it fluctuates greatly before reaching a C value of 10^3. The adoption of 10-fold cross-validation facilitates more accurate and precise assessments of model performance.

3.4 Task D10



Best C Value	0.005736152510448681
Cost of Training Set	0.15859269586801322
Cost of Validation Set	0.19486201167117834
Misclassification Fraction of Training Set	0.0309697601668405
Misclassification Fraction of Validation Set	0.041284403669724745

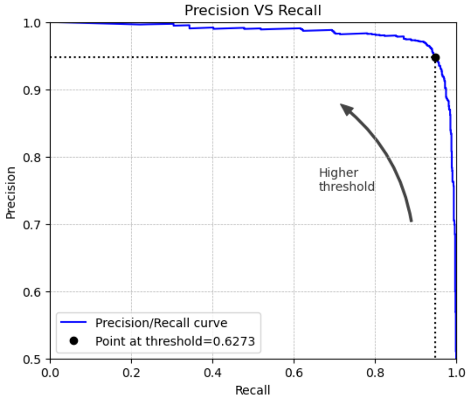
Figure 9. GridSearchCV Results

3.5 Task D11

Due to the different parameter interpretations across scikit-learn models, I opted to directly input the C value into LogisticRegressionCV, while passing its inverse into SGDClassifier.

The best C value identified in D10 is smaller than the optimal value determined in D8, yet it closely approximates 1/C determined in D8. Moreover, the cost function and misclassification values obtained in D10 closely resemble the minimum cost and misclassification fractions observed in D8. While both the LogisticRegressionCV and SGDClassifier with GridSearchCV approaches effectively explore and identify the optimal hyperparameter value C, there is a discrepancy in the optimal values found by these two methods. Notably, GridSearchCV offers a more straightforward and direct outcome.

3.6 Task D12



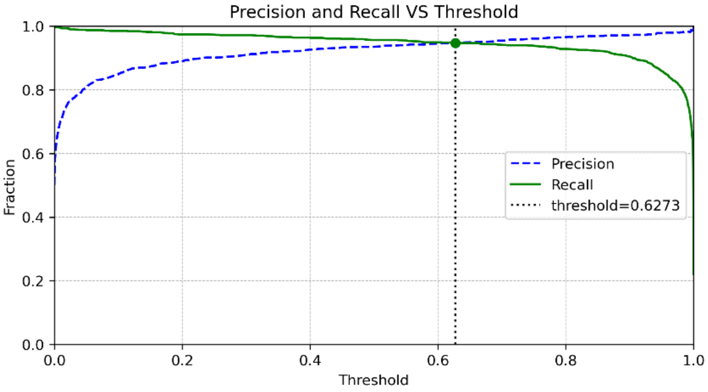


Figure 10

Figure 11

The precision and recall curves intersect at approximately the threshold value of 0.6273, indicating a point of balance between precision and recall. Initially, the precision value starts at 0.5 and then gradually ascends to a high fraction at around 0.95. Conversely, the recall curve exhibits a steep decrease following its intersection with the precision curve.

I opt for a threshold value of 0.6273 (rounded to 4 decimal places) which closely aligns with the intersection point of the precision and recall curves. This choice is motivated by the fact that at this threshold, both precision and recall hover around 95%. Such high values imply that instances predicted as class label 1 are mostly correct, and nearly all actual instances with class label 1 are accurately predicted. Consequently, the model demonstrates high accuracy and completeness in its positive predictions.

3.7 Task D13

The threshold value obtained from my custom grid search implementation, is approximately 0.5455, which differs slightly from the optimal value I selected in D10.

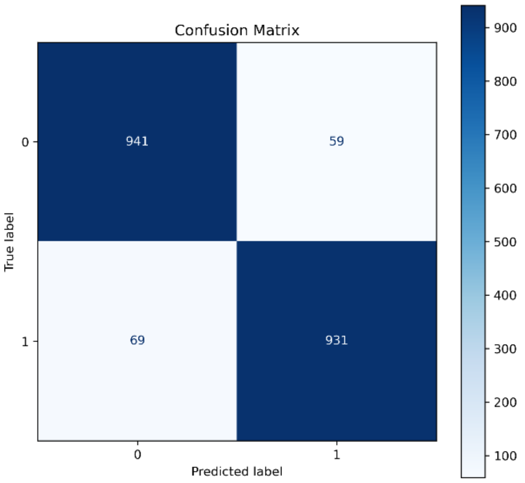
In D10, I chose the threshold approximately based on the curve where both precision and recall exceed 0.95. However, the threshold value found in D13 is derived from maximizing the AUC score and accuracy score, offering a comprehensive evaluation of the model's overall performance.

When determining the optimal threshold value, utilising a customised grid search alongside key metrics such as AUC score, F1 score and accuracy score serves as a more robust and scientific approach compared to solely relying on the precision versus recall curve for threshold determination.

4. Analysing performance

4.1 Task D14

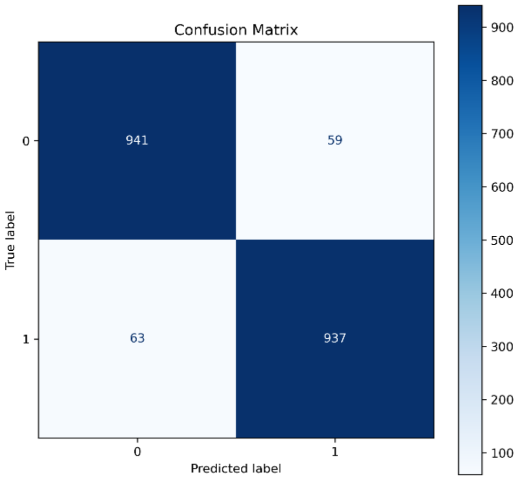
LR1



Precision	0.9404040404040404
Recall	0.931
FPR	0.059

Figure 12

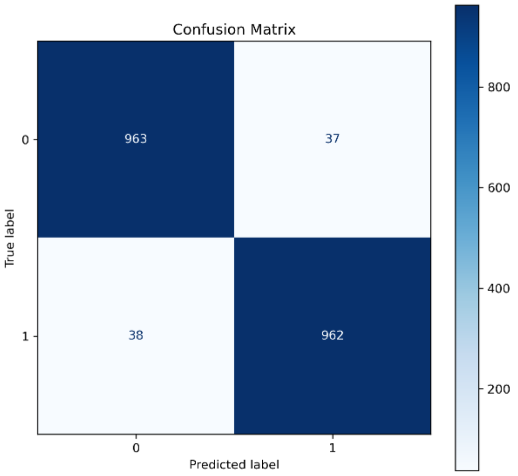
LR2



Precision	0.9407630522088354
Recall	0.937
FPR	0.059

Figure 13

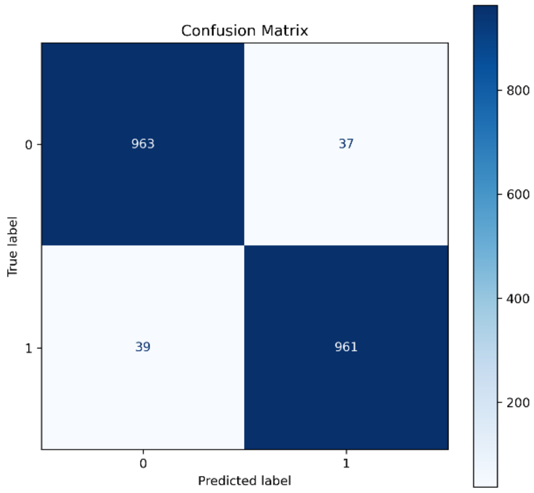
LR3



Precision	0.9629629629629629
Recall	0.962
FPR	0.037

Figure 14

LR4



Precision	0.9629258517034068
Recall	0.961
FPR	0.037

Figure 15

4.2 Task D15

Among the four models examined, LR3 and LR4 showcase commendable performance metrics, boasting high precision, recall, and notably low False Positive Rates (FPR). The confusion matrices underscore their proficiency, as indicated by the dominant presence of darker shades in the True Positive (TP) and True Negative (TN) blocks.

LR3 and LR4 exhibit minimal misclassifications, with fewer than 80 samples inaccurately labelled out of the 2000 test instances. Combining with their superior precision and recall rates, they exhibit robust generalization capabilities. Conversely, LR1 and LR2 demonstrate a relatively higher misclassification rate, indicating a lesser ability to generalize.

In terms of generalization capability ranking, LR3 and LR4 emerge as the top performers, followed by LR2, with LR1 exhibiting slightly inferior generalization capability.

4.3 Task D16

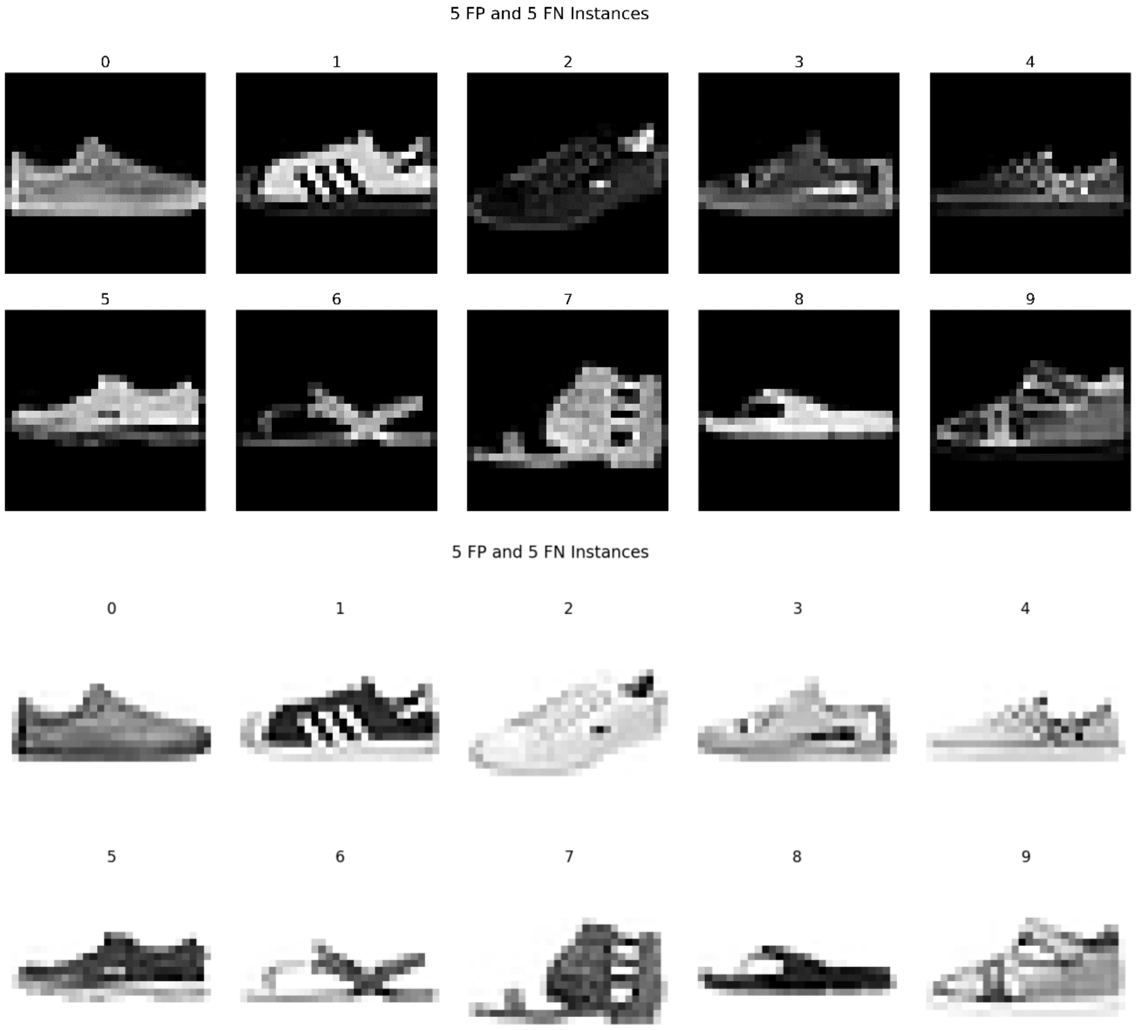


Figure 16

4.4 Task D17

A sneaker (show 0) that points to the right is misclassified as sandals. Sneakers (shoe 1) that have a high colour contrast are misclassified as sandals. Sneakers (shoe 2, 3, 4) that are mostly in the colour of white are misclassified as sandals.

Conversely, sandals characterized by predominantly dark tones (shoe 8) are mislabelled as sneakers. Sandals (shoe 5, 9) that have thick soles and in the shape of sneakers are mislabelled as sneakers.

4.5 Task D18

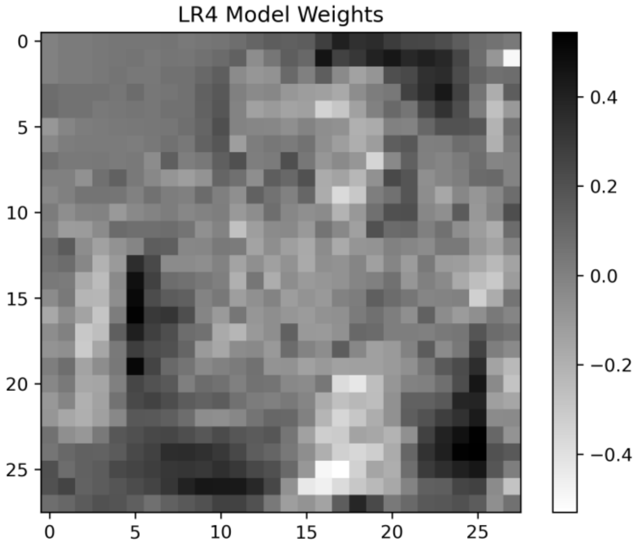


Figure 17

4.6 Task D19

The values in the array range from very small (close to 0) to relatively larger values (below 1). The smallest value is negative and approximately 0.

The dark regions indicate large positive weights, while the lighter areas indicate large negative weights. Samples that have bigger pixel values (lighter shades) within these dark areas are prone to be predicted to be class label 1 (sandal), because ϑ^T x is larger, resulting in a larger predicted probability computed by the sigmoid function. Conversely, intances that have smaller pixel values (darker hues) within those dark areas tend to be predicted to be class label 0 (sneaker).

Similarly, samples that have bigger pixel values (whiter colour) within the white areas are more inclined to be categorised as class label 0 (sneakers) since the weights are negative in these regions.

5. Comparing models

5.1 Task D20

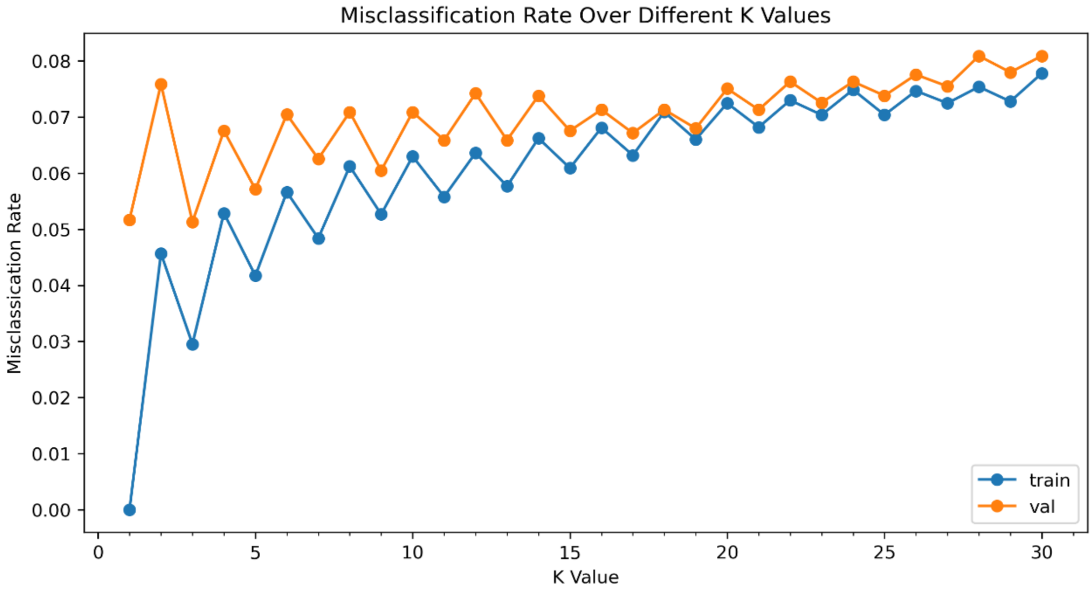


Figure 18

5.2 Task D21

The misclassification rates exhibit fluctuations as the k value escalates, showcasing a zigzag pattern in both the training and validation sets.

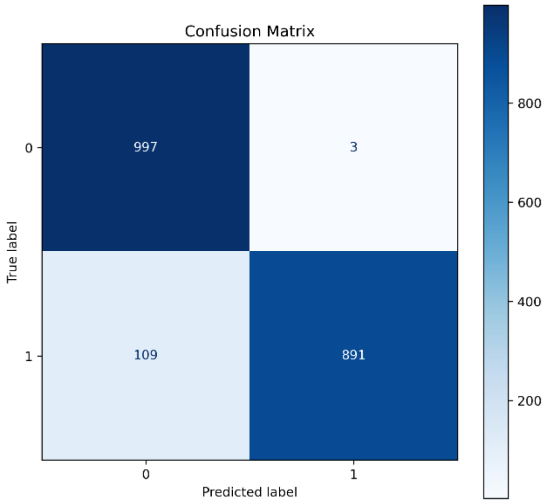
I opt for a k value of 3. At that point, the validation set demonstrates the lowest misclassification rate, with a narrower discrepancy between the training and validation set compared to when k equals 1.

The preference for k=3 over k=1 stems from the significant disparity between the training and validation misclassification rates when k=1 which suggests overfitting of the model to the training data.

In contrast to LR1 model, this k-NN model exhibits lower misclassification rates and significantly faster training speed, resulting in an enhanced overall performance. With consistently minimal misclassification rates (below 0.08) across all k values, coupled with a significantly shorter training time, the k-NN model emerges an optimal choice.

5.3 Task D22

KNN



Precision	0.9966442953020134
Recall	0.891
FPR	0.003

Figure 19

5.4 Task D23

The k-NN model showcases an impressive precision score of approximately 0.9967, coupled with an exceptionally low false positive rate of 0.003. However, its recall score falls short compared to all the logistic regression models. Despite its minimal false-positive predictions, the k-NN model exhibits over 100 false-negative predictions, the highest among all models.

While the total number of misclassifications in the k-NN model is fewer than LR1 and LR2, indicating a superior generalization capacity compared to these models, it falls short when compared to LR3 and LR4. This suggests a weaker generalization capacity relative to LR3 and LR4.

6. Limitations and further improvements

The machine learning pipeline refers to the sequence of steps involved in building, training, evaluating, and deploying a machine learning model. It typically contains the processes of data collection, data preprocessing, feature engineering, model selection, model training, hyperparameter tuning, model evaluation, data augmentation etc.

1. Data Collection
When collecting data, we can enhance the dataset by collecting and targeting images that pose challenges to the classification algorithms, such as those prone to misclassification. This could involve gathering more instances that resemble those false positive and false negative samples observed in task D16, allowing the model to identify those samples more accurately.

2. Data Scaling
Experiment with different preprocessing methods to identify the most effective approach. Options include scaling pixel values from [0,255] to [0,1] by dividing by 255 via MinMax scaling, or standardises features using the Standard Scaler, which subtracts the mean value and divides by the standard deviation of training samples.

3. Data Augmentation
Increase dataset diversity and improve model generalization by applying techniques like rotation, flipping, and adding noise to images.

4. Feature Engineering
Expand beyond pixel values by extracting additional features from images. This could involve identifying edges or specific regions within images that contribute discriminative information to the classification task.

5. Feature Selection (Dimensionality Reduction)
Not all pixel values are crucial for classifying the images, we can select part of the pixels for classifying. Through observation, we can find that the corners of the images are usually blank. And there are areas that most sample will have similar pixel values, which provides fewer useful information. We can explore techniques for interpreting and visualizing the model's predictions and decision boundaries to help in understanding which features are most important for the classification task and identifying potential areas for improvement.

Since the images have high-dimensional pixel values, dimensionality reduction techniques such as Principal Component Analysis (PCA) or t-distributed Stochastic Neighbour Embedding (t-SNE) can be applied to reduce the dimensionality of the data while preserving most of the variance. This can help in visualizing and understanding the data better and may improve the model's performance.

6. Model Selection
Experiment with a wider array of machine learning models suited for binary classification tasks, such as decision trees, random forests, support vector machines, and ensemble methods, beyond logistic regression and k-nearest neighbours.

7. Model Evaluation
Utilize advanced evaluation metrics like ROC curve, AUC score, and F1 score alongside precision, recall, FPR, and accuracy. Also, incorporate cross-validation to obtain a more accurate score when measuring the performance of the model, instead of using a fixed validation set.

8. Model Training

Optimize training efficiency by setting a tolerance for stopping criteria and considering Mini-batch Gradient Descent over Batch Gradient Descent to accelerate convergence.

9. Hyperparameter Tuning

Experiment various combinations of hyperparameters to find the optimal values for different models using techniques such as grid search or random search, instead of determining one hyperparameter each time. For instance, find the optimal combination of learning rate, regularisation strength, number of epochs and so on.

10. Transfer Learning

Instead of training a model from scratch, we can leverage pre-trained deep learning models such as VGG, ResNet, or Inception, which have been trained on large datasets like ImageNet. We can fine-tune these models on our dataset, potentially achieving better performance with less training data.

By implementing these strategies and carefully analysing the results, we can iteratively improve the modelling process and achieve better performance for the classification of sandals and sneakers.