**Logistic Regression**

**Introduction**

Logistic Regression is a supervised machine learning algorithm used for classification tasks. Unlike Linear Regression, which predicts continuous values, Logistic Regression predicts the probability that a given input belongs to a particular class. It is widely used for binary classification problems such as spam detection, disease diagnosis, and sentiment analysis.

**What is Logistic Regression?**

* Logistic Regression is a classification algorithm, not regression (despite the name).
* It is used when the dependent variable (target) is categorical (like Yes/No, Spam/Not Spam, 0/1).
* Instead of predicting a continuous value (like Linear Regression), it predicts the probability that an instance belongs to a class.

**Why Logistic Regression?**

* Linear Regression isn’t suitable for classification because predicted values can go outside [0,1].
* Logistic Regression solves this by using the sigmoid function to "squash" predictions into the range [0,1].

**Logistic regression works**

* Logistic (Sigmoid) Function

The logistic function is:

1

σ(Z)= -------------

1+e^-z

* Z= b0​+b1​x1​+b2​x2​+...+bn​xn​
* Output is always between **0 and 1**.

Interpretation:

* If σ(z)>0.5\sigma(z) > 0.5σ(Z)>0.5 → class 1 (Yes/Positive)
* If σ(z)≤0.5\sigma(z) ≤ 0.5σ(Z)≤0.5 → class 0 (No/Negative)

**Logistic Regression Equation**

1

P(y=1∣X)=-----------------------------------------------

1+e^−(b0​+b1​x1​+b2​x2​+...+bn​xn​)

Where:

* P(y=1∣X) = probability of class 1 given input X
* b0 = intercept (bias)
* bi = coefficient for feature xi

**Decision Boundary**

* Logistic regression outputs probabilities.
* A **threshold** (usually 0.5) is applied to decide the final class.
* Example: If P>0.5 → predict **Yes**, else **No**.

**Cost Function (Loss Function)**

Unlike Linear Regression (MSE), Logistic Regression uses Log Loss (Binary Cross-Entropy Loss):

J(0)= -1/m∑(i=1 to m)[y^(i)log(h0(x^(i)))+(1-y^(i))log(1-h0(x^(i)))]

Where:

* m= number of training examples
* y^(i) = actual label (0 or 1)
* hθ(x^(i)) = predicted probability

This ensures penalty is high for wrong confident predictions.

**Optimization**

* We use **Gradient Descent** to minimize the cost function.
* Update rule:

θ := θ- α(∂J(θ)/ ∂θ)

where :

α= learning rate.

**Types of Logistic Regression**

1. Binary Logistic Regression → 2 classes (Yes/No).
2. Multinomial Logistic Regression → more than 2 classes (e.g., predicting fruits: Apple/Orange/Banana).
3. Ordinal Logistic Regression → classes have an order (e.g., ratings: Poor, Average, Good)

**Advantages**

* Easy to implement & interpret
* Works well with linearly separable classes
* Outputs probabilities (not just class labels)
* Less computation than complex models

**Disadvantages**

* Assumes linear relationship between independent variables & log-odds
* Not suitable for complex, non-linear problems
* Sensitive to outliers
* Needs large sample size for stable results

**Applications**

* Email spam detection (Spam/Not Spam)
* Medical diagnosis (Disease/No Disease)
* Customer churn prediction (Leave/Stay)
* Sentiment analysis (Positive/Negative)
* Credit risk prediction (Default/Not Default)