Logistic Regression

By
Dr Ravi Prakash Verma
Professor
Department of CSAI
ABESIT

Logistic Regression

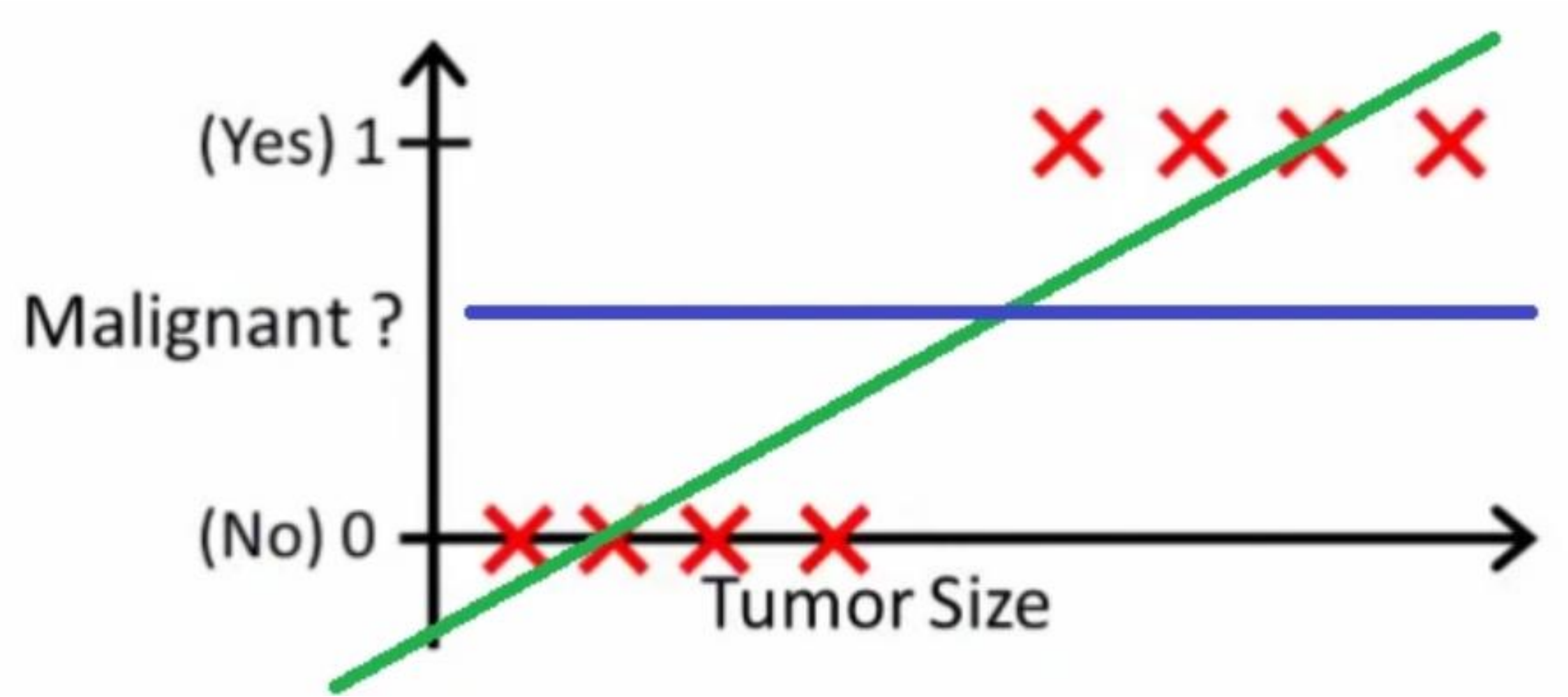
- **Introduction**
- Logistic Regression is a supervised learning algorithm used for classification problems.
- Despite its name, it is not a regression algorithm in the traditional sense (like Linear Regression) but rather a probabilistic classification model.
- It is widely used in binary classification tasks, where the goal is to predict one of two possible outcomes.

Why Not Linear Regression for Classification?

- Linear Regression models continuous values and is unsuitable for classification because:
  1. It can output values beyond the range of 0 to 1, which are not meaningful for classification.
  2. It is sensitive to outliers, which can distort decision boundaries.
  3. It does not model probabilities directly.
- To address these issues, Logistic Regression applies a transformation to map outputs into a probability range of (0,1).

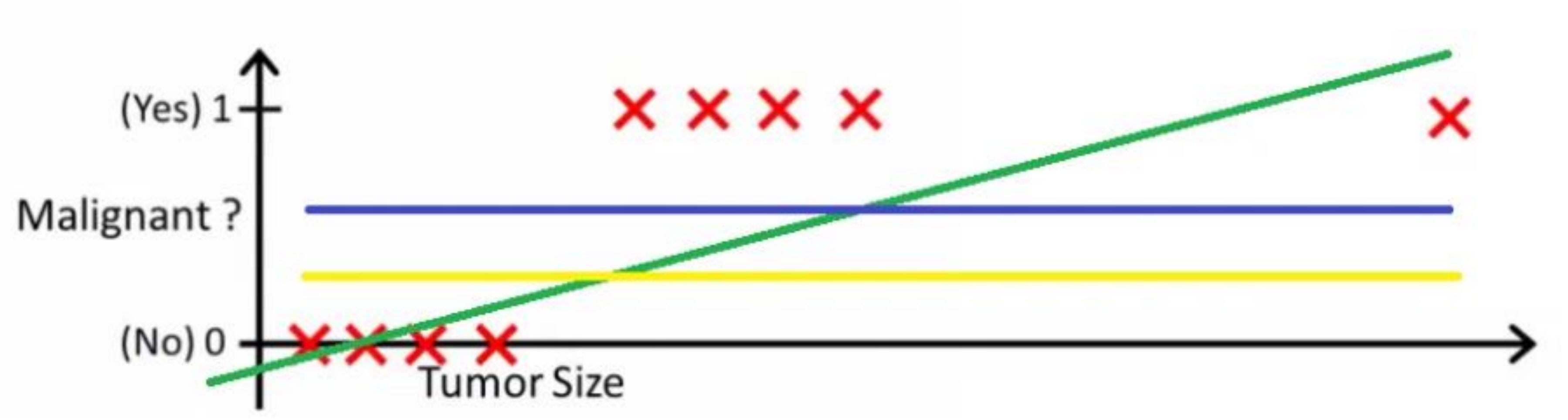
Why Not Linear Regression for Classification?

- Here the threshold value is 0.5, which means if the value of h(x) is greater than 0.5 then we predict malignant tumor (1) and if it is less than 0.5 then we predict benign tumor (0).



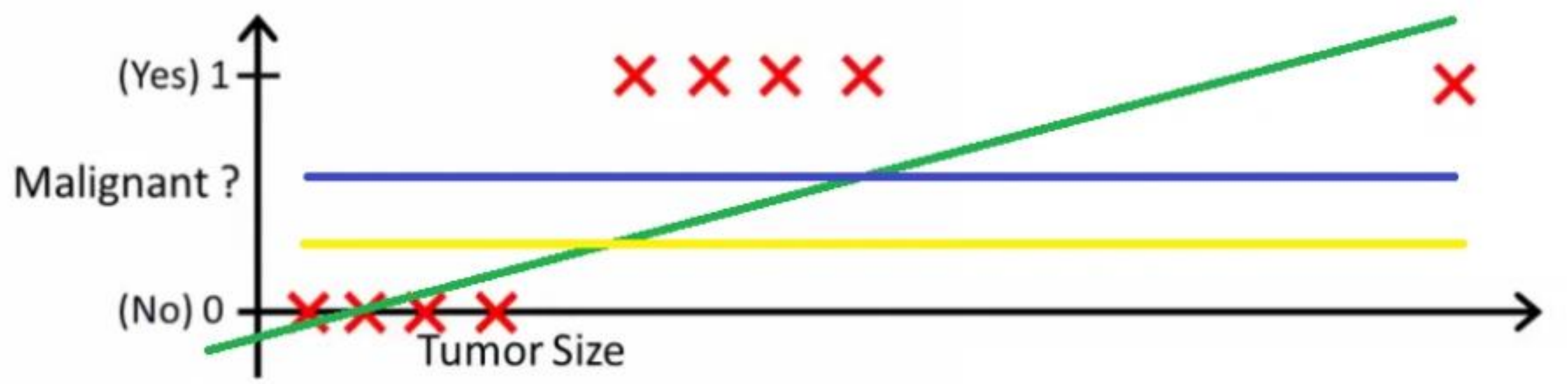
Why Not Linear Regression for Classification?

- Add some outliers in our dataset, now this best fit line will shift to that point. Hence the line will be somewhat like this:



Why Not Linear Regression for Classification?

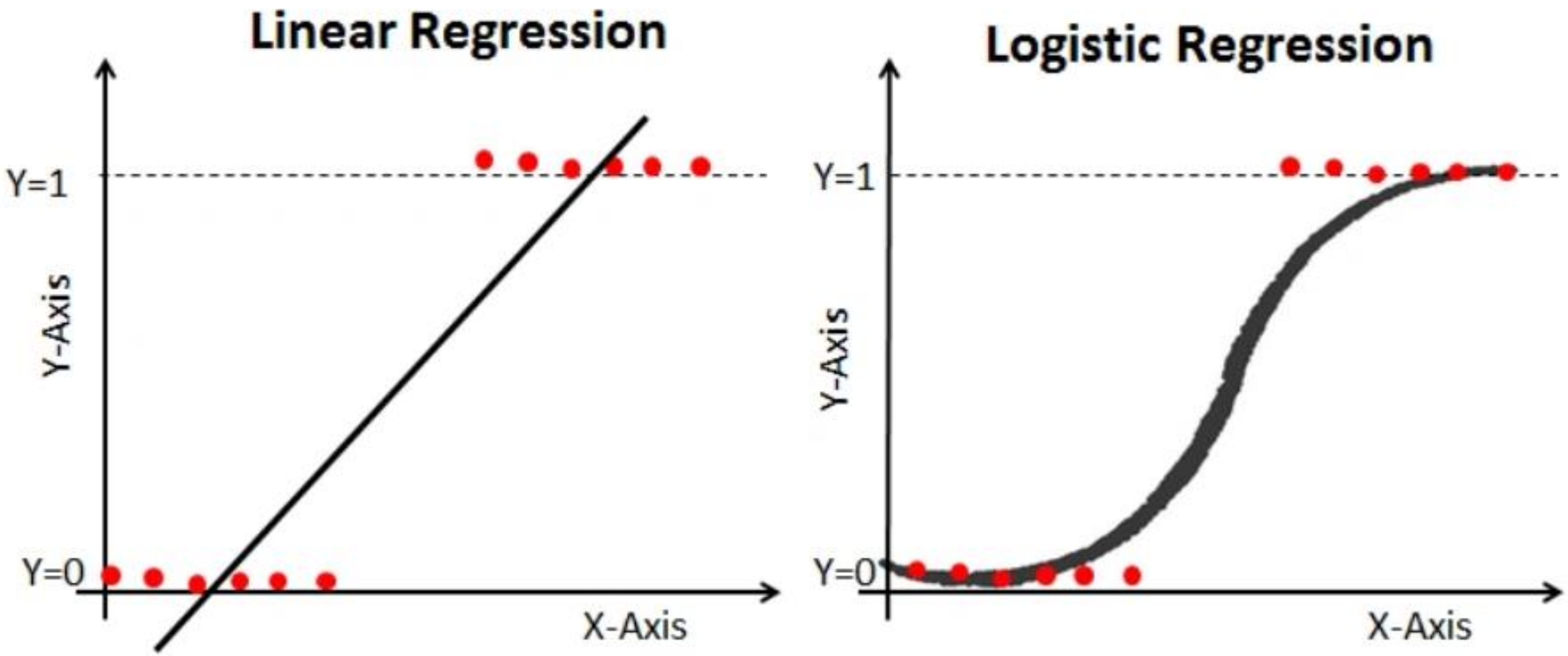
- The blue line represents the old threshold, and the yellow line represents the new threshold, which is maybe 0.2.
- To keep predictions right, lower threshold value.
- Hence, we can say that linear regression is prone to outliers.



Why Not Linear Regression for Classification?

- Now, if h(x) is greater than 0.2, only this regression will give correct outputs.
- Another problem with linear regression is that the predicted values may be out of range.
- Probability can be between 0 and 1, but if we use linear regression, this probability may exceed 1 or go below 0.
- To overcome these problems, we use Logistic Regression, which converts this straight best-fit line in linear regression to an S-curve using the sigmoid function, which will always give values between 0 and 1.

Why Not Linear Regression for Classification?



Types of Logistic Regression

- On the basis of the categories, Logistic Regression can be classified into three types:

  1. **Binomial**: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

  2. **Multinomial**: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

  3. **Ordinal**: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High ".

- “

Assumptions of Logistic Regression

1. **Independent observations:** Each observation is independent of the other. meaning there is no correlation between any input variables.

2. **Binary dependent variables:** It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values. For more than two categories SoftMax functions are used.

3. **Linearity relationship between independent variables and log odds:** The relationship between the independent variables and the log odds of the dependent variable should be linear.

4. **No outliers:** There should be no outliers in the dataset.

5. **Large sample size:** The sample size is sufficiently large

The Logistic (Sigmoid) Function

- The core of Logistic Regression is the **Sigmoid Function**, which maps any real-valued number into the range **(0,1)**:
  - $\sigma(z)=1/1+e^{-z}$
- where:
  - z is the linear combination of input features: $z=w_0+w_1x_1+w_2x_2+...+w_nx_n$
  - w are the weights (or coefficients) associated with each feature.
  - x are the input feature values.
  - The sigmoid function ensures that outputs represent probabilities.
- For example:
  - If $z \rightarrow +\infty$, then $\sigma(z) \approx 1$
  - If $z \rightarrow -\infty$, then $\sigma(z) \approx 0$
  - If $z=0$, then $\sigma(z)=0.5$

Binary Classification with Logistic Regression

Logistic Regression predicts the probability that a given input belongs to a certain class:

$$P(Y=1|X) = \sigma(w^T X) = \frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2+\ldots+w_nx_n)}}$$

where:

- $P(Y=1|X)$ is the probability of belonging to class **1**.
- The decision rule is:
  - If $P(Y=1|X) \geq 0.5$, classify as **1**.
  - If $P(Y=1|X) < 0.5$, classify as **0**.

Cost Function in Logistic Regression

Unlike Linear Regression (which uses Mean Squared Error), Logistic Regression uses **Log Loss (Cross-Entropy Loss)** to measure model performance:

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(h(x_i)) + (1 - y_i)\log(1 - h(x_i))]$$

where:

- m is the number of training examples.
- y_i is the actual label (0 or 1).
- $h(x_i)$ is the predicted probability.

The Log Loss function ensures that:

- If the predicted probability is close to the actual class, the loss is small.
- If the predicted probability is far from the actual class, the loss is large.

Optimization Using Gradient Descent

To minimize the cost function, we use **Gradient Descent**:

1. Compute the gradient of the cost function with respect to the weights w.
2. Update the weights iteratively using:

$$w_j = w_j - \alpha \cdot \frac{\partial J(w)}{\partial w_j}$$

   where:

   - α is the learning rate.
   - $\frac{\partial J(w)}{\partial w_j}$ is the derivative of the cost function.

3. Repeat until convergence.

Alternatively, we can use **Stochastic Gradient Descent (SGD)** or **Adam Optimizer** for faster convergence.

Multi-Class Logistic Regression (Softmax Regression)

For more than two classes, we extend Logistic Regression to **Softmax Regression**:

- Instead of using a single sigmoid function, we use the **Softmax Function**:

$$P(y = k|X) = \frac{e^{w_k^T X}}{\sum_{j=1}^{K} e^{w_j^T X}}$$

  where:

  - K is the number of classes.
  - The denominator ensures probabilities sum to 1.

- The model predicts the class with the highest probability.

Regularization in Logistic Regression

To prevent overfitting, we add **L1 (Lasso) or L2 (Ridge) Regularization**:

1. **L2 Regularization (Ridge Regression)**:

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(h(x_i)) + (1 - y_i)\log(1 - h(x_i))] + \lambda\sum_{j=1}^{n} w_j^2$$

   - Penalizes large weights, preventing overfitting.

2. **L1 Regularization (Lasso Regression)**:

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(h(x_i)) + (1 - y_i)\log(1 - h(x_i))] + \lambda\sum_{j=1}^{n} |w_j|$$

   - Leads to feature selection by shrinking some weights to zero.

Performance Evaluation

Common metrics to evaluate Logistic Regression:

1. **Accuracy** – Percentage of correctly classified instances.
2. **Precision** – $\frac{TP}{TP+FP}$ (focuses on positive predictions).
3. **Recall (Sensitivity)** – $\frac{TP}{TP+FN}$ (important for imbalanced datasets).
4. **F1-score** – Harmonic mean of Precision and Recall.
5. **ROC Curve** – Shows the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR).
6. **AUC (Area Under Curve)** – Measures the classifier's ability to distinguish classes.

Applications of Logistic Regression

- **Medical Diagnosis**: Predicting disease presence (e.g., cancer detection).
- **Fraud Detection**: Identifying fraudulent transactions.
- **Marketing**: Predicting customer conversion.
- **Spam Detection**: Classifying emails as spam or not spam.
- **Credit Scoring**: Determining loan approvals.

Conclusion

- Logistic Regression is a simple yet powerful classification algorithm.
- It provides probabilistic predictions, making it interpretable.
- Regularization techniques help prevent overfitting.
- It works well for binary and multiclass classification (via Softmax).
- Suitable for problems where relationships are linear in log-odds space.

Example Logistic Regression

- Predicting if a Student Passes an Exam

Study Hours (X)	Pass (Y)
1	0
2	0
3	0
4	1
5	1
6	1
7	1

Example Logistic Regression

- Predicting if a Student Passes an Exam

Step 1: Define the Logistic Regression Model

Logistic regression uses the **sigmoid function:**

$$P(Y = 1|X) = \frac{1}{1 + e^{-(wX+b)}}$$

Where:

- X = Study Hours
- Y = Outcome (Pass/Fail)
- w = Weight (to be learned)
- b = Bias (to be learned)

Example Logistic Regression

- Predicting if a Student Passes an Exam

Step 2: Initialize Parameters

We start with:

- $w = 0$
- $b = 0$

Using these initial values, let's compute predictions.

Step 3: Compute Predictions Using Sigmoid

For each X, we compute:

$$\hat{Y} = \frac{1}{1 + e^{-(wX+b)}}$$

Since $w = 0$ and $b = 0$:

$$\hat{Y} = \frac{1}{1 + e^0} = \frac{1}{2} = 0.5$$

For all X, the initial prediction is 0.5.

Example Logistic Regression

- Predicting if a Student Passes an Exam

Step 4: Compute Cost (Log Loss)

The **log loss function** (cost function) is:

$$J = -\frac{1}{m}\sum_{i=1}^{m}\left[Y_i \log(\hat{Y}_i) + (1 - Y_i)\log(1 - \hat{Y}_i)\right]$$

Plugging in our values:

$$J = -\frac{1}{7}\sum[Y\log(0.5) + (1 - Y)\log(0.5)]$$

Since $\log(0.5) = -0.693$:

$$J = -\frac{1}{7}\sum(-0.693) = 0.693$$

This is our initial cost.

Example Logistic Regression

- Predicting if a Student Passes an Exam

Step 5: Compute Gradients

For gradient descent, we update w and b using:

$$\frac{\partial J}{\partial w} = \frac{1}{m} \sum (\hat{Y} - Y) X$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum (\hat{Y} - Y)$$

Computing gradients:

$$\frac{\partial J}{\partial w} = \frac{1}{7} \sum (0.5 - Y) X$$

$$\frac{\partial J}{\partial b} = \frac{1}{7} \sum (0.5 - Y)$$

Using the dataset:

Example Logistic Regression

- Predicting if a Student Passes an Exam

$$\frac{\partial J}{\partial w} = \frac{1}{7}[(0.5-0)(1) + (0.5-0)(2) + (0.5-0)(3) + (0.5-1)(4) + (0.5-1)(5) + (0.5-1)(6) + (0.5-1)(7)]$$

$$= \frac{1}{7}[0.5 + 1 + 1.5 - 2 - 2.5 - 3 - 3.5] = \frac{1}{7}[-8] = -1.14$$

Similarly, for b:

$$\frac{\partial J}{\partial b} = \frac{1}{7}[(0.5-0) + (0.5-0) + (0.5-0) + (0.5-1) + (0.5-1) + (0.5-1) + (0.5-1)]$$

$$= \frac{1}{7}[0.5 + 0.5 + 0.5 - 0.5 - 0.5 - 0.5 - 0.5] = \frac{1}{7}[-0.5] = -0.07$$

Example Logistic Regression

- Predicting if a Student Passes an Exam

Step 6: Update Parameters

Using **gradient descent** with **learning rate** $\alpha = 0.1$:

$$w = w - \alpha \frac{\partial J}{\partial w} = 0 - 0.1(-1.14) = 0.114$$

$$b = b - \alpha \frac{\partial J}{\partial b} = 0 - 0.1(-0.07) = 0.007$$

New parameters:

- $w = 0.114$
- $b = 0.007$

Example Logistic Regression

- Predicting if a Student Passes an Exam
- **Step 7: Repeat Until Convergence**
  - We continue updating w and b in multiple iterations until the cost function J minimizes.

Example Logistic Regression

- Predicting if a Student Passes an Exam

Iteration 2: Compute Predictions

Using the updated values from the previous iteration:

- $w = 0.114$
- $b = 0.007$

We compute predictions using:

$$\hat{Y} = \frac{1}{1 + e^{-(0.114X + 0.007)}}$$

Example Logistic Regression

- Predicting if a Student Passes an Exam

For each X:

Study Hours (X)	Linear Function $wX + b$	Sigmoid Output $\hat{Y}$
1	$0.114(1) + 0.007 = 0.121$	$\frac{1}{1+e^{-0.121}} \approx 0.530$
2	$0.114(2) + 0.007 = 0.235$	$\frac{1}{1+e^{-0.235}} \approx 0.558$
3	$0.114(3) + 0.007 = 0.349$	$\frac{1}{1+e^{-0.349}} \approx 0.587$
4	$0.114(4) + 0.007 = 0.463$	$\frac{1}{1+e^{-0.463}} \approx 0.613$
5	$0.114(5) + 0.007 = 0.577$	$\frac{1}{1+e^{-0.577}} \approx 0.640$
6	$0.114(6) + 0.007 = 0.691$	$\frac{1}{1+e^{-0.691}} \approx 0.666$
7	$0.114(7) + 0.007 = 0.805$	$\frac{1}{1+e^{-0.805}} \approx 0.690$

Example Logistic Regression

- Predicting if a Student Passes an Exam

Iteration 2: Compute Gradients

Using the updated predictions $\hat{Y}$, we recalculate gradients.

$$\frac{\partial J}{\partial w} = \frac{1}{m}\sum(\hat{Y} - Y)X$$

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum(\hat{Y} - Y)$$

Summing for each sample:

$$\frac{\partial J}{\partial w} = \frac{1}{7}[(0.530 - 0)(1) + (0.558 - 0)(2) + (0.587 - 0)(3) + (0.613 - 1)(4) + (0.640 - 1)(5) + (0.666 - 1)(6) + (0.690 - 1)(7)]$$

$$= \frac{1}{7}[0.530 + 1.116 + 1.761 - 1.548 - 1.8 - 2.004 - 2.17]$$

$$= \frac{1}{7}[-4.115] = -0.588$$

Example Logistic Regression

- Predicting if a Student Passes an Exam

Similarly, for b:

$$\frac{\partial J}{\partial b} = \frac{1}{7}[(0.530 - 0) + (0.558 - 0) + (0.587 - 0) + (0.613 - 1) + (0.640 - 1) + (0.666 - 1) + (0.690 - 1)]$$

$$= \frac{1}{7}[0.530 + 0.558 + 0.587 - 0.387 - 0.36 - 0.334 - 0.31]$$

$$= \frac{1}{7}[0.284] = 0.041$$

Example Logistic Regression

- Predicting if a Student Passes an Exam

Iteration 2: Update Parameters

Using **gradient descent** with **learning rate** $\alpha = 0.1$:

$$w = w - \alpha \frac{\partial J}{\partial w} = 0.114 - 0.1(-0.588) = 0.172$$

$$b = b - \alpha \frac{\partial J}{\partial b} = 0.007 - 0.1(0.041) = 0.003$$

New parameters:

- $w = 0.172$
- $b = 0.003$

Example Logistic Regression

- Predicting if a Student Passes an Exam

Further Iterations

We repeat the same process for more iterations until the cost function stabilizes and the model converges.

Iteration	w	b	Cost Function J
1	0.114	0.007	0.693
2	0.172	0.003	0.631
3	0.224	-0.001	0.580
4	0.268	-0.005	0.537
5	0.307	-0.008	0.500
6	0.341	-0.011	0.468
...	...	...	...

Eventually, w **and** b **converge to optimal values**, and the cost function J stops decreasing significantly.

Multivariate Logistic Regression

- So far, we used only **one feature** (study hours). In real-world scenarios, we have **multiple features**.
- The logistic regression equation for **multiple variables** is:

$$P(Y=1|X_1, X_2, ..., X_n) = \frac{1}{1+e^{-(w_1X_1+w_2X_2+...+w_nX_n+b)}}$$

Where:

- $X_1, X_2, ..., X_n$ are features (e.g., study hours, attendance, previous grades)
- $w_1, w_2, ..., w_n$ are weights
- b is bias
- $P(Y=1)$ is the probability that the student passes.

Multivariate Logistic Regression

Example: Predicting Student Pass or Fail with Multiple Features

Study Hours (X_1)	Attendance (X_2)	Pass (Y)
1	2	0
2	3	0
3	1	0
4	4	1
5	5	1
6	6	1
7	7	1

Using **two features**, the equation becomes:

$$\hat{Y} = \frac{1}{1 + e^{-(w_1 X_1 + w_2 X_2 + b)}}$$

Multivariate Logistic Regression

We follow the **same process** as before:

1. Initialize w_1, w_2, b.
2. Compute predictions using the sigmoid function.
3. Calculate cost function J.
4. Compute gradients for w_1, w_2, b.
5. Update parameters using gradient descent.
6. Repeat until convergence.

This generalizes logistic regression to **multiple variables**, allowing us to make more accurate predictions!