Week-8

Linear Discriminant Analysis Recap and Project Implementation

V Semester - ML and AIUP, Aug-Oct 2024

Session Date and Time: 10th Nov 2024 10:30 AM IST – 12:00 Noon IST

Venkateswar Reddy Melachervu

Visiting Faculty and CTO, Brillium Technologies

Department of Computer Science Engineering – AI and ML (CSM)

Email: venkat.reddy.gf@gprec.ac.in

G.Pulla Reddy Engineering College (Autonomous)

G.Pulla Reddy Nagar, Nandyal Road, Kurnool, AP 518007, India

Website: https://www.gprec.ac.in

Disclaimer and Confidentiality Notice

The content of this guest lecture, including all discussions, materials, and demonstrations, code is intended for educational purposes only and reflects the views and opinions of the speaker. While every effort has been made to ensure the accuracy and relevance of the information presented, it should not be considered as legal, financial, or professional advice.

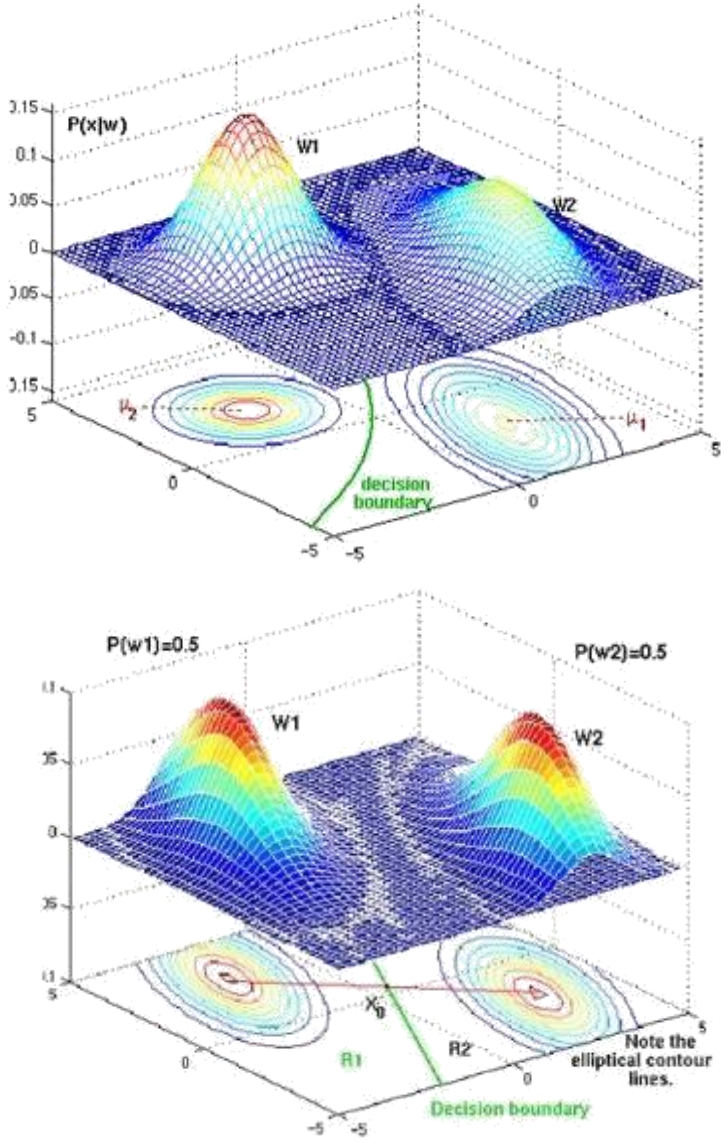
Brillium Technologies retains unrestricted ownership of all information shared during this session. Participants must not record, reproduce, distribute, or disclose any part of the lecture or materials without prior written permission from Brillium Technologies. Unauthorized use or distribution of the content may result in legal action.

Additionally, all trademarks, service marks, and intellectual properties referenced or used in this presentation are the property of their respective owners. No ownership or rights over such third-party content are claimed or implied by the author or Brillium Technologies or by GPREC.

By attending this lecture, you agree to respect the confidentiality of the information shared and refrain from using it in any unauthorized manner. Failure to comply with these terms may result in legal action.

Thank you for your understanding and cooperation.

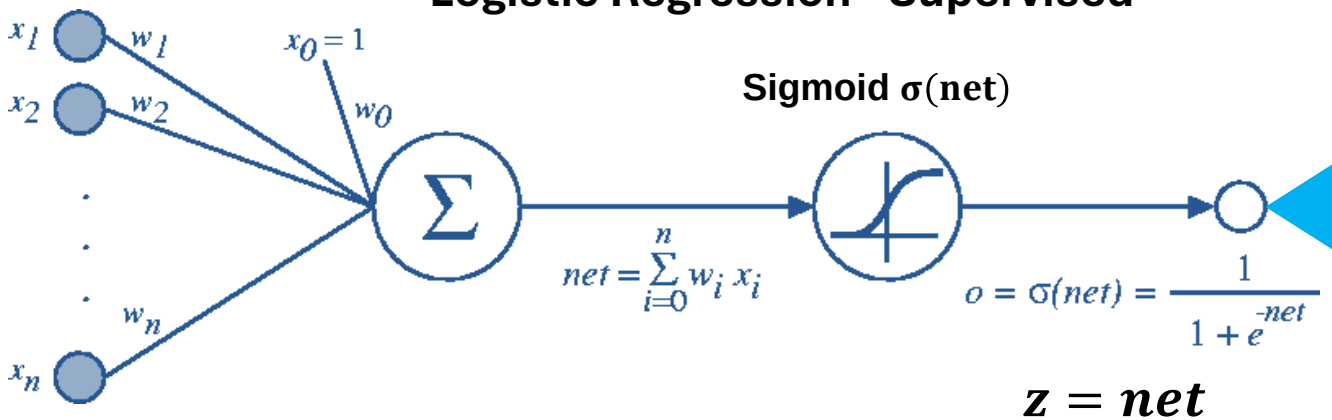
Lecture Outline



- Classical Classification
- Classification Recap
- Challenges of Clear Class Separation: Beyond Probabilities and Margins
- Discriminant Functions
- Probability and Likelihood in Discriminant Functions
- LDA and GDA Overview
- Discriminant Analysis Classification Rule Derivation
- LDA/GDA Efficiency Metrics
- LDA/GDA Peer Comparison
- Key Takeaways
- Project Announcement
- Project Implementation Steps
- Project Demo
- Q&A

Classical Classification

Logistic Regression - Supervised

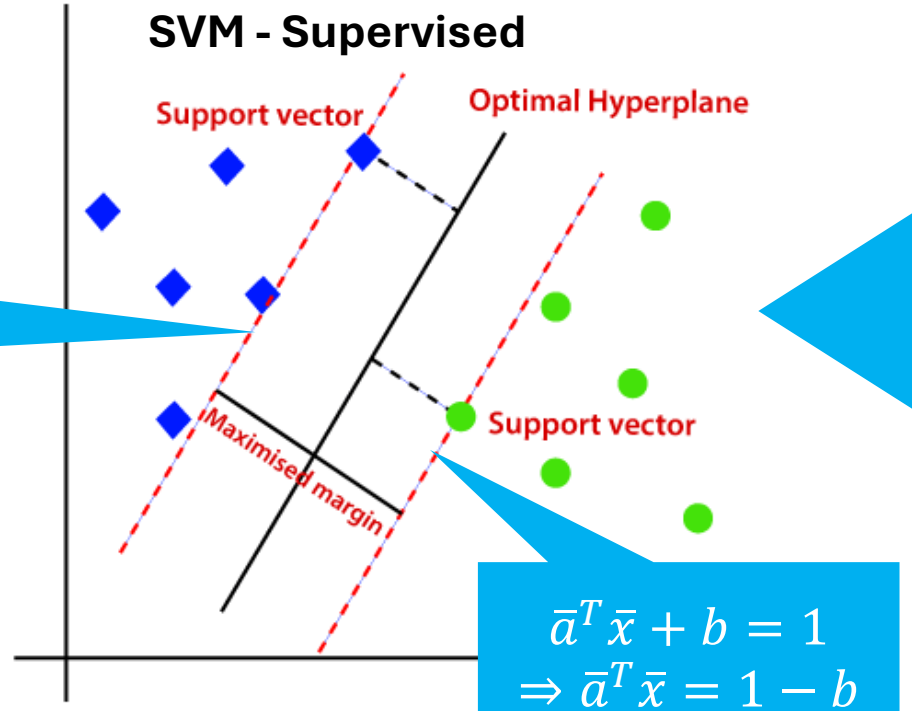


$$z = net$$

Class Probabilities

- $P(y = 1 \mid \bar{x}) = \frac{1}{\left(1+e^{-\bar{x}^T \bar{w}}\right)}$
- $P(y = 0 \mid \bar{x}) = \frac{e^{-\bar{x}^T \bar{w}}}{\left(1+e^{-\bar{x}^T \bar{w}}\right)}$

SVM - Supervised



$$\bar{a}^T \bar{x} + b = -1$$
$$\Rightarrow \bar{a}^T \bar{x} = -1 - b$$

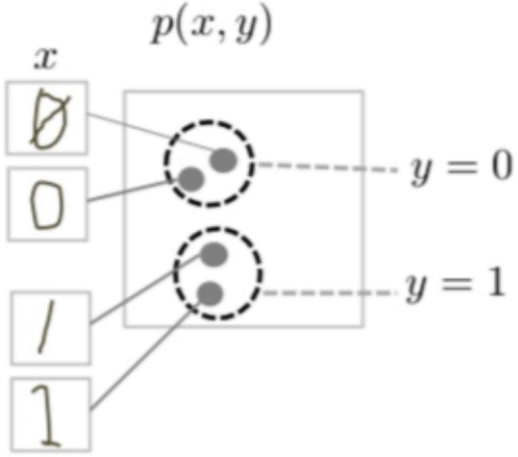
$$\bar{a}^T \bar{x} + b = 1$$
$$\Rightarrow \bar{a}^T \bar{x} = 1 - b$$

Decision Boundary

- $f(\bar{x}, b) = \bar{a}^T \bar{x} + b > 0 \Rightarrow y = 1$
- $f(\bar{x}, b) = \bar{a}^T \bar{x} + b < 0 \Rightarrow y = 0$

Classical Classification

Naïve Bayes (Generative Model) - Supervised



Joint Probabilities

- $\boldsymbol{P(y = 1|\bar{x} = \bar{v}) > P(y = 0|\bar{x} = \bar{v}) \Rightarrow y = 1}$
- $\boldsymbol{P(y = 1|\bar{x} = \bar{v}) < P(y = 0|\bar{x} = \bar{v}) \Rightarrow y = 0}$

- Given a set of features and labeled training data, new instances is classified into predefined category
- Key Terms
  - Feature Space - The n-dimensional space where data points (feature vectors) reside
  - Class Boundaries - Hypothetical boundaries that separate instances of different classes
- Often assign instances to classes based on probabilities derived from training data, or by optimizing a boundary that separates the classes
  - **Probability-Based Approach -** Algorithms like Logistic Regression and Naive Bayes classify based on calculated class probabilities
  - **Decision Boundary-based Approach -** Algorithms like SVM directly optimize boundaries that separate classes in feature space

Classifier Categories - Recap

- Probabilistic Classifiers
  - Given dataset $\mathcal{D} = \{(\bar{x}_i, y_i)\}_{i=1}^{N}$
  - Goal is to compute the **class posterior** $\boldsymbol{p(y = c \mid \bar{x})}$ which models the mapping $y = f(\bar{x})$
- Generative Classifiers
  - Given dataset $\mathcal{D} = \{(\bar{x}_i, y_i)\}_{i=1}^{N}$
  - Goal is to model how the data is generated for each class using **joint probability** $\boldsymbol{p(y, \bar{x}|\theta)}$
  - Assumes each class has a certain data distribution that is defined by a parameter set $\boldsymbol{\theta}$
  - θ represents each class's data distribution characteristics
    - Class conditional distribution $p(\bar{x}|\, y = c; \theta)$
    - Class prior probability $p(y = c; \theta)$
  - The joint probability of class label y, feature vector $\bar{x}$ given class parameter set θ is $\boldsymbol{p((y, \bar{x})|\theta)}$
  - **Class-conditional density** $\boldsymbol{p(\bar{x} \mid y = c; \theta)}$ models the likelihood of observing a **feature vector** $\boldsymbol{\bar{x}}$ given a specific **class** $\boldsymbol{y = c}$ and parameter $\boldsymbol{\theta}$
  - Bayes' Theorem $p(y = c \mid \bar{x}) = \frac{p(\bar{x}; y=c) \times p(y=c)}{p(\bar{x}=\bar{x}_i)}$
  - Applying Bayes' theorem, the posterior probability $\boldsymbol{p(y = c|\bar{x}; \theta) \propto p(\bar{x}; c) \times p(y = c)}$
  - This posterior probability represents the probability of class c for the observed feature vector $\bar{x}$
  - Feature vector $\bar{x}$ is generated for each class using class-conditional density $p(x|y = c; \theta) \Rightarrow$ Generative Classifier
  - θ is estimated by fitting the model to training data by maximizing the joint log-likelihood
  - $\boldsymbol{\theta^* = arg \max_i log\big(p(y_i, \bar{x}_i|\theta)\big)}$

Classifier Categories - Recap

- Discriminative Classifier
  - Given dataset $\mathcal{D} = \{(\bar{x}_i, y_i)\}_{i=1}^{N}$
  - Goal is to compute the discriminative probability $p(y = c|\bar{x})$ directly from the training data
  - The model is usually fit to training data by maximising the conditional log-likelihood
    - $\boldsymbol{\theta^* = arg\ \max_{\theta} \sum_i log\left(p(y_i|x_i, \theta)\right)}$

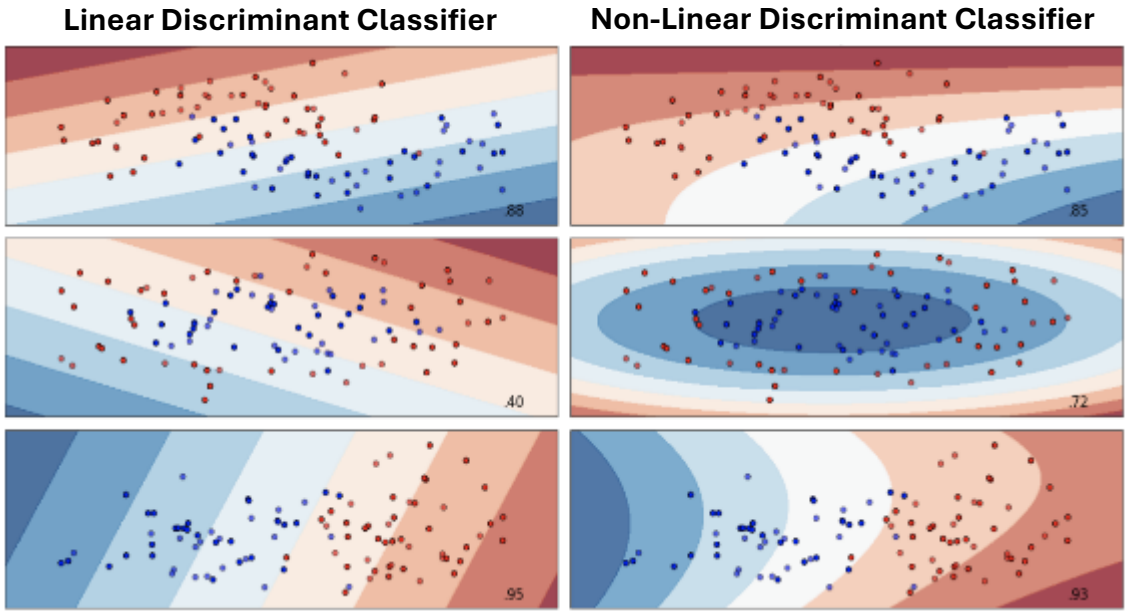
Challenges of Clear Class Separation: Beyond Probabilities and Margins

- Probability and decision boundary optimization-based methods are effective in general
- But aren't always ideal for **maximizing the multi-class separation for overlapping complex datasets**

Is there a better approach to maximize the spread between overlapping classes to reduce misclassification risk and maximizing multi-class separation?

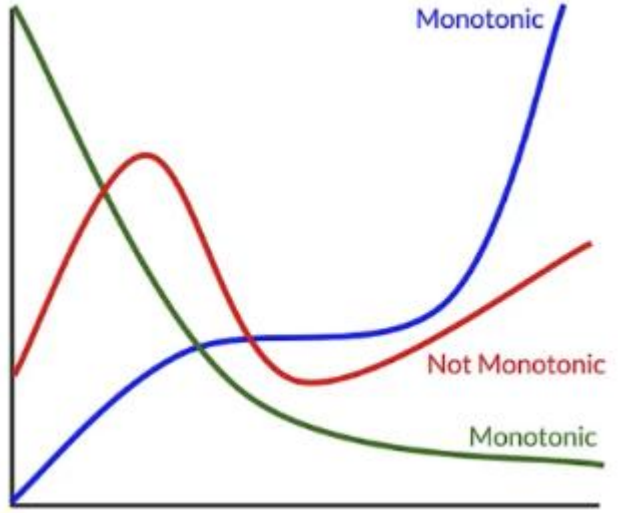
Discriminant Functions

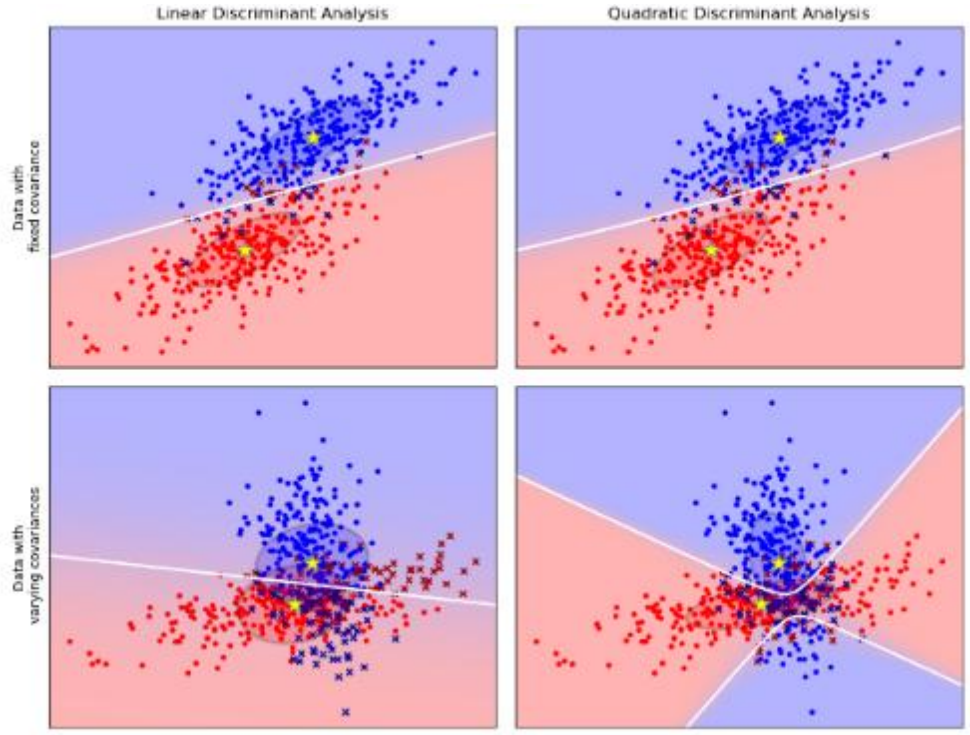
- Statistical mathematical functions to separate data points into different categories
- Assigns a score to each class for a given data point, the data point is assigned to the class with the highest score
- Data distribution plays a key role in defining the discriminant functions
- Based on the assumption that **data follows/generated from a distributions** like normal distribution
- Decision boundaries are defined based on statistical characteristics of data distribution
  - Mean, covariance
- **Covariance** describes the degree to which features vary together
  - Represented by covariance matrix for each class
- It is a statistical canonical form of multi-class classifiers



Discriminant Functions

- Simplifies class probabilities with an equivalent function that assigns a score to each class $\boldsymbol{c_i}$ for a given $\overline{\boldsymbol{x}}$
  - $g_i(\bar{x}) = f\left(P(c_i|\bar{x})\right), i = 1,2, \dots k$
  - $f(.)$ – Monotonically Increasing Function
    - If $x_1 < x_2, \Rightarrow f(x_1) \leq f(x_2)$
    - $f(x)$ can increase or stay constant but never decreases
  - $g_i(\bar{x})$ – Discriminant Function of class i
- $\overline{\boldsymbol{x}} \in \boldsymbol{c_l}$
  - $\boldsymbol{if\ g_l(\bar{x}) > g_j(\bar{x})\ \forall\ j \neq i\ \text{and}\ i,j = 1,2, \dots k}$
  - $\boldsymbol{g_l(\bar{x}) = arg\ \max_{1 \leq i \leq k} g_i(\bar{x}) \Rightarrow \bar{x} \in c_l}$
- Types
  - **Linear Discriminants** - Linear boundaries between classes
  - **Non-linear Discriminants**- Flexible boundaries for higher dimensional datasets

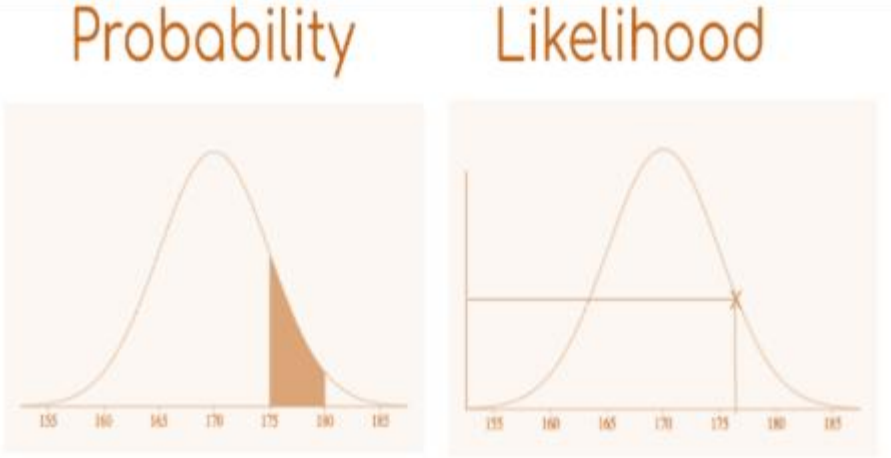




Probability and Likelihood in Discriminant Functions

- **Probability** is the likelihood of a possible result
- **Likelihood** is the process of determining the best data distribution for a given situation/data
- The discriminant scores are derived from log-likelihood functions of each class
- Likelihoods incorporate mean and covariance of each class
- The discriminant functions maximize the likelihood
- A point belonging to a particular class is decided by taking mean and covariance of each class into account
- **Probability** measures the fitness of data given a specific distribution
- **Likelihood** measures the fitness of a model given some data

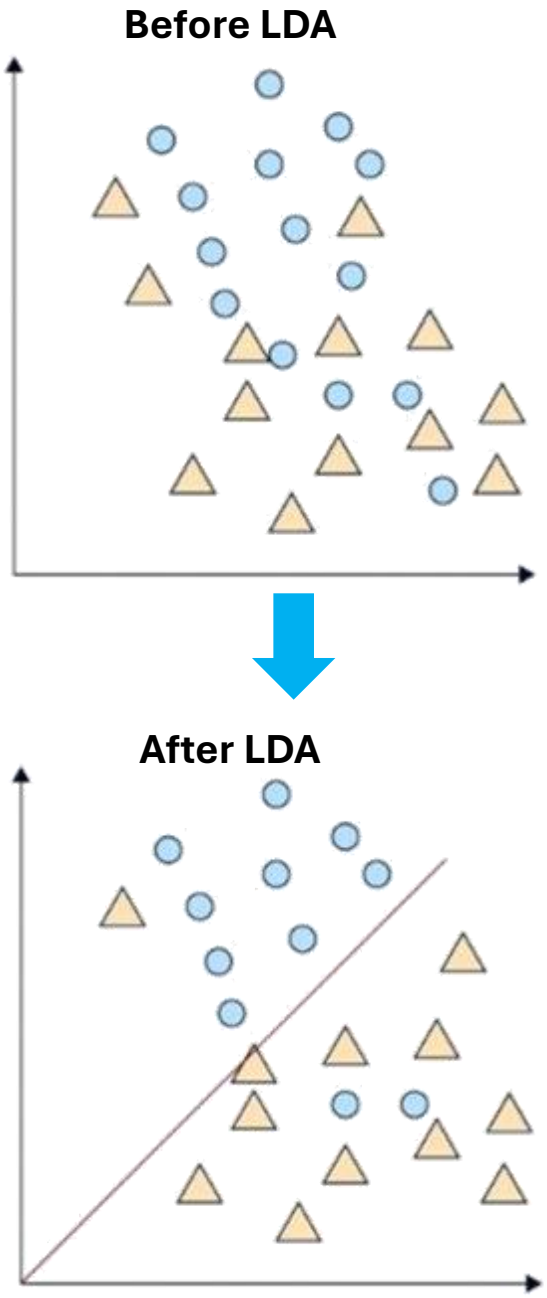
Probability	Likelihood
Predicts the chance of future or unknown outcomes	Measures how well a model or parameter explains observed data
Fixed model, variable data	Fixed data, variable model parameters
Used in predictive contexts, like Bayesian inference	Used in parameter estimation, like MLE
"Given a fair die, what's the probability of rolling a 3?"	"Given the rolls we observed, how likely is it that this die is fair?"



Data Distributions

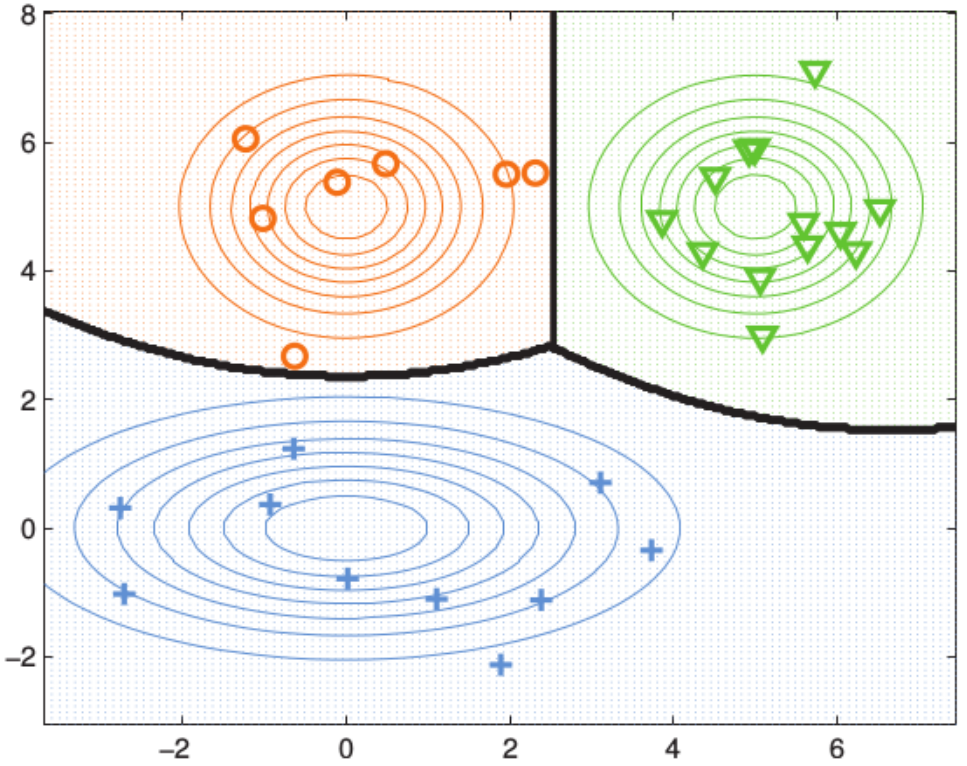
Linear Discriminant Analysis - LDA

- A supervised learning model
- Uses linear discriminants to find a linear combination of features for multi-class separate
- Simplifies the discriminant function to a linear form
- Classes have the same geometric shape and orientation in the feature space
- Underlying Assumptions
  - Distribution Normality - Each class is normally distributed
  - Distribution Covariance - All classes share the same covariance matrix
- Models the difference between classes by focusing on maximizing the variance between them
- Shared covariance leads to linear decision boundaries
- Decision boundary is a line/hyperplane that lies midway between the means of the two classes



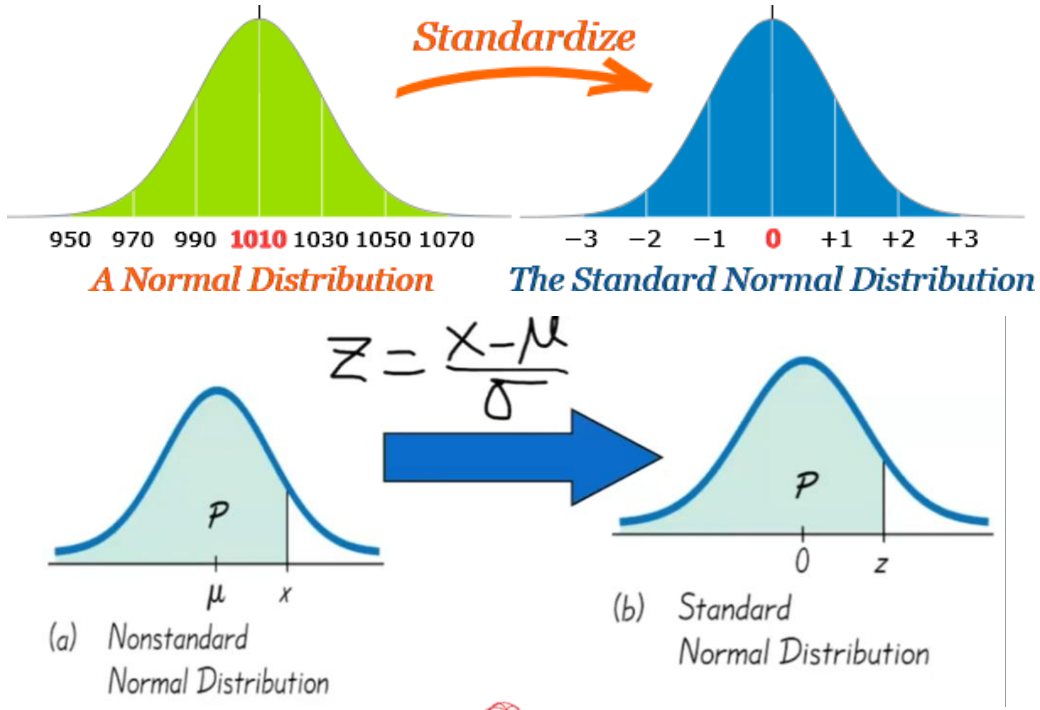
Gaussian Discriminant Analysis - GDA

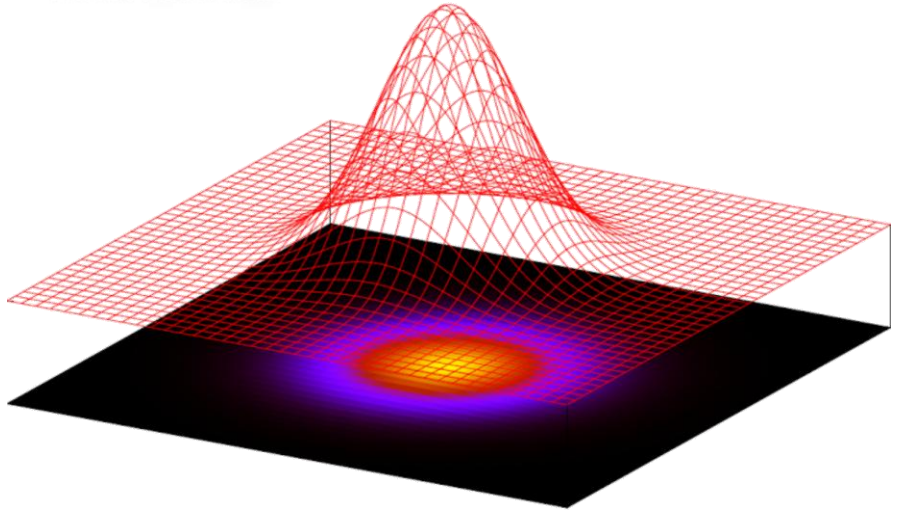
- Generalization of LDA allowing for different covariances for classes
- Underlying Assumptions
  - Distribution Normality - Each class is normally distributed
  - Distribution Covariance - Each class is allowed to have its own covariance matrix
- Shape and orientation of each class in the feature space can vary
- Models more complex class distributions with more parameters to estimate the covariance for each class individually
- Decision boundary is quadratic and curves based on the relative shape and spread of each class's covariance
- Individual covariances lead to **flexible (quadratic/non-linear) decision boundaries**



Discriminant Analysis Classification Rule Derivation

- Data is assumed to have Normal distribution ⇒ Gaussian Distribution $\mathcal{N}(\mu, \sigma)$
- PDF of a univariate continuous variable $\boldsymbol{X} \in \mathbb{R}$ with Gaussian distribution $\mathrm{X} \sim \mathcal{N}(\mu, \sigma^2)$
  - PDF - $f_X(x) = \mathrm{P_X}(X = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}$
  - Mean - $E\{X\} = \mu$
  - Variance - $E\{(X-\mu)^2\} = \sigma^2$
- PDF of Multi-variate continuous variable $\bar{X} \in \mathbb{R}^n$ with Gaussian Distribution $\bar{\mathrm{X}} \sim \mathcal{N}(\bar{\mu}, R)$
  - PDF - $f_{\bar{X}}(\bar{x}) = P_{\bar{X}}(\bar{x}) = \frac{1}{\sqrt{(2\pi)^n |R|}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu})^T R^{-1} (\bar{x}-\bar{\mu})}$
  - Mean - $E\{\bar{X}\} = \bar{\mu}$
  - Covariance Matrix - $E\{(\bar{x}-\bar{\mu})(\bar{x}-\bar{\mu})^T\} = R \in \mathbb{R}^{n \times n}, R = R^T, R \geq 0$
  - n – number of variables/dimensions/features
- Consider two classes of multi-variate (n-variate) Gaussian distributions
  - $\mathcal{C}_0 : \mathcal{N}_{\mathcal{C}_0}(\bar{\mu}_0, R)$
  - $\mathcal{C}_1 : \mathcal{N}_{\mathcal{C}_1}(\bar{\mu}_1, R)$
  - Class Prior Probabilities: p_0, p_1

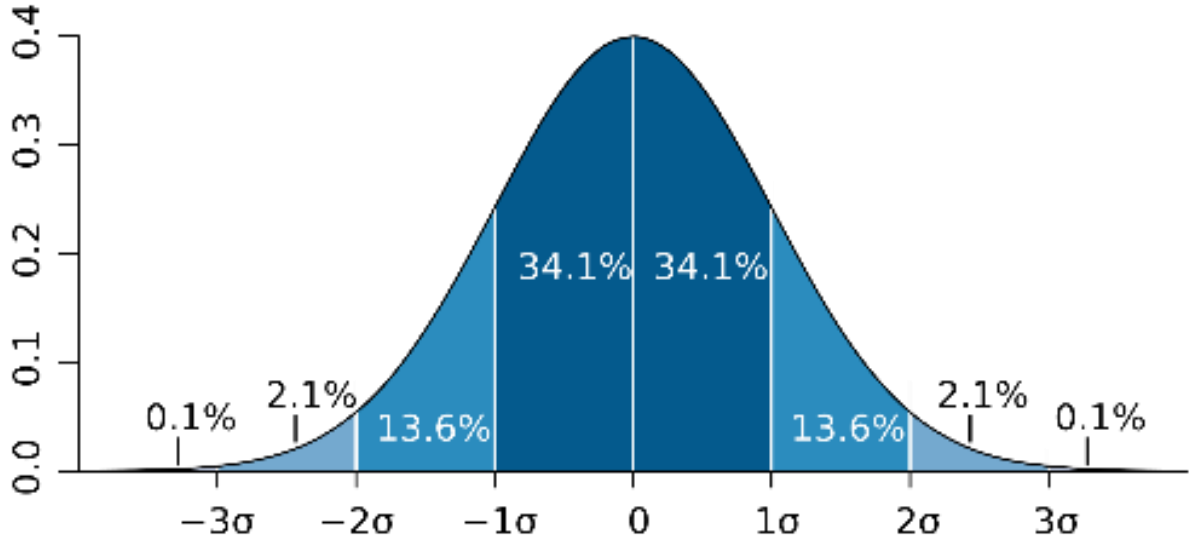


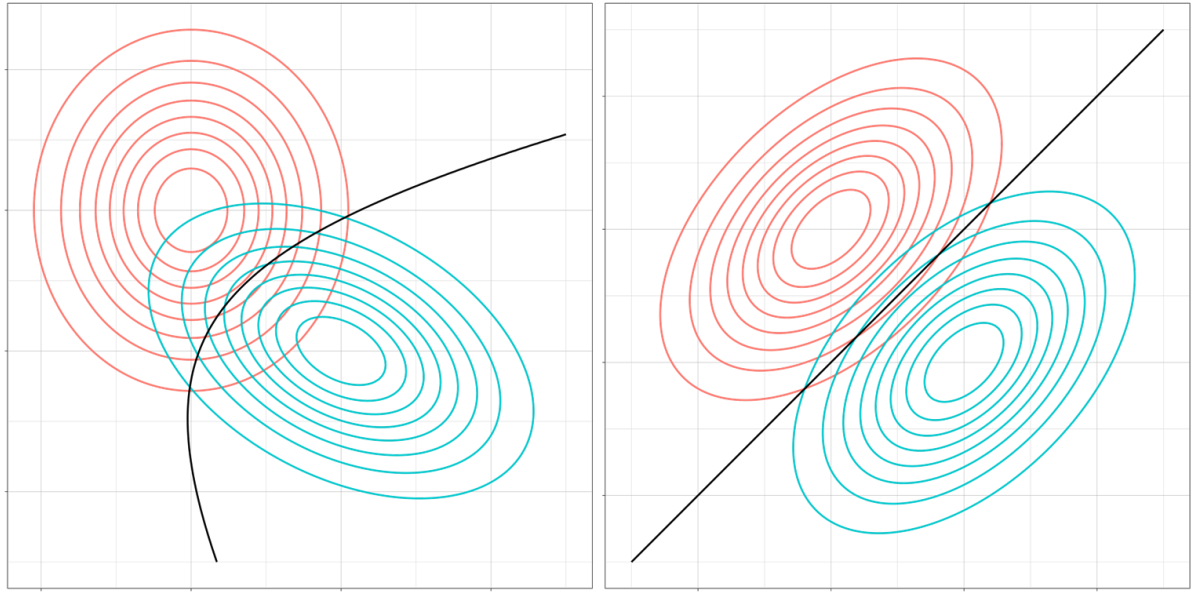


Multi-variate Gaussian Distribution

Discriminant Analysis Classification Rule Derivation

- Likelihoods of two classes
  - $p(\bar{x}; \mathcal{C}_0) = \frac{1}{\sqrt{(2\pi)^n |R|}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu}_0)^T R^{-1} (\bar{x}-\bar{\mu}_0)}$
  - $p(\bar{x}; \mathcal{C}_1) = \frac{1}{\sqrt{(2\pi)^n |R|}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu}_1)^T R^{-1} (\bar{x}-\bar{\mu}_1)}$
- Choosing the class for classification
  - Maximum likelihood rule - choose the class that **maximizes the posterior probability**
- Posterior probabilities
  - For generative classifier, $\boldsymbol{p(y = c|\bar{x}; \theta) \propto p(\bar{x}; c) \times p(y = c)}$
  - Class $\boldsymbol{\mathcal{C}_0} \Rightarrow \boldsymbol{p(y = \mathcal{C}_0|\bar{x}; \theta_{\mathcal{C}_0}) \propto p(\bar{x}; \mathcal{C}_0) \times p_0}$
  - Class $\boldsymbol{\mathcal{C}_1} \Rightarrow \boldsymbol{p(y = \mathcal{C}_1|\bar{x}; \theta_{\mathcal{C}_1}) \propto p(\bar{x}; \mathcal{C}_1) \times p_1}$
- Choose $\boldsymbol{\mathcal{C}_0}$: $\boldsymbol{p(\bar{x}; \mathcal{C}_0) \times p_0 \geq p(\bar{x}; \mathcal{C}_1) \times p_1}$
- $\frac{1}{\sqrt{(2\pi)^n |R|}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu}_0)^T R^{-1} (\bar{x}-\bar{\mu}_0)} \geq \frac{1}{\sqrt{(2\pi)^n |R|}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu}_1)^T R^{-1} (\bar{x}-\bar{\mu}_1)}$
- Taking natural log both sides
- $ln\, p_0 - \frac{1}{2}(\bar{x} - \bar{\mu}_0)^T R^{-1} (\bar{x} - \bar{\mu}_0) \geq ln\, p_1 - \frac{1}{2}(\bar{x} - \bar{\mu}_1)^T R^{-1} (\bar{x} - \bar{\mu}_1)$
- $\Rightarrow ln\, p_0 - \frac{1}{2}(\bar{x} - \bar{\mu}_0)^T R^{-1} (\bar{x} - \bar{\mu}_0) \geq ln\, p_1 - \frac{1}{2}(\bar{x} - \bar{\mu}_1)^T R^{-1} (\bar{x} - \bar{\mu}_1)$
- $\Rightarrow ln\, p_0 - ln\, p_1 \geq \frac{1}{2}(\bar{x} - \bar{\mu}_0)^T R^{-1} (\bar{x} - \bar{\mu}_0) - \frac{1}{2}(\bar{x} - \bar{\mu}_1)^T R^{-1} (\bar{x} - \bar{\mu}_1)$
- $\Rightarrow ln \frac{p_0}{p_1} \geq \frac{1}{2}(\bar{x} - \bar{\mu}_0)^T R^{-1} (\bar{x} - \bar{\mu}_0) - \frac{1}{2}(\bar{x} - \bar{\mu}_1)^T R^{-1} (\bar{x} - \bar{\mu}_1)$



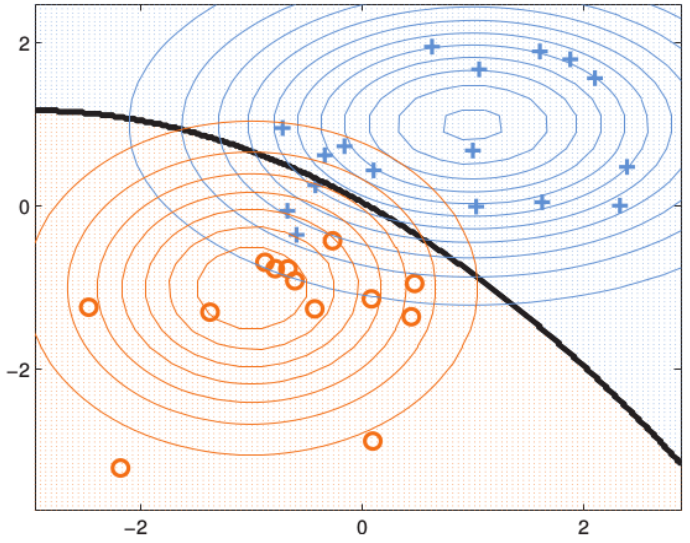


Discriminant Analysis Classification Rule Derivation

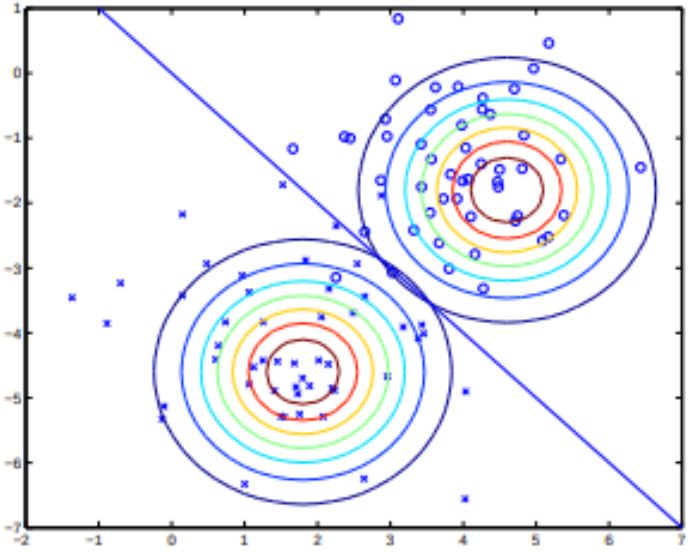
- $\Rightarrow (\bar{x}-\bar{\mu}_0)^T R^{-1}(\bar{x}-\bar{\mu}_0)-(\bar{x}-\bar{\mu}_1)^T R^{-1}(\bar{x}-\bar{\mu}_1) \le 2\,ln\frac{p_0}{p_1}$
- $(\bar{x}-\bar{\mu}_0)^T R^{-1}(\bar{x}-\bar{\mu}_0), (\bar{x}-\bar{\mu}_1)^T R^{-1}(\bar{x}-\bar{\mu}_1)$ are quadratic forms
- $\Rightarrow (\bar{x}^T-\bar{\mu}_0^T)R^{-1}(\bar{x}-\bar{\mu}_0)-(\bar{x}^T-\bar{\mu}^T)R^{-1}(\bar{x}-\bar{\mu}_1) \le 2\,ln\frac{p_0}{p_1}$
- $\Rightarrow (\bar{x}^T-\bar{\mu}_0^T)R^{-1}\bar{x}-(\bar{x}^T-\bar{\mu}_0^T)R^{-1}\bar{\mu}_0-(\bar{x}^T-\bar{\mu}_1{}^T)R^{-1}\bar{x}+(\bar{x}^T-\bar{\mu}_1{}^T)R^{-1}\bar{\mu}_1 \le 2\,ln\frac{p_0}{p_1}$
- $\Rightarrow \bar{x}^T R^{-1}\bar{x}-\bar{\mu}_0^T R^{-1}\bar{x}-\bar{x}^T R^{-1}\bar{\mu}_0+\bar{\mu}_0^T R^{-1}\bar{\mu}_0 \sout{-\bar{x}^T R^{-1}\bar{x}}+\bar{\mu}_1^T R^{-1}\bar{x}+\bar{x}^T R^{-1}\bar{\mu}_1-\bar{\mu}_1^T R^{-1}\bar{\mu}_1 \le 2\,ln\frac{p_0}{p_1}$
- $(ABC)^T = C^T B^T A^T$ and $A^T = B$ and $B^T = A \Rightarrow A = B$
  - $\Rightarrow (\bar{x}^T R^{-1}\bar{\mu}_0)^T = \bar{\mu}_0^T R^{-1}\bar{x}$ and $(\bar{\mu}_0^T R^{-1}\bar{x})^T = \bar{x}^T R^{-1}\bar{\mu}_0 \Rightarrow \boldsymbol{\bar{\mu}_0^T R^{-1}\bar{x} = \bar{x}^T R^{-1}\bar{\mu}_0}$
  - $\Rightarrow (\bar{x}^T R^{-1}\bar{\mu}_1)^T = \bar{\mu}_1^T R^{-1}\bar{x}$ and $(\bar{\mu}_1^T R^{-1}\bar{x})^T = \bar{x}^T R^{-1}\bar{\mu}_1 \Rightarrow \boldsymbol{\bar{\mu}_1^T R^{-1}\bar{x} = \bar{x}^T R^{-1}\bar{\mu}_1}$
- $\Rightarrow -2\bar{\mu}_0^T R^{-1}\bar{x}+\bar{\mu}_0^T R^{-1}\bar{\mu}_0+2\bar{\mu}_1^T R^{-1}\bar{x}-\bar{\mu}_1^T R^{-1}\bar{\mu}_1 \le 2\,ln\frac{p_0}{p_1}$
- $\Rightarrow 2(\bar{\mu}_1^T-\bar{\mu}_0^T)R^{-1}\bar{x}+\bar{\mu}_0^T R^{-1}\bar{\mu}_0-\bar{\mu}_1^T R^{-1}\bar{\mu}_1 \le 2\,ln\frac{p_0}{p_1} \Rightarrow (\bar{\mu}_1^T-\bar{\mu}_0^T)R^{-1}\bar{x}+\frac{1}{2}(\bar{\mu}_0^T R^{-1}\bar{\mu}_0-\bar{\mu}_1^T R^{-1}\bar{\mu}_1) \le ln\frac{p_0}{p_1}$
  - $\frac{1}{2}(\bar{\mu}_0^T R^{-1}\bar{\mu}_0-\bar{\mu}_1^T R^{-1}\bar{\mu}_1) = -\frac{1}{2}(\bar{\mu}_1-\bar{\mu}_0)^T R^{-1}(\bar{\mu}_1+\bar{\mu}_0)$
- $\Rightarrow (\bar{\mu}_1^T-\bar{\mu}_0^T)R^{-1}\bar{x}-\frac{1}{2}(\bar{\mu}_1-\bar{\mu}_0)^T R^{-1}(\bar{\mu}_1+\bar{\mu}_0) \le ln\frac{p_0}{p_1}$
- $\Rightarrow (\bar{\mu}_1-\bar{\mu}_0)^T R^{-1}\left(\bar{x}-\frac{1}{2}(\bar{\mu}_1+\bar{\mu}_0)\right) \le ln\frac{p_0}{p_1} \Rightarrow (\bar{\mu}_0-\bar{\mu}_1)^T R^{-1}\left(\bar{x}-\frac{1}{2}(\bar{\mu}_1+\bar{\mu}_0)\right) \ge ln\frac{p_0}{p_1}$

Discriminant Analysis Classification Rule Derivation

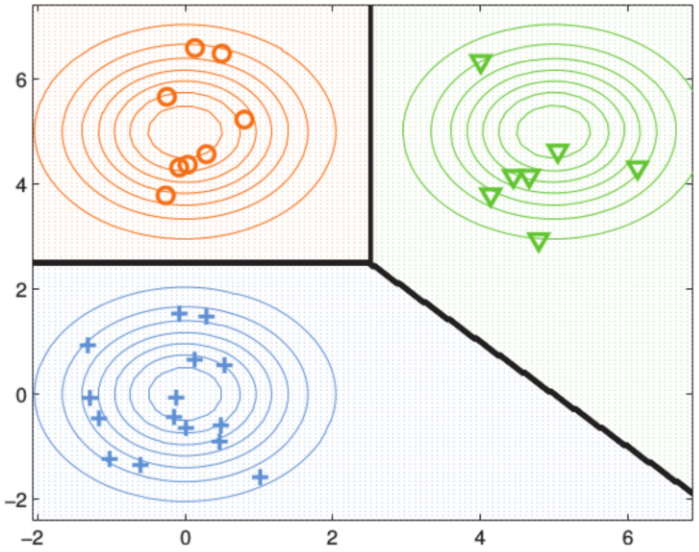
- Choose $\mathcal{C}_0$: $(\bar{\mu}_0 - \bar{\mu}_1)^T R^{-1}\left(\bar{x} - \frac{1}{2}(\bar{\mu}_1 + \bar{\mu}_0)\right) \geq ln\frac{p_0}{p_1}$
- Simplifying, GDA classification rule reduces to:
  - Choose $\mathcal{C}_0$: $\bar{h}^T(\bar{x} - \tilde{\mu}) \geq ln\frac{p_1}{p_0}$
  - Choose $\mathcal{C}_1$: $\bar{h}^T(\bar{x} - \tilde{\mu}) < ln\frac{p_1}{p_0}$
  - $\bar{h} = R^{-1}(\bar{\mu}_0 - \bar{\mu}_1)$
  - $\tilde{\mu} = \frac{1}{2}(\bar{\mu}_0 + \bar{\mu}_1)$
- $\bar{h}^T(\bar{x} - \tilde{\mu}) = ln\frac{p_1}{p_0}$ is a Linear Equation ⇒ Hyperplane ⇒ Linear Classifier



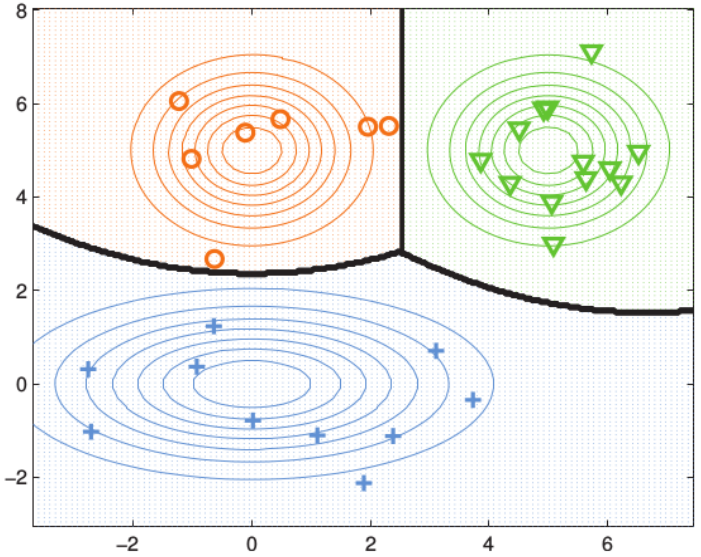
Parabolic Boundary



Linear Boundary



All Linear Boundaries



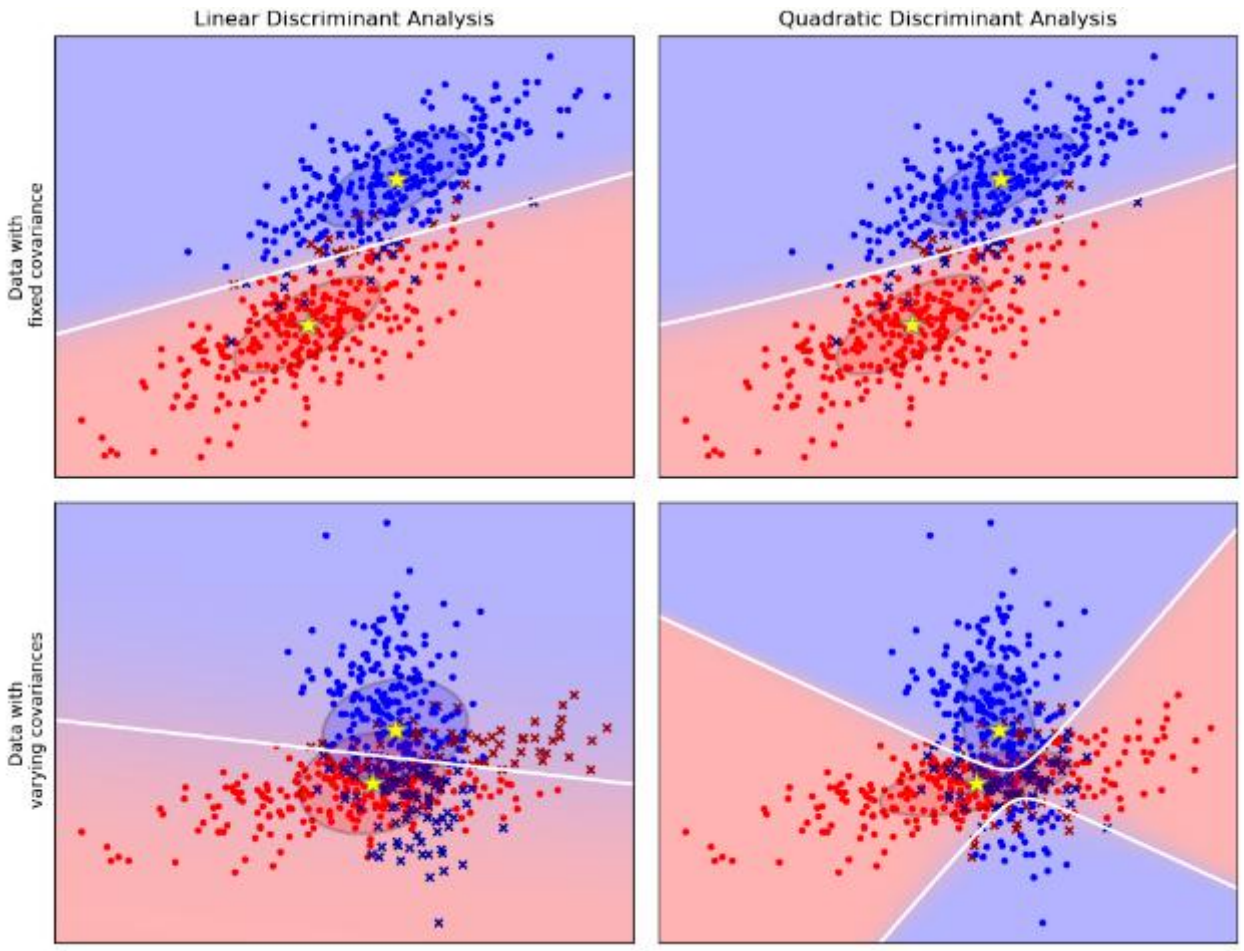
Some Linear, Some Quadratic Boundaries

Efficiency Metrics

- Training Time
  - LDA and GDA have relatively low computational complexity, as they involve calculating mean and covariance for each class
  - For LDA with shared covariance, complexity is $O(Nd^2)$
    - N – Number of training samples
    - d – feature dimension
- Prediction Time
  - Linear – Proportional to number of feature dimensions
- Accuracy
  - The percentage of correctly classified instances out of the total instances.
- Precision
  - The ratio of true positives to the sum of true positives and false positives
- Recall – Sensitivity
  - The ratio of true positives to the sum of true positives and false negatives
- F1 Score
  - The harmonic mean of precision and recall
  - Between 0 and 1
    - 1 – Perfect precision and recall - model correctly identifies all true positives with no false positives or false negatives
    - 0 - Model is entirely ineffective for positive class detection
- ROC-AUC
  - Measures the model's ability to distinguish between classes across different thresholds
  - A higher AUC indicates better performance, with a value of 1 indicating perfect classification

Key Takeaways

- LDA
  - Best suited for problems where class distributions have equal covariance
  - Linear decision boundaries and high efficiency make it suitable for high-dimensional datasets
- GDA
  - Flexible in handling different covariance matrices for each class, leading to quadratic decision boundaries
  - Better suited for problems with complex, non-linear class boundaries



Project Announcement

- Algorithm of Application
  - LDA/GDA
- Project Title
  - Inferring Breast Cancer/Malignancy from Diagnosis Data and Classification Models Comparative Analysis
- Project Objective
  - Implement LDA/GDA for inferring breast cancer disease and analyze comparative classification models performance you have learnt in AAM-IPL on the Breast Cancer Wisconsin Dataset
  - Optionally, use wine data set from scikit-learn and see the comparative performance of all classification models you have learnt in AAM-IPL
- Dataset – Breast Cancer
  - Description:
    - Implements an LDA/GDA classifier to predict the class of breast tumor (breast cancer) from the provided dataset – malignant or benign.
    - The dataset should be used for training a SVM classifier and evaluate its performance using various metrics such as accuracy, precision, recall, F1-score.
  - Dataset Details:
    - The breast cancer dataset consists of 569 samples, each representing a patient with a set of features
  - Data Source/Published By:
    - Breast Cancer Wisconsin (Diagnostic) - UCI Machine Learning Repository
    - Nuclear feature extraction for breast tumor diagnosis | Semantic Scholar
    - Supported by CS Department, University of Wisconsin-Madison
  - Features & Data Download Link
    - Breast Cancer Wisconsin (Diagnostic) - UCI Machine Learning Repository
    - Alternatively, you can load the data directly using sklearn datasets as shown on the side code snippet

```python
from sklearn.datasets import load_breast_cancer
import pandas as pd

# Load the dataset
data = load_breast_cancer()
X = pd.DataFrame(data.data,
columns=data.feature_names)
y = pd.Series(data.target, name="target")
```

Project Announcement

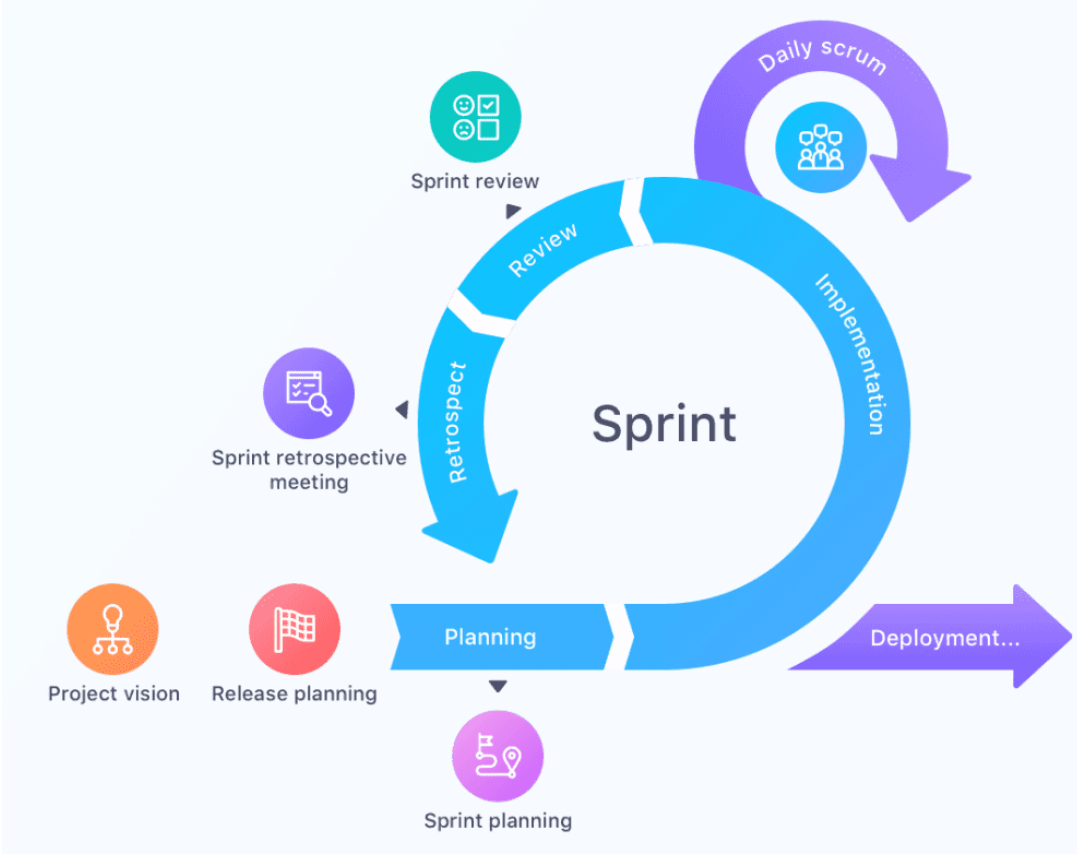
- Dataset – Wine Data
  - Description:
    - The Wine dataset contains information about chemical analysis of wines from the Italian region of Piedmont, which are produced by three different cultivars. This dataset is widely used for classification tasks, particularly for distinguishing wine types based on chemical properties.
  - Dataset Details:
    - Number of Instances: 178
    - Number of Features: 13 (not including the target class)
    - Target: The dataset has 3 classes, corresponding to three wine cultivars (Class 0, Class 1, and Class 2)
  - Data Source/Published By:
    - Published By: This dataset was made available by the UCI Machine Learning Repository.
    - Original Source: Forina, M., et al. (1991) in "PARVUS - An Extendible Package for Data Exploration, Classification and Correlation"
  - Features
    - Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids
    - Non-flavanoid phenols, Proanthocyanins, Color intensity, Hue
    - OD280/OD315 of diluted wines, Proline
    - Each feature represents a chemical property used to classify the wine samples
  - Data Download Link
    - You can download the Wine dataset from the UCI Machine Learning Repository - UCI Wine Dataset
    - Alternatively, you can load the data directly using sklearn datasets as below

```
from sklearn.datasets import load_wine
import pandas as pd

# Load the dataset
data = load_wine()
X = pd.DataFrame(data.data,
columns=data.feature_names)
y = pd.Series(data.target, name="target")
```

Project Implementation Steps

1. Import necessary libraries
- Load and preprocess dataset
- Split dataset into training and test sets
- Standardize features
- Define models
- Initialize results list
- Define function to add watermark to plots
- Train models and evaluate
- Convert results to data frame
- Plotting metrics and times with AAM-IPL watermark
- Plot combined ROC curves with AAM-IPL watermark
- Plot combined confusion matrices with AAM-IPL watermark
- Generate the PDF of code and output of project Jupyter file



Project Implementation Demo

Interested in building a Gen AI application?

Reach out to venkat@brillium.in

AAML-IPL Brought You in Partnership with:

Brillium Technologies

Sector 7, HSR Layout, Bengaluru 560102, Karnataka, India
Website: www.brillium.in | Email: connect@brillium.in