ALGORITHM: LOGISTIC REGRESSION

Logistic Regression is a supervised machine learning algorithm used for classification problems. It predicts the probability that an input belongs to a certain class.

- It is used when the target variable is categorical, such as Yes/No, 0/1, True/False.
- It uses the sigmoid (logistic) function to output probabilities between 0 and 1.
- The model learns a linear boundary and then applies the sigmoid function to map predictions to a probability.
- It is mainly used for binary classification, but can be extended to multiclass problems using methods like One-vs-Rest.
- The decision boundary is based on a threshold (commonly 0.5) to classify the output

Algorithm Type & Objective

- Algorithm type
    - Refers to the category or class the algorithm belongs to base on how it learns from data.
    - It tells how the model operates.
      Ex:
        - Supervised Learning (uses labelled data)
        - Unsupervised Learning (uses unlabelled data)
        - Classification, Regression, Clustering, etc.

  For Logistic Regression:
  Algorithm Type = Supervised Learning → Classification

- Objective.
    - Refers to the goal the algorithm is trying to achieve.
    - It defines what the model is predicting or optimizing for.

      Common objectives:

        - Predicting a class label
        - Estimating a probability
        - Minimizing error or loss

  For Logistic Regression:
  Objective = Estimate the probability that an input belongs to a certain class (e.g., spam or not spam), then classify it based on a threshold.

Mathematical Intuition.

- Logistic Regression models the log-odds of the probability of the default class (usually class 1).
- It applies a linear model:
- It then uses the sigmoid function to squash this value into a range between 0 and
- The output is interpreted as the probability of class 1.
- A threshold (usually 0.5) is applied to decide the final class.

1.Probablity Mapping.

- Binary classification uses the sigmoid function:

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

- Multiclass classification uses the SoftMax function

$$P(y=k \mid \mathbf{x}) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

2.Decision Objective.

- The decision rule in logistic regression is used to convert the predicted probability into a final class label.

3.Training Objective.

- Minimize the cross-entropy loss between predicted possibilities and actual labels:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y^{(i)}\log(p^{(i)}) + (1-y^{(i)})\log(1-p^{(i)})\right]$$

4.Objective.

- Update parameters w and b using gradient descent or other optimization solvers
- Optimization in logistic regression refers to the process of finding the best model parameters (weights w\math{w}w and bias) that minimize the loss function.

$$w_j := w_j - \alpha \cdot \frac{\partial \mathcal{L}}{\partial w_j} \quad , \quad b := b - \alpha \cdot \frac{\partial \mathcal{L}}{\partial b}$$

Key Hyperparameters.

Hyperparameter	Description	Default	Tips
C (inverse of regularization strength)	Controls overfitting; smaller values specify stronger regularization	1.0	Lower for noisy data
penalty	Type of regularization	'l2'	Try 'l1' for sparse data (e.g., text features)
solver	Optimization algorithm	'lbfgs'	Use 'liblinear' for small datasets or 'saga' for L1
max_iter	Max number of iterations	100	Increase if convergence warning occurs
class_weight	Weighs classes differently	None	Use 'balanced' for imbalanced datasets

Strengths & Limitations (Especially in Text Classification)

Strengths:

- Simple, fast, and interpretable.
- Performs well on high-dimensional, sparse data (e.g., TF-IDF vectors from text).
- Works well as a baseline model.
- Probabilistic outputs are useful for ranking and threshold tuning.

Limitations:

- Assumes linearity in feature relationships.
- Struggles with complex non-linear patterns.
- Sensitive to irrelevant features or outliers.
- Can underperform if features are highly correlated or not scaled properly.

When to Use / When Not to Use.

Use When:

- You need a simple, explainable model.
- Data is linearly separable or close.
- You are working with text data (e.g., TF-IDF).
- You want probabilistic outputs for ranking or decision thresholds.

Avoid When:

- The data has complex non-linear relationships.
- Accuracy is critical and other models outperform.
- You need feature interactions or deep structures (try tree-based models or neural networks).