Multi-Dimensional Logistic Regression Analysis
From 1D Point Separation to 3D Decision Surfaces

Logistic Regression Project Report

December 2025

Abstract

This project explores the application of Logistic Regression across three distinct datasets of increasing dimensionality (1D, 2D, and 3D). The primary objective was to implement a classification model that determines the optimal decision boundary—represented as a point, a line, or a surface—to separate binary classes. Using numerical optimization techniques and sigmoid activation, the project successfully demonstrates how classification complexity scales with feature dimensionality and how non-linear boundaries can be captured through feature engineering.

1 Introduction

The core objective of this project was to achieve effective binary classification by separating labeled data points using the Binary Cross-Entropy (Log-Loss) cost function. The study was designed to observe the mathematical and visual adaptation of the model as data transitions from a simple one-dimensional line to a complex three-dimensional spatial volume.

A central focus was understanding the **Sigmoid Function** as the probabilistic heart of the model. The Sigmoid function transforms a linear predictor Z (the logit) into a probability value between 0 and 1. As the dimensionality of the dataset transitions, the complexity of Z evolves:

- **1D Transition:** Z is a linear equation of one variable ($Z = \beta_0 + \beta_1 x$), where the decision boundary ($Z = 0$) is a single **point**.
- **2D Transition:** Z expands to include multiple features and quadratic terms ($Z = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2$). Here, the point of separation evolves into a **line** or a **circular curve**.
- **3D Transition:** Z becomes a multivariate quadratic logit ($Z = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 z + \beta_4 x^2 + \beta_5 y^2 + \beta_6 z^2$). The separation logic reaches its highest complexity, manifesting as a **3D decision surface** (such as an ellipsoid or cylinder).

2 Theory

The classification framework relies on mapping the logit Z through the Sigmoid activation function and minimizing the error using Binary Cross-Entropy.

2.1 The Sigmoid Function

The probability of a sample belonging to Class 1 is given by:

$$P(y = 1|x) = \sigma(Z) = \frac{1}{1 + e^{-Z}} \tag{1}$$

2.2 Binary Cross-Entropy Loss

To optimize the weights (β_n), we minimize the average Log-Loss:

$$J(\beta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{2}$$

where $\hat{y}$ is the predicted probability and y is the actual label.

3 Implementation

The implementation utilized Python's `SciPy.optimize.minimize` for parameter estimation. Below is the representative logic for the 3D Multivariate Logistic Regression:

```python
def binary_cross_loss(params, x, y, z, m):
    b0, b1, b2, b3, b4, b5, b6 = params
    # Multivariate Quadratic Predictor
    Z = b0 + b1*x + b2*y + b3*z + b4*x**2 + b5*y**2 + b6*z**2
    # Sigmoid Activation
    m_hat = 1 / (1 + np.exp(-Z))
    # Log-Loss Calculation
    loss = -(m * np.log10(m_hat + 1e-5) + (1 - m) * np.log10(1 - m_hat
   + 1e-5))
    return np.mean(loss)
```

Listing 1: 3D Logistic Regression Optimization

4 Results

4.1 Logistic Regression: 1 and 2 Dimensional Data

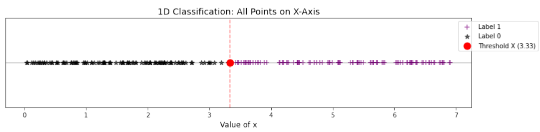

Figure 1: Point of Separation

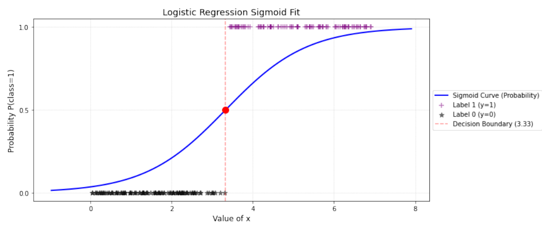

Figure 2: 1D Sigmoid Curve

Figure 1 shows the points as $+$ and $\star$ separated by red dot $\bullet$ along x-axis. Figure 2 depicts the sigmoid function's graph.

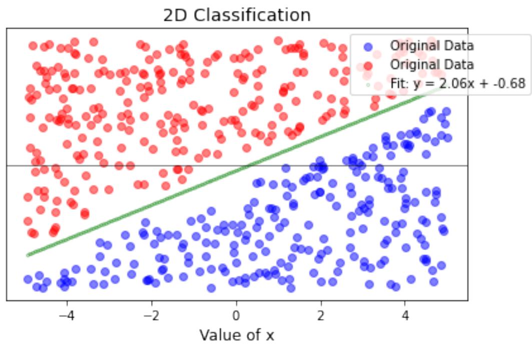

Figure 3: Line of Separation

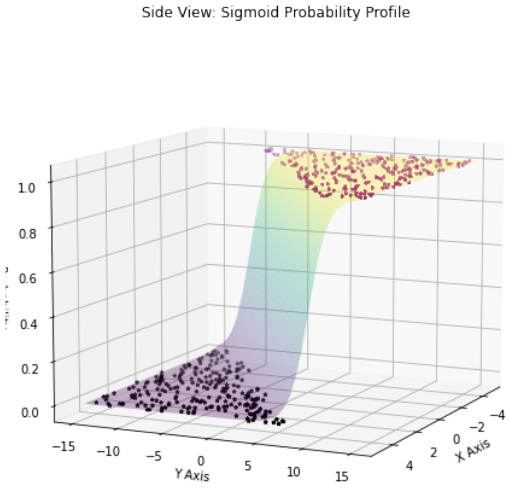

Figure 4: 2D Sigmoid Curve

Figure 3 shows the green line as separating line x-y plane. Figure 4 depicts the sigmoid function's graph in 3D.

4.1.1 Effect of Noise on Data

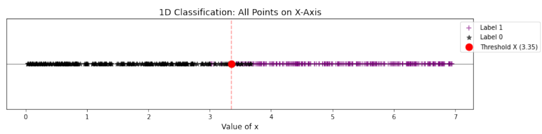

Figure 5: Point of Separation

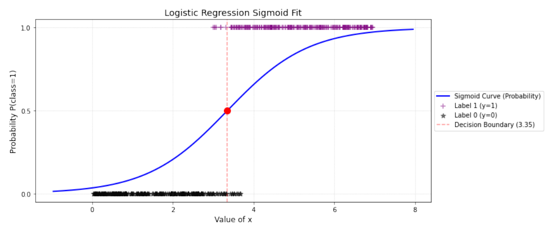

Figure 6: 1D Sigmoid Curve

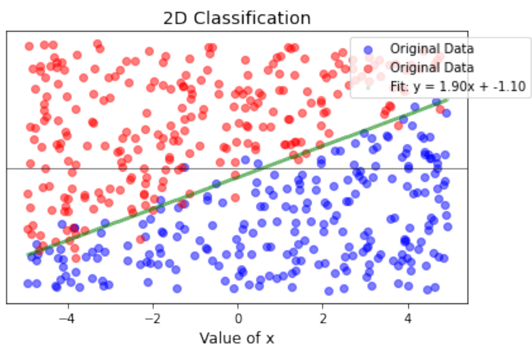

Figure 7: Line of Separation

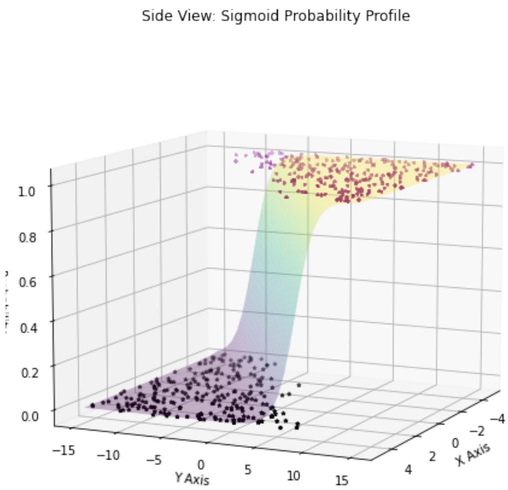

Figure 8: 2D Sigmoid Curve

Figure 5-8 shows the how greater noise leads to data overlaping.

4.2 Logistic Regression: 2D Circular Dimensional Data

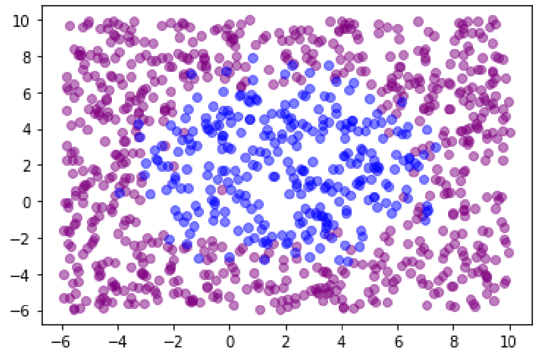

Figure 9: 3D Distribution of Labeled Data

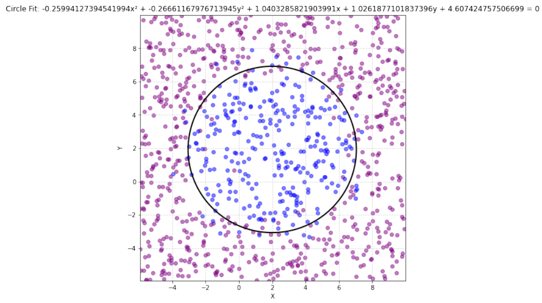

Figure 10: Cylindrical Surface as Separating Surface

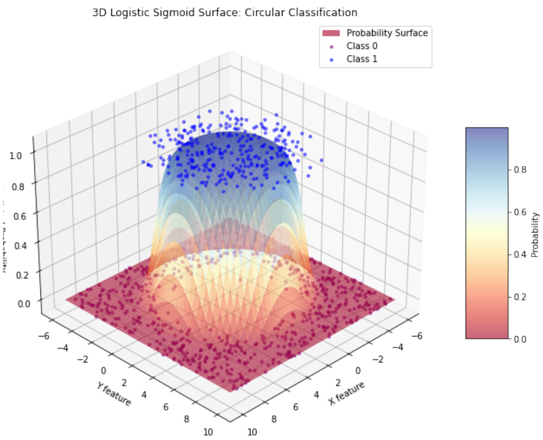

Figure 11: Top View

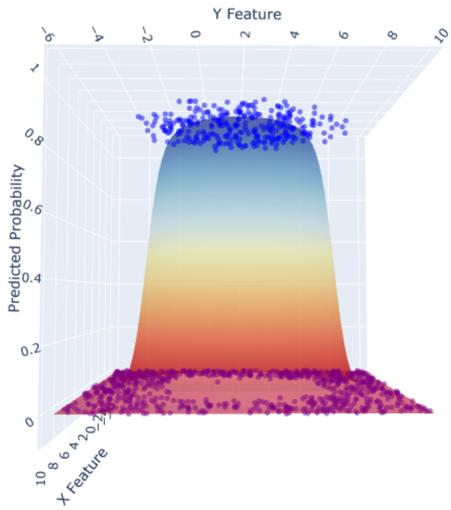

Figure 12: View depicting Red Labeled Data

Figure 9-12 shows the 2D distribution of labeled data and how a circle act as separating curve for the labeled data.

4.3 Logistic Regression: 3D Dimensional Data

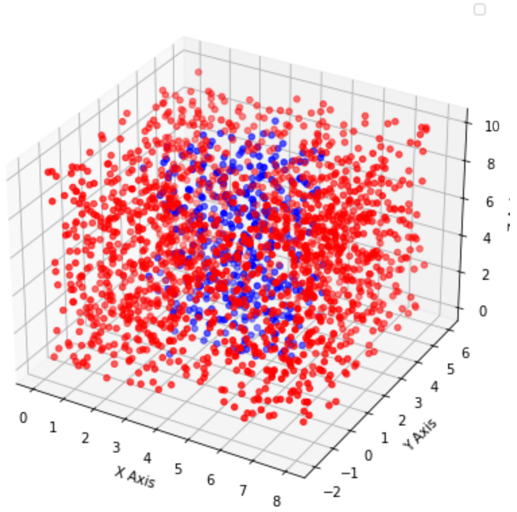

Figure 13: 3D Distribution of Labeled Data

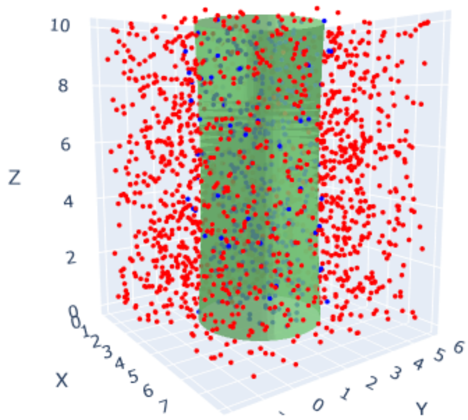

Figure 14: Cylindrical Surface as Separating Surface

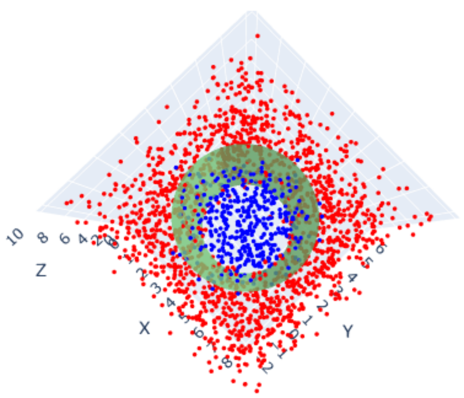

Figure 15: Top View

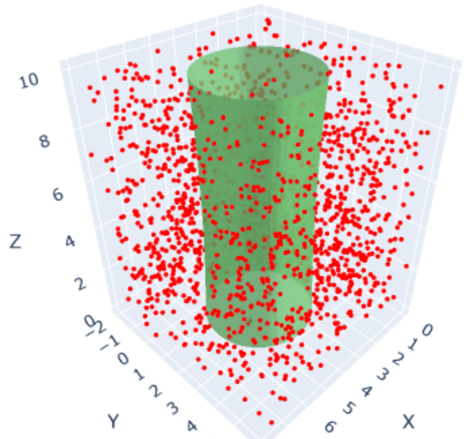

Figure 16: View depicting Red Labeled Data

Figure 13-16 shows the 3D distribution of labeled data and how a cylindrical surface act as separating surface for the labeled data.

Dimension	Data Structure	Boundary Type	Logit Form (Z)
1D	Linear / Overlapping	Point	$\beta_0 + \beta_1 x$
2D	Circular Clusters	Line / Curve	$\beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2$
3D	Spatial Volume	Surface	$\sum \beta_i X_i + \sum \beta_j X_j^2$

5 Discussion

The transition from 1D to 3D highlighted the necessity of **Feature Expansion**. While 1D data could be separated by a single weight, the circular 2D data and the volumetric 3D data required quadratic terms to define "closed" boundaries. In the 3D phase, we specifically analyzed the significance of the z and z^2 coefficients to determine if "height" influenced the classification, effectively uncovering the hidden 3D geometry of the dataset.

6 Conclusion

This project successfully demonstrated that Logistic Regression, when paired with appropriate feature engineering and optimization of the Binary Cross-Entropy loss, is a powerful tool for finding hidden boundaries in multi-dimensional space. We successfully mapped the evolution of classification from a 0D point to a 2D surface.