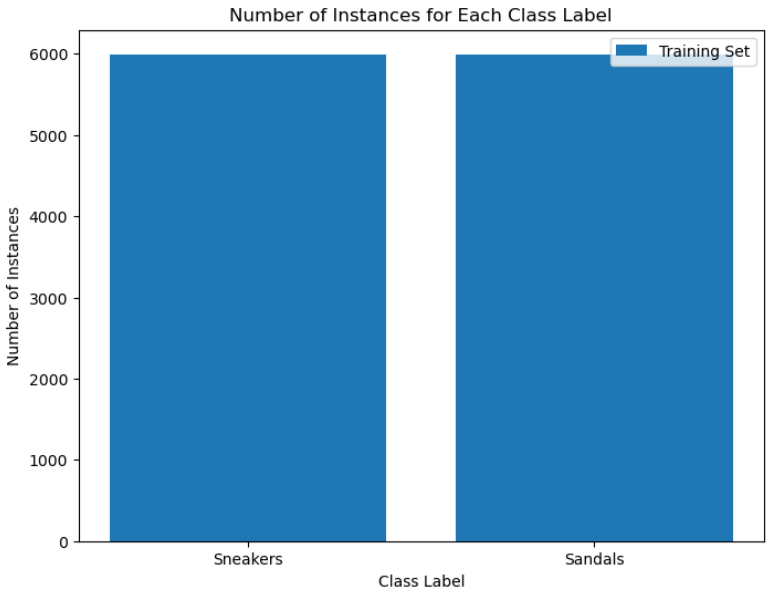
Problem: Classifying Sneakers versus Sandals

1 Summarising the datasets

D1 [3 marks] Table of the number of instances in the dataset

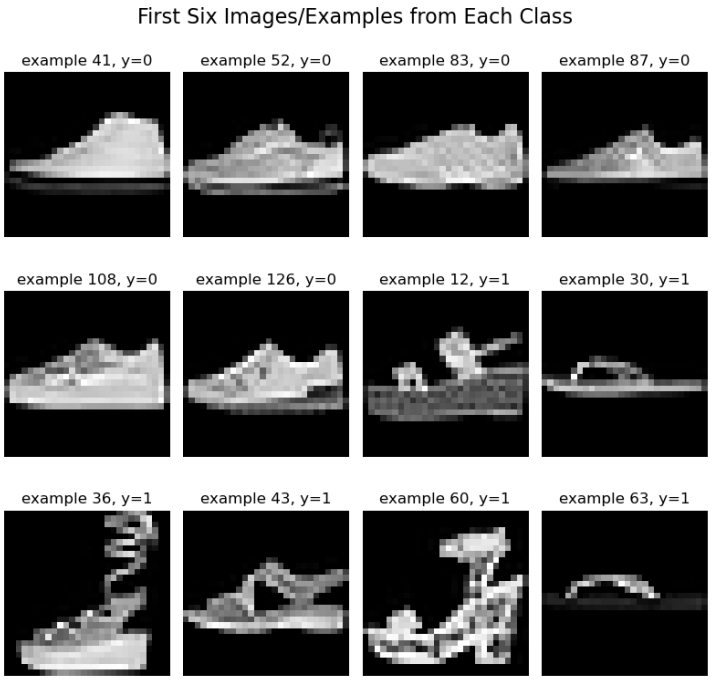
The number of instances in the training set	11988
The number of instances in the test set	2000
The total number of instances	13988

D2 [2 marks] Bar plot showing the number of instances for each class label



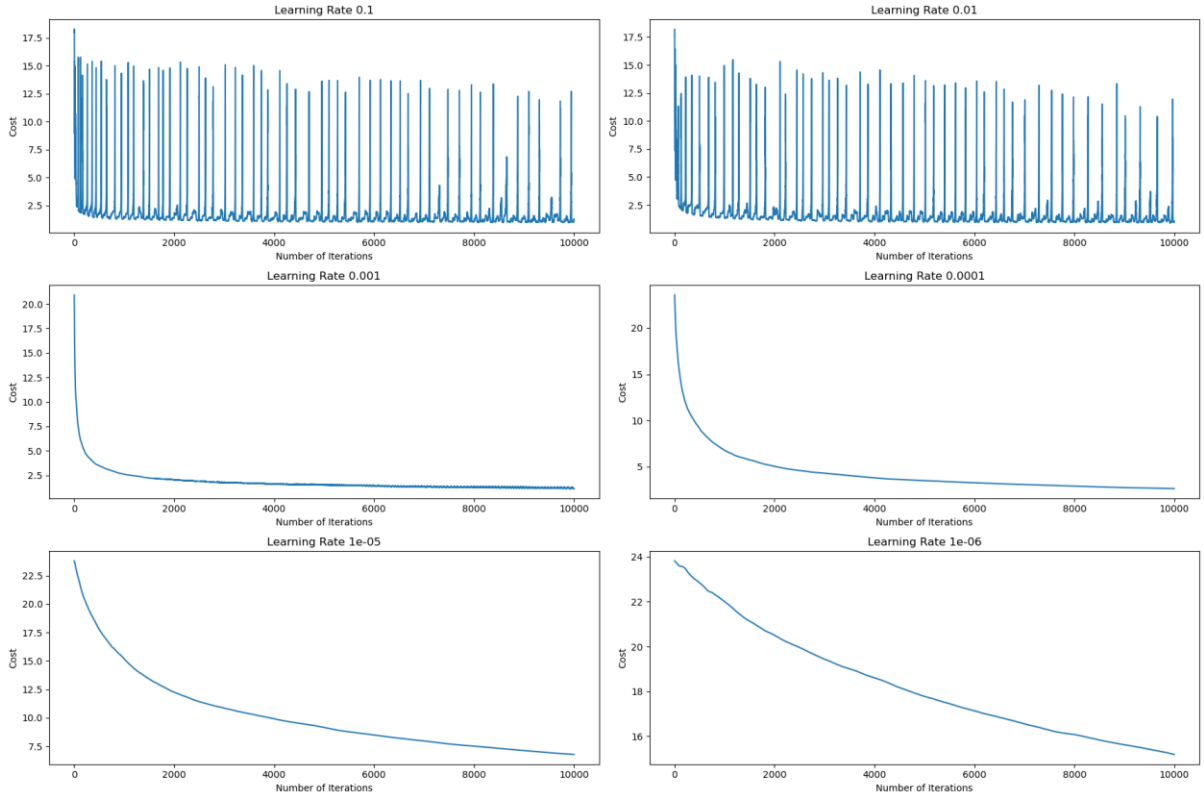
The training set is balanced. There are approximately 6000 number of sneakers and sandals in the training set.

D3 [3 marks] First six images from each class



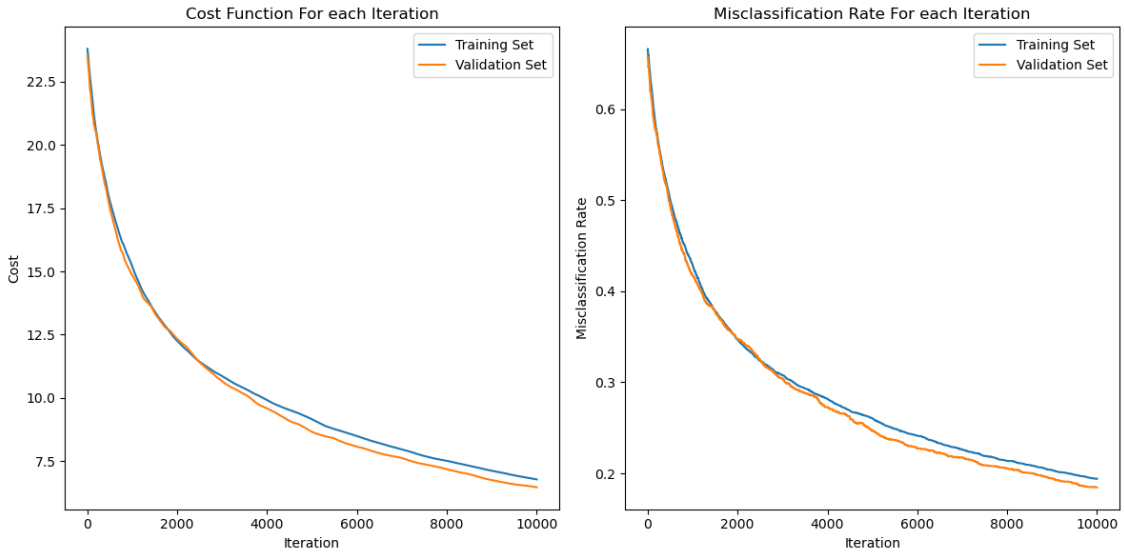
2 Fitting your logistic regression classifier

D4 [3 marks] Experiment with different learning rates



A learning rate that is too high, such as 0.1 and 0.01, results in erratic behaviour, with the cost function bouncing around over time, causing the algorithm to overshoot the minimum and fail to converge. Whereas a learning rate that is too low, such as 10^{-5} and 10^{-6}, results in a very slow decrease in the cost function, resulting in slow convergence, requiring a large number of iterations to converge. The chosen learning rate is 10^{-4} because it allows for a stable and consistent decrease in the cost function to achieve a low cost within 10000 iterations without erratic behaviour.

D5 [10 marks] Plots of Costs and Misclassification rates for learning rate of 1e-5

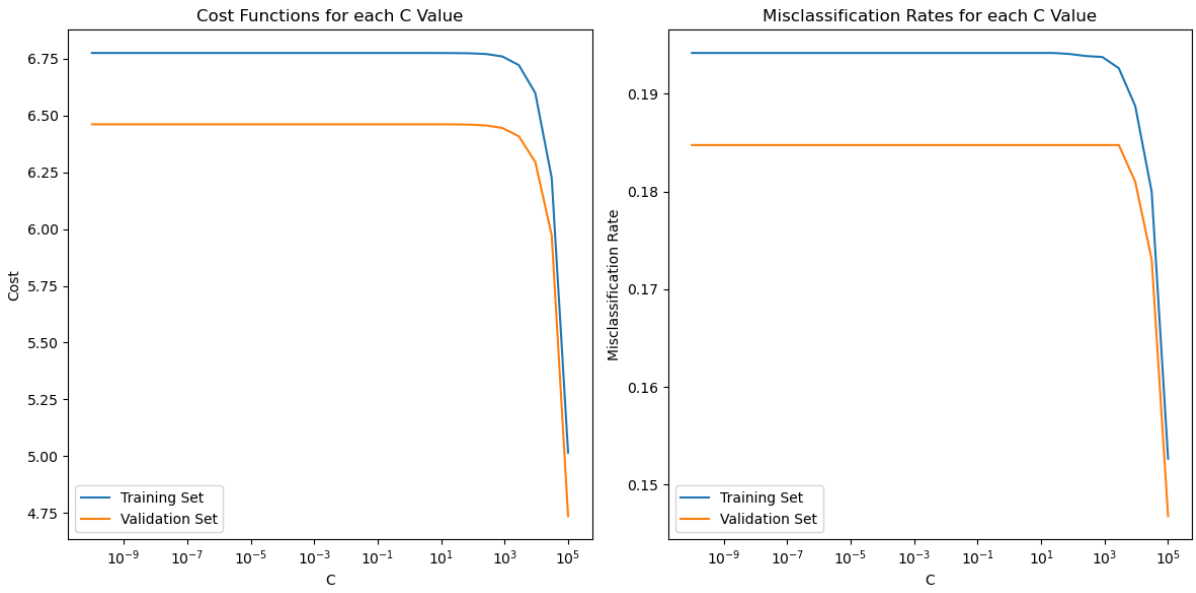


D6 [3 marks] Interpretation of above results

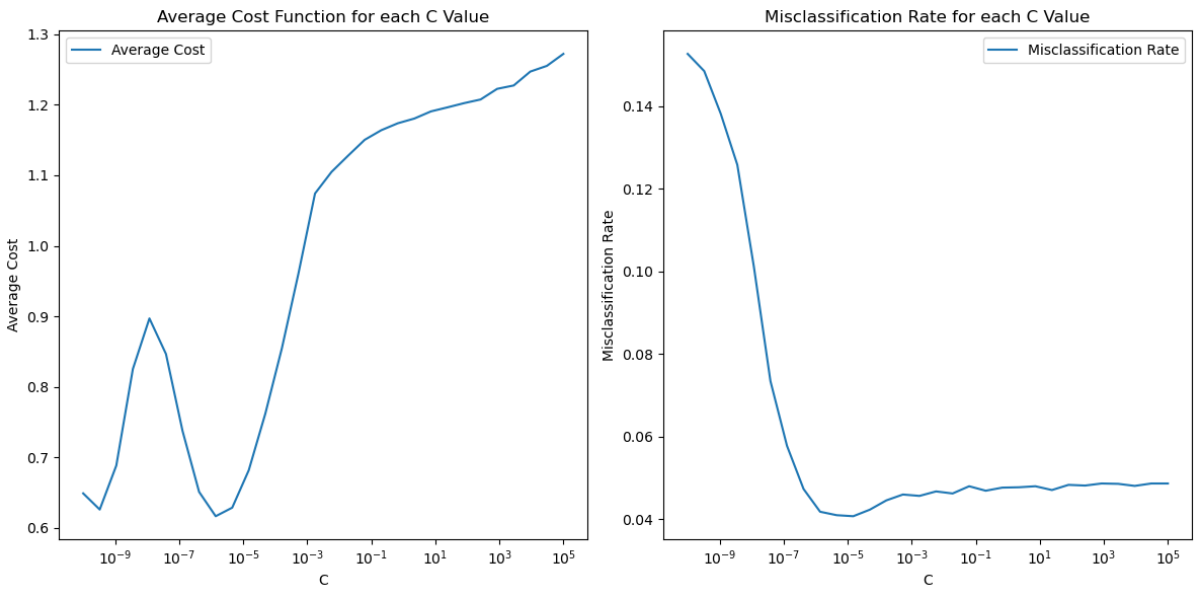
The model is able to generalise to new data as shown by the decreasing costs and misclassification rates for the validation set. The cost function decreased steadily over the iterations, meaning that the algorithm was minimising the cost and approaching a more optimal solution. Similarly, the misclassification rate decreased steadily over the iterations, meaning that the model was improving its predictive performance on both the training and validation set.

3 Using cross-validation and Scikit-learn logistic regression classifier

D7 [6 marks] Plots of Costs and Misclassification rates for each C value

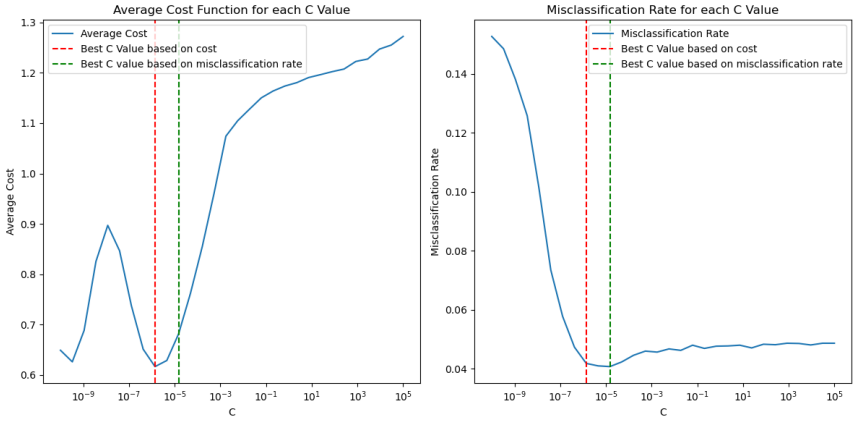


D8 [4 marks] Plots of Average Costs and Misclassification rates



D9 [3 marks]

As C in increases, misclassification rates decrease to a minimum before slightly increasing, whereas average cost values oscillate before reaching a minimum, then increasing sharply. This may be because higher C values leads to weaker regularisation, allowing the model to fit the training data closely and potentially overfit as C becomes too high. The best C value based on the model that achieved the lowest cost is 1.373e-6 (red line), while the best C value based on the model that achieved the lowest misclassification rate is 1.487e-5 (green line).



In this case, I would select the C value of 1.373e-06, found in D8 using 10-fold cross-validation. This C value improves the model's performance with the lowest cost of 0.617 and a low misclassification rate of 0.042. In contrast, the best C value of 10000 found in D7 has a cost of 5.016 and misclassification rate of 0.153. The C value in D8 obtained significantly lower costs and misclassification rates. This may be due to using a fixed validation set that might not be representative of the overall dataset. Using 10-fold cross-validation averages over multiple validation sets. It avoids overfitting hyperparameters to a fixed validation set and makes use of the entire dataset for training and validation in each fold.

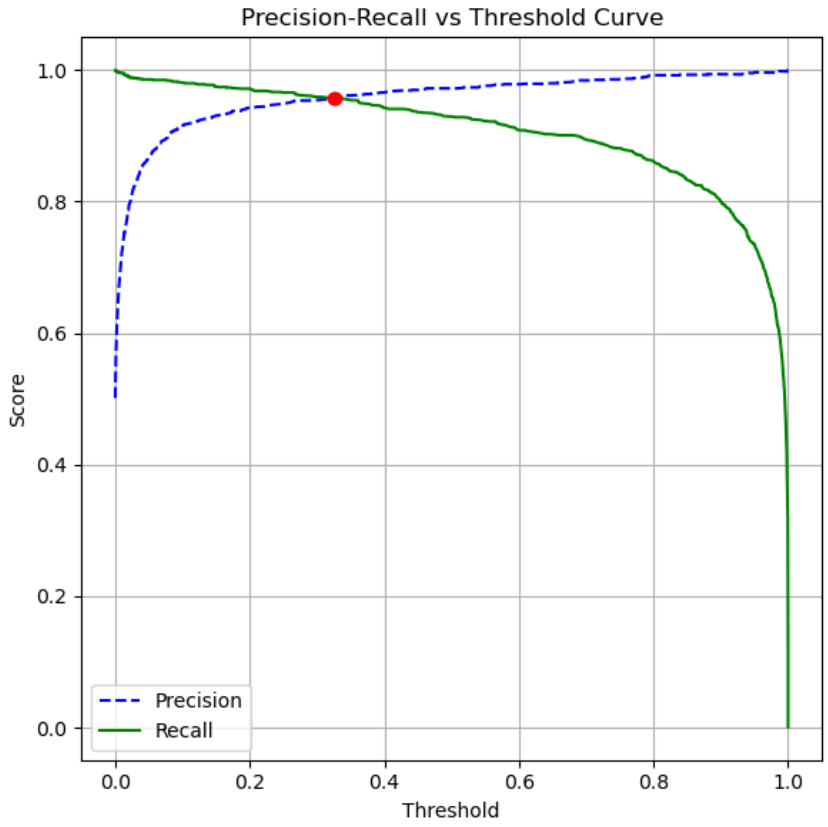
D10 [5 marks] Table of values

Optimal C Value	23.950266
Training Cost Function Value	0.098397
Validation Cost Function Value	0.131889
Training Misclassification Rate	0.035975
Validation Misclassification Rate	0.048791

D11 [3 marks] Interpretation of above results

Task D8 uses LogisticRegressionCV, whereas Task D10 uses SGDClassifier with GridSearchCV. Both approaches use 10-fold cross-validation but uses different search methods to select an optimal value for the regularisation hyperparameter C. The cost value and misclassification rates for the model using the optimal C value from D10 is lower than D8 for both training and validation sets. Using grid search for hyperparameter tuning in D10 can lead to improved model performance as it explores the set of C values more exhaustively and systematically to find a more optimal value.

D12 [5 marks] Plot of precision versus recall for different threshold values



As the threshold increases, the precision increases but the recall decreases. This is because there is a trade-off between the precision and recall. A lower threshold prioritises recall, ensuring that more positive instances are captured, even if it means more false positives. A higher threshold prioritises precision, reducing false positives at the cost of missing some positive instances.

In this case, the selected threshold is 0.3260 which is the intersection between the precision and recall curves. This threshold is chosen because it represents a balanced trade-off between precision and recall.

D13 [4 marks] Optimal threshold value using grid search

The optimal threshold obtained using the grid search on the validation set is 0.2676, with the highest corresponding F1 score of 0.9595. This threshold value represents the balance point between precision and recall, maximising the harmonic mean of both the precision and recall.

Both approaches in D12 and D13 aim to find a threshold that balances precision and recall. D12 uses a visual method to find the intersection between precision and recall, and averaging them to obtain an optimal threshold. D13 uses a quantitative method to maximise the F1 score.

4 Analysing the performance closer

D14 [8 marks] Four Logistic Regression Model's performance on the test set

LR1 Model

Confusion Matrix:

		Actual Values	
		Positive	Negative
Predicted Values	Positive	753	161
	Negative	247	839

Precision : 0.824
Recall : 0.753
False positive rate : 0.161

LR2 Model

Confusion Matrix:

		Actual Values	
		Positive	Negative
Predicted Values	Positive	812	131
	Negative	188	869

Precision : 0.861
Recall : 0.812
False positive rate : 0.131

LR3 Model

Confusion Matrix:

		Actual Values	
		Positive	Negative
Predicted Values	Positive	927	25
	Negative	73	975

Precision : 0.974
Recall : 0.927
False positive rate : 0.025

LR4 Model

Confusion Matrix:

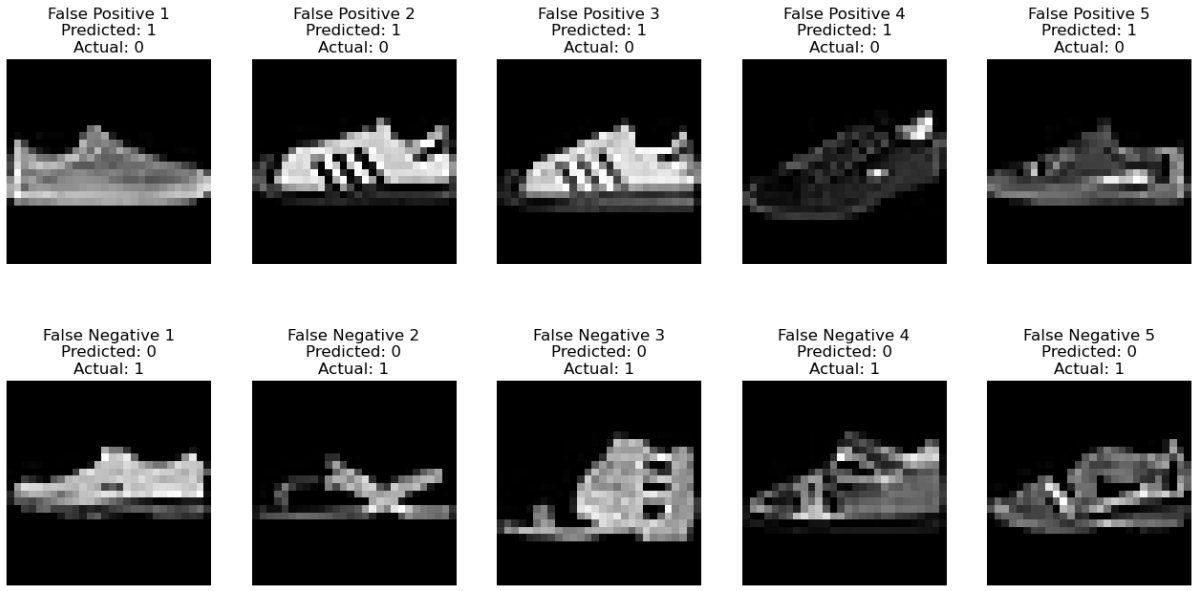
		Actual Values	
		Positive	Negative
Predicted Values	Positive	960	51
	Negative	40	949

Precision : 0.950
Recall : 0.960
False positive rate : 0.051

D15 [3 marks] Interpretation of above results

LR1 model achieves a precision of 0.824, recall of 0.753, and false positive rate of 0.161, indicating a moderate level of performance. **LR2 model** shows improvement in generalisation capacity with a higher precision of 0.861, recall of 0.812, and false positive rate of 0.131, indicating better performance in correctly identifying positive instances. This shows that regularisation helps to improve the generalisation capacity of the model by reducing overfitting. **LR3 model** further improves with a high precision of 0.974, high recall of 0.927, and low false positive rate of 0.025, indicating even better performance in correctly identifying positive instances while minimising false positives. This model has the best generalisation ability due to the use of cross-validation and optimised regularisation hyperparameter C. **LR4 model** uses an optimal threshold fine-tuned using grid search method. It has a lower precision, higher recall, and higher false positive rate than LR3 model. The precision is lower because the false positive rate is higher, meaning that there are more instances predicted as positive (sandals) that are actually negative (sneakers) compared to LR3, hence has a lower generalisation capacity than LR3.

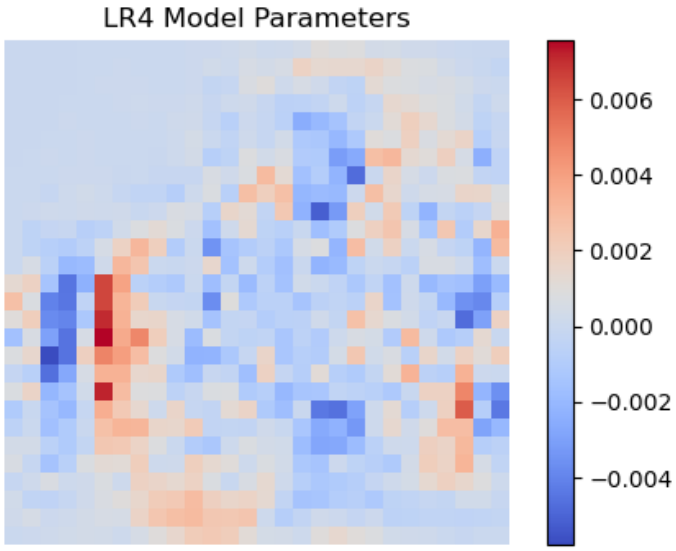
D16 [4 marks] Five images of false positives and false negatives on the test set

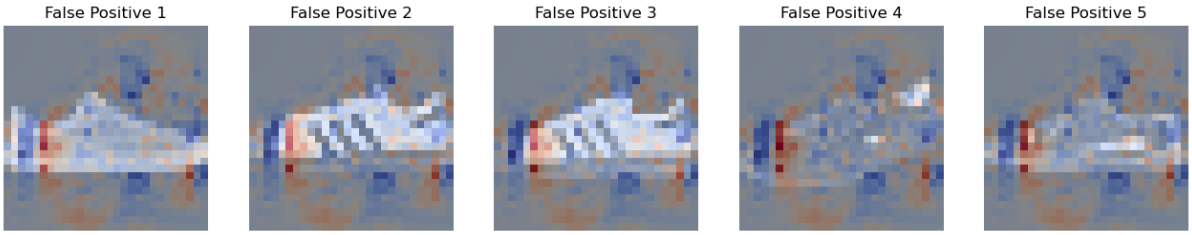


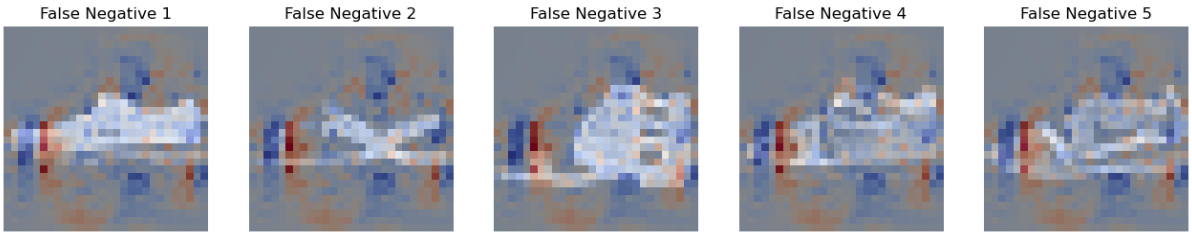
D17 [2 marks] Interpretation of above results

The LR4 model is making mistakes as it struggles to distinguish between certain features that differentiate the two classes. For example, some sandals have more straps that causes them to resemble sneaker's closed shoe design. Likewise, some sneakers have line designs on them which resembles the straps on sandals. Sneakers also often have thicker soles, but some sandals may have similar sole structures.

D18 [2 marks] LR4's estimated weights (parameters)





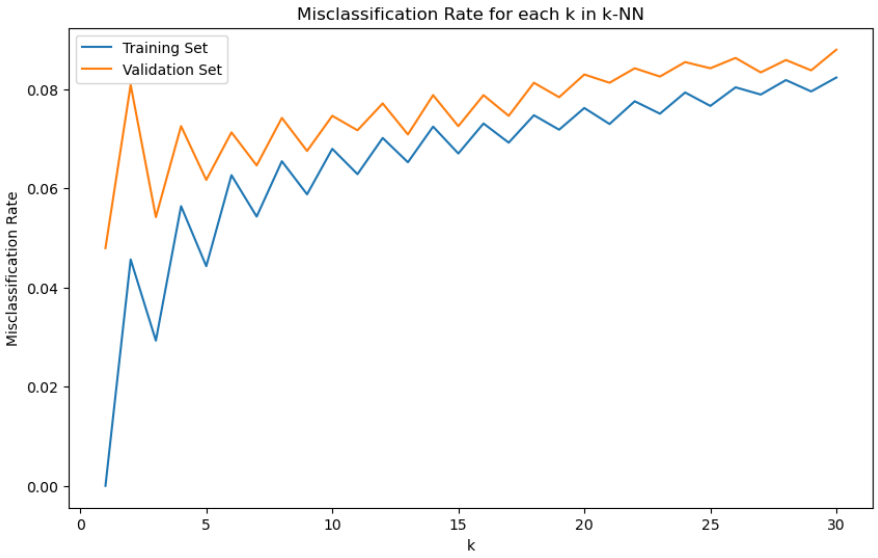


D19 [3 marks] Interpretation of above results

The blue and red colours represent the weights of the coefficients in the LR4 model, which is trained to classify sandals. In the plot, the blue areas indicate negative weights, whereas the red areas indicate positive weights. The intensity of the blue and red colours reflects the magnitude of the weights, with darker shades representing larger magnitudes. Positive weights indicate that higher pixel values contribute to predicting sandals, while negative weights indicate that higher pixel values contribute to predicting sneakers. The model seems to have learned to distinguish between sandals and sneakers using the images, with higher positive weights for pixels corresponding to features like straps or open areas (indicative of sandals), and higher negative weights for pixels corresponding to features like laces or closed areas (indicative of sneakers).

5 Comparing models

D20 [3 marks] Plot of fraction of misclassifications for each k-value



D21 [4 marks] Interpretation of above results

When k equals to 1, the misclassification rate for the validation set is significantly higher than the training set. This means that the model is overfitting on the training data. As k increases, the gap decreases and overfitting reduces as the model tends to become more biased. However, the misclassification rates for both the training and validation set increases. This suggests that the model is underfitting. I would choose a k value of 13 to balance the trade-off between bias and variance. This model has significantly lower misclassification rates than the model in section 4.2.

D22 [2 marks] k-NN performance for chosen k-value

k-NN Model for k = 13

Confusion Matrix:

		Actual Values	
		Positive	Negative
Predicted Values	Positive	845	3
	Negative	155	997

Precision : 0.996
Recall : 0.845
False positive rate : 0.003

D23 [2 marks] Comparing k-NN performance with Logistic Regression Models

13-NN model has a precision of 0.996, recall of 0.845, and false positive rate of 0.003. The 13-NN model outperforms all logistic regression models in terms of precision and false positive rate, but its recall is lower compared to LR3 and LR4. This suggests that while 13-NN is very good at classifying true positives, it may miss some positive cases that the logistic regression models can capture. The logistic regression models has overall good generalisation capacities, showing a progressive improvement from LR1 to LR4. The 13-NN model also shows strong performance, particularly in precision, indicating its ability to generalise well to unseen data.

6 Exploring the ML pipeline

D24 [5 marks]

Feature engineering techniques to extract more meaningful features such as edge detection, texture analysis, or shape analysis. This can better capture the distinct characteristics of sandals and sneakers.

Dimensionality reduction techniques such as Principal Component Analysis (PCA) to reduce the number of features while retaining important information, to reduce overfitting and improve generalisation. Also can use t-distributed Stochastic Neighbour Embedding (t-SNE) to visualise the high-dimensional feature space and identify clusters that correspond to different classes.

Implement early stopping by monitoring the model's performance on the validation set and stopping the training process when the performance starts to degrade. This can help to prevent overfitting the model to the training data.

Use mini-batch gradient descent instead of batch gradient descent or stochastic gradient descent. Mini-batch gradient descent computes the gradient on small random sets of instances at each step, which can lead to faster convergence and better generalisation ability compared to batch or stochastic gradient descent.