Supervised Learning: Classification, Regression and Decision Trees

UNIVERSITY OF OXFORD
Mathematical Institute

Lida Kanari

Mathematical Institute
University of Oxford

Introduction to Machine Learning, October 2025

Oxford Mathematics

Table of Contents

In this lecture, we will cover the following topics:

- Key concepts of Supervised Learning
- Classification
- Bayes Classifier
- Decision Trees
- Regression

Supervised Learning

Supervised learning consists of algorithms that are trained on labeled data. It is suitable for solving problems where the goal is to predict the output value based on the input data.

- There is a clear relationship between the input data and the output labels.
- Sufficient labeled data is available to train the algorithm.
- The problem can be defined as a classification or regression task.
- The goal is to make accurate predictions on new, unseen data.

Supervised Learning: Problem Setup

Definition. Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ the output space. Given a training set

$$S = \{(x_i, y_i)\}_{i=1}^{n} \subset \mathcal{X} \times \mathcal{Y},$$

the goal of supervised learning is to find a function

$$f : \mathcal{X} \to \mathcal{Y}$$

such that f generalizes well to unseen data:

$$S' = \{(x_i', y_i')\}_{i=1}^{m} \subset \mathcal{X} \times \mathcal{Y},$$

Setup. Given the training data

$$(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, \quad i = 1, \ldots, n,$$

the goal is to predict a new $y \in \mathcal{Y}$ from a previously unseen $x \in \mathcal{X}$.
Prediction rule. A supervised learning algorithm outputs a function

$$f : \mathcal{X} \to \mathcal{Y}.$$

Challenge: Good predictions on unseen (test) data.

Question: What would be the best possible predictor for $p(\mathcal{Y} \mid \mathcal{X})$ if the joint distribution $p(\mathcal{X}, \mathcal{Y})$ is known?

- Random variables $(\mathcal{X}, \mathcal{Y}) \sim p$ on $\mathcal{X} \times \mathcal{Y}$.
- Prediction function $f : \mathcal{X} \to \mathcal{Y}$.
- Performance measured by a *loss function* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Loss Function

Definition. A *loss function* measures the penalty for predicting $z \in \mathcal{Y}$ when the true label is $y \in \mathcal{Y}$:

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}, \qquad (y, z) \mapsto \ell(y, z).$$

Interpretation:

- $\ell(y, z) = 0$ when prediction z is correct.
- Larger $\ell(y, z)$ means worse prediction.

Role in learning: The choice of ℓ determines what we consider a "good" predictor and drives both theoretical analysis and practical algorithms.

Supervised Learning Problems and Loss Functions

Example	**Task**	**Loss function** $\ell(y, z)$
Binary classification	$y \in \{-1, 1\}$	$\mathbf{1}_{y \neq z}$ (0–1 loss)
Multiclass classification	$y \in \{1, \ldots, K\}$	$\mathbf{1}_{y \neq z}$ (0–1 loss)
Regression	$y \in \mathbb{R}$	$(y - z)^2$ (squared loss)
Regression	$y \in \mathbb{R}$	$\lvert y - z \rvert$ (absolute loss)

Supervised Learning Problems

Examples of supervised learning problems:

- **Binary classification:** Is this image a Cat $y = 1$ or a Dog $y = -1$?
- **Multiclass classification:** Given a set of medical exam measurements classify the patient into the following categories: Healthy $y = 1$, Disease 1 $y = 2$, Disease 2 $y = 3, \ldots$
- **Regression:** Predicting house prices from a list of features (house size, location, dates).

Definition of errors (risks)

Expected risk (generalization error):

$$R(f) = \mathbb{E}[\ell(\mathcal{Y}, f(\mathcal{X}))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, f(x))\, dp(x, y).$$

Empirical risk (training error):

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)).$$

Classification of a Medical Condition

We want to predict the condition of a patient, based on some medical observations. If the patient is healthy (H), the doctor does not need to prescribe anything. If the patient has a virus (V), the doctor will prescribe painkillers. If the patient has a bacterial infection (B) the doctor needs to prescribe antibiotics. The doctor will assess the condition based on the following features:

- Temperature
- Blood pressure
- Coughing frequency

Classification of a Medical Condition

We want to predict the condition of a patient, based on some medical observations.

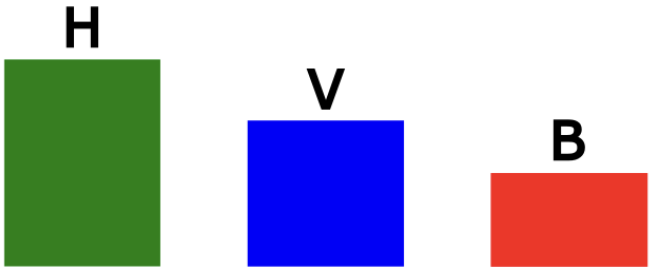


Figure: Patients distributed in three medical conditions: healthy (H), viral (V) or bacterial (B) infection.

Classification of a Medical Condition

We want to predict the condition of a patient, based on some medical observations:

	PatientID	Temperature	BloodPressure	CoughFreq	Condition
0	81	39.73	118.5	14.0	B
1	85	38.61	133.7	13.6	B
2	34	35.75	115.9	2.9	H
3	82	40.70	139.3	13.8	B
4	94	38.80	147.1	15.0	B
5	18	36.85	128.0	0.6	H
6	37	36.77	127.3	0.4	H
7	83	39.83	130.1	16.7	B
8	70	38.70	114.6	6.4	V
9	66	38.21	114.1	3.9	V

Figure: Medical conditions: healthy (H), viral (V) or bacterial (B) infection.

Classification of a Medical Condition

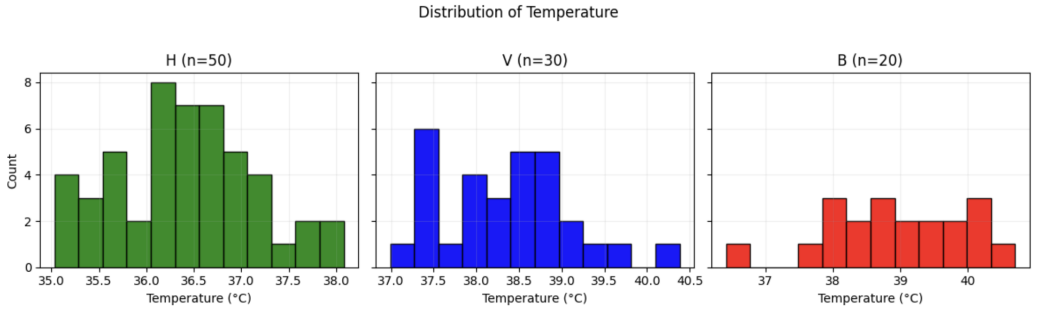


Figure: Temperature by condition: healthy (H), viral (V) or bacterial (B) infection.

Classification of a Medical Condition

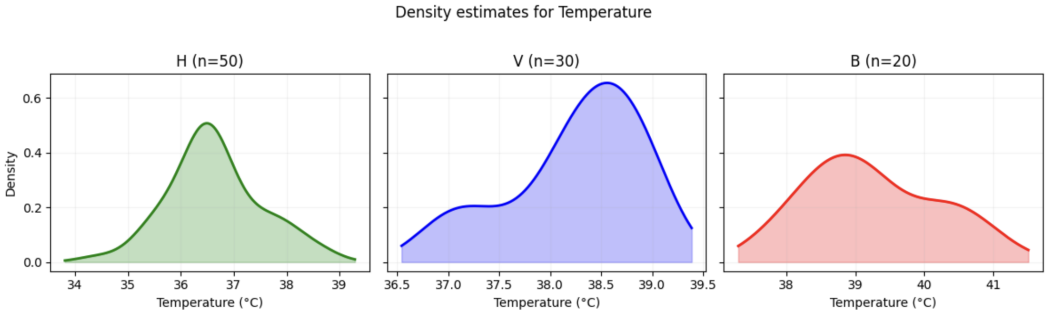


Figure: Temperature density by condition: healthy (H), viral (V) or bacterial (B) infection.

Probabilistic view on the classification task

We will make some assumptions so that we can use probabilities on our problem.

- Independence between the features.
- Each feature within each condition is statistically distributed according to a specific known distribution.

Probabilistic view on the classification task

Let's assume that the features can be characterized by a normal distribution. For example, the temperature of each patient is normally distributed according to the following:

$$p(T \mid c, \boldsymbol{w}) \approx N(\mu_c, \sigma_c^2)$$

where $\boldsymbol{w}$ is a set of parameters:

$$\boldsymbol{w} = (\mu_H, \mu_V, \mu_B, \sigma_H^2, \sigma_V^2, \sigma_B^2)$$

Probabilistic view on the classification task

Let's assume that the features can be characterized by a normal distribution for each condition:

$$N(\mu_H, \sigma_H^2)$$
$$N(\mu_V, \sigma_V^2)$$
$$N(\mu_B, \sigma_B^2)$$

Probabilistic view on the classification task

Let's assume that the features can be characterized by a normal distribution for each condition:

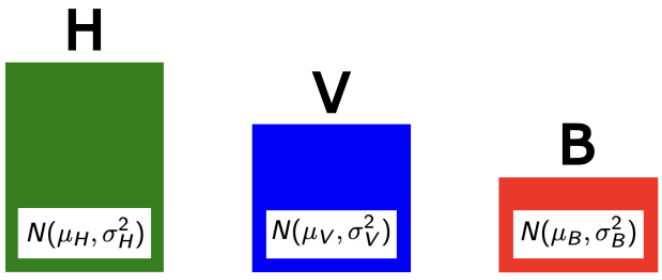


Figure: Patients distributed in three medical conditions: healthy (H), viral (V) or bacterial (B) infection.

Different views on the classification problem

The probability of an event represents the frequency over a large number of samples. For example, if I keep taking measurements of the temperature T of patients, the proportion of temperatures of healthy patients in the interval $[37, 38]^\circ C$ degrees will converge to:

$$P(T = [37, 38] \mid H, \boldsymbol{w}) = \int_{37}^{38} p(T \mid H, \boldsymbol{w}) dT$$

Different views on the classification problem

The probability of an event, represents the degree of belief about the event. For example, if I take the temperature of a patient, I cannot predict the exact value, but I can estimate the probability that it will be between $[37, 38]^\circ C$ degrees as:

$$P(T = [37, 38] \mid H, \boldsymbol{w})$$

Furthermore, if $P(T = [37, 38] \mid H, \boldsymbol{w}) > P(T = [38, 39] \mid H, \boldsymbol{w})$ it is more likely that the measurement will be between $[37, 38]^\circ C$ degrees.

Bayes classification task

How do we infer the posterior $p(\boldsymbol{w} \mid \mathcal{X})$ from the input data $\mathcal{X}$?

Bayes classification task

How do we infer the posterior $p(\boldsymbol{w} \mid \mathcal{X})$ from the input data $\mathcal{X}$?

Let's recall from Bayes probability:

$$p(\mathcal{X}, \boldsymbol{w}) = p(\boldsymbol{w} \mid \mathcal{X}) p(\mathcal{X})$$

Bayes classification task

How do we infer the posterior $p(\boldsymbol{w} \mid \mathcal{X})$?

Let's recall from Bayes probability:

$$p(\mathcal{X}, \boldsymbol{w}) = p(\boldsymbol{w} \mid \mathcal{X})p(\mathcal{X}) = p(\mathcal{X} \mid \boldsymbol{w})p(\boldsymbol{w})$$

Bayes classification task

How do we infer the posterior $p(\boldsymbol{w} \mid \mathcal{X})$?

Let's recall from Bayes probability:

$$p(\mathcal{X}, \boldsymbol{w}) = p(\boldsymbol{w} \mid \mathcal{X})p(\mathcal{X}) = p(\mathcal{X} \mid \boldsymbol{w})p(\boldsymbol{w})$$

Bayes theorem:

$$p(\boldsymbol{w} \mid \mathcal{X}) = \frac{p(\mathcal{X} \mid \boldsymbol{w})p(\boldsymbol{w})}{p(\mathcal{X})}$$

Bayes classification task

Bayes theorem:

$$p(\boldsymbol{w} \mid \mathcal{X}) = \frac{p(\mathcal{X} \mid \boldsymbol{w})p(\boldsymbol{w})}{p(\mathcal{X})}$$

where,
$p(\boldsymbol{w})$ is the prior representing the initial belief about $\boldsymbol{w}$.
$p(\boldsymbol{w} \mid \mathcal{X})$ is the posterior representing the updated belief about $\boldsymbol{w}$ after observations of data D.
$p(\mathcal{X} \mid \boldsymbol{w})$ is the likelihood that the data $\mathcal{X}$ are observed for the distributions with parameters $\boldsymbol{w}$.

Bayes classification task

Two types of uncertainty:

1. Epistemic: which model is correct, given the data we have observed? The more patient data I collect, the more certain I am about μ_H, σ_H^2.

Bayes classification task

Two types of uncertainty:

1. Epistemic: which model is correct, given the data we have observed? The more patient data I collect, the more certain I am about μ_H, σ_H^2.
2. Aleatoric: related to the stochastic nature of the variables. I will never be able to predict a precise temperature measurement even when I know exactly the condition of the patient.

Bayes classification task

Based on our assumptions:

- Independence of features
- Normal distribution of features within a class

Bayes classification task

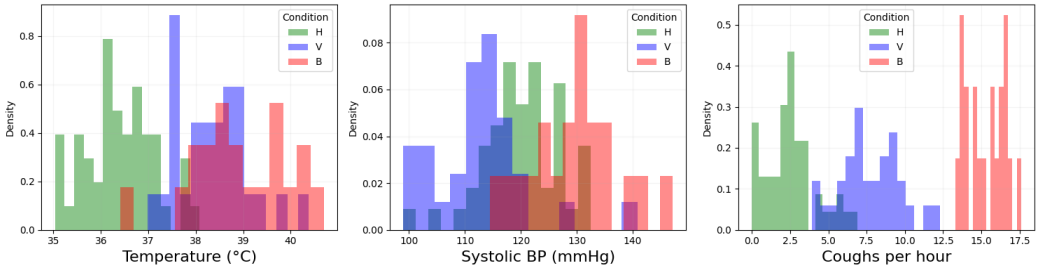
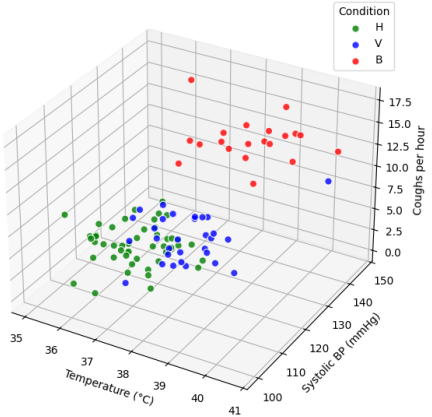


Figure: Temperature, blood pressure and coughing frequency for: healthy (H), viral (V) or bacterial (B) infection.

Three dimensional representation of the medical conditions



Bayes Classifier

Definition. Suppose $\mathcal{Y} = \{1, \ldots, K\}$ is a finite label set. The *Bayes classifier* assigns to each input $x \in \mathcal{X}$ the class with the highest posterior probability:

$$f^*(x) = \arg\max_{y \in \mathcal{Y}} \mathbb{P}(\mathcal{Y} = y \mid \mathcal{X} = x).$$

Intuition: The Bayes classifier uses complete knowledge of the data distribution to minimize the classification error.

Key property: Assuming perfect knowledge of $p(x, y)$ then f^* is the optimal classifier, achieving the lowest possible error (the *Bayes error*).

Bayes Classifier: Example

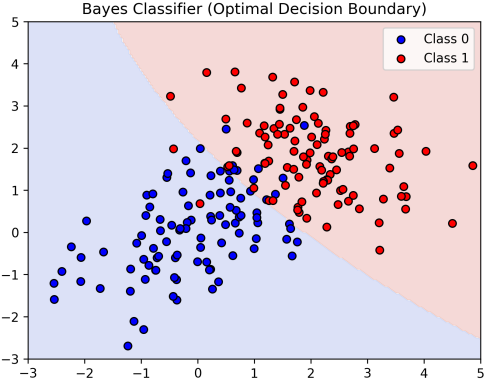


Figure: Example of a Bayes classifier in a binary classification problem

Bayes Classifier: Medical use case

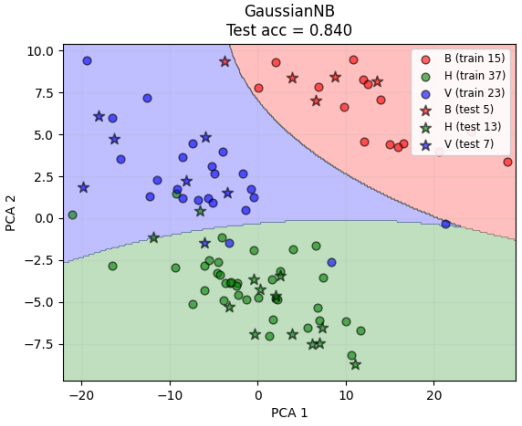


Figure: Bayes classifier to distinguish medical conditions.

Bayes Risk

Definition. The lowest achievable risk given full knowledge of p is the *Bayes risk*:

$$R^* = \mathbb{E}_X \left[\inf_{z \in \mathcal{Y}} \mathbb{E}[\ell(y, z) \mid \mathcal{X}] \right].$$

Excess risk is defined as:

$$R(f) - R^* \geq 0.$$

Bayes Predictor

Definition. The Bayes optimal predictor minimizes conditional risk:

$$R^*(x) = \arg\min_{z \in \mathcal{Y}} \mathbb{E}[\ell(Y, z) \mid X = x].$$

Special cases:

- **Classification:** $0 - 1$ loss
- **Regression:** squared loss

Example: Binary Classification

Let $\eta(x) = p(\mathcal{Y} = 1 \mid \mathcal{X} = x)$.

Bayes classifier:

$$f^*(x) = \begin{cases} 1 & \text{if } \eta(x) > 1/2, \\ -1 & \text{if } \eta(x) < 1/2. \end{cases}$$

Bayes risk:

$$R^* = \mathbb{E}[\min\{\eta(x),\, 1 - \eta(x)\}]\,.$$

Bayes Classifier: Error

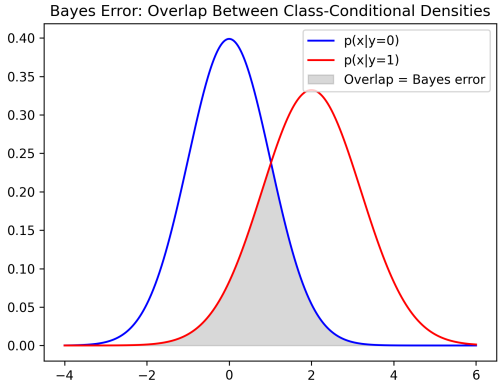


Figure: Example of a Bayes error

Bayes Classifier: Error on Example

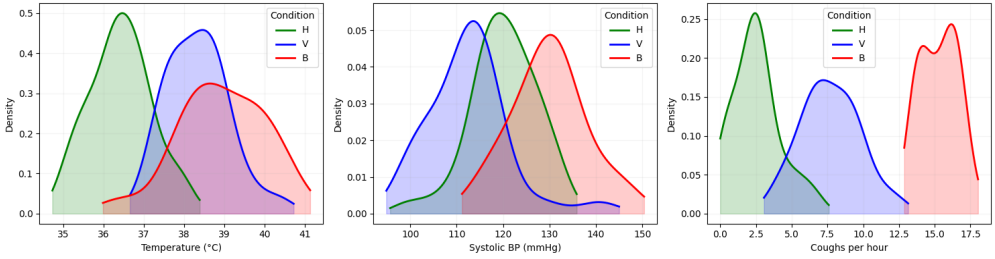


Figure: Bayes error in the classification of medical conditions.

Key idea: The Bayes predictor is optimal but requires full knowledge of $p(x, y)$.
In practice: The distribution $p(x, y)$ is unknown.

Objective: Approximate the Bayes predictor from finite training data:

- **Generative methods:** approximate $p(x \mid y)$ and $p(y)$, then apply Bayes rule.
- **Discriminative methods:** approximate f^* which corresponds to $p(y \mid x)$ directly from the data.

Generative vs. Discriminative Models

Discriminative models:

- Model the conditional distribution $p(y \mid x)$ directly.
- Learn a decision boundary between classes.
- Examples: Logistic regression, SVMs, neural networks.

Generative models:

- Model the joint distribution $p(x, y) = p(y)\, p(x \mid y)$.
- Capture how the data x is generated for each class y.
- Use Bayes' rule to compute $p(y \mid x)$:

$$p(y \mid x) = \frac{p(y)\, p(x \mid y)}{\sum_{y'} p(y')\, p(x \mid y')}.$$

- Examples: Naive Bayes, LDA, QDA.

Focus on Generative Models

We model:

$$p(x, y \mid \theta, \pi) = p(y \mid \pi)\, p(x \mid y, \theta)$$

where

$$p(y = c \mid \pi) = \pi_c, \qquad \sum_c \pi_c = 1.$$

Procedure:

1. Choose a model for $p(x \mid y = c, \theta_c)$ (e.g. Gaussian, multinomial).
2. Estimate parameters π_c, θ_c from training data.
3. Predict using:

$$\hat{y} = \arg\max_c \; \pi_c\, p(x \mid y = c, \theta_c).$$

Naïve Bayes Classifier (NBC)

Assumption: Features are conditionally independent given the class.

$$p(x \mid y = c, \theta_c) = \prod_{j=1}^{D} p(x_j \mid y = c, \theta_{jc})$$

Advantages:

- Simple and fast to train.
- Surprisingly effective for text and categorical data.

Limitations:

- Independence assumption rarely true.
- Model is often misspecified but works well for classification.

Linear Discriminant Analysis (LDA)

Goal: classify by modelling how each class *generates* the data, then use Bayes' rule. Key idea:

- For each class c assume the features $x \in \mathbb{R}^D$ are Gaussian with mean μ_c and a *shared* covariance Σ.
- Because covariances are shared, class-conditional contours are ellipses with the same shape, only their centers differ.
- Decision boundaries between classes become **linear** in x.

LDA

Assume for each class c:

$$p(x \mid y = c) = \mathcal{N}(x \mid \mu_c, \Sigma).$$

Prior class probabilities: $\pi_c = p(y = c)$, $\sum_c \pi_c = 1$.
Posterior (unnormalised):

$$p(y = c \mid x) \propto \pi_c \exp\Big(-\tfrac{1}{2}(x - \mu_c)^\top \Sigma^{-1}(x - \mu_c)\Big).$$

Algebra (rearrange quadratic term):

$$p(y = c \mid x) \propto \exp\big(\beta_c^\top x + \gamma_c\big),$$

with

$$\beta_c = \Sigma^{-1}\mu_c, \qquad \gamma_c = -\tfrac{1}{2}\mu_c^\top \Sigma^{-1}\mu_c + \log \pi_c.$$

LDA

- μ_c: empirical mean of class c (centre of class c).
- Σ: shared covariance across classes: controls feature scaling and correlation.
- $\beta_c = \Sigma^{-1}\mu_c$: a weight vector: direction in which class c pulls the log-odds.
- γ_c: offset combining class prior π_c and a quadratic correction $-\frac{1}{2}\mu_c^\top \Sigma^{-1}\mu_c$.
- Decision function for class c: $g_c(x) = \beta_c^\top x + \gamma_c$. Predict class with largest $g_c(x)$.

Training (ML estimates):

1. Compute class counts N_c and priors $\hat{\pi}_c = N_c/N$.
2. Compute class means: $\hat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} x_i$.
3. Compute pooled covariance:

$$\hat{\Sigma} = \frac{1}{N-C} \sum_{c} \sum_{i:y_i=c} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^\top$$

4. Set $\hat{\beta}_c = \hat{\Sigma}^{-1}\hat{\mu}_c$, $\hat{\gamma}_c = -\frac{1}{2}\hat{\mu}_c^\top \hat{\Sigma}^{-1}\hat{\mu}_c + \log \hat{\pi}_c$.

Quadratic Discriminant Analysis (QDA)

Goal: Similar to LDA but allows more flexibility in the shapes.

- Each class c has its own covariance Σ_c and mean μ_c.
- Class-conditional contours can have different shapes and orientations.
- Decision boundaries between classes are in general **quadratic** surfaces (ellipses, parabolas, etc).

Assume for each class c:

$$p(x \mid y = c) = \mathcal{N}(x \mid \mu_c, \Sigma_c).$$

Using Bayes' rule:

$$p(y = c \mid x) \propto \pi_c \, |\Sigma_c|^{-1/2} \exp\Big(-\tfrac{1}{2}(x - \mu_c)^\top \Sigma_c^{-1}(x - \mu_c) \Big).$$

Decision rule (take log and compare):

$$\hat{y} = \arg\max_c \Big[-\tfrac{1}{2}(x - \mu_c)^\top \Sigma_c^{-1}(x - \mu_c) - \tfrac{1}{2}\log|\Sigma_c| + \log \pi_c \Big].$$

Because of the quadratic term $x^\top \Sigma_c^{-1} x$ that depends on c, the boundary is quadratic.

QDA

- μ_c: mean (center) of class c.
- Σ_c: class-specific covariance (controls shape/orientation of class c's density).
- $-\frac{1}{2}(x-\mu_c)^\top \Sigma_c^{-1}(x-\mu_c)$: log-likelihood for class c.
- $-\frac{1}{2}\log|\Sigma_c|$: volume term, penalises classes with large covariance.
- $\log \pi_c$: prior log-probability (accounts for class imbalance).
- The quadratic term in x depends on Σ_c^{-1}, hence different Σ_c leads to quadratic decision boundaries.

QDA Algorithm

Training (ML estimates):

1. Compute class counts N_c and priors $\hat{\pi}_c = N_c/N$.
2. Compute class means: $\hat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} x_i$.
3. Compute class covariances:

$$\hat{\Sigma}_c = \frac{1}{N_c} \sum_{i:y_i=c} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^\top.$$

Prediction: for a test x compute scores

$$s_c(x) = -\tfrac{1}{2}(x - \hat{\mu}_c)^\top \hat{\Sigma}_c^{-1}(x - \hat{\mu}_c) - \tfrac{1}{2} \log |\hat{\Sigma}_c| + \log \hat{\pi}_c,$$

and predict $\arg\max_c s_c(x)$.

Comparing LDA and QDA

Summary table:

Model	Covariance	Decision boundary
LDA	Σ (shared)	Linear
QDA	Σ_c (class-specific)	Quadratic
Naïve Bayes	Diagonal / independent	Linear/naive

Comparing LDA and QDA

Trade-offs:

- QDA more flexible (can model complex class shapes) but needs more data to estimate Σ_c.
- LDA is a compromise: fewer parameters $\rightarrow$ lower variance but possibly higher bias if covariances truly differ.
- Naïve Bayes imposes stronger independence assumptions and is very parameter-efficient.

Practical rule: use QDA if you have lots of data and evidence that the covariances differ.

Generative classifiers: LDA and QDA

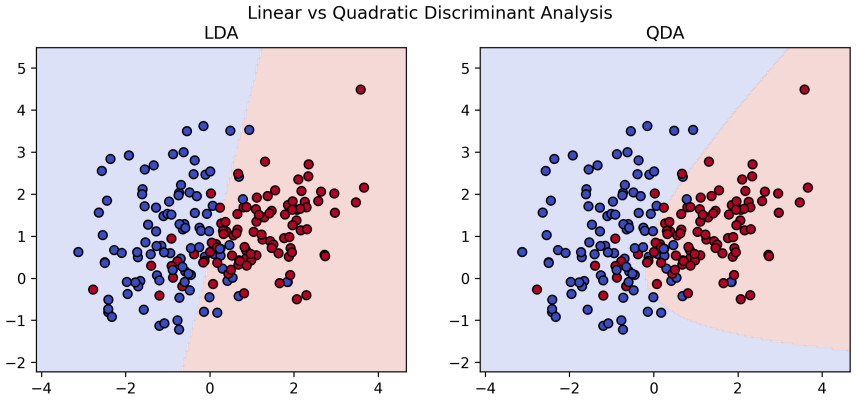
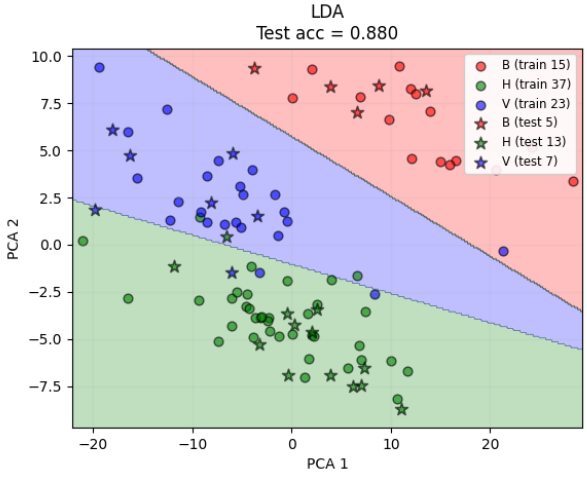
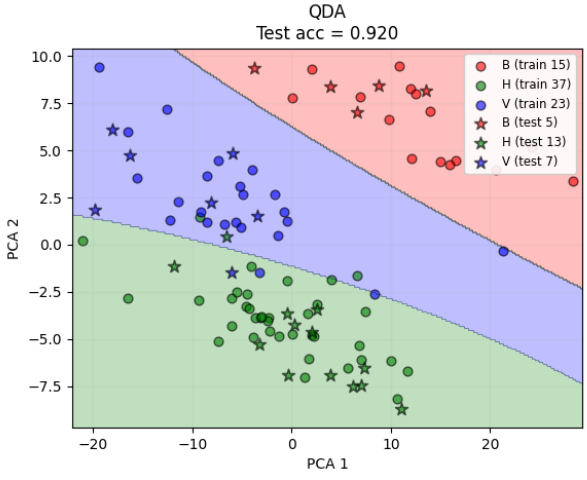


Figure: Example of LDA vs QDA

LDA on the medical example



QDA on the medical example



Definition. In regression, the output space is continuous, $\mathcal{Y} \subseteq \mathbb{R}$.
The goal is to find a function $f : \mathcal{X} \to \mathbb{R}$ that predicts $\mathcal{Y}$ from $\mathcal{X}$ by minimizing the expected squared error:

$$f^*(x) = \arg\min_f \ \mathbb{E}\big[(\mathcal{Y} - f(\mathcal{X}))^2 \mid \mathcal{X} = x\big] .$$

Result: The optimal predictor is the conditional expectation:

$$f^*(x) = \mathbb{E}[\mathcal{Y} \mid \mathcal{X} = x].$$

Example: Linear regression assumes $f(x) = w^\top x + b$.

Linear Regression

Given $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, we model

$$f(x) = w^\top x + b.$$

The least-squares estimator minimizes

$$\hat{R}_n(w, b) = \frac{1}{n} \sum_{i=1}^{n} (y_i - w^\top x_i - b)^2.$$

Solution:

$$\hat{w} = (X^\top X)^{-1} X^\top y, \qquad \hat{b} = \bar{y} - \hat{w}^\top \bar{x},$$

provided $X^\top X$ is invertible.

Linear Regression

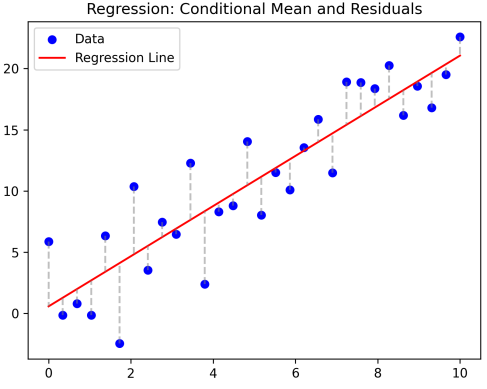


Figure: Example of Linear Regression

Logistic Regression (Classification problem)

Setup: Binary classification with $\mathcal{Y} = \{-1, 1\}$.
Model: Posterior probability

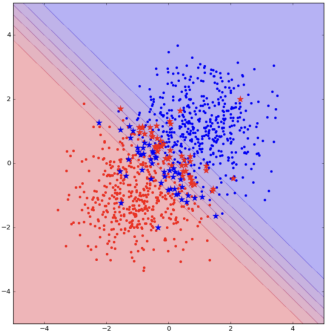
$$p(y = 1 \mid x) = \sigma(w^\top x + b), \quad \sigma(t) = \frac{1}{1 + e^{-t}}.$$

Decision rule:

$$f(x) = \text{sign}(w^\top x + b).$$

Learning: Parameters (w, b) are estimated by minimizing the *logistic loss*, a convex surrogate of the 0–1 loss.

Logistic Regression



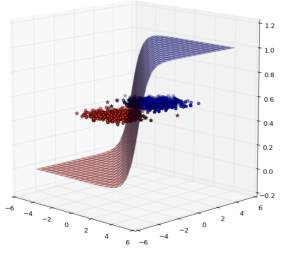


Figure: Example of Logistic Regression (image from Prof. Seth Flaxman)

Decision Trees

Idea: Partition the input space $\mathcal{X}$ into regions and assign a prediction in each region.

Construction:

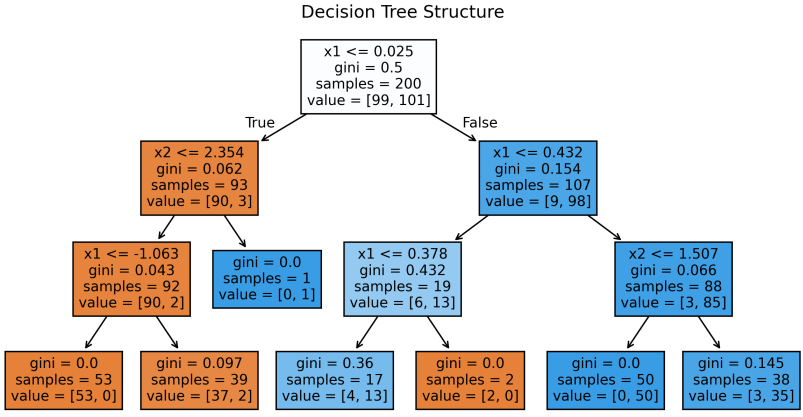
- Recursively split $\mathcal{X}$ along feature dimensions.
- At each node, choose the split that maximizes class separation (e.g., Gini index, entropy).

Prediction: A new point x is assigned to the region it falls into; prediction is the majority label (classification) or mean response (regression).

Remarks:

- Easy to interpret.
- High variance (often reduced with ensembles: random forests, boosting).

Decision Trees



Decision Trees

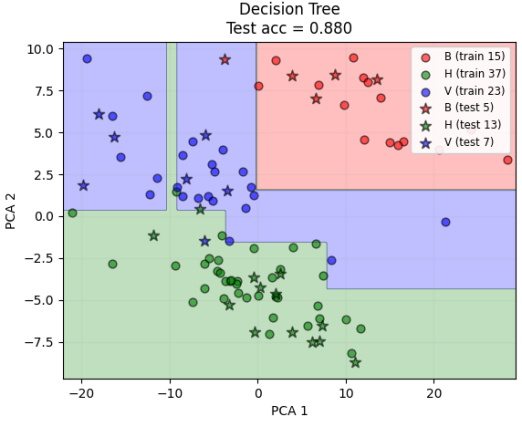


Figure: Decision Trees of medical example

Thank you! Questions?