Logistic Regression

---

1. Introduction

- **Logistic Regression** is a **supervised machine learning algorithm** used for **classification**, not regression.
- It predicts the **probability** that a data point belongs to a particular category.
- The output is **discrete** (e.g., 0 or 1, Yes or No).

Examples:

- Email classification → Spam (1) or Not Spam (0)
- Disease prediction → Positive (1) or Negative (0)
- Customer churn → Yes (1) or No (0)

---

2. Difference Between Linear and Logistic Regression

Feature	Linear Regression	Logistic Regression
Purpose	Regression (predict continuous value)	Classification (predict category)
Output	Continuous (e.g., salary, height)	Probability (0 to 1)
Equation	$y = \beta_0 + \beta_1 x$	$h(x) = 1 / (1 + e^{-(\beta_0 + \beta_1 x)})$
Output Range	$(-\infty, +\infty)$	(0, 1)
Cost Function	Mean Squared Error (MSE)	Log Loss / Cross-Entropy
Example	Predicting house prices	Predicting spam vs. non-spam

---

3. Logistic (Sigmoid) Function

The logistic regression model uses the **sigmoid (logistic)** function to map any real-valued number into a probability between **0 and 1**.

h(x)=11+e−(β0+β1x)h(x) = \frac{1}{1 + e^{-(β_0 + β_1x)}}h(x)=1+e−(β0+β1x)1

Where:

- **h(x)** = Predicted probability
- **β_0** = Intercept (bias)
- **β_1** = Coefficients (weights)
- **x** = Independent variable

Interpretation:

- If **h(x) ≥ 0.5**, predict class **1**
- If **h(x) < 0.5**, predict class **0**

---

4. Logit Function (Odds Representation)

Instead of modeling probability directly, logistic regression models the **log-odds** (logit) of the event:

logit(p)=log(1−pp)=β0+β1x

Where:

- **p** is the probability of the event occurring (e.g., class 1).
- The relationship between log-odds and predictors is **linear**.

---

5. Cost Function – Log Loss

The model is trained by minimizing the **Log Loss (Cross-Entropy Loss):**

J(β)=−n1i=1∑n[yilog(y^i)+(1−yi)log(1−y^i)]

Explanation of Terms

- **J(β)** The cost function (or loss) for a given set of model parameters β. It represents the average error across all samples.
- **n** The total number of samples in the dataset.
- **∑i=1n** A summation that iterates through each sample in the dataset, from the 1st to the n-th sample.
- **yi** The actual binary label for the i-th sample (it can only be 0 or 1).
- **y^i** The model's predicted probability that the i-th sample belongs to class 1. This value is always between 0 and 1.
- **log(·)** The natural logarithm. The cost function heavily penalizes predictions that are confident but wrong.

The goal is to **minimize** this cost function.

---

6. Optimization – Gradient Descent

- The algorithm adjusts coefficients (**β_0, β_1, …**) to minimize the cost function.

- Uses **Gradient Descent**, which updates parameters as:

βj:=βj−α∂βj∂J(β)

This update is performed simultaneously for all parameters j=0,1,...,n.

---

7. Types of Logistic Regression

1. **Binary Logistic Regression** – Two possible outcomes (e.g., Yes/No).

2. **Multinomial Logistic Regression** – More than two unordered outcomes (e.g., predicting fruit type).

3. **Ordinal Logistic Regression** – Ordered categories (e.g., ratings: low, medium, high).

---

8. Assumptions of Logistic Regression

1. **Binary or categorical dependent variable**
   - The response variable should be binary (0 or 1) or categorical.
2. **Linear relationship between independent variables and log-odds**
   - The independent variables are linearly related to the logit (log of odds), not directly to the probability.
3. **Independent observations**
   - Observations should be independent of each other (no autocorrelation).
4. **Low multicollinearity**
   - Predictors should not be highly correlated with each other.
   - Use **VIF (Variance Inflation Factor)** to detect multicollinearity.
5. **Large sample size**
   - Logistic regression performs best with sufficient data points to ensure stability of estimates.
6. **No extreme outliers**
   - Outliers can distort coefficients and decision boundaries.
7. **Predictors need not be normally distributed**
   - Logistic regression does not assume normality of predictors (unlike Linear Regression).

---

9. Evaluation Metrics

Metric	Formula / Concept	Purpose
Accuracy	(TP + TN) / (TP + TN + FP + FN)	Overall correctness
Precision	TP / (TP + FP)	Fraction of predicted positives that are true
Recall (Sensitivity)	TP / (TP + FN)	Fraction of actual positives detected

F1-Score	2 × (Precision × Recall) / (Precision + Recall)	Balance between precision and recall
ROC-AUC	Area under the ROC curve	Measures how well model distinguishes between classes

10. Advantages

- Easy to implement and interpret.
- Works well for linearly separable classes.
- Provides probabilities (not just labels).
- Can handle large datasets efficiently.
- Supports regularization to avoid overfitting.

11. Limitations

- Assumes linearity in log-odds (not suitable for complex non-linear problems).
- Sensitive to outliers.
- May perform poorly with high multicollinearity.
- Can't capture non-linear relationships without feature engineering.