Linear Discriminant Analysis

🧠 What is Linear Discriminant Analysis (LDA)?

🔷 Simple Definition:

> Linear Discriminant Analysis (LDA) is a supervised learning algorithm that finds a linear combination of features that best separates two or more classes.

It can be used for:

- **Classification** (like Logistic Regression)
- **Dimensionality Reduction** (like PCA, but supervised)

🎯 Intuition

Imagine you have data from two classes (say, cats 🐱 and dogs 🐶) scattered in 2D space.

- Each class forms a cluster (like two Gaussian blobs).
- You want to find a **line (or direction)** that best separates these two blobs.

LDA finds the **best line** — a direction where:

- The distance between the class means is **maximized** (means are far apart), and
- The variance within each class is **minimized** (points within a class stay tight together).

Visually:

> It’s like finding the angle from which, if you look at the data, the two classes appear most distinct.

📊 The Core Idea (Intuition Formula)

LDA finds a projection vector w such that when you project your data x onto w:

$z = w^T x$

The separation between classes (in 1D space) is as large as possible.

It does this by **maximizing the ratio:**

$$J(w) = \frac{\text{between-class variance}}{\text{within-class variance}}$$

🔶 Between-class variance:

Measures **how far apart** the class means are after projection:

$S_B = (\mu_1 - \mu_0)(\mu_1 - \mu_0)^T$

🔶 Within-class variance:

Measures **how spread out** points are *within* each class:

$S_W = \Sigma_0 + \Sigma_1$

Then, LDA finds w that **maximizes:**

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

The solution turns out to be:

$w = S_W^{-1}(\mu_1 - \mu_0)$

That's the **direction** that best separates the two classes linearly.

🧩 Connection with GDA

Here's the key insight 🔑:

> LDA and GDA are mathematically connected.

- Both assume **each class is Gaussian**.
- $\text{Both assume the covariance matrices are shared}((\Sigma_0 = \Sigma_1 = \Sigma))$.

The only difference:

- **GDA** models the full probability distribution $P(x|y)$ and uses Bayes' rule.
- **LDA** directly computes the **linear boundary (projection)** that achieves maximum separation.

So:

📘 GDA = Probabilistic modeling version

📘 LDA = Geometric optimization version

Both lead to the same *linear decision boundary*.

🧮 Classification Rule (after projection)

Once we find w, we project all data onto it:

$z = w^T x$

Then we compute thresholds to decide class boundaries:

$$\text{If } z > \text{threshold} \Rightarrow y = 1, \text{ else } y = 0$$

For multiple classes, LDA generalizes to finding multiple discriminant directions (like PCA but with class labels).

🧭 LDA vs PCA

Aspect	LDA	PCA
Type	Supervised	Unsupervised
Goal	Maximize class separation	Maximize data variance
Uses class labels?	✅ Yes	❌ No
Output	Directions that separate classes	Directions of greatest variance
Typical use	Classification, supervised dimension reduction	Feature compression, visualization

---

🧩 Summary

Concept	LDA Insight
Assumption	Each class follows a Gaussian with shared covariance
Goal	Find direction that best separates class means while minimizing overlap
Result	Linear boundary or projection
Output	Lower-dimensional representation or classifier
Related to	GDA (same assumptions, different approach)

🧠 In one sentence:

> LDA finds the best line (or plane) to project your data onto so that different classes are as far apart and as tight within themselves as possible.

🧠 Similarities between LDA and Logistic Regression

Aspect	LDA	Logistic Regression	Common Ground
Goal	Classify samples into discrete classes	Classify samples into discrete classes	Both are classification algorithms
Decision Boundary	Linear (if covariance is shared)	Linear	Both produce a linear separator
Output	Class probabilities $P(y\|x)$	Class probabilities $P(y\|x)$	Both estimate probability of each class
Data Type	Continuous input, categorical target	Continuous input, categorical target	Same data types used
When they agree	When data is Gaussian with equal covariance	When model fits the same data	Both give almost identical predictions under certain conditions

---

⚙️ Differences between LDA and Logistic Regression

Concept	LDA	Logistic Regression
Model Type	**Generative model** (models how data is generated)	**Discriminative model** (models the decision boundary directly)
What it learns	Learns $P(x\|y)$ (how each class generates features) and $P(y)$	Learns $P(y\|x)$ (how likely a label is given features)
Assumptions	Assumes data from each class follows a Gaussian (Normal) distribution with same covariance matrix Σ	Makes **no assumption** about data distribution
Mathematical Form	Derived using **Bayes theorem** + Gaussian assumption	Derived from **maximum likelihood** using the **sigmoid** model
Training Objective	Maximize joint likelihood P(x, y)	Maximize conditional likelihood $P(y\|x)$
Parameters Estimated	Class means μ_k , shared covariance Σ, and priors π_k	Weight vector w and bias b
Robustness	Sensitive to non-Gaussian data and outliers	More robust to non-Gaussian data
Computation	Closed-form (analytical) solution	Iterative optimization (e.g., gradient descent)
Interpretation	Finds decision boundary by modeling each class distribution and using Bayes rule	Finds decision boundary directly from data
When to Use	When data roughly follows Gaussian distributions	When you want fewer assumptions or more flexibility

🧩 Intuitive Difference

Analogy	LDA	Logistic Regression
How they "think"	"Let me model how each class *produces* its data (like two Gaussian clouds) and then find the dividing line."	"Let me directly draw the best line that separates the two classes — no need to assume a distribution."
If you visualize data	It fits Gaussian ellipses to each class, then finds where they intersect.	It fits a sigmoid curve that transitions smoothly between class probabilities.

🧭 When to Use LDA, GDA, and Logistic Regression

Algorithm	Type	Best Used When	Why It Works Well	When to Avoid
LDA (Linear Discriminant Analysis)	Generative models $P(x\|y)$	- Classes are **normally distributed** - **Covariance matrices are similar** across classes - Data is **low-dimensional** or **small sample size**	- Provides a **linear boundary** that's optimal for Gaussian data - Has a **closed-form solution** (no need for gradient descent) - Works well with **small datasets** because it uses distributional assumptions	- Data is **non-Gaussian** - Classes have **different covariance structures** - Dataset is **large and complex** (assumptions break down)
GDA (Gaussian Discriminant Analysis)	Generative models $P(x\|y)$	- Each class has its **own Gaussian distribution** - Covariance matrices **can differ** between classes - You want a **non-linear decision boundary**	- More flexible than LDA - Can model **elliptical class boundaries** (different spreads/orientations)- Good for **moderate-sized datasets** where Gaussian fits are reasonable	- If data is **not Gaussian at all** - If classes overlap heavily or have too few samples - More parameters → risk of **overfitting**

Algorithm	Type	Best Used When	Why It Works Well	When to Avoid
Logistic Regression	Discriminative models $P(y\|x)$	- You **don't want to assume** any data distribution - Dataset is **large** and **features aren't Gaussian** - You care mainly about **predictive performance**	- More **robust and flexible** - Works well with **non-Gaussian**, **skewed**, or **high-dimensional** data - Regularization (L1/L2) helps avoid overfitting	- Very **small datasets** where distributional modeling helps (LDA/GDA might outperform) - When you want to **understand the data generation process**, not just classify