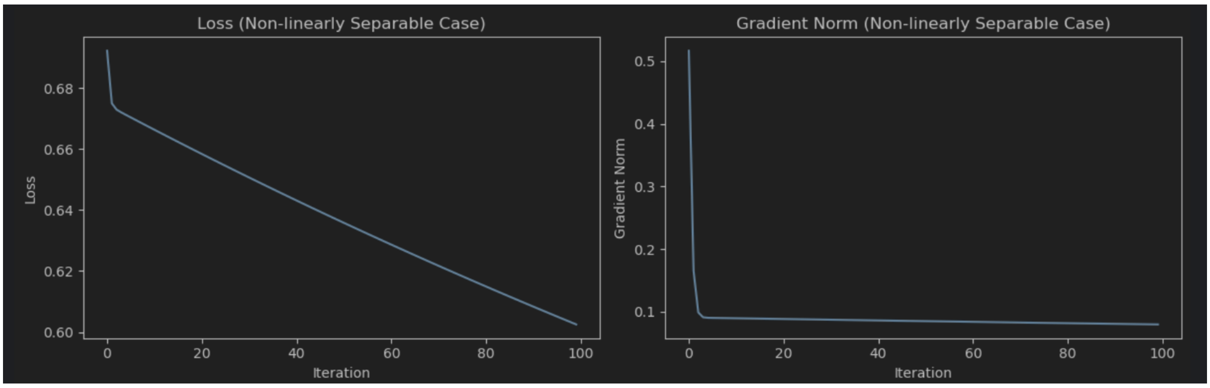
Lab 3

Ruoxin Jia
rj3u25@soton.ac.uk

October 24, 2025

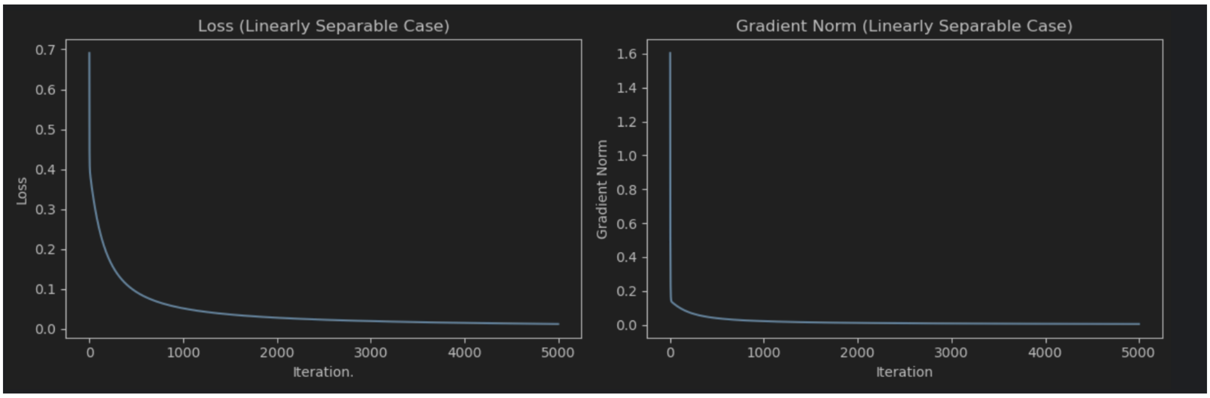
Part 1: Logistic Regression

Task 1: Non-Linearly Separable Data



The logistic regression model was first trained on a non-linearly separable dataset.

Task 2: Linearly Separable Data



Next, the model was trained on linearly separable dataset. The loss and gradient norm plots are shown below.

Analysis of Linearly Separable Case

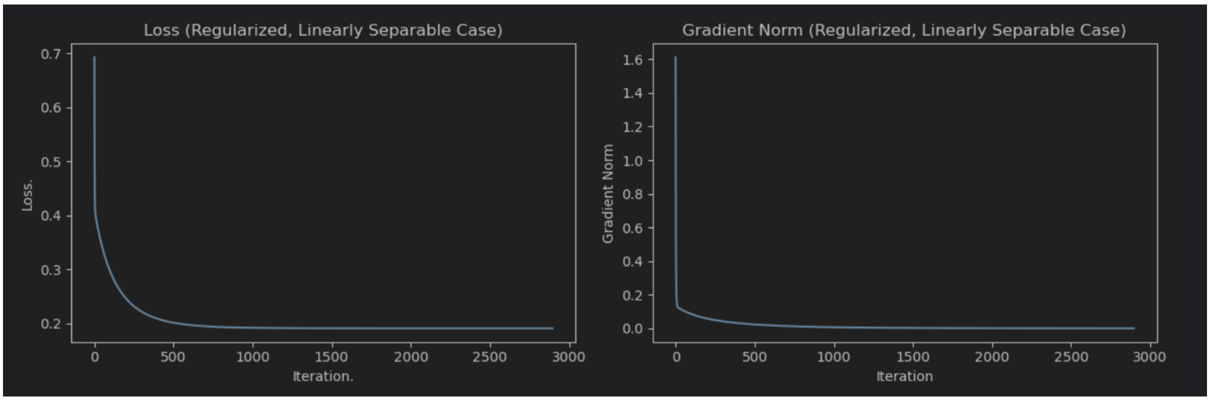
A distinct behavior is observed. The loss decreases to a very small value (**0.0126**) and the gradient norm also becomes very small (**0.0047**), but neither converges fully to zero. Most notably, the magnitude of the weight vector grows significantly to **8.7353** and would continue to grow if not for the 5000 iteration limit.

This phenomenon occurs because the logistic loss function for perfectly separable dataset can be minimized indefinitely by increasing the magnitude of the weights. As the weights w grow larger, the term $w^T x$ goes to $\pm\infty$, which pushes the sigmoid function's output $\sigma(w^T x)$ closer and closer to 1 (for class 1) or 0 (for class 0).

This makes the model more confident and pushes the log-loss $-\log(p(y_i|x_i))$ ever closer to 0. Since the theoretical optimal solution is at infinity, the gradient descent algorithm continuously updates the weights, causing their magnitude to grow without bound. The gradient never vanishe because the loss never truly reaches zero.

Task 3: L2 Regularization on Linearly Separable Data

To address the issue from Task 2, L2 regularization was applied to the model on the same linearly separable dataset.



Analysis of Regularized Model

The L2 regularization term adds a penalty ($\lambda||w||^2$) to the loss function, which penalizes large weights. This penalty term directly counteracts the problem observed in the non-regularized case.

Comparing this to the non-regularized version, the effect is clear:

- **Weight Magnitude:** The final weight magnitude (**3.8243**) is significantly smaller and more stable than in the non-regularized case (**8.7353**). The penalty successfully "pulls" the weights back from infinity.
- **Convergence:** The gradient norm converges effectively to zero (**0.0001**), indicating that the optimizer has found a true, finite minimum for the new, regularized loss function.
- **Final Loss:** The final loss (**0.1906**) is higher than the non-regularized loss (0.0126). This is expected, as the loss now includes the penalty term, and the model is prevented from becoming "overly confident."

Regularization successfully solves the convergence problem for linearly separable data by introducing a trade-off between classification accuracy and model complexity (weight size).

Task 4: Final Model Evaluation

The final regularized model was evaluated on the test set, yielding strong performance metrics.

Classification Report

The precision, recall, and f1-score are all high, with an overall accuracy of **97%**.

	precision	recall	f1-score	support
0	1.00	0.94	0.97	17
1	0.93	1.00	0.96	13
accuracy			0.97	30
macro avg	0.96	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

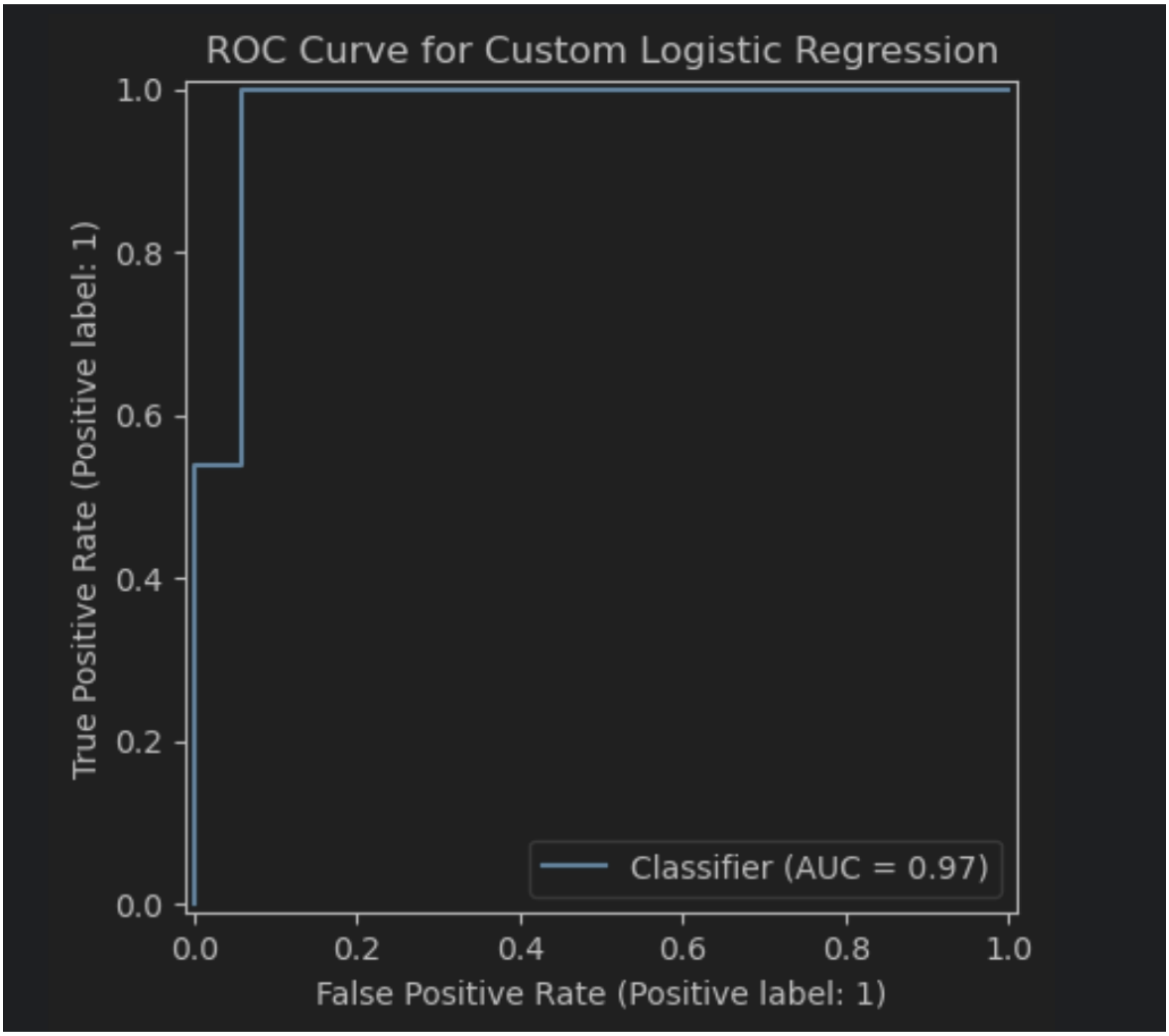
Confusion Matrix

The confusion matrix shows excellent results, with 16 true negatives, 13 true positives, and only one misclassification (a false positive).

$$\begin{bmatrix} 16 & 1 \\ 0 & 13 \end{bmatrix}$$

ROC Curve and AUC Score

The model's ability to distinguish between the two classes is excellent, confirmed by an Area Under the Curve (AUC) score of **0.9729**. The ROC curve plot is shown in Figure below.



Part 2: Naive Bayes

Task 1: Manual Spam Classification

1. Class Priors

$$P(\text{Spam}) = \frac{2}{5} = 0.4$$
$$P(\text{Ham}) = \frac{3}{5} = 0.6$$

2. Vocabulary and Word Likelihoods

Vocabulary size $V = 14$

Word counts:

- Spam: total words = 7
- Ham: total words = 13

With Laplace smoothing ($\alpha = 1$):

$$
\begin{aligned}
P(\text{"free"}|\text{Spam}) &= \frac{1+1}{7+14} = \frac{2}{21} \\
P(\text{"call"}|\text{Spam}) &= \frac{1+1}{7+14} = \frac{2}{21} \\
P(\text{"for"}|\text{Spam}) &= \frac{0+1}{7+14} = \frac{1}{21} \\
P(\text{"you"}|\text{Spam}) &= \frac{0+1}{7+14} = \frac{1}{21}
\end{aligned}
$$

$$
\begin{aligned}
P(\text{"free"}|\text{Ham}) &= \frac{0+1}{13+14} = \frac{1}{27} \\
P(\text{"call"}|\text{Ham}) &= \frac{1+1}{13+14} = \frac{2}{27} \\
P(\text{"for"}|\text{Ham}) &= \frac{0+1}{13+14} = \frac{1}{27} \\
P(\text{"you"}|\text{Ham}) &= \frac{2+1}{13+14} = \frac{3}{27}
\end{aligned}
$$

3. Posterior Probabilities

$$
\begin{aligned}
\text{Score(Spam)} &\propto \frac{2}{5} \times \frac{2}{21} \times \frac{2}{21} \times \frac{1}{21} \times \frac{1}{21} \approx 8.227 \times 10^{-6} \\
\text{Score(Ham)} &\propto \frac{3}{5} \times \frac{1}{27} \times \frac{2}{27} \times \frac{1}{27} \times \frac{3}{27} \approx 6.774 \times 10^{-6}
\end{aligned}
$$

4. Classification

Since Score(Spam) > Score(Ham), the message "free call for you" is classified as **SPAM**.

Task 2: Real-World Text Classification Report

	precision	recall	f1-score	support
sci.space	0.97	0.99	0.98	394
talk.religion.misc	0.99	0.96	0.97	251
accuracy			0.98	645
macro avg	0.98	0.98	0.98	645
weighted avg	0.98	0.98	0.98	645

Task 3: Manual Verification

- Model's Prediction: 'talk.religion.misc'
- Manual Log-Score for sci.space: -698.6741132
- Manual Log-Score for talk.religion.misc: -617.91260235

Confirmation: The manual calculation confirms the model's prediction, as the log-score for 'talk.religion.misc' (-617.91) is higher (less negative) than for 'sci.space' (-698.67).