Logistic Regression Explained

Logistic regression is a **supervised learning algorithm** used for **binary classification problems**, where the outcome is dichotomous (e.g., yes/no, true/false, 0/1). Unlike linear regression, which predicts a continuous value, logistic regression predicts the **probability** of a data point belonging to a particular class.

Core Concepts

1. **Logistic Function (Sigmoid Function):** Logistic regression uses the **sigmoid function** to map predictions to a probability range between 0 and 1.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

   where z is the linear combination of input features:

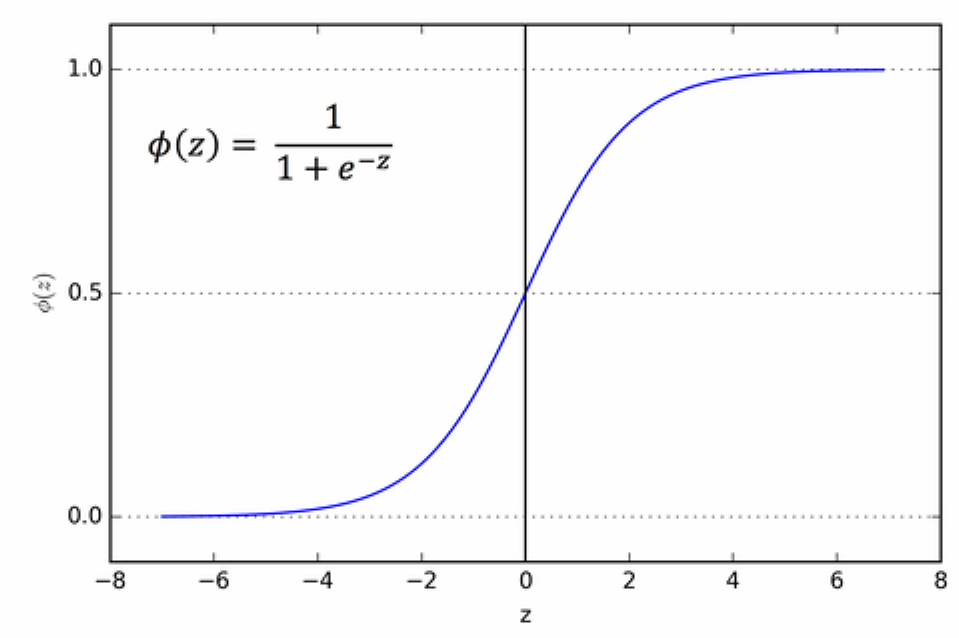
$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

   Here:

   - β_0 is the intercept.
   - $\beta_1, \beta_2, \ldots, \beta_n$ are coefficients of the model.
   - $x_1, x_2, \ldots, x_n$ are the input features.

2. **Probability Prediction:** The sigmoid function outputs a probability:

$$P(y = 1|x) = \sigma(z) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{n} \beta_i x_i)}}$$



3. Loss Function for Logistic Regression

The **loss function** in logistic regression quantifies the difference between the predicted probabilities and the actual labels. It serves as the objective function that the model minimizes during training. For logistic regression, the **log-loss** or **binary cross-entropy loss** is used.

Why is the Loss Function Needed?

1. Logistic regression predicts probabilities ($\hat{y}$) rather than direct classes.
2. The loss function ensures that the model assigns high probabilities to correct classes and penalizes wrong predictions, especially when they are confident but incorrect.

Mathematics of Log-Loss

The loss function for a single prediction in binary logistic regression is:

$$\text{Loss}(\hat{y}, y) = \begin{cases} -\log(\hat{y}), & \text{if } y = 1 \\ -\log(1-\hat{y}), & \text{if } y = 0 \end{cases}$$

This can be combined into a single formula:

$$\text{Loss}(\hat{y}, y) = -\left[y \log(\hat{y}) + (1-y)\log(1-\hat{y})\right]$$

For N samples, the average loss (Log-Loss) is:

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

How It Works

1. **For $y = 1$:**
   The term $-\log(\hat{y})$ penalizes the model when $\hat{y}$ (predicted probability of class 1) is small.
2. **For $y = 0$:**
   The term $-\log(1-\hat{y})$ penalizes the model when $\hat{y}$ (predicted probability of class 1) is large.

This encourages the model to assign high probabilities to the correct class and low probabilities to the incorrect one.

Example

Data

Sample	Actual Label (y)	Predicted Probability ($\hat{y}$)
1	1	0.9
2	0	0.2
3	1	0.4
4	0	0.8

Loss Calculation

1. For Sample 1 ($y = 1, \hat{y} = 0.9$):

$$\text{Loss} = -\log(0.9) = 0.105$$

2. For Sample 2 ($y = 0, \hat{y} = 0.2$):

$$\text{Loss} = -\log(1 - 0.2) = -\log(0.8) = 0.223$$

3. For Sample 3 ($y = 1, \hat{y} = 0.4$):

$$\text{Loss} = -\log(0.4) = 0.916$$

4. For Sample 4 ($y = 0, \hat{y} = 0.8$):

$$\text{Loss} = -\log(1 - 0.8) = -\log(0.2) = 1.609$$

Average Log Loss

$$\text{Log Loss} = \frac{0.105 + 0.223 + 0.916 + 1.609}{4} = 0.713$$

Interpreting Log Loss

- **Lower Log Loss:** Indicates better model performance.
- **High Log Loss:** Implies the model is making incorrect or overconfident predictions.
- **Perfect Predictions:** Achieve a log loss of 0 when $\hat{y} = 1$ for $y = 1$ and $\hat{y} = 0$ for $y = 0$.

Logistic Regression Using the Gradient Descent Approach: A Step-by-Step Example

In this example, we demonstrate how logistic regression works using the **gradient descent** approach for optimization. The goal is to iteratively update the coefficients ($\beta_0, \beta_1, \ldots$) to minimize the **log-loss function.**

Problem Setup

Suppose we are predicting whether a student passes ($y = 1$) or fails ($y = 0$) based on their study hours (x).

Training Data

Hours Studied (x)	Passed (y)
2	0
4	0
6	1
8	1

---

1. Logistic Regression Model

The logistic regression model is:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{where } z = \beta_0 + \beta_1 x$$

- $\hat{y}$: Predicted probability of passing.
- β_0: Intercept.
- β_1: Coefficient for x (study hours).

---

2. Loss Function

The **log-loss function** for m data points is:

$$L(\beta_0, \beta_1) = -\frac{1}{m} \sum_{i=1}^{m} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right]$$

---

3. Gradients

To minimize the loss, compute the gradients with respect to β_0 and β_1:

- Gradient for β_0:

$$\frac{\partial L}{\partial \beta_0} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)$$

- Gradient for β_1:

$$\frac{\partial L}{\partial \beta_1} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) x_i$$

---

4. Gradient Descent Update Rule

Update the coefficients:

$$\beta_j := \beta_j - \alpha \frac{\partial L}{\partial \beta_j}$$

where:

- α: Learning rate.

---

5. Example Walkthrough

Initialize Parameters

- $\beta_0 = 0, \beta_1 = 0$
- Learning rate (α) = 0.1

Iteration 1

1. **Compute z:**

$$z_i = \beta_0 + \beta_1 x_i \quad \text{(Initially, } z = 0 \text{ for all samples).}$$

2. **Compute Predicted Probabilities $\hat{y}_i$:**

$$\hat{y}_i = \sigma(z_i) = \frac{1}{1 + e^{-z_i}} = 0.5 \quad \text{(for all samples, since } z = 0\text{).}$$

3. **Compute Gradients:**
   - For β_0:

$$\frac{\partial L}{\partial \beta_0} = \frac{1}{4} \sum_{i=1}^{4} (\hat{y}_i - y_i) = \frac{1}{4} [(0.5 - 0) + (0.5 - 0) + (0.5 - 1) + (0.5 - 1)] = -0.25$$

   - For β_1:

$$\frac{\partial L}{\partial \beta_1} = \frac{1}{4} \sum_{i=1}^{4} (\hat{y}_i - y_i) x_i = \frac{1}{4} [(0.5 - 0) \cdot 2 + (0.5 - 0) \cdot 4 + (0.5 - 1) \cdot 6 + (0.5 - 1) \cdot 8] = -1.75$$

4. **Update Parameters:**
   - Update β_0:

$$\beta_0 := \beta_0 - \alpha \frac{\partial L}{\partial \beta_0} = 0 - 0.1 \cdot (-0.25) = 0.025$$

   - Update β_1:

$$\beta_1 := \beta_1 - \alpha \frac{\partial L}{\partial \beta_1} = 0 - 0.1 \cdot (-1.75) = 0.175$$

Iteration 2

1. **Compute z:**

$$z_i = \beta_0 + \beta_1 x_i = 0.025 + 0.175 \cdot x_i$$

   For each x_i:

   - $z_1 = 0.025 + 0.175 \cdot 2 = 0.375$
   - $z_2 = 0.025 + 0.175 \cdot 4 = 0.725$
   - $z_3 = 0.025 + 0.175 \cdot 6 = 1.075$
   - $z_4 = 0.025 + 0.175 \cdot 8 = 1.425$

2. **Compute Predicted Probabilities $\hat{y}_i$:**

$$\hat{y}_i = \frac{1}{1 + e^{-z_i}}$$

   - $\hat{y}_1 = \frac{1}{1+e^{-0.375}} = 0.592$
   - $\hat{y}_2 = \frac{1}{1+e^{-0.725}} = 0.673$
   - $\hat{y}_3 = \frac{1}{1+e^{-1.075}} = 0.745$
   - $\hat{y}_4 = \frac{1}{1+e^{-1.425}} = 0.806$

3. **Compute Gradients** (repeating Step 1 with updated $\hat{y}_i$).
4. **Update Parameters** (repeating Step 2 with new gradients).

Convergence

Repeat the above steps until the loss function converges (stabilizes or reaches a pre-defined tolerance level).

Final Model

After convergence, the coefficients β_0 and β_1 can be used to predict probabilities for new data points. For example:

$$\hat{y} = \sigma(\beta_0 + \beta_1 x)$$