Understanding Loss Functions in Linear and Logistic Regression:

By: Hind Al Araimi

Introduction

In machine learning, a loss function is a mathematical tool used to measure how well a model's predictions match the actual values. It acts as a guide for the learning process: the lower the loss, the better the model.

Two common models—Linear Regression and Logistic Regression—use different loss functions because they solve different types of problems:

- Linear Regression predicts continuous values (e.g., house prices).
- Logistic Regression predicts probabilities for classification tasks (e.g., spam or not spam).

1- Loss Function in Linear Regression

Purpose:

To minimize the difference between predicted and actual numeric values.

Common Loss Function:

$$\text{Mean Squared Error} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where:

- yi is the actual value.
- y^i is the predicted value.
- n is the number of observations.

Why MSE?

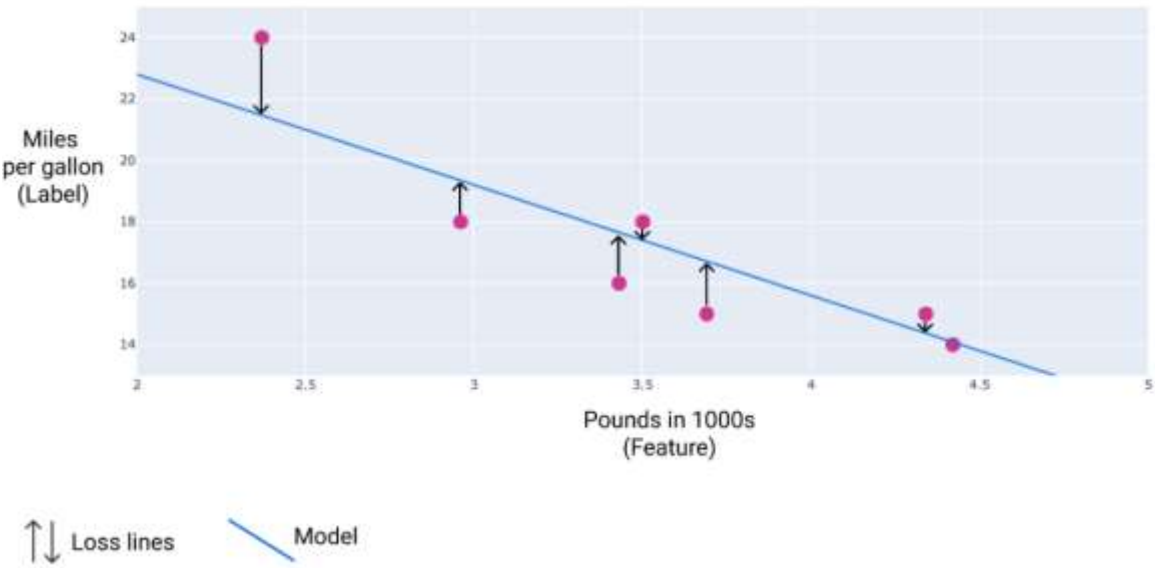
- Penalizes large errors more than small ones (squares the errors).
- Smooth and differentiable—great for gradient-based optimization.

Example:

If the actual values are [2, 4] and the predictions are [3, 5]:

$$\text{MSE} = \frac{1}{2}\left((2-3)^2 + (4-5)^2\right) = \frac{1}{2}(1+1) = 1$$

📉 The following visual shows a **linear regression model** (blue line) with actual data points (pink dots). The **vertical arrows** represent the errors — the distance between predicted and actual values — which are squared and averaged in the MSE formula.



2- Loss Function in Logistic Regression

Purpose:

To minimize the difference between predicted **probabilities** and actual **binary labels (0 or 1)**.

Common Loss Function:

Binary Cross-Entropy (Log Loss)

$$\text{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right]$$

Where:

- y is the actual label (0 or 1).
- p is the predicted probability of class 1.

Why Cross-Entropy?

- It heavily penalizes **confident and incorrect predictions**.
- It works well with outputs from a **sigmoid activation function**, which maps predictions to the [0, 1] range.

Example:

- If the actual label is 1 and the model predicts a probability of 0.9:

$$\text{Loss} = -[1 \cdot \log(0.9) + 0 \cdot \log(1 - 0.9)] = -\log(0.9) \approx 0.105$$

If it predicted 0.1 instead:

$$\text{Loss} = -\log(0.1) \approx 2.302$$

🧠 Comparison Summary:

Feature	Linear Regression	Logistic Regression
Problem Type	Regression	Classification
Output	Continuous	Probability (0 to 1)
Common Loss Function	Mean Squared Error (MSE)	Binary Cross-Entropy (Log Loss)
Sensitive to Outliers	Yes (squares error)	Yes (penalizes confident wrong preds)
Prediction Formula	y^=wx+b	y^=σ(wx+b)

📌 **Conclusion**

The choice of loss function is crucial because it directly impacts **how the model learns**. Linear regression uses **MSE** because it cares about numerical distances, while logistic regression uses **cross-entropy** because it cares about how well probabilities match binary labels.

Understanding these functions allows us to **design better models**, optimize them properly, and interpret results more meaningfully.

References:

Google Developers. (n.d.). *Logistic regression: Loss and regularization*. Google. Retrieved May 17, 2025, from https://developers.google.com/machine-learning/crash-course/logistic-regression/loss-regularization

GeeksforGeeks. (2022, December 14). *Mean squared error*. Retrieved May 17, 2025, from https://www.geeksforgeeks.org/mean-squared-error/

GeeksforGeeks. (2023, May 9). *Binary cross entropy / log loss for binary classification*. Retrieved May 17, 2025, from https://www.geeksforgeeks.org/binary-cross-entropy-log-loss-for-binary-classification/

Ejable. (n.d.). *Logistic and linear regression*. Ejable. Retrieved May 17, 2025, from https://www.ejable.com/tech-corner/ai-machine-learning-and-deep-learning/logistic-and-linear-regression/