GATE – Data Science and Artificial Intelligence (DA)

Machine Learning: Supervised Learning

Regression Problems

simple linear regression

multiple linear regression

ridge regression

Classification Problems

logistic regression

k-nearest neighbour

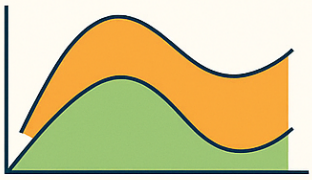
naive Bayes classifier

linear discriminant analysis

support vector machine

decision trees

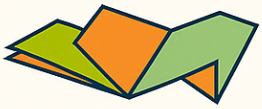
Bias-Variance Trade-off Graph



Cross-Validation Methods

leave-one-out cross-validation

k-fold cross-validation



Neural Networks

multi-layer perceptron

feed-forward neural network

GateXAIML

2025

Contents

About the Book

Artificial Intelligence and Machine Learning (AI/ML) are transforming industries across the globe — from healthcare and finance to transportation and education. From medical diagnosis systems and fraud detection to personalized recommendations and autonomous vehicles, AI/ML is shaping the way we live, work, and interact with technology.

To support this rapidly growing field, the GATE Data Science and Artificial Intelligence (DA) exam was introduced as a national-level gateway to higher studies, research, and employment opportunities in top institutions and organizations. The exam tests a candidate's proficiency in mathematics, programming, data handling, machine learning, and AI fundamentals.

This book is a compact and comprehensive guide for GATE DA aspirants. It is designed to help learners build a strong conceptual foundation while developing the problem-solving skills required for the exam. Many solved examples are included to illustrate key concepts, and each chapter features carefully crafted problems for practice.

Solutions to selected problems and topic-wise lectures will be discussed in detail on my YouTube channel (@GATEXAiml). All the concepts covered in the book will also be taught step-by-step through video tutorials, making this a complete learning resource for GATE DA preparation.

This book is designed for aspirants of the GATE DA exam focusing on **Machine Learning: Supervised Learning**. It systematically covers theory, solved examples, and practice problems aligned with the official syllabus.

Dedicated to all my Gurus and Students.
"Knowledge grows only when shared — and it must remain free, for that is how it thrives."

Machine Learning: Supervised Learning - Syllabus

Supervised Learning: regression and classification problems, simple linearregression, multiple linear regression, ridge regression, logistic regression, k-nearest neighbour,naive Bayes classifier, linear discriminant analysis, support vector machine, decision trees, bias-variance trade-off, cross-validation methods such as leave-one-out (LOO) cross-validation, k-foldscross-validation, multi-layer perceptron, feed-forward neural network;

STOP!

Attention!

Some examples solved in video lectures are different from those given in this book.

The procedure to solve problems and examples is well explained in the video lectures, and it is highly recommended to go through the video lectures for complete understanding.

Official Video Playlist

Watch on YouTube

Chapter 1

Introduction to Machine Learning

1.1 What is Machine Learning

What is Machine Learning?

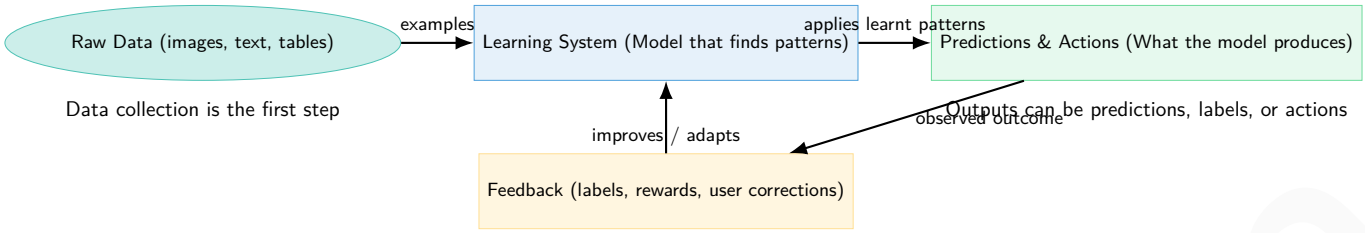
Machine Learning (ML) is the practice of building systems that improve their behaviour through experience (data). Instead of encoding exact rules for every case, ML systems detect patterns, form general rules from examples, and apply those rules to new situations. In simple terms: ML finds useful patterns in data and uses them to make predictions, decisions, or generate useful outputs.

Core Intuition

- **Learning from examples:** Provide examples that show input and desired output or let the system find structure in inputs alone.
- **Generalization:** The ability to perform well on new, unseen examples that come from the same kind of data.
- **Feedback loop:** The system improves when it receives signals telling it how well it did (explicit labels, rewards, or implicit signals like clicks).

A classroom analogy

Think of teaching a child to recognize apples. Rather than giving a strict rule ("red and round equals apple"), you show many examples (red apples, green apples, sliced apples). The child learns a flexible concept from examples and can correctly identify new apples despite differences in color, shape, or lighting.

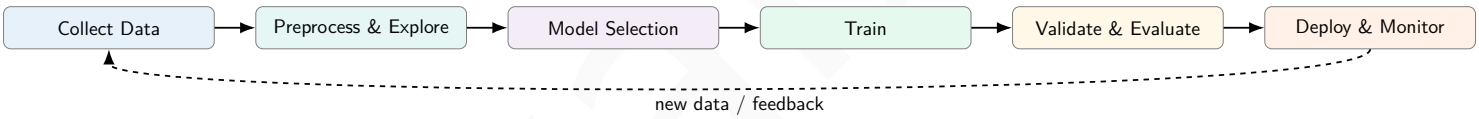


1.2 Overview of Machine Learning Paradigm

Big-picture learning loop

ML development typically follows a repeatable loop:

1. **Collect data** relevant to the problem.
2. **Preprocess and explore** data to understand its structure.
3. **Select or design** an appropriate model architecture.
4. **Train** the model on the data (learn patterns).
5. **Validate and evaluate** on held-out examples.
6. **Deploy and monitor** in production and iterate with new data.



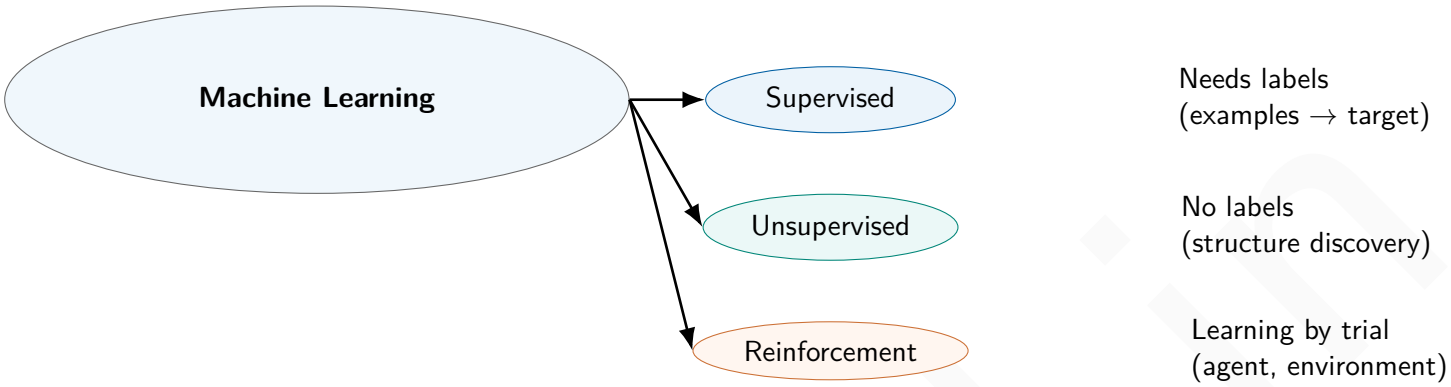
Why this loop matters

Iterating the loop allows improvement. Real-world ML rarely works on the first try — understanding data quirks, evaluating fairness, and updating models as environments change are essential steps in practical ML.

1.3 Types: supervised, unsupervised, and reinforcement learning

High-level categories

Supervised learning: Learn from examples where the desired outcome is given. Useful when labeled examples are available.
Unsupervised learning: Find structure in unlabeled data (grouping, representations).
Reinforcement learning: Learn to act through trial-and-error interactions with an environment using feedback in the form of rewards.



Quick relatable examples

- **Supervised:** Email labelled as "spam" or "not spam" → train a spam filter.
- **Unsupervised:** Customer purchase histories → group (cluster) customers into segments for marketing.
- **Reinforcement:** Robot learns to walk by trying actions and receiving rewards for forward progress and penalties for falling.

1.4 Data

1.4.1 Basic Types of Data

Kinds of data you will meet

- **Tabular:** Rows and columns (structured) — common in business data.
- **Text:** Documents, chat logs, reviews — sequential and high-dimensional.
- **Images:** Pixel grids with spatial structure.
- **Audio:** Waveforms and spectrograms, temporal patterns.
- **Time-series:** Measurements over time (stock prices, sensor readings).
- **Graphs:** Entities connected by edges (social networks, molecules).

Implications of data type

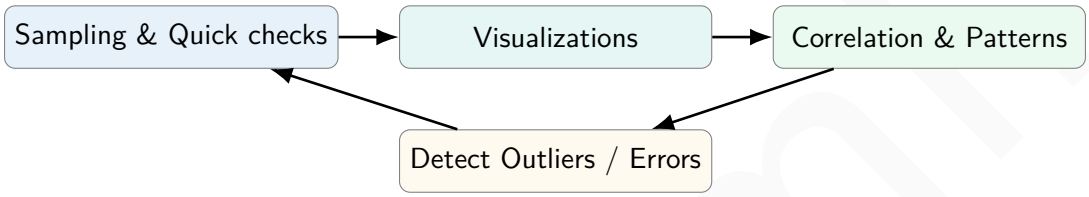
Different data types require different methods and preprocessing. For example, images benefit from convolutional approaches that exploit spatial locality; text needs tokenization and representations that handle order; graphs require methods that use connectivity structure.

1.4.2 Exploring Structure of Data

Why exploration matters

Before building models you must understand:

- What features exist and their distributions.
- Which features are noisy, missing, or correlated.
- Whether labels (if any) are balanced or skewed.
- Which exploratory visualizations reveal structure (histograms, scatter plots, example images).



1.4.3 Data Quality

Key dimensions of data quality

- **Completeness:** Are important values missing?
- **Consistency:** Do records contradict each other?
- **Accuracy:** Do values reflect reality or measurement error?
- **Timeliness:** Are data current and relevant?
- **Representativeness:** Does the data capture the population you care about?

Practical check-list for quality

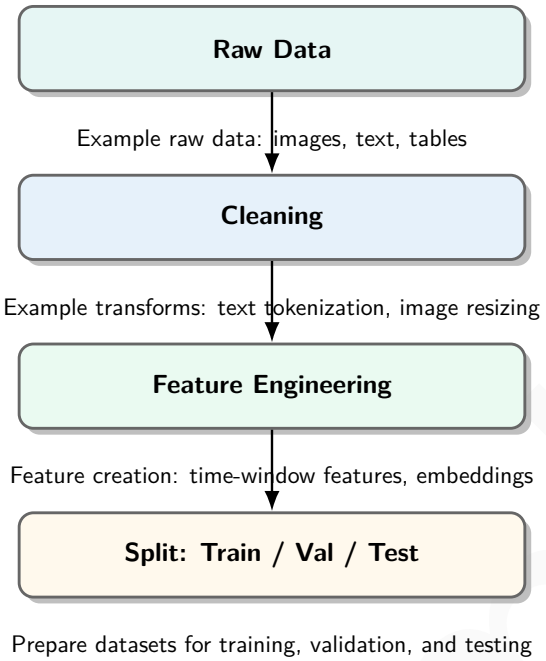
1. Inspect missing values and patterns of missingness.
2. Check for duplicated records.
3. Validate ranges (dates, values) and enforce constraints.
4. Sample labels for human verification if labels are noisy.
5. Consider whether historical data reflects the present.

1.4.4 Data Pre-Processing

Common preprocessing steps

Before training, data is commonly cleaned and transformed:

- **Cleaning:** remove or fix corrupted records, handle missing values.
- **Normalization / scaling:** make ranges consistent where needed.
- **Encoding:** transform categorical variables into numerical representations.
- **Feature creation:** derive higher-level features from raw inputs.
- **Splitting:** keep holdout data for fair evaluation (validation / test).



1.5 Parametric and Non-Parametric Functions

1.5.1 Parametric Functions

Parametric Functions

Parametric models assume a specific functional form for the relationship between input features X and output Y, characterized by a finite set of parameters θ.

$$Y = f(X; \theta)$$

Key Points:

- Requires assumption about the underlying distribution or functional form (e.g., linear, polynomial).
- Number of parameters is fixed and does not depend on the number of training samples.
- Learning reduces to estimating the parameters θ from data.
- Examples: Linear regression, Logistic regression, Polynomial regression.

Example 1:

Simple linear regression:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

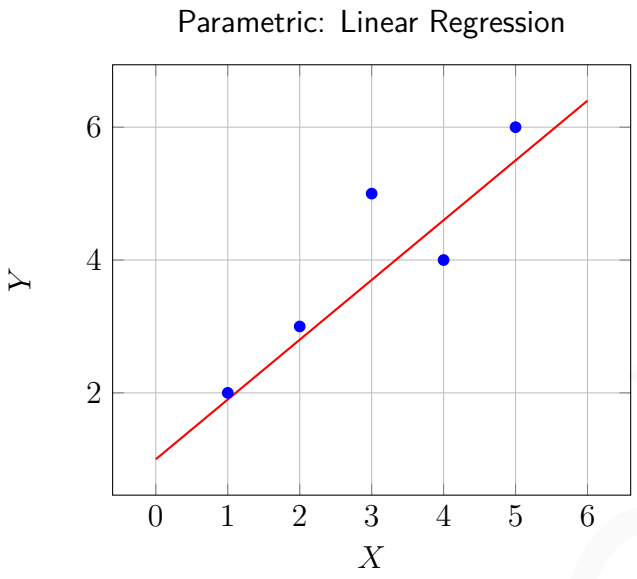
Here, the model is linear, and we estimate only β_0 (intercept) and β_1 (slope). Predictions for new X values are made using these parameters.

Example 2:

Logistic regression:

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

The model is parametric, with parameters β_0 and β_1 controlling the sigmoid function shape.



Parametric model: Linear regression line fitting the data.

1.5.2 Non-Parametric Functions

Non-Parametric Functions

Non-parametric models do not assume a fixed functional form. Instead, they learn the function directly from data.

Key Points:

- Model complexity can grow with the dataset.
- Often requires more data to avoid overfitting.
- Can capture complex, nonlinear relationships.
- Examples: k-Nearest Neighbors (k-NN), Decision Trees, Kernel Density Estimation.

Example 3:

k-NN regression predicts the value at $X = x$ as the average of Y values of the k nearest neighbors:

$$\hat{Y}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} Y_i$$

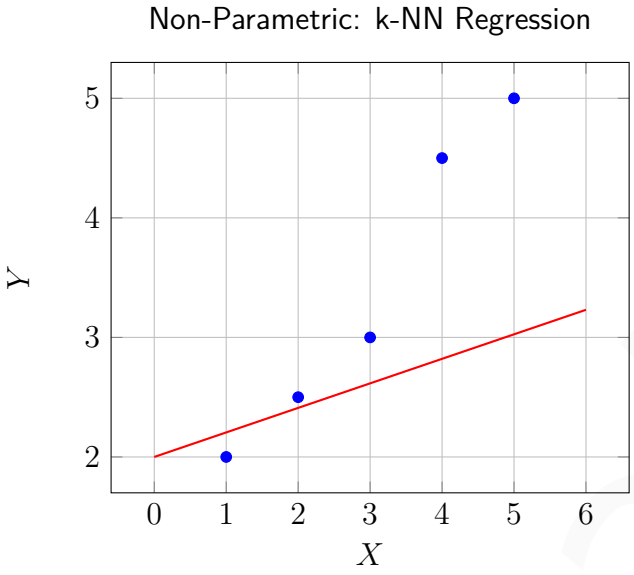
No explicit parameters are estimated; predictions depend entirely on the data points.

Example 4:

Decision trees split the feature space into regions. For features X_1 and X_2:

$$Y = \begin{cases} 5 & \text{if } X_1 < 3 \text{ and } X_2 > 2 \\ 7 & \text{if } X_1 \geq 3 \end{cases}$$

The function form emerges from the splits, not predefined parameters.



Non-parametric model: k-NN regression curve adapting to data points.

1.5.3 Comparison: Parametric vs Non-Parametric

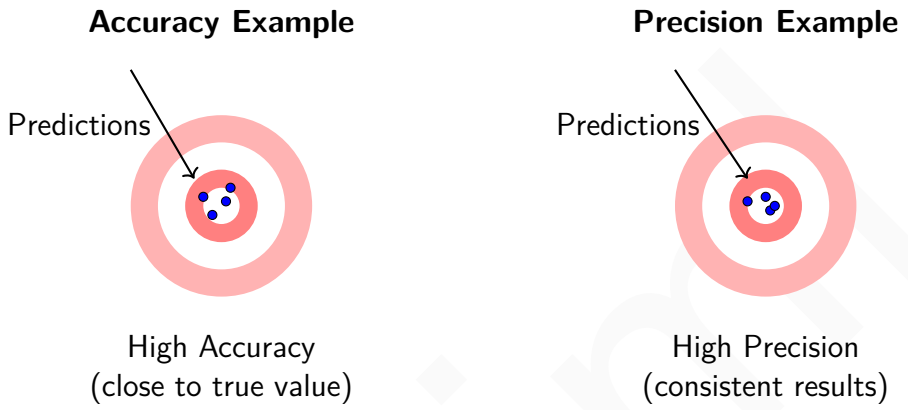
Comparison

Feature	Parametric	Non-Parametric
Assumption	Fixed functional form	No functional form assumed
Parameters	Finite, fixed number	Can grow with data
Data requirement	Less	More
Flexibility	Limited	High
Example	Linear regression, Logistic regression	k-NN, Decision Trees

1.6 Issues in ML

1.6.1 Accuracy and Precision

- **Accuracy:** Is the model giving the correct results overall (close to true value)?
- **Precision:** MIs the model consistent in giving the same correct result each time (consistent results)?

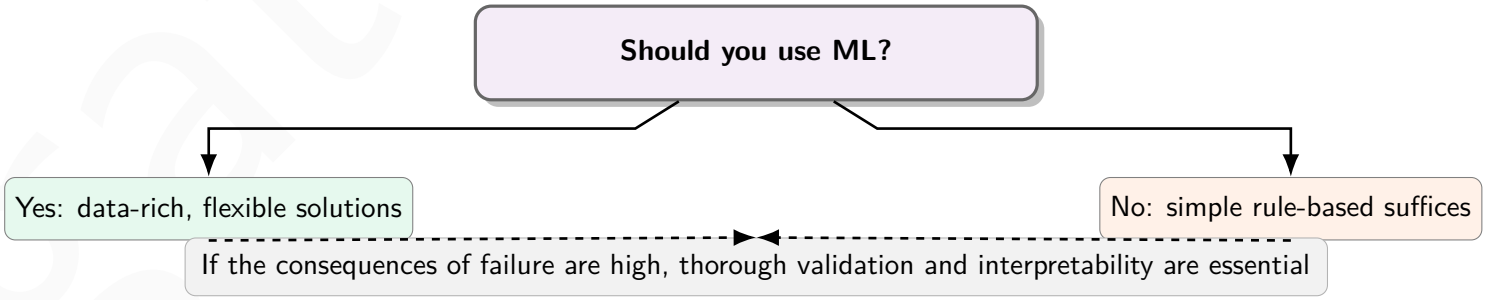


1.6.2 Issues

When not to use ML

ML is powerful, but it is not always the best tool. Avoid ML when:

- The task can be solved reliably with deterministic rules or simple logic.
- You lack sufficient, representative data covering real-world variability.
- You need fully explainable, provably-correct behaviour for safety-critical tasks (unless the ML system has been heavily validated).
- Data privacy, legal, or ethical constraints prevent the collection or usage of the required data.



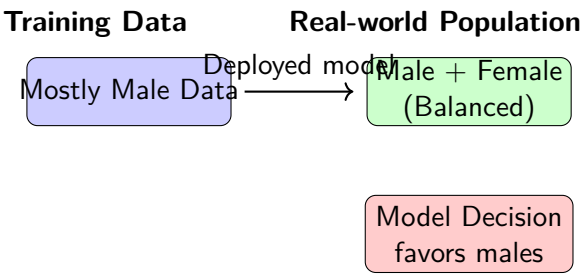
Common issues in practice

- **Bias in data:** Historical biases in training data lead to biased outputs.
- **Concept drift:** Data distributions change over time; models must be updated.
- **Overfitting:** Model learns noise or dataset quirks that don't generalize.

- **Underfitting:** Model is too simple to capture important structure.
- **Interpretability:** Some models (deep nets) are harder to explain.

Bias in Data

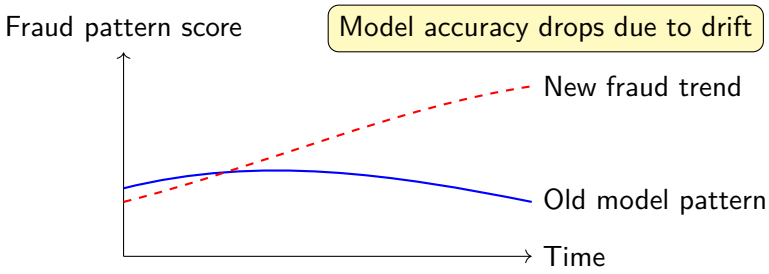
Definition: Data bias occurs when the training data does not fairly represent the real-world population. This leads the model to make systematically unfair or inaccurate predictions.
Example: A hiring model is trained mostly on male applicants. When used for new applicants, it prefers male candidates even with identical qualifications.



Consequence: The model generalizes poorly across groups, causing unfair treatment or legal risks.

Concept Drift

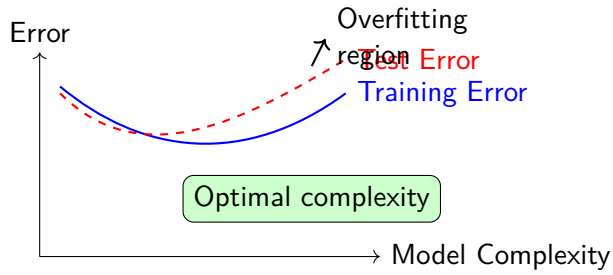
Definition: Concept drift occurs when the relationship between input features and output labels changes over time.
Example: A credit card fraud detection system trained in 2018 fails in 2025 because fraud patterns evolve.



Consequence: Models degrade unless retrained periodically with new data.

Overfitting

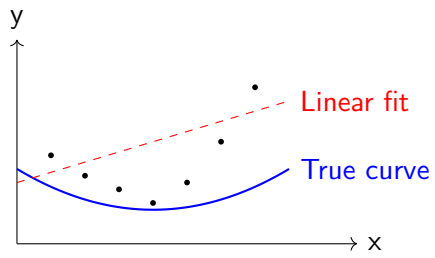
Definition: Overfitting happens when the model learns both the true pattern *and* the noise of the training set, performing poorly on unseen data.
Example: A decision tree memorizes every training example, achieving 100% accuracy on training but 60% on test data.



Consequence: Model appears perfect during training but fails in real applications.

Underfitting

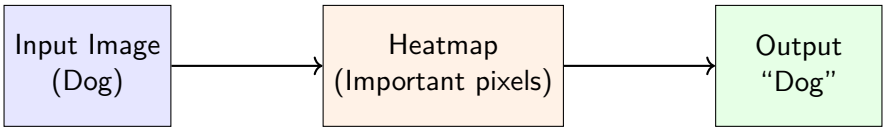
Definition: Underfitting occurs when a model is too simple to capture the true underlying pattern in data.
Example: Using a straight line to model a quadratic relationship $y = x^2$.



Consequence: Both training and test accuracy remain poor due to model simplicity.

Interpretability

Definition: Interpretability means understanding *why* a model makes certain predictions. Complex models like deep neural networks are powerful but often opaque.
Example: A CNN classifies an image as "dog", but we don't know which pixels or regions influenced the decision.



Consequence: Lack of interpretability makes debugging, fairness evaluation, and trust difficult.

1.6.3 Bias and Variance

Bias

In any predictive task, the model's error can be understood as a combination of three components:

- **Bias:** The systematic error introduced by approximating a complex problem with a simplified model. High bias implies the model consistently misses the target in a certain direction.
- **Variance:** The error due to sensitivity to small changes in the training data. High variance implies the model's predictions fluctuate a lot when data changes.
- **Irreducible noise:** Random variability in the system or inherent unpredictability.

The total expected error at a point x can be decomposed as:

$$\mathbb{E}[(\hat{f}(x) - f(x))^2] = \underbrace{(\text{Bias}[\hat{f}(x)])^2}_{\text{Squared Bias}} + \underbrace{\text{Var}[\hat{f}(x)]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible Noise}}$$

Example: Bias and Variance in Predictions

Suppose we try to predict a quantity y multiple times using a model and obtain these predictions:

$$\hat{y}_1 = 5.2, \quad \hat{y}_2 = 4.9, \quad \hat{y}_3 = 5.1$$

The true value is $y = 5$.

- **Bias:** Average difference between predictions and true value:

$$\text{Bias} = \bar{\hat{y}} - y = \frac{5.2 + 4.9 + 5.1}{3} - 5 = 5.067 - 5 = 0.067$$

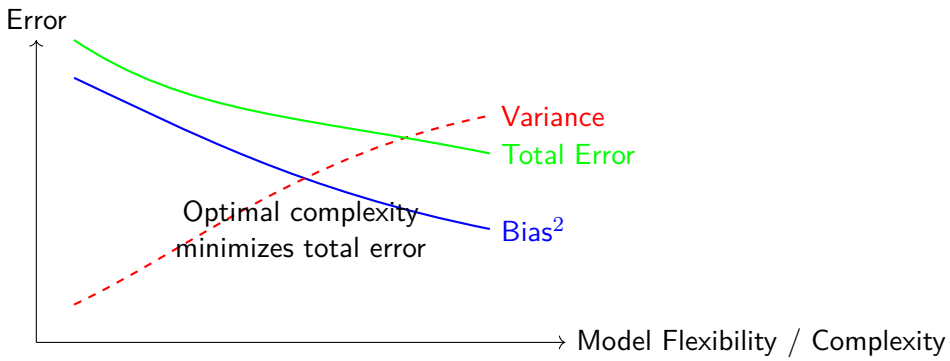
- **Variance:** Spread of predictions around their mean:

$$\text{Variance} = \frac{(5.2 - 5.067)^2 + (4.9 - 5.067)^2 + (5.1 - 5.067)^2}{3} \approx 0.0089$$

- **Total Error:** $\text{Bias}^2 + \text{Variance}$:

$$\text{Error} \approx 0.067^2 + 0.0089 \approx 0.0133$$

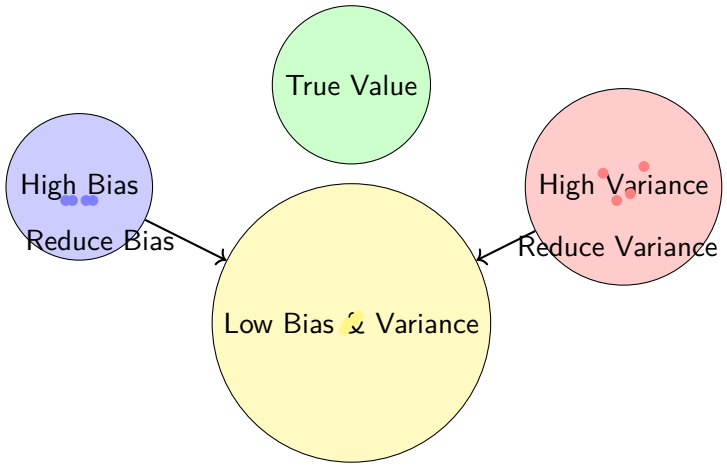
Bias-Variance Tradeoff



Practical Scenarios for Bias and Variance

1. A simple constant model predicting the average: High bias, low variance.
2. A model that outputs random predictions: Low bias on average, high variance.
3. Averaging several independent models: Reduces variance without increasing bias.
4. Overly simplistic formula for forecasting: High bias, may underfit data.
5. Highly sensitive formula reacting to every minor fluctuation: Low bias, high variance.

Visualization of Bias and Variance



1.7 Evaluation Metrics for Classification

Classification Metrics

Classification metrics evaluate the performance of supervised learning models by measuring how well predicted labels match actual labels.
They are derived from the **confusion matrix**, which summarizes predictions:

- **True Positive (TP)**: Correctly predicted positive class.
- **False Positive (FP)**: Incorrectly predicted positive class.
- **True Negative (TN)**: Correctly predicted negative class.

- **False Negative (FN)**: Incorrectly predicted negative class.

Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

This matrix forms the basis for computing all classification metrics.

Accuracy

Accuracy measures the overall correctness of the model:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Example: TP=40, TN=50, FP=5, FN=5

$$\text{Accuracy} = \frac{40 + 50}{40 + 50 + 5 + 5} = 0.9 = 90\%$$

Precision

Precision measures how many predicted positives are actually positive:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Example: TP=40, FP=5

$$\text{Precision} = \frac{40}{40 + 5} \approx 0.8889 = 88.89\%$$

High precision indicates few false positives.

Recall (Sensitivity)

Recall measures how many actual positives were correctly predicted:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Example: TP=40, FN=5

$$\text{Recall} = \frac{40}{40 + 5} \approx 0.8889 = 88.89\%$$

High recall indicates few false negatives.

F1-Score

F1-Score is the harmonic mean of Precision and Recall:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Example: Precision=0.8889, Recall=0.8889

$$\text{F1} = 2 \cdot \frac{0.8889 \cdot 0.8889}{0.8889 + 0.8889} = 0.8889 = 88.89\%$$

It balances Precision and Recall.

Specificity

Specificity measures how many actual negatives were correctly predicted:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Example: TN=50, FP=5

$$\text{Specificity} = \frac{50}{50 + 5} \approx 0.909 = 90.9\%$$

False Positive Rate (FPR)

FPR measures incorrect positive predictions:

$$\text{FPR} = \frac{FP}{FP + TN} = 1 - \text{Specificity}$$

Example: FP=5, TN=50

$$\text{FPR} = \frac{5}{50 + 5} \approx 0.0909 = 9.09\%$$

Matthews Correlation Coefficient (MCC)

MCC combines TP, TN, FP, FN into a single metric:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Example: TP=40, TN=50, FP=5, FN=5

$$\text{MCC} = \frac{40 \cdot 50 - 5 \cdot 5}{\sqrt{(40 + 5)(40 + 5)(50 + 5)(50 + 5)}} \approx 0.889$$

MCC ranges from -1 (worst) to 1 (perfect prediction).

Numerical Example of Classification Metrics

Suppose we have a dataset of 20 patients where a model predicts whether each patient has a disease (Positive) or not (Negative).
The predictions vs actual outcomes are summarized as:

	Predicted Positive	**Predicted Negative**
Actual Positive	6	2
Actual Negative	3	9

From this confusion matrix:

- True Positives (TP) $= 6$
- False Negatives (FN) $= 2$
- False Positives (FP) $= 3$
- True Negatives (TN) $= 9$

Now we calculate the common metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{6 + 9}{6 + 9 + 3 + 2} = \frac{15}{20} = 0.75 = 75\%$$

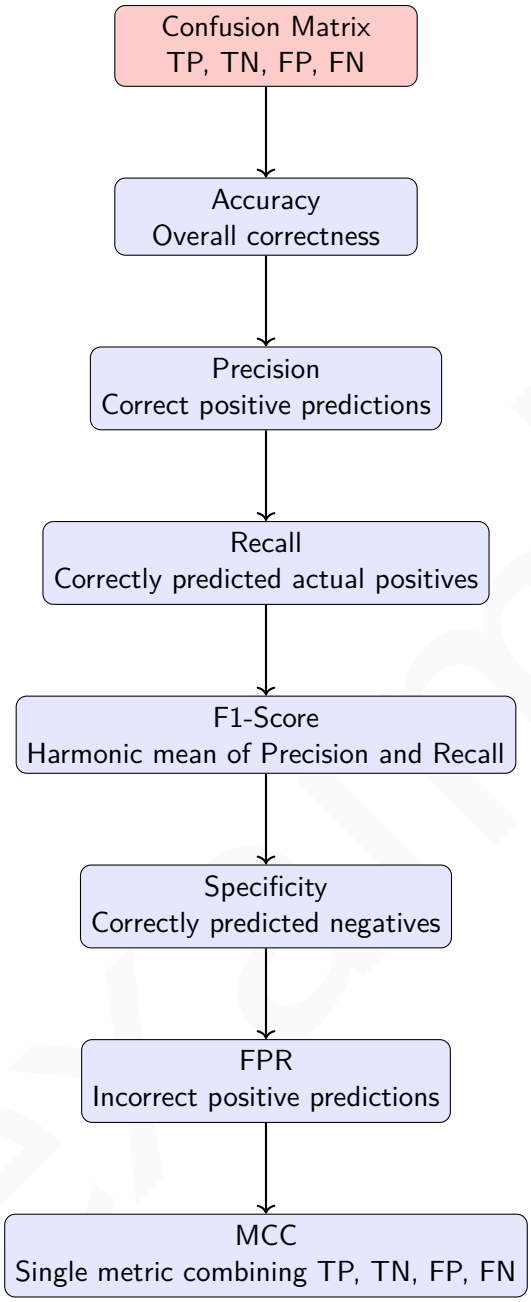
$$\text{Precision} = \frac{TP}{TP + FP} = \frac{6}{6 + 3} = \frac{6}{9} \approx 0.667 = 66.7\%$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} = \frac{6}{6 + 2} = \frac{6}{8} = 0.75 = 75\%$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \cdot \frac{0.667 \cdot 0.75}{0.667 + 0.75} \approx 0.706 = 70.6\%$$

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{9}{9 + 3} = \frac{9}{12} = 0.75 = 75\%$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} = \frac{3}{3 + 9} = \frac{3}{12} = 0.25 = 25\%$$



Example 5:

A multi-class classifier (4 classes) is evaluated on 400 samples. The confusion matrix is partially given:

Class	Pred A	Pred B	Pred C	Pred D
A	90	5	?	?
B	10	80	?	?
C	?	?	70	?
D	?	?	?	50

Fill missing entries assuming total samples per class: A=100, B=100, C=100, D=100. Compute per-class Precision, Recall, and macro-F1.

Solution:

We are given each row's total = 100. We'll fill missing entries by distributing each row's remaining misclassifications across the unknown predicted columns. (Any reasonable integer allocation that makes each row sum to 100 is acceptable — here we use a simple balanced distribution; explainable choices are fine in exam settings.)

Fill rows (one possible integer allocation):

- Row A: known 90 (Pred A) + 5 (Pred B) = 95, remainder 5 → distribute as Pred C = 3, Pred D = 2.
- Row B: known 10 (Pred A) + 80 (Pred B) = 90, remainder 10 → Pred C = 5, Pred D = 5.
- Row C: known 70 (Pred C), remainder 30 → distribute Pred A = 10, Pred B = 10, Pred D = 10.
- Row D: known 50 (Pred D), remainder 50 → distribute Pred A = 17, Pred B = 17, Pred C = 16 (sums to 50).

So the completed confusion matrix (Actual rows, Predicted columns):

Actual \ Pred	A	B	C	D	Row sum
A	90	5	3	2	100
B	10	80	5	5	100
C	10	10	70	10	100
D	17	17	16	50	100

Column sums (predicted counts):

$$\text{Pred A} = 90 + 10 + 10 + 17 = 127$$

$$\text{Pred B} = 5 + 80 + 10 + 17 = 112$$

$$\text{Pred C} = 3 + 5 + 70 + 16 = 94$$

$$\text{Pred D} = 2 + 5 + 10 + 50 = 67$$

Now compute per-class metrics (for class X: TP = diagonal entry, Precision = TP / (predicted X column sum), Recall = TP / 100 (actual class size)).

Class A:

$$\text{TP}_A = 90, \quad \text{Precision}_A = \frac{90}{127} \approx 0.7087\ (70.87\%), \quad \text{Recall}_A = \frac{90}{100} = 0.9\ (90\%).$$

$\text{F1}_A = 2 \cdot \frac{0.7087 \cdot 0.9}{0.7087 + 0.9} \approx 0.7930$ (79.30%).

Class B:

$$\text{TP}_B = 80, \quad \text{Precision}_B = \frac{80}{112} \approx 0.7143\ (71.43\%), \quad \text{Recall}_B = \frac{80}{100} = 0.8\ (80\%).$$

$\text{F1}_B \approx 0.7547$ (75.47%).

Class C:

$$\text{TP}_C = 70, \quad \text{Precision}_C = \frac{70}{94} \approx 0.7447\ (74.47\%), \quad \text{Recall}_C = \frac{70}{100} = 0.7\ (70\%).$$

$\text{F1}_C \approx 0.7216$ (72.16%).

Class D:

$$\text{TP}_D = 50, \quad \text{Precision}_D = \frac{50}{67} \approx 0.7463\ (74.63\%), \quad \text{Recall}_D = \frac{50}{100} = 0.5\ (50\%).$$

$\text{F1}_D \approx 0.5988$ (59.88%).

Macro-F1: average of per-class F1s:

$$\text{Macro-F1} = \frac{0.7930 + 0.7547 + 0.7216 + 0.5988}{4} \approx 0.7170\ (71.70\%).$$

Interpretation: - Recalls show class A is recognized very well (90%), class D lags (50%). - Precision values are roughly similar (71–75%), but class D's low recall penalizes its F1. - If you need to prioritize a particular class (say D), adjust model or collect more data for that class.

Example 6:

A machine learning model classifies emails as spam or ham. Out of 1000 emails: 300 spam, 700 ham. The model predicts 280 spam correctly, 40 ham as spam, 20 spam as ham, and remaining correctly as ham. Compute Accuracy, Precision, Recall, F1, Specificity, and MCC.

Solution:

First derive confusion-matrix entries:

- Actual positive (spam) = 300. Model predicts 280 correctly ⇒ TP = 280.
- Among spam, 20 predicted ham ⇒ FN = 20.
- Model mislabels 40 ham as spam ⇒ FP = 40.

- Remaining ham correctly predicted: TN = total ham - FP = 700 - 40 = 660.

Confusion matrix:

	Pred Spam	Pred Ham
Actual Spam	TP = 280	FN = 20
Actual Ham	FP = 40	TN = 660

Now compute metrics:

$$\text{Accuracy} = \frac{TP + TN}{\text{Total}} = \frac{280 + 660}{1000} = \frac{940}{1000} = 0.94 = 94\%.$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{280}{280 + 40} = \frac{280}{320} = 0.875 = 87.5\%.$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{280}{280 + 20} = \frac{280}{300} \approx 0.9333 = 93.33\%.$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \cdot \frac{0.875 \cdot 0.9333}{0.875 + 0.9333} \approx 0.9032 = 90.32\%.$$

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{660}{660 + 40} = \frac{660}{700} \approx 0.9429 = 94.29\%.$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} = \frac{280 \cdot 660 - 40 \cdot 20}{\sqrt{(320)(300)(700)(680)}} \approx 0.8608.$$

Interpretation: High accuracy and recall show the model catches most spam; precision ¡ recall means of all predicted spam, some fraction are false alarms. MCC ≈ 0.86 indicates strong overall classification quality accounting for all four cells.

Example 7:

A rare event detection model evaluates 50,000 events. Only 100 are positive. The model flags 120 events as positive, including 90 true positives. Compute FPR, Precision, Recall, and discuss the effect of class imbalance.

Solution:

From problem statement:

$$\text{Total positives} = 100, \quad TP = 90, \quad \text{model flagged positives} = 120 \Rightarrow FP = 120 - 90 = 30.$$

$$\text{Total negatives} = 50{,}000 - 100 = 49{,}900, \quad TN = 49{,}900 - FP = 49{,}900 - 30 = 49{,}870.$$

Compute metrics:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{90}{90 + 30} = \frac{90}{120} = 0.75 = 75\%.$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{90}{100} = 0.9 = 90\%.$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} = \frac{30}{30 + 49{,}870} = \frac{30}{49{,}900} \approx 0.0006012 \approx 0.06012\%.$$

Discussion (class imbalance):

- The recall (90%) and precision (75%) are fairly good. However, even a tiny FPR (0.06%) can produce many false positives when the dataset is huge. Here FP=30 is small in absolute terms — but if the negative population were even larger or the FPR slightly higher, false positives would grow quickly.
- In rare-event detection, Precision and FPR are critical: a low FPR is necessary to avoid an overwhelming number of false alarms. Also consider metrics like Precision-Recall curve or use specialized loss/thresholding to control FP.

Example 8:

A medical imaging system detects tumors in CT scans. Out of 500 scans, 120 contain tumors. The system correctly identifies 100 tumors and misclassifies 30 healthy scans. Compute Accuracy, Precision, Recall, F1, and suggest which metric should guide deployment.

Solution:

Given:

$$\text{Total scans} = 500, \quad \text{Positives (tumor)} = 120, \quad TP = 100, \quad FP = 30.$$

Compute remaining cells:

$$FN = 120 - 100 = 20,$$

$$\text{Negatives} = 500 - 120 = 380, \quad TN = 380 - FP = 380 - 30 = 350.$$

Confusion matrix:

	Pred Tumor	Pred Healthy
Actual Tumor	TP = 100	FN = 20
Actual Healthy	FP = 30	TN = 350

Metrics:

$$\text{Accuracy} = \frac{TP + TN}{500} = \frac{100 + 350}{500} = \frac{450}{500} = 0.9 = 90\%.$$

$$\text{Precision} = \frac{100}{100 + 30} = \frac{100}{130} \approx 0.7692 = 76.92\%.$$

$$\text{Recall (Sensitivity)} = \frac{100}{100 + 20} = \frac{100}{120} \approx 0.8333 = 83.33\%.$$

$$\text{F1} = 2 \cdot \frac{0.7692 \cdot 0.8333}{0.7692 + 0.8333} \approx 0.8000 = 80.00\%.$$

Which metric should guide deployment?

- In medical detection tasks, missing a true positive (i.e., a tumor) can have severe consequences. Thus **Recall (sensitivity)** is often the top priority — we want to minimize FN. An acceptable approach is to maximize recall while keeping precision at an operationally acceptable level (to avoid overwhelming radiologists with false positives).
- However, extremely low precision (many false positives) will create excessive follow-ups and costs; so balance is needed — use recall as priority with constraints on acceptable precision. Use ROC / Precision-Recall curves for threshold selection, and engage clinical stakeholders to set acceptable trade-offs.

Example 9:

A social media model detects offensive posts. Out of 10,000 posts, 200 are offensive. The model flags 180 correctly, but 400 normal posts as offensive. Compute confusion matrix, Precision, Recall, F1-score, and MCC.

Solution:

Given:

$$\text{Total} = 10{,}000, \quad \text{Positives (offensive)} = 200, \quad TP = 180, \quad FP = 400.$$

Then:

$$FN = 200 - 180 = 20.$$

Negatives:

$$\text{Negatives} = 10{,}000 - 200 = 9{,}800, \quad TN = 9{,}800 - 400 = 9{,}400.$$

Confusion matrix:

	Pred Offensive	Pred Normal
Actual Offensive	TP = 180	FN = 20
Actual Normal	FP = 400	TN = 9,400

Compute metrics:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{180}{180 + 400} = \frac{180}{580} \approx 0.3103 = 31.03\%.$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{180}{180 + 20} = \frac{180}{200} = 0.9 = 90\%.$$

$$\text{F1} = 2 \cdot \frac{0.3103 \cdot 0.9}{0.3103 + 0.9} \approx 0.4615 = 46.15\%.$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} = \frac{180 \cdot 9400 - 400 \cdot 20}{\sqrt{(580)(200)(9800)(9200)}} \approx 0.5146.$$

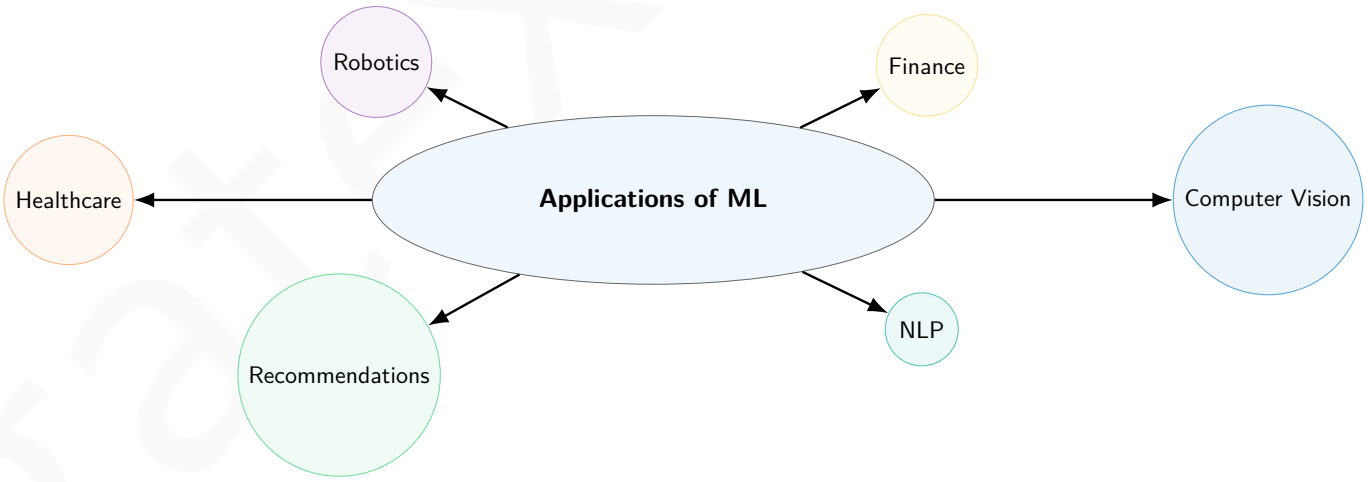
Interpretation:

- High recall (90%) — the model catches most offensive posts.
- Low precision (31%) — many false positives, so moderation teams will see many normal posts flagged.
- F1 around 46% shows overall balance is poor due to low precision. MCC ≈ 0.51 indicates moderate classification performance accounting for imbalance.
- For production, consider raising threshold, filtering, or a human-in-loop to reduce false positives; also consider cost of false positives vs false negatives for policy decisions.

1.8 Applications

Where ML is applied today

Machine Learning powers many modern services and systems: search, recommendations, voice assistants, medical imaging, fraud detection, personalized education, robotics, weather forecasting, and many more. The value comes from converting raw or structured data into predictions, insights, or actions that support human decisions or automate tasks.



Selected use-cases

- **Computer Vision:** Detect objects in images to help autonomous vehicles or medical image analysis to highlight abnormal tissue.
- **NLP:** Translate languages, summarize documents, or extract information from text.
- **Recommendations:** Suggest relevant products, songs, or videos by learning user preferences.

- **Healthcare:** Predict patient risk and assist diagnostics (requires domain validation).
- **Finance:** Detect fraudulent transactions or forecast market signals (with caution).

1.9 Problems

Problem 1 *If Precision* $= \dfrac{TP}{TP+FP}$ *and Recall* $= \dfrac{TP}{TP+FN}$*, then F1-score is maximum when:*

(A) $\text{Precision} = \text{Recall}$

(B) $\text{Precision} > \text{Recall}$

(C) $\text{Recall} > \text{Precision}$

(D) $FP = FN$

Problem 2 *If False Positive Rate (FPR) decreases while True Positive Rate (TPR) remains constant, which metric will definitely increase?*

(A) Accuracy

(B) Recall

(C) Precision

(D) None of the above

Problem 3 *Given two classifiers A and B:*

$$A\text{: } (TP, FP, FN, TN) = (x, y, z, w), \quad B\text{: } (TP, FP, FN, TN) = (2x, 2y, 2z, 2w)$$

Which statement is TRUE?

(A) Both have identical Precision and Recall

(B) Classifier B has higher Accuracy

(C) F1-score of B is double that of A

(D) Precision of B is half that of A

Problem 4 *A model's F1-score is equal to its Precision. Then which of the following relations is TRUE?*

(A) $\text{Precision} = \text{Recall} = 1$ *always*

(B) $\text{Recall} = 0.5$

(C) $\text{Recall} = \text{Precision}$

(D) $\text{Recall} = 0$

Problem 5 *If a classifier predicts all instances as positive, then:*

(A) *Recall* $= 1$

(B) *Precision* $= 1$

(C) *Accuracy* $= 1$

(D) *F1-score* $= 0$

Problem 6 *A model's Precision is higher than Recall. Which of the following could explain this?*

(A) *Many False Negatives*

(B) *Many False Positives*

(C) *Few True Negatives*

(D) *High FPR*

Problem 7 *If a confusion matrix is such that* $TN = 0$*, which of the following statements must be true?*

(A) *Specificity* $= 0$

(B) *Accuracy* $=$ *Recall*

(C) *Precision* $=$ *F1-score*

(D) *FPR* $= 1$

Problem 8 *In a binary classification task, doubling all entries in the confusion matrix* (TP, FP, TN, FN) *will:*

(A) *Change Accuracy*

(B) *Change Precision*

(C) *Change Recall*

(D) *Change none of the above metrics*

Problem 9 *A medical test is applied to 1000 patients. The prevalence of a disease is 10%. The test correctly identifies 85% of sick patients and wrongly classifies 5% of healthy patients as sick. Compute Accuracy, Precision, Recall, and F1-score.*

Problem 10 *A rare disease affects 1 in 500 people. A screening test has 99% sensitivity and 98% specificity. If a person tests positive, calculate the probability they actually have the disease (Precision). Also compute expected False Positive Rate for a population of 10,000.*

Problem 11 *In an autonomous driving system, out of 1000 frames, 300 contain pedestrians. The detector flags 280 correctly but has 50 false alarms. Compute Precision and Recall.*

1.10 Try it Yourself

If $F1 = \dfrac{2PR}{P+R}$ and $F1 = 0.8$ when $P = 0.9$, the Recall must be:

(A) 0.7

(B) 0.8

(C) 0.9

(D) 1.0

Exercise 1 *If Accuracy* $= \dfrac{TP+TN}{TP+TN+FP+FN}$ *is constant, but Precision increases, then which of the following must hold?*

(A) FP *decreases while* TP *constant*

(B) TP *increases while* FP *constant*

(C) TN *decreases while* FN *increases*

(D) FN *decreases while* TN *increases*

Exercise 2 *A hospital screening test is applied to 500 patients. Disease prevalence is 15%. The test correctly identifies 60% of sick patients and misdiagnoses 5% of healthy patients. Find Accuracy, Precision, Recall.*

Exercise 3 *In a quality control process, 500 products are inspected. 20% are defective. The classifier identifies 90% of defective products correctly but mislabels 8% of good products as defective. Calculate the confusion matrix and F1-score.*

Exercise 4 *A spam filter scans 200 emails. 40% are actually spam. It correctly detects 95% of spam and incorrectly flags 10% of non-spam as spam. Compute Precision and Recall.*

Exercise 5 *A model predicts whether a student passes an exam. Out of 150 students, 60 pass. The model correctly predicts 50 passing students and wrongly predicts 10 failing students as passing. Find Accuracy and F1-score.*

Exercise 6 *In a city, 5% of people are infected with a virus. A test identifies 98% of infected people but 2% of healthy people are misdiagnosed. Calculate Precision and Specificity for this test.*

Exercise 7 *A classifier predicts three classes (A, B, C). In a dataset of 900 items: 400 A, 300 B, 200 C. The model correctly predicts 350 A, 200 B, 150 C, and misclassifies remaining items. Compute per-class Precision, Recall, and macro F1-score.*

Exercise 8 *An imbalanced dataset has 950 negative samples and 50 positive samples. A model predicts 40 positive correctly but also predicts 60 false positives. Compute Accuracy, Precision, Recall, F1-score, and discuss why Accuracy is misleading.*

Exercise 9 *In a medical test, the prevalence of a condition is 5%. The test has 90% sensitivity and 85% specificity. Compute the Matthews Correlation Coefficient (MCC) for a population of 2000 patients.*

Exercise 10 *A spam filter has to balance Precision and Recall. Out of 500 emails, 100 are spam. The system detects 90 spam correctly but mislabels 50 ham emails. Find F1-score and discuss the trade-off if you increase threshold to reduce false positives.*

Exercise 11 *A fraud detection system sees 10000 transactions with 2% fraudulent. The model identifies 150 fraudulent and 50 legitimate transactions incorrectly. Compute Accuracy, Precision, Recall, F1, and interpret in the context of low prevalence.*

1.11 YouTube Links and QR Codes

Lecture	Details	YouTube Link	QR Code
1	Overview of Machine Learning: Types, Workflow & Data Explained	https://youtu.be/d5mAnQUZe50	
2	Issues in ML: Bias, Underfitting & Overfitting Explained	https://youtu.be/APedsqZeZRI	
3	Confusion Matrix Explained: Accuracy, Precision, Recall & F1 Score	https://youtu.be/2jCQxN5cP8Y	
4	Problem Solving – Confusion Matrix & Metrics (Solutions 1 - 11)	https://youtu.be/ouXrZEQPM8Q	

Chapter 2

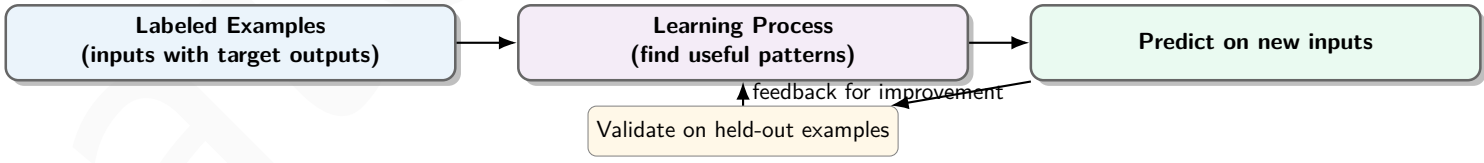
Regression Models

What supervised learning does

Supervised learning uses examples where the desired outcome is known to teach a model to predict or output the correct result for new inputs. The key idea is generalization: the model captures useful regularities from examples that hold across new, similar cases.

How training feels in plain language

Think of supervised learning as training an assistant: you show many examples of the correct answer for different situations. The assistant learns patterns in those examples so they can handle future situations they haven't seen before. During training you evaluate them on practice questions (validation) and adjust their training strategy until their performance is satisfactory.



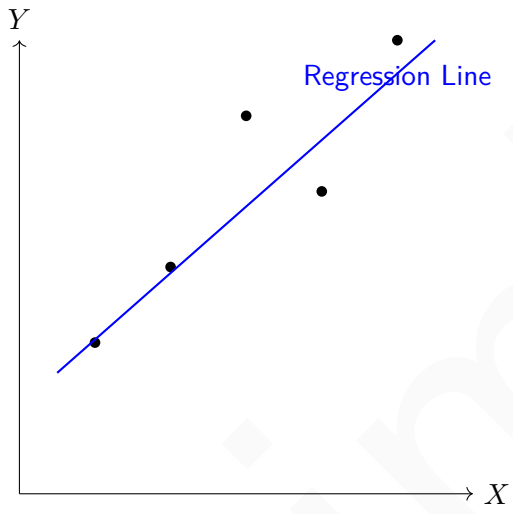
Practical notes

- Always start with simple baselines to set realistic expectations.
- Use appropriate evaluation metrics (accuracy, precision, recall, F1, etc.) depending on the task and consequences.
- Beware of overfitting: good training performance does not guarantee real-world success.
- Keep interpretability and fairness considerations in mind, especially for decisions affecting people.

2.1 Simple Linear Regression

2.1.1 Model

Simple Linear Regression predicts a quantitative response Y from a single predictor X assuming a linear relationship. The model represents a straight line, and the predicted values are points on this line.



Simple Linear Regression Model

Simple Linear Regression predicts a quantitative response Y from a single predictor X assuming a linear relationship:

$$Y \approx \beta_0 + \beta_1 X$$

where

- β_0: Intercept (expected Y when $X = 0$)
- β_1: Slope (change in Y per unit change in X)

Predicted value for a given $X = x$:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Here $\hat{\beta}_0$ and $\hat{\beta}_1$ are estimates of the true coefficients β_0 and β_1.

To find the best line, we minimize the vertical distances (residuals) from each point to the line. Squaring these distances and summing gives the Residual Sum of Squares (RSS).

2.1.2 Estimating Coefficients: Ordinary Least Squares (OLS)

Estimating Coefficients: Least Squares (OLS)

We estimate β_0 and β_1 by minimizing the Residual Sum of Squares (RSS):

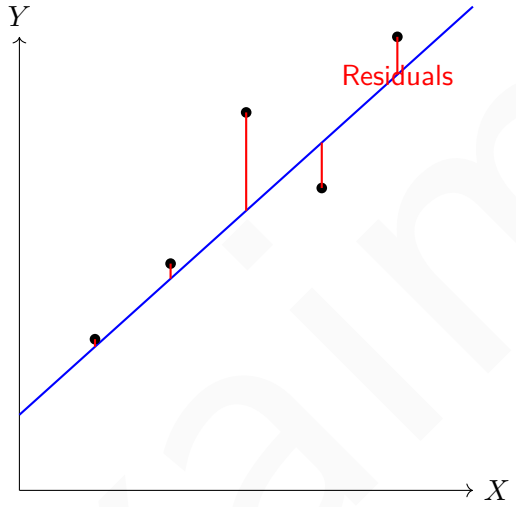
$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

The least squares estimates are:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where $\bar{x}$ and $\bar{y}$ are sample means of X and Y.

2.1.3 Residuals and Residual Standard Error



Residuals and Residual Standard Error

Residual for observation i:

$$e_i = y_i - \hat{y}_i$$

Residual Sum of Squares:

$$RSS = \sum_{i=1}^{n} e_i^2$$

Residual Standard Error (RSE) measures the average deviation of observed Y from the regression line:

$$RSE = \sqrt{\frac{RSS}{n-2}}$$

It indicates model's lack of fit in the units of Y.

2.1.4 R-Squared (R^2)

R-squared measures how much of the variance in Y is explained by X.

Before understanding R^2, we need to know what TSS and RSS are:

- TSS (Total Sum of Squares): Measures the total variability of Y around its mean $\bar{Y}$.

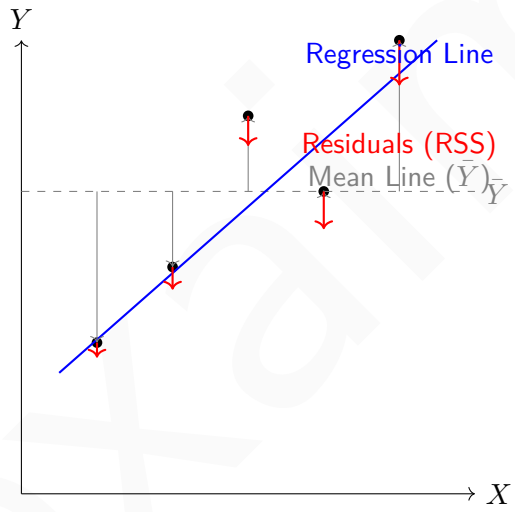
$$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

  Think of it as how "spread out" the data points are from the average.
- RSS (Residual Sum of Squares): Measures the variability of Y that is **not explained by the regression line**, i.e., the sum of squared residuals:

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

  This shows how far the points are from the fitted line.
- R^2 then tells us the proportion of total variability explained by the model:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$



R-Squared (R^2)

R^2 measures the proportion of variance in Y explained by X:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}, \quad TSS = \sum(y_i - \bar{y})^2$$

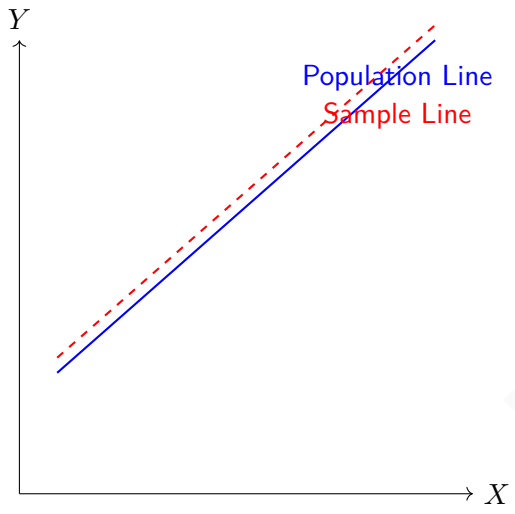
- $R^2 \approx 1$: model explains most variability in Y
- $R^2 \approx 0$: model explains little variability

For simple linear regression, R^2 is the square of correlation:

$$R^2 = \text{Cor}(X, Y)^2$$

2.1.5 Population vs Sample Regression

The population regression line represents the true relationship with random error. The sample line estimates it from data.



Population vs Sample Regression

True model with random error:

$$Y = \beta_0 + \beta_1 X + \varepsilon, \quad \mathbb{E}[\varepsilon] = 0$$

- Population regression line: $\beta_0 + \beta_1 X$ (unknown)
- Sample least squares line: $\hat{\beta}_0 + \hat{\beta}_1 X$ (estimated from data)

Different samples yield slightly different $\hat{\beta}_0, \hat{\beta}_1$, but all estimate the same population line.

2.1.6 Standard Errors and Confidence Intervals

Confidence intervals indicate the likely range of true coefficients.

Standard Errors and Confidence Intervals (in terms of $Z_{\alpha/2}$)

Standard Errors of Coefficients:

$$SE(\hat{\beta}_1) = \sqrt{\frac{\sigma^2}{\sum(x_i - \bar{x})^2}}, \quad SE(\hat{\beta}_0) = \sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2} \right)}$$

Confidence Intervals at $(1-\alpha)$ level:

$$\hat{\beta}_1 \pm Z_{\alpha/2} \cdot SE(\hat{\beta}_1), \quad \hat{\beta}_0 \pm Z_{\alpha/2} \cdot SE(\hat{\beta}_0)$$

where:

- $Z_{\alpha/2}$ is the critical value from the standard normal distribution.
- For a 95% confidence level, $\alpha = 0.05 \Rightarrow Z_{\alpha/2} = 1.96 \, (\approx 2)$.

$$P\left(\hat{\beta}_j - Z_{\alpha/2} SE(\hat{\beta}_j) \leq \beta_j \leq \hat{\beta}_j + Z_{\alpha/2} SE(\hat{\beta}_j)\right) = 1 - \alpha, \quad j = 0, 1$$

Example 10:

A company collects data on advertising spend (X, in \$1000) and sales (Y, in \$1000) for 6 markets:

$$(X, Y) = (1, 2), (2, 3), (3, 5), (4, 4), (5, 6), (6, 7)$$

Answer the following:

1. Compute the least squares estimates of the regression coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$.
2. Predict sales for $X = 7$.
3. Compute the residuals e_i and Residual Sum of Squares (RSS).
4. Compute the Residual Standard Error (RSE).
5. Compute standard errors of $\hat{\beta}_0$ and $\hat{\beta}_1$ assuming $\sigma^2 = 0.49$.
6. Compute the 95% confidence intervals for $\hat{\beta}_0$ and $\hat{\beta}_1$.
7. Compute R^2 and interpret it.
8. Explain what large positive and negative residuals indicate.

Solution:

1. **Least squares estimates:**

$$\bar{X} = \frac{1+2+3+4+5+6}{6} = 3.5, \quad \bar{Y} = \frac{2+3+5+4+6+7}{6} = 4.5$$

$$\hat{\beta}_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = \frac{15.75}{17.5} = 0.9$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} = 4.5 - 0.9 \cdot 3.5 = 1.35$$

2. **Prediction for $X = 7$:**

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X = 1.35 + 0.9 \cdot 7 = 7.65$$

3. **Residuals and RSS:**

$$\hat{Y}_i = 1.35 + 0.9 X_i = [2.25, 3.15, 4.05, 4.95, 5.85, 6.75]$$

$$e_i = Y_i - \hat{Y}_i = [-0.25, -0.15, 0.95, -0.95, 0.15, 0.25]$$

$$RSS = \sum e_i^2 = 0.0625 + 0.0225 + 0.9025 + 0.9025 + 0.0225 + 0.0625 = 1.975$$

4. **Residual Standard Error (RSE):**

$$RSE = \sqrt{\frac{RSS}{n-2}} = \sqrt{\frac{1.975}{6-2}} = \sqrt{0.49375} \approx 0.703$$

5. **Standard errors of coefficients:**

$$SE(\hat{\beta}_1) = \sqrt{\frac{\sigma^2}{\sum(x_i - \bar{x})^2}} = \sqrt{\frac{0.49}{17.5}} \approx 0.167$$

$$SE(\hat{\beta}_0) = \sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{X}^2}{\sum(x_i - \bar{x})^2}\right)} = \sqrt{0.49 \left(\frac{1}{6} + \frac{3.5^2}{17.5}\right)} \approx 0.44$$

6. **95% Confidence intervals:**

$$\hat{\beta}_1 \pm 2SE(\hat{\beta}_1) = 0.9 \pm 2 \cdot 0.167 = [0.566, 1.234]$$

$$\hat{\beta}_0 \pm 2SE(\hat{\beta}_0) = 1.35 \pm 2 \cdot 0.44 = [0.47, 2.23]$$

7. **R-Squared (R^2):**

$$TSS = \sum(Y_i - \bar{Y})^2 = 17.5, \quad RSS = 1.975$$

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{1.975}{17.5} \approx 0.887$$

Interpretation: 88.7% of variability in sales is explained by advertising.

8. **Interpretation of residuals:**
   - $e_i > 0 \rightarrow$ observed Y_i is above regression line $\rightarrow$ model underestimates
   - $e_i < 0 \rightarrow$ observed Y_i is below regression line $\rightarrow$ model overestimates
   - Large magnitude $\rightarrow$ poor prediction or outlier

Example 11:

A startup collects data on number of hours spent on online ads (X, in hours) and number of website signups (Y) over 7 campaigns:

$$(X, Y) = (1, 3), (2, 5), (3, 4), (4, 6), (5, 7), (6, 8), (7, 7)$$

Answer the following:

1. Compute the least squares estimates of the regression coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$.
2. Predict signups for $X = 8$ hours.
3. Compute the residuals e_i and Residual Sum of Squares (RSS).
4. Compute the Residual Standard Error (RSE).
5. Compute standard errors of $\hat{\beta}_0$ and $\hat{\beta}_1$ assuming $\sigma^2 = 0.64$.

6. Compute the 95% confidence intervals for $\hat{\beta}_0$ and $\hat{\beta}_1$.
7. Compute R^2 and interpret it.
8. Explain the interpretation of large positive and negative residuals in this case.

Solution:

1. **Least squares estimates:**

$$\bar{X} = \frac{1+2+3+4+5+6+7}{7} = 4, \quad \bar{Y} = \frac{3+5+4+6+7+8+7}{7} = 5.714$$

$$\hat{\beta}_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = \frac{(1-4)(3-5.714) + \cdots + (7-4)(7-5.714)}{(1-4)^2 + \cdots + (7-4)^2} = 0.786$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} = 5.714 - 0.786 \cdot 4 \approx 2.570$$

2. **Prediction for $X = 8$:**

$$\hat{Y} = 2.570 + 0.786 \cdot 8 = 9.838$$

3. **Residuals and RSS:**

$$\hat{Y}_i = 2.570 + 0.786 X_i = [3.356, 4.142, 4.928, 5.714, 6.5, 7.286, 8.072]$$

$$e_i = Y_i - \hat{Y}_i = [-0.356, 0.858, -0.928, 0.286, 0.5, 0.714, -1.072]$$

$$RSS = \sum e_i^2 = 0.127 + 0.736 + 0.861 + 0.082 + 0.25 + 0.51 + 1.149 \approx 3.715$$

4. **Residual Standard Error (RSE):**

$$RSE = \sqrt{\frac{RSS}{n-2}} = \sqrt{\frac{3.715}{7-2}} = \sqrt{0.743} \approx 0.862$$

5. **Standard errors of coefficients:**

$$SE(\hat{\beta}_1) = \sqrt{\frac{\sigma^2}{\sum(x_i - \bar{x})^2}} = \sqrt{\frac{0.64}{28}} \approx 0.151$$

$$SE(\hat{\beta}_0) = \sqrt{0.64\left(\frac{1}{7} + \frac{4^2}{28}\right)} = \sqrt{0.64\,(0.1429 + 0.5714)} = \sqrt{0.457} \approx 0.676$$

6. **95% Confidence intervals:**

$$\hat{\beta}_1 \pm 2SE(\hat{\beta}_1) = 0.786 \pm 0.302 = [0.484, 1.088]$$

$$\hat{\beta}_0 \pm 2SE(\hat{\beta}_0) = 2.570 \pm 1.352 = [1.218, 3.922]$$

7. **R-Squared (R^2):**

$$TSS = \sum(Y_i - \bar{Y})^2 = 0.735 + 0.020 + 2.938 + 0.082 + 1.653 + 5.204 + 1.653 \approx 12.285$$

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{3.715}{12.285} \approx 0.697$$

Interpretation: Approximately 69.7% of variability in signups is explained by online ad hours.

8. **Interpretation of residuals:**
   - Positive residual $e_i > 0 \rightarrow$ observed Y_i above regression line $\rightarrow$ model underestimates
   - Negative residual $e_i < 0 \rightarrow$ observed Y_i below regression line $\rightarrow$ model overestimates
   - Large residuals indicate potential outliers or poor fit for some campaigns

2.2 Multiple Linear Regression

Multiple Linear Regression (MLR) models the relationship between a quantitative response Y and multiple predictors $X_1, X_2, \ldots, X_p$:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon, \quad \mathbb{E}[\varepsilon] = 0$$

Here, β_0 is the intercept, β_j represents the effect of predictor X_j on Y, and ε is the random error. The goal is to estimate the coefficients to minimize the difference between observed and predicted values.

Matrix Formulation

We can write MLR compactly in matrix form:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Reasoning: The column of ones in $\mathbf{X}$ corresponds to the intercept. Using matrix notation simplifies calculation of predictions, residuals, and coefficient estimates.

Example 12:

Consider 3 houses with Size (X_1) and Age (X_2) and Price (Y):

X_1	X_2	Y
1200	5	240
1500	10	300
1700	15	325

Matrix form:

$$\mathbf{X} = \begin{bmatrix} 1 & 1200 & 5 \\ 1 & 1500 & 10 \\ 1 & 1700 & 15 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 240 \\ 300 \\ 325 \end{bmatrix}$$

Example 13:

A fourth house with Size=2000, Age=20:

$$\mathbf{x}_{new} = \begin{bmatrix} 1 & 2000 & 20 \end{bmatrix}$$

OLS Estimation: Multiple Linear Regression

In multiple linear regression, the response Y depends on multiple predictors $X_1, X_2, \ldots, X_p$:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

For n observations, in matrix form:

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & X_{11} & X_{12} & \ldots & X_{1p} \\ 1 & X_{21} & X_{22} & \ldots & X_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \ldots & X_{np} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

OLS Estimate: The coefficients are chosen to minimize the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

Closed-form solution:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

Interpretation: Each $\hat{\beta}_j$ represents the expected change in Y per unit change in X_j, holding all other predictors constant.

OLS Estimation and Normal Equations

OLS estimates minimize the Residual Sum of Squares:

$$RSS = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

Set derivative w.r.t $\boldsymbol{\beta}$ to zero:

$$\frac{\partial RSS}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

$$\Rightarrow \mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{Y}, \quad \hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

Reasoning: The normal equations provide a closed-form solution for OLS coefficients. $\mathbf{X}^T\mathbf{X}$ must be invertible; otherwise, predictors are linearly dependent.

Example 14:

Using previous 3-house data:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 50 \\ 0.15 \\ 5 \end{bmatrix}$$

Regression line: $\hat{Y} = 50 + 0.15X_1 + 5X_2$

Goodness of Fit: R^2 and Adjusted R^2 (Matrix–Vector Form)

Symbol Definitions:

Symbol	Type	Meaning
$Y \in \mathbb{R}^{n\times 1}$	Vector	Observed dependent variable values
$X \in \mathbb{R}^{n\times(p+1)}$	Matrix	Design matrix (first column for intercept)
$\hat{\beta} = (X^\top X)^{-1}X^\top Y$	Vector	Estimated regression coefficients
$X\hat{\beta}$	Vector	Predicted (fitted) response values $\hat{Y}$
$V \in \mathbb{R}^{n\times 1}$	Vector	Vector of ones
$\bar{y} = \frac{V^\top Y}{n}$	Scalar	Mean of observed responses
n	Scalar	Number of observations
p	Scalar	Number of predictors (excluding intercept)

Formulas:

$$TSS = (Y - \bar{y}V)^\top (Y - \bar{y}V)$$

$$RSS = (Y - X\hat{\beta})^\top (Y - X\hat{\beta})$$

$$R^2 = 1 - \frac{(Y - X\hat{\beta})^\top (Y - X\hat{\beta})}{(Y - \bar{y}V)^\top (Y - \bar{y}V)}$$

$$\bar{R}^2 = 1 - \frac{\left[(Y - X\hat{\beta})^\top (Y - X\hat{\beta})\right]/(n-p-1)}{\left[(Y - \bar{y}V)^\top (Y - \bar{y}V)\right]/(n-1)}$$

Residual Sum of Squares (RSS):
RSS tells us how far the model's predictions are from the actual data points.
It adds up all the squared differences between what we predicted and what really happened.
A smaller RSS means our model fits the data better.

Why RSS Always Goes Down:
When we add more predictors (extra variables) to the model, it gets more freedom to adjust itself.
Each new variable gives the model another way to slightly bend or twist to fit the data.
Even if that new variable is just random noise, the model can still use it to reduce some errors.
So, the total error (RSS) can only go down or stay the same — it never goes up.

The Problem:
Because RSS keeps getting smaller when we add more variables, the regular R^2 value (which depends on RSS) always goes up.
This can make a model *look better* on paper even if it is actually overfitting — learning noise instead of real patterns.

Adjusted R^2:
Adjusted R^2 fixes this problem. It rewards models that truly explain the data well but penalizes those that just keep adding useless variables.
If a new variable genuinely improves predictions, Adjusted R^2 will increase.
If it only adds noise, Adjusted R^2 will go down.
In short, Adjusted R^2 gives a fairer picture of how good your model really is.

Example 15:

For 3-house data:

$$TSS = (240 - 295)^2 + (300 - 295)^2 + (325 - 295)^2 = 2450, \quad RSS = 0$$

$$R^2 = 1 - \frac{0}{2450} = 1$$

Perfect fit.

Example 16:

A company wants to predict the sales (Y in \$1000) of a product based on two factors: Advertising Spend (X_1 in \$1000) and Number of Stores (X_2). The data for 4 regions is:

X_1	X_2	Y
2	10	50
3	12	65
5	15	80
4	11	70

Step 1: Matrix Formulation

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 10 \\ 1 & 3 & 12 \\ 1 & 5 & 15 \\ 1 & 4 & 11 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 50 \\ 65 \\ 80 \\ 70 \end{bmatrix}$$

Step 2: Compute OLS Estimates

Normal equation: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$

Compute $\mathbf{X}^T\mathbf{X}$:

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 4 & 14 & 48 \\ 14 & 54 & 170 \\ 48 & 170 & 650 \end{bmatrix}$$

Compute $\mathbf{X}^T\mathbf{Y}$:

$$\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 265 \\ 945 \\ 3485 \end{bmatrix}$$

Solve $(\mathbf{X}^T\mathbf{X})\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{Y}$ to get:

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} 20 \\ 5 \\ 2 \end{bmatrix}$$

Step 3: Regression Equation

$$\hat{Y} = 20 + 5X_1 + 2X_2$$

Step 4: Predictions
For $X_1 = 4$, $X_2 = 13$:

$$\hat{Y} = 20 + 5 \cdot 4 + 2 \cdot 13 = 20 + 20 + 26 = 66$$

Step 5: Residuals and R^2
Observed $Y = 70$, residual $e = 70 - 66 = 4$
Compute $TSS = \sum(Y_i - \bar{Y})^2$, $RSS = \sum(Y_i - \hat{Y}_i)^2$ and

$$R^2 = 1 - \frac{RSS}{TSS} \approx 0.95$$

Excellent model fit.

2.3 Regularized Regression

2.3.1 Need for Regularization

In ordinary least squares (OLS) regression, the coefficients $\boldsymbol{\beta}$ are estimated by minimizing the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

However, when predictors are highly correlated (**multicollinearity**) or when the number of predictors p is close to or exceeds the number of observations n, the matrix $\mathbf{X}^T\mathbf{X}$ becomes nearly singular (non-invertible), making OLS estimates unstable and highly sensitive to small changes in data.

To overcome this instability, we introduce **regularization**, which penalizes large coefficient values by adding a constraint to the loss function. Regularization discourages overly complex models that overfit the training data and improves generalization on unseen data.

Motivation for Regularization

Regularization adds a penalty term to the objective function that discourages large coefficients:

$$\text{Minimize: } \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\, P(\boldsymbol{\beta})$$

where $P(\boldsymbol{\beta})$ is a penalty function and $\lambda \geq 0$ is the **regularization parameter** controlling the strength of the penalty.

- $\lambda = 0$: reduces to ordinary least squares (no penalty)
- Large λ: coefficients shrink toward zero (simpler model)

2.3.2 Ridge Regression (L2 Regularization)

Ridge Regression Objective

Ridge regression (also called L2 regularization) adds the squared magnitude of the coefficients as a penalty term:

$$\text{Minimize: } J(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

or equivalently in matrix form:

$$J(\boldsymbol{\beta}) = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}$$

Here, λ controls how much we shrink the coefficients. Larger λ values lead to greater shrinkage (smaller coefficients).

Taking derivative with respect to $\boldsymbol{\beta}$ and setting to zero:

$$\frac{\partial J}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = 0$$

$$\Rightarrow (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{Y}$$

$$\therefore \hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$$

Interpretation of Ridge Estimator

- $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})$ is always invertible for $\lambda > 0$, ensuring numerical stability.
- Ridge regression shrinks coefficients towards zero but never exactly to zero.
- It reduces variance at the cost of introducing slight bias — a bias–variance trade-off.

Example 17:

A dataset has predictors X_1, X_2 and response Y:

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{bmatrix}, \quad Y = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

Compute the ridge regression coefficients for $\lambda = 1$.

Solution:

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 14 & 20 \\ 20 & 29 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 38 \\ 55 \end{bmatrix}$$

$$\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} = \begin{bmatrix} 15 & 20 \\ 20 & 30 \end{bmatrix}$$

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} = \frac{1}{(15)(30) - (20)(20)} \begin{bmatrix} 30 & -20 \\ -20 & 15 \end{bmatrix} = \frac{1}{50} \begin{bmatrix} 30 & -20 \\ -20 & 15 \end{bmatrix}$$

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \frac{1}{50} \begin{bmatrix} 30 & -20 \\ -20 & 15 \end{bmatrix} \begin{bmatrix} 38 \\ 55 \end{bmatrix} = \frac{1}{50} \begin{bmatrix} (30)(38) - (20)(55) \\ (-20)(38) + (15)(55) \end{bmatrix} = \frac{1}{50} \begin{bmatrix} (1140 - 1100) \\ (-760 + 825) \end{bmatrix} = \frac{1}{50} \begin{bmatrix} 40 \\ 65 \end{bmatrix}$$

$$\therefore \hat{\boldsymbol{\beta}}_{\text{ridge}} = [0.8, 1.3]^T$$

—

Example 18:

To study effect of regularization, consider $\lambda = 0$ (OLS) and $\lambda = 5$ (Ridge) using the same dataset as before.

Solution:

For $\lambda = 0$ (OLS):

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix}$$

For $\lambda = 5$:

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + 5\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 0.65 \\ 1.12 \end{bmatrix}$$

As λ increases, coefficients shrink toward zero, reducing variance and model complexity.

2.3.3 Bias–Variance Trade-off in Ridge Regression

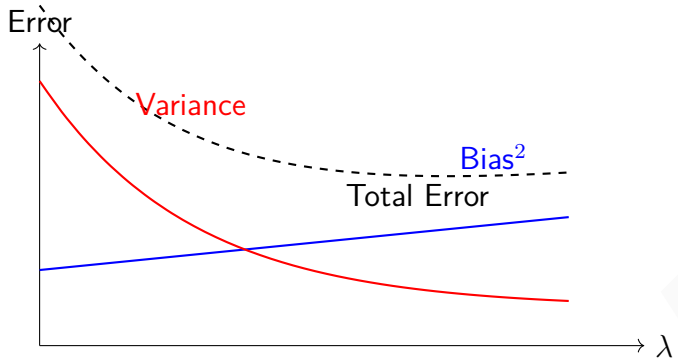
The total expected error in a model can be decomposed as:

$$E[(Y - \hat{f}(X))^2] = \text{Bias}^2[\hat{f}(X)] + \text{Var}[\hat{f}(X)] + \sigma^2$$

Bias–Variance Decomposition

- Bias: Difference between the true model and the expected prediction.
- Variance: How much the model predictions vary across different training sets.
- Ridge regression reduces variance by shrinking coefficients, but introduces bias since coefficients are pulled toward zero.

The optimal λ minimizes the total error (bias^2 + variance).



Bias–Variance trade-off as a function of λ.

2.3.4 Comparison with OLS Regression

Ridge vs OLS Regression

Property	OLS Regression	Ridge Regression
Objective Function	$\min RSS = \sum(y_i - \hat{y}_i)^2$	$\min RSS + \lambda \sum \beta_j^2$
Penalty Term	None	L2 penalty ($\sum \beta_j^2$)
Solution	$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$	$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$
Handling Multicollinearity	Unstable if $\mathbf{X}^T\mathbf{X}$ is singular	Always stable ($\lambda > 0$ ensures invertibility)
Effect on Coefficients	Can be large (unstable)	Shrunk towards zero
Interpretation	Unbiased, high variance	Slight bias, lower variance

2.4 Problems

Problem 12 *A simple linear regression model is fitted to a dataset. Which of the following statements is always true?*

A. *The regression line passes through* $(\bar{X}, \bar{Y})$.

B. R^2 *can be negative.*

C. *Residuals always sum to 1.*

D. *Standard error of* $\hat{\beta}_1$ *decreases with increasing sample size.*

Problem 13 *Consider a dataset with* $X = [1, 2, 3, 4]$ *and* $Y = [2, 3, 5, 4]$. *The least squares (OLS) estimate of slope* $\hat{\beta}_1$ *is:*

A. *0.5*

B. *0.7*

C. *0.8*

D. *1.0*

Problem 14 *In linear regression, increasing* X *variance while keeping* σ^2 *constant:*

A. *Increases* $SE(\hat{\beta}_1)$

B. *Decreases* $SE(\hat{\beta}_1)$

C. *Increases* $SE(\hat{\beta}_0)$

D. *Has no effect on* $SE(\hat{\beta}_1)$

Problem 15 *If* $R^2 = 0.95$ *in a simple linear regression, which is correct?*

A. *Model explains 95% of variability in* Y

B. *Residuals are all zero*

C. *Prediction interval for new* Y *will have zero width*

D. *Model always has high slope*

Problem 16 *Which of the following statements about residuals in simple linear regression are correct? (Select all that apply)*

A. *Sum of residuals is zero.*

B. *Residuals are independent of* X.

C. *Residual variance equals* σ^2 *if model is correctly specified.*

D. *Residuals can be used to detect non-linearity.*

Problem 17 *Regarding standard errors of regression coefficients:*

A. $SE(\hat{\beta}_1)$ *decreases if sample size increases.*

B. $SE(\hat{\beta}_0)$ *depends on* $\bar{X}$.

C. $SE(\hat{\beta}_1)$ *depends on variance of* X.

D. $SE(\hat{\beta}_0)$ *is independent of* X *values.*

Problem 18 *Which of the following influence* R^2 *in simple linear regression?*

A. *Slope magnitude*

B. *Spread of* X *values*

C. *Residual variance*

D. *Sample size*

Problem 19 *Which of the following statements about OLS in simple linear regression are true?*

A. *Minimizes RSS*

B. *Slope depends on covariance between* X *and* Y

C. *Intercept ensures the fitted line passes through* $(\bar{X}, \bar{Y})$

D. *Sum of squares of residuals equals TSS*

Problem 20 *A dataset contains advertising spend (*X *in \$1000) and sales (*Y *in units):* $\{(1,5),(2,7),(3,8),(4,10)\}$. *Which coefficient represents the change in sales per \$1000 spent?*

(A) $\hat{\beta}_0$

(B) $\hat{\beta}_1$

(C) *RSE*

(D) *RSS*

Problem 21 *In simple linear regression, which of the following decreases with increasing sample size?*

(A) *Residual Standard Error*

(B) *RSS*

(C) *TSS*

(D) $\bar{Y}$

Problem 22 *Given data* $\{(1,2),(2,4),(3,6)\}$, *which of the following is TRUE about* R^2*?*

(A) $R^2 = 0$

(B) $R^2 = 1$

(C) $R^2 < 0$

(D) $R^2 > 1$

Problem 23 *For dataset* $\{(-1,1),(2,-5),(3,5)\}$, *fit regression* $Y = wX$ *through origin. Compute* w *rounded to 3 decimal places.*

Problem 24 *Consider the multiple regression model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$. Which of the following is true for OLS estimates?*

(A) *They minimize $\sum (Y_i - \hat{Y}_i)$*

(B) *They minimize $\sum (Y_i - \hat{Y}_i)^2$*

(C) *They maximize R^2*

(D) *They minimize $\sum |\epsilon_i|$*

Problem 25 *In the model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$, if X_1 and X_2 are perfectly correlated, then:*

(A) *$\hat{\beta}_1$ and $\hat{\beta}_2$ are unique*

(B) *$R^2 = 0$*

(C) *$\mathbf{X}^T\mathbf{X}$ is singular*

(D) *Residuals are zero*

Problem 26 *For a model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$, if X_2 is constant for all observations:*

(A) *$\hat{\beta}_2 = 0$*

(B) *$\hat{\beta}_2$ cannot be estimated*

(C) *$\hat{\beta}_1 = 0$*

(D) *Model reduces to simple regression*

Problem 27 *Adjusted R^2:*

(A) *Always increases when new predictors are added*

(B) *Always decreases when new predictors are added*

(C) *May increase or decrease when new predictors are added*

(D) *Is independent of the number of predictors*

Problem 28 *If one predictor is a linear combination of others, it leads to:*

(A) *High R^2*

(B) *Perfect multicollinearity*

(C) *Low variance of coefficients*

(D) *Model overfitting*

Problem 29 *The number of parameters estimated in a multiple regression with p predictors (not including intercept) is:*

(A) *p*

(B) $p + 1$

(C) $n - p$

(D) $n - p - 1$

Problem 30 *If* $R^2 = 0.95$ *and Adjusted* $R^2 = 0.80$*, this indicates:*

(A) Excellent model

(B) Underfitting

(C) Possible overfitting

(D) None

Problem 31 *Which of the following are assumptions in multiple linear regression?*

(A) Errors have constant variance

(B) Predictors are uncorrelated with residuals

(C) Relationship is linear in parameters

(D) Predictors are uncorrelated with each other

Problem 32 *In multiple regression* $Y = X\beta + \epsilon$*, increasing* p *while keeping* n *fixed:*

(A) Increases R^2

(B) Decreases Adjusted R^2

(C) May cause overfitting

(D) Reduces degrees of freedom

Problem 33 *If two predictors are highly correlated:*

(A) Removing one may stabilize coefficients

(B) R^2 *will decrease drastically*

(C) Coefficient signs may become unpredictable

(D) Standard errors of coefficients increase

Problem 34 *Which of the following can be consequences of omitting a relevant predictor?*

(A) Biased estimates

(B) Increased residual variance

(C) Lower R^2

(D) Reduced multicollinearity

Problem 35 *Which are true about Adjusted* R^2*?*

(A) *Penalizes inclusion of irrelevant predictors*

(B) *Can be negative*

(C) *Always less than or equal to* R^2

(D) *Independent of sample size*

Problem 36 *Given data* $(X_1, X_2, Y) = (1,2,3), (2,1,2), (3,2,4), (4,3,6)$, *compute* $\hat{\beta}_1$ *and* $\hat{\beta}_2$ *using OLS (rounded to 3 decimals).*

Problem 37 *For model* $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$, $R^2 = 0.9$, $n = 20$, $p = 2$. *Compute Adjusted* R^2.

Problem 38 *If* $\hat{\beta} = (X^T X)^{-1} X^T Y$, *and*

$$X^T X = \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix}, \quad X^T Y = \begin{bmatrix} 10 \\ 8 \end{bmatrix},$$

find $\hat{\beta}$.

Problem 39 *A regression model* $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$ *on* $n = 10$ *observations gives:*

$$\sum (Y_i - \hat{Y}_i)^2 = 45, \quad \sum (Y_i - \bar{Y})^2 = 100.$$

Compute R^2 *and Adjusted* R^2 *(report Adjusted* R^2*, 3 decimals).*

Problem 40 *A model* $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$ *is fitted with* $n = 8$. *Residual Sum of Squares (RSS)* $=$ *36. Compute Residual Standard Error (RSE) up to 3 decimals.*

Problem 41 *Given* $R^2 = 0.92$, *Adjusted* $R^2 = 0.89$, *and sample size* $n = 25$. *Find number of predictors* p.

Problem 42 *In ridge regression, the addition of the penalty term* $\lambda \sum_j \beta_j^2$ *primarily helps to:*

(A) *Reduce bias at the cost of increased variance*

(B) *Reduce variance at the cost of increased bias*

(C) *Reduce both bias and variance*

(D) *Increase both bias and variance*

Problem 43 *If* $\lambda = 0$, *the ridge regression estimator becomes:*

(A) *Biased and unstable*

(B) *Unbiased and minimum variance*

(C) *Equivalent to ordinary least squares*

(D) *Undefined*

Problem 44 *The ridge estimator $\hat{\beta}_{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$ exists for:*

(A) All $\lambda > 0$

(B) Only when $\mathbf{X}^T\mathbf{X}$ is invertible

(C) $\lambda = 0$

(D) None of the above

Problem 45 *As $\lambda \to \infty$ in ridge regression, the coefficients:*

(A) Approach OLS estimates

(B) Approach zero

(C) Diverge to infinity

(D) Become random

Problem 46 *Ridge regression is particularly useful when:*

(A) The predictors are uncorrelated

(B) The predictors are highly correlated (multicollinear)

(C) The model is perfectly specified

(D) The sample size is extremely large

Problem 47 *Which of the following statements about ridge regression are correct?*

(A) Ridge regression shrinks coefficient magnitudes.

(B) Ridge regression can handle perfect multicollinearity.

(C) Ridge regression sets some coefficients exactly to zero.

(D) Ridge regression always improves test set performance.

Problem 48 *The regularization parameter λ in ridge regression controls:*

(A) The bias–variance trade-off

(B) The amount of shrinkage applied to coefficients

(C) The learning rate in optimization

(D) The number of predictors used in the model

Problem 49 *Consider the cost function $J(\boldsymbol{\beta}) = \|Y - X\beta\|^2 + \lambda\|\beta\|^2$. Which of the following are true?*

(A) It is convex in β

(B) It guarantees a unique global minimum for $\lambda > 0$

(C) It is non-differentiable for $\lambda > 0$

(D) *It cannot be solved in closed form*

Problem 50 *Compared to OLS, ridge regression tends to:*

(A) *Reduce overfitting*

(B) *Increase training error*

(C) *Reduce test error*

(D) *Increase variance*

Problem 51 *Which of the following statements about ridge and OLS are true?*

(A) *Ridge regression always has lower variance than OLS*

(B) *Ridge regression may have higher bias than OLS*

(C) *Ridge estimator approaches OLS when* $\lambda = 0$

(D) *Ridge regression always has lower bias than OLS*

Problem 52 *Given* $X = [1, 2, 3]$, $Y = [2, 2.5, 4]$, *compute the ridge regression coefficient* $\hat{\beta}$ *for model* $y = \beta x$ *with* $\lambda = 1$. *(Round off to three decimal places.)*

Problem 53 *For a model with* $\mathbf{X}^T\mathbf{X} = 100$, $\mathbf{X}^T\mathbf{Y} = 200$, *and* $\lambda = 25$, *compute the ridge estimator* $\hat{\beta}$.

Problem 54 *In ridge regression, the shrinkage factor is* $\frac{1}{1+\lambda}$. *If* $\lambda = 4$, *what proportion of the OLS coefficient remains?*

Problem 55 *Given true coefficient* $\beta = 3$ *and estimated ridge coefficient* $\hat{\beta} = 2.4$, *compute the squared bias.*

Problem 56 *If the variance of* $\hat{\beta}_{OLS}$ *is* 9 *and* λ *reduces it by* 60%, *what is the new variance under ridge regression?*

Problem 57 *For* $n = 3$, $\mathbf{X}^T\mathbf{X} = 50$, $\mathbf{X}^T\mathbf{Y} = 100$, *and* $\lambda = 10$, *find* $\hat{\beta}_{ridge}$.

Problem 58 *Assume* X_1 *and* X_2 *are perfectly correlated. Ridge regression with* $\lambda = 2$ *gives coefficient estimates* $\hat{\beta}_1 = \hat{\beta}_2 = 1.5$. *You are also given the sample means:* $\bar{X}_1 = 3$, $\bar{X}_2 = 3$, *and* $\bar{Y} = 12$.

(a) *Compute the intercept term* $\hat{\beta}_0$.

(b) *Using the ridge regression model, compute the predicted value* $\hat{y}$ *for input* $(x_1, x_2) = (3, 3)$.

Problem 59 *A ridge regression model yields coefficients* $[1.2, 0.8, 0.6]$ *for* $\lambda = 1$. *When* $\lambda = 5$, *coefficients become* $[0.6, 0.4, 0.3]$. *Compute approximate shrinkage ratio.*

Problem 60 *If* $R^2_{OLS} = 0.85$ *and ridge regression with* $\lambda = 2$ *gives* $R^2 = 0.82$, *compute the relative reduction in explained variance.*

Problem 61 *Given* $\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix}$, $\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 8 \\ 7 \end{bmatrix}$, *and* $\lambda = 1$, *compute* $\hat{\boldsymbol{\beta}}_{ridge}$ *(two decimal places).*

Problem 62 *For a ridge model, the effective degrees of freedom are defined as* trace(S), *where* $S = X(X^TX + \lambda I)^{-1}X^T$. *If* S *has eigenvalues* $[0.9, 0.8, 0.5]$, *compute the effective model complexity.*

Problem 63 *Given that increasing* λ *from* 1 *to* 10 *reduces test MSE from* 12 *to* 9, *but increasing to* 50 *increases MSE to* 14, *find the optimal* λ *region qualitatively.*

Problem 64 *For a dataset with* $p = 5$ *predictors and* $n = 10$ *samples,* $\mathbf{X}^T\mathbf{X}$ *is singular. What is the smallest* λ *value (in theory) needed to ensure invertibility of* $(\mathbf{X}^T\mathbf{X} + \lambda I)$*?*

2.5 Try it Yourself

Exercise 12 *Number of workers* (X) *and packages processed* (Y)*:*
$\{(2, 52), (3, 71), (4, 79), (5, 97), (6, 111)\}$. *Compute* $\hat{\beta}_0$ *and* $\hat{\beta}_1$.

Exercise 13 *Daily sales* (X *in \$1000) and profits* (Y *in \$1000):*
$\{(1, 10), (2, 20), (3, 30), (4, 40), (5, 50)\}$. *Compute* R^2.

Exercise 14 *Dataset:* $\{(5, 110), (6, 130), (7, 150), (8, 170), (9, 190)\}$. *Predict* Y *at* $X = 10$ *using linear regression.*

Exercise 15 *Advertising spend* (X) *and revenue* (Y)*:*
$\{(1, 50), (3, 90), (4, 120), (6, 150), (7, 180)\}$. *Compute residuals and RSS.*

Exercise 16 *Daily hours worked* (X) *and number of units* (Y)*:*
$\{(2, 100), (3, 130), (5, 170), (6, 200), (7, 220)\}$. *Compute 95% confidence interval for* $\hat{\beta}_1$.

Exercise 17 *Dataset:* $\{(1, 2), (2, 5), (3, 7), (4, 10)\}$. *Fit regression* $Y = \beta_0 + \beta_1 X$. *Compute* $\hat{\beta}_1$.

Exercise 18 *Dataset:* $\{(2, 3), (3, 4), (4, 6), (5, 8), (6, 11)\}$. *Compute residuals and RSS.*

Exercise 19 *Daily temperature* (X) *and ice cream sales* (Y)*:* $\{(25, 120), (28, 130), (30, 150), (32, 165), (35, 180)\}$. *Compute predicted* Y *for* $X = 33$.

Exercise 20 *Dataset:* $\{(1, 3), (2, 4), (3, 5), (4, 6), (5, 8), (6, 9)\}$. *Compute 95% prediction interval for* $X = 7$ *using RSE* $= 1$.

Exercise 21 *A multiple regression has RSE* $= 2.5$, $n = 50$, $p = 4$. *Compute residual variance.*

Exercise 22 *For model* $\hat{Y} = 2 + 1.2X_1 + 0.8X_2$, *compute predicted* Y *for* $(X_1, X_2) = (3, 4)$.

Exercise 23 *Given* $R^2 = 0.92$, *Adjusted* $R^2 = 0.89$, $n = 30$. *Find number of predictors* p.

Exercise 24 *Data:* $Y = [4, 5, 7, 10]$, $\hat{Y} = [3.8, 5.2, 6.9, 9.8]$. *Compute* R^2.

Exercise 25 *In a regression with intercept,* $\sum(Y_i - \hat{Y}_i)^2 = 50$ *and* $\sum(Y_i - \bar{Y})^2 = 200$. *Compute* R^2.

Exercise 26 *A regression gives residual variance $\sigma^2 = 1.8$, $n = 12$, $p = 3$. Compute Residual Standard Error (RSE).*

Exercise 27 *Given the regression $\hat{Y} = 3 + 1.2X_1 + 0.8X_2$, find $\hat{Y}$ for $(X_1, X_2) = (2, 5)$ and compute residual if $Y = 11$.*

Exercise 28 *In multiple regression, $R^2 = 0.90$, $n = 30$, $p = 4$. Compute Adjusted R^2.*

Exercise 29 *A dataset has $\sum X_1^2 = 50$, $\sum X_2^2 = 30$, $\sum X_1X_2 = 10$, $\sum X_1Y = 80$, $\sum X_2Y = 60$. Estimate $\hat{\beta}_1$ and $\hat{\beta}_2$ for $Y = \beta_1X_1 + \beta_2X_2$ (no intercept). Report $\hat{\beta}_1$ rounded to 3 decimals.*

Exercise 30 *Regression output:*

$$\hat{\beta}_0 = 2.4, \quad \hat{\beta}_1 = 1.3, \quad SE(\hat{\beta}_1) = 0.25.$$

At 95% confidence level ($t_{0.025,df=18} = 2.101$), compute 95% confidence interval for β_1. Report upper limit rounded to 3 decimals.

Exercise 31 *A simple linear regression model gives $\hat{\beta}_{OLS} = 2.5$. Using ridge regression with penalty $\lambda = 3$ and $\mathbf{X}^T\mathbf{X} = 12$, compute $\hat{\beta}_{ridge}$. (Round off to two decimal places.)*

Exercise 32 *Given $\mathbf{X}^T\mathbf{X} = 40$, $\mathbf{X}^T\mathbf{Y} = 160$, and $\lambda = 10$, find $\hat{\beta}_{ridge}$. Also compute the ratio $\frac{\hat{\beta}_{ridge}}{\hat{\beta}_{OLS}}$.*

Exercise 33 *A ridge regression model has bias $= 0.8$ and variance $= 1.6$. Compute the expected mean squared error (MSE) of the estimator.*

Exercise 34 *For a dataset with $\mathbf{X}^T\mathbf{X} = 25$, $\mathbf{X}^T\mathbf{Y} = 100$, $\lambda = 5$, compute the ridge estimate $\hat{\beta}$ and the corresponding shrinkage factor.*

Exercise 35 *Suppose the true model is $y = 3x + \epsilon$ and the ridge estimate is $\hat{\beta}_{ridge} = 2.4$. Find the percentage shrinkage of the coefficient.*

Exercise 36 *In a ridge regression, $\hat{\beta}_{OLS} = 5$ and $\lambda = 4$, $\mathbf{X}^T\mathbf{X} = 16$. Compute $\hat{\beta}_{ridge}$ and the reduction in magnitude relative to OLS (in percentage).*

Exercise 37 *Given $\mathbf{X}^T\mathbf{X} = 9$, $\mathbf{X}^T\mathbf{Y} = 27$, and $\lambda = 3$, compute $\hat{\beta}_{ridge}$. Then, if λ doubles, compute the new $\hat{\beta}_{ridge}$.*

Exercise 38 *A ridge regression yields $\hat{\beta}_{ridge} = 1.6$ for $\lambda = 2$. If $\hat{\beta}_{OLS} = 2.0$, and the error variance $\sigma^2 = 0.5$, compute the bias introduced by ridge.*

Exercise 39 *For multiple regression with two predictors, $\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 10 & 4 \\ 4 & 6 \end{bmatrix}$, $\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} 14 \\ 10 \end{bmatrix}$, and $\lambda = 2$. Compute $\hat{\boldsymbol{\beta}}_{ridge}$ (round to two decimals).*

Exercise 40 *Let the eigenvalues of $\mathbf{X}^T\mathbf{X}$ be $[10, 2, 0.5]$. If $\lambda = 1$, compute the effective degrees of freedom for ridge regression: $d_{eff} = \sum_{j=1}^{p} \frac{\lambda_j}{\lambda_j + \lambda}$.*

2.6 YouTube Links and QR Codes

Lecture	Details	YouTube Link	QR Code
5	Linear Regression Explained: OLS, RSS, R^2 & Error Metrics	https://youtu.be/0VokMba7Zsk	
6	Linear Regression Standard Error Explained with Examples	https://youtu.be/lKrLXDW9zhI	
7	Problem Solving – Linear Regression Standard Error (Solutions 12 - 23)	https://youtu.be/KT0gwV4ca7U	
8	Multiple Regression Explained: OLS, RSS & Step-by-Step Examples	https://youtu.be/MtAV8DV1_fY	

9	Multiple Regression Explained: R^2 & Adjusted R^2 (Goodness of Fit)	https://youtu.be/ntkwx47h3dM	
10	Problem Solving – Multiple Linear Regression (Solutions 24 - 41)	https://youtu.be/z0ffEOi5TRk	
11	Ridge Regression Explained: L2 Regularization & OLS Issues	https://youtu.be/YHxJY7JE1BE	
12	Problem Solving – Ridge Regression (L2) (Solutions 42 - 64)	https://youtu.be/-NYU26KhVkg	

Chapter 3

Classification Models - I

3.1 Logistic Regression

3.1.1 Sigmoid Function and Log-Odds Interpretation

Concept: Logistic Regression

Logistic Regression is used when the dependent variable Y is **categorical (binary)**, i.e., it takes only two possible values — typically 0 or 1. It models the probability that $Y = 1$ as a function of input features $X_1, X_2, \ldots, X_p$.

Unlike linear regression, where the output can take any real value, in classification we need outputs between 0 and 1, representing probabilities. Thus, the model uses the sigmoid (logistic) function to map real-valued predictions into the range $(0, 1)$.

$$p = P(Y = 1 \mid X) = \frac{1}{1 + e^{-z}}$$

where $z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$

Sigmoid Function Properties

- $\sigma(z) = \frac{1}{1+e^{-z}}$
- $\sigma(z)$ is bounded between 0 and 1.
- $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ (important for gradient derivations)

Log-Odds Interpretation: The odds of an event are defined as $\frac{p}{1-p}$. Taking the logarithm gives the **logit** (log-odds):

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

This means that logistic regression assumes the **log-odds of the probability** are linearly related to the predictors.

Example 19:

A model for disease prediction is:

$$\log\left(\frac{p}{1-p}\right) = -3 + 0.05 \times (\text{Age}) + 1.2 \times (\text{weight})$$

For a 40-year-old weight:

$$z = -3 + 0.05(40) + 1.2(1) = 0.2$$

$$p = \frac{1}{1+e^{-0.2}} = 0.5498$$

Hence, the model predicts a 54.98% probability of having the disease.

Example 20:

Suppose $\beta_1 = 0.7$ for the predictor "Study Hours." Then, increasing study hours by 1 increases the **log-odds** of passing by 0.7. In terms of odds:

$$e^{0.7} \approx 2.01$$

This means the odds of passing roughly double for each additional study hour.

3.1.2 Decision Boundary Analysis

The decision boundary separates predicted classes 0 and 1.

At boundary: $p = 0.5$

$$\Rightarrow \frac{1}{1+e^{-z}} = 0.5 \implies z = 0$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

Concept: Decision Boundary

The decision boundary is linear in the input space for logistic regression. However, applying non-linear transformations to features (like polynomial terms) can produce non-linear boundaries.

Example 21:

For a 2D feature space:

$$z = -3 + 2x_1 + x_2$$

Decision boundary:

$$-3 + 2x_1 + x_2 = 0 \implies x_2 = 3 - 2x_1$$

This is a straight line dividing the classes.

Example 22:

If a polynomial term is added:

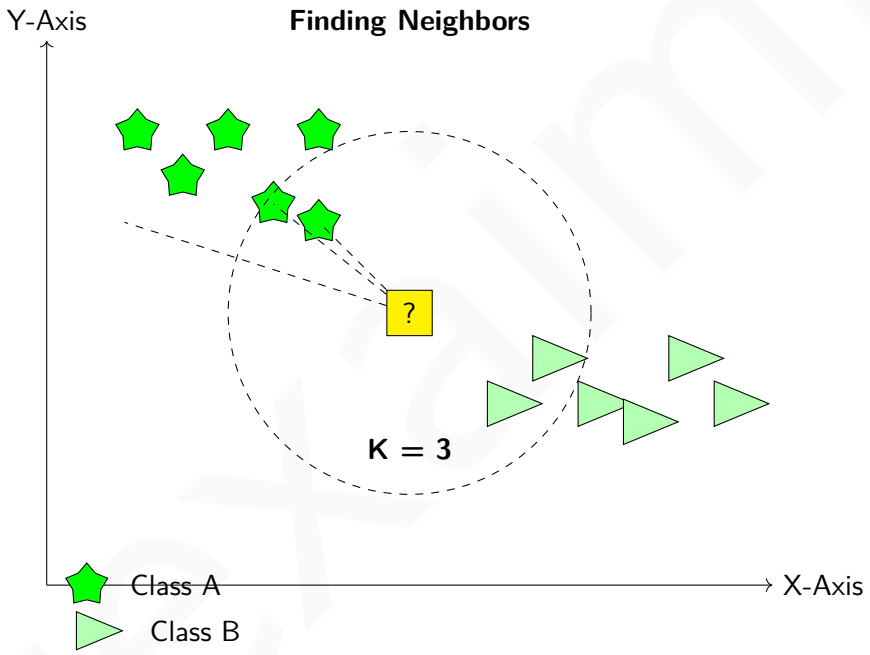
$$z = 1 + 0.5x_1^2 + x_2$$

Then, the decision boundary:

$$x_2 = -1 - 0.5x_1^2$$

is **non-linear**, showing the flexibility of logistic regression with feature transformations.

3.2 k-Nearest Neighbour (k-NN)



3.2.1 Distance Metrics in k-NN

Distance Metrics in k-NN

k-NN classification/regression relies on the notion of distance to identify nearest neighbors. Common distance metrics:

- **Euclidean Distance (L2 norm):**

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Most common; sensitive to scale and outliers.

- **Manhattan Distance (L1 norm):**

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

  Less sensitive to outliers; distances are sum of absolute differences.
- **Minkowski Distance:** Generalized metric:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

  Special cases: $p = 1 \rightarrow$ Manhattan, $p = 2 \rightarrow$ Euclidean.

Key Point: Feature scaling is important because large-magnitude features dominate Euclidean distance.

Example 23:

Compute distances between points $A = (1, 2)$ and $B = (4, 6)$:

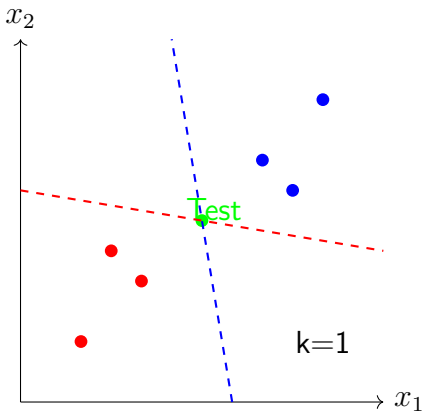
- Euclidean: $\sqrt{(4-1)^2 + (6-2)^2} = \sqrt{9+16} = \sqrt{25} = 5$
- Manhattan: $|4-1| + |6-2| = 3 + 4 = 7$
- Minkowski with $p = 3$: $((3)^3 + (4)^3)^{1/3} = (27 + 64)^{1/3} = 91^{1/3} \approx 4.48$

3.2.2 Choice of k and Model Complexity

Choice of k and Model Complexity

- **Small k (e.g., 1)**: Model follows data closely, low bias, high variance, sensitive to noise.
- **Large k**: Smoother decision boundaries, higher bias, lower variance, may underfit.
- **Trade-off:** Choose k using cross-validation to balance bias and variance.

Graphical Intuition:



Example 24:

Given 9 points in 2D with two classes (0 and 1). Test point at center:

- For $k = 1$: Closest neighbor is class 0 $\rightarrow$ prediction 0
- For $k = 5$: 3 neighbors class 1, 2 neighbors class 0 $\rightarrow$ prediction 1

Observation: Increasing k can change prediction and smooth decision boundary.

3.2.3 Effect of Dimensionality on Performance

Effect of Dimensionality on k-NN Performance

- As the number of features (dimensions) increases:
  - Distances between points become similar (*curse of dimensionality*)
  - Nearest neighbors may be far away $\rightarrow$ poor generalization
- Feature scaling becomes critical.
- Dimensionality reduction (PCA, feature selection) improves performance.

Intuition: In high dimensions, most points lie near the corners of a hypercube; Euclidean distance loses discriminative power.

Example 25:

Consider points in 10D: $x_1 = (1, 1, ..., 1), x_2 = (2, 2, ..., 2)$. Compute Euclidean distance:

$$d = \sqrt{\sum_{i=1}^{10}(2-1)^2} = \sqrt{10} \approx 3.16$$

If all 100 points are randomly distributed in 10D cube $[0, 10]^10$, nearest neighbor distances are around 7-8 units. Distances become similar $\rightarrow$ difficulty in distinguishing neighbors $\rightarrow$ accuracy drops.

3.3 Naive Bayes Classifier

3.3.1 Bayes' Theorem and Conditional Independence

Bayes' Theorem

Bayes' theorem provides a way to update our belief about a class C given observed features $X = (X_1, X_2, \ldots, X_n)$:

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

Explanation:

- $P(C)$ is the prior probability of class C, representing our knowledge before seeing data.

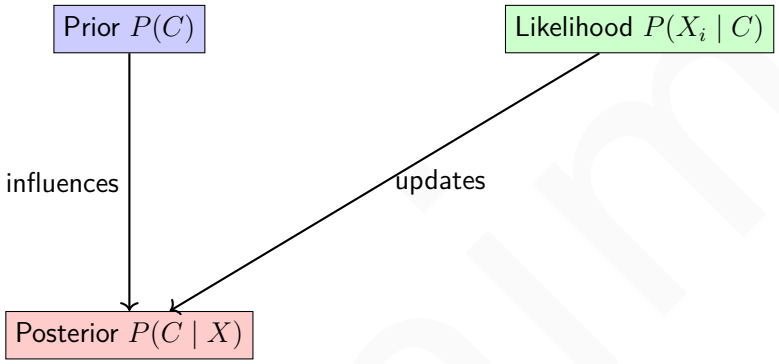
- $P(X \mid C)$ is the likelihood: probability of observing X assuming class C.
- $P(X)$ is the marginal probability of features: ensures probabilities sum to 1.
- $P(C \mid X)$ is the posterior: updated probability of class C after observing features.

Conditional Independence Assumption

Naive Bayes assumes features are conditionally independent given the class:

$$P(X \mid C) = \prod_{i=1}^{n} P(X_i \mid C)$$

This reduces the exponential complexity of joint probability estimation to linear, making computation feasible.



Posterior combines prior and likelihood for prediction

Example 26:

A weather dataset for predicting **PlayTennis** has features `Outlook` and `Humidity`. Suppose:

- $P(\text{Yes}) = 0.6$, $P(\text{No}) = 0.4$
- $P(\text{Sunny} \mid \text{Yes}) = 0.3$, $P(\text{High} \mid \text{Yes}) = 0.4$

Then posterior for $C = \text{Yes}$:

$$P(\text{Yes} \mid X) \propto 0.6 \cdot 0.3 \cdot 0.4 = 0.072$$

Compute similarly for No and select the class with higher posterior.

3.3.2 Gaussian, Multinomial, and Bernoulli Naive Bayes

Gaussian Naive Bayes

For continuous features $X_i \in \mathbb{R}$, model the likelihood as a Gaussian:

$$P(X_i \mid C) = \frac{1}{\sqrt{2\pi\sigma_{iC}^2}} \exp\left[-\frac{(X_i - \mu_{iC})^2}{2\sigma_{iC}^2}\right]$$

Posterior:

$$P(C \mid X) \propto P(C) \prod_{i=1}^{n} P(X_i \mid C)$$

Reasoning: This allows modeling continuous features without discretization.

Multinomial Naive Bayes

Used for discrete count data (e.g., word frequencies in documents):

$$P(X \mid C) = \frac{(\sum_i X_i)!}{\prod_i X_i!} \prod_i \theta_{iC}^{X_i}$$

where $\theta_{iC} = P(\text{word}_i \mid C)$.

Bernoulli Naive Bayes

For binary features $X_i \in \{0, 1\}$:

$$P(X_i \mid C) = \theta_{iC}^{X_i}(1 - \theta_{iC})^{1-X_i}$$

Useful when modeling presence/absence of features.

Example 27:

Email spam classification with binary features:

$$X_1 = \text{“contains ‘offer’”}, X_2 = \text{“contains ‘free’”}$$

Given $P(\text{Spam}) = 0.4$, $P(X_1 = 1 \mid \text{Spam}) = 0.7$, $P(X_2 = 1 \mid \text{Spam}) = 0.6$, test email: $X_1 = 1, X_2 = 0$.

$$P(\text{Spam} \mid X) \propto 0.4 \cdot 0.7 \cdot (1 - 0.6) = 0.112$$

Similarly compute $P(\text{Ham} \mid X)$ and choose higher.

3.3.3 Parameter Estimation and MAP Decision Rule

Maximum A Posteriori (MAP) Decision Rule

Definition: The MAP decision rule selects the class C that is most probable given the observed features $X = (X_1, X_2, \ldots, X_n)$. Formally:

$$\hat{C} = \arg\max_C P(C \mid X)$$

Using Bayes' Theorem:

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

Since $P(X)$ is the same for all classes, the MAP decision reduces to:

$$\hat{C} = \arg\max_C P(C)P(X \mid C)$$

Assuming conditional independence of features (Naive Bayes assumption):

$$P(X \mid C) = \prod_{i=1}^{n} P(X_i \mid C) \quad \Rightarrow \quad \hat{C} = \arg\max_C P(C) \prod_{i=1}^{n} P(X_i \mid C)$$

Reasoning:

- The MAP rule balances **prior knowledge** $P(C)$ with the **likelihood** of the observed features $P(X \mid C)$.
- It chooses the class that is most likely given the evidence, not just the one with the highest prior probability.

Example: Suppose a dataset has two classes, C_1 and C_2, with priors $P(C_1) = 0.4, P(C_2) = 0.6$. For a new observation X, the likelihoods are $P(X \mid C_1) = 0.5$ and $P(X \mid C_2) = 0.3$.

MAP decision: $\hat{C} = \arg\max_C P(C)P(X \mid C) = \arg\max\{0.4{\cdot}0.5, 0.6{\cdot}0.3\} = \arg\max\{0.2, 0.18\} = C_1$

Hence, the new observation is classified as C_1 even though C_2 had higher prior probability.

Parameter Estimation (MLE)

Goal: Estimate the parameters of the Naive Bayes model from data using Maximum Likelihood Estimation (MLE).

- **1. Multinomial Naive Bayes (word counts):**

$$\hat{\theta}_{iC} = P(\text{word}_i \mid C) = \frac{\text{count}(\text{word}_i \text{ in class } C)}{\sum_j \text{count}(\text{word}_j \text{ in class } C)}$$

  - $\hat{\theta}_{iC}$ = probability of word i appearing in class C
  - Used in text classification (Bag-of-Words, TF–IDF)

- **2. Bernoulli Naive Bayes (binary presence/absence):**

$$\hat{\theta}_{iC} = P(X_i = 1 \mid C) = \frac{\text{count}(X_i = 1 \text{ and } C)}{\text{count}(C)}$$

  - $\hat{\theta}_{iC}$ = probability that feature i is present in class C
  - Features are Yes/No, Present/Absent, 0/1

- **3. Gaussian Naive Bayes (continuous features):**

$$\hat{\mu}_{iC} = \frac{1}{n_C} \sum_{j:X_j \in C} X_{ij}, \qquad \hat{\sigma}^2_{iC} = \frac{1}{n_C} \sum_{j:X_j \in C} (X_{ij} - \hat{\mu}_{iC})^2$$

  - $\hat{\mu}_{iC}$ = mean of feature i in class C
  - $\hat{\sigma}^2_{iC}$ = variance of feature i in class C

Example 28:

Weather dataset for tennis decision:

Outlook	Temperature	PlayTennis	Count
Sunny	Hot	Yes	3
Sunny	Hot	No	2
Rainy	Mild	Yes	4
Rainy	Mild	No	1

Estimate $P(\text{PlayTennis=Yes} \mid X = \text{Sunny, Hot})$:

$$P(\text{Yes} \mid X) \propto P(\text{Yes})P(\text{Sunny} \mid \text{Yes})P(\text{Hot} \mid \text{Yes}) = \frac{7}{10} \cdot \frac{3}{7} \cdot \frac{3}{7} \approx 0.184$$

Similarly compute $P(\text{No} \mid X)$ and choose class with higher posterior.

Example 29: Bayes Theorem with Categorical Features

A medical dataset has patients with a symptom S and a disease D. Suppose:

$$P(D = \text{Yes}) = 0.1, \quad P(D = \text{No}) = 0.9$$
$$P(S = \text{Present} \mid D = \text{Yes}) = 0.8, \quad P(S = \text{Present} \mid D = \text{No}) = 0.2$$

Compute $P(D = \text{Yes} \mid S = \text{Present})$.

Solution:

$$P(D = \text{Yes} \mid S) = \frac{P(S \mid D = \text{Yes})P(D = \text{Yes})}{P(S)}$$

First, compute marginal $P(S)$:

$$P(S) = P(S \mid D = \text{Yes})P(D = \text{Yes}) + P(S \mid D = \text{No})P(D = \text{No}) = 0.8 \cdot 0.1 + 0.2 \cdot 0.9 = 0.26$$

$$\Rightarrow P(D = \text{Yes} \mid S) = \frac{0.8 \cdot 0.1}{0.26} \approx 0.308$$

Interpretation: Even though the symptom is present, the probability of disease is only 30.8%.

Example 30: Gaussian Naive Bayes with Continuous Features

Suppose we have a dataset of students with features: hours studied X_1 and marks obtained X_2, and class $C = \{\text{Pass}, \text{Fail}\}$.
Data statistics:

$$\text{Pass: } \mu_{X_1} = 5, \sigma_{X_1} = 1.5, \mu_{X_2} = 60, \sigma_{X_2} = 10$$

$$\text{Fail: } \mu_{X_1} = 2, \sigma_{X_1} = 1, \mu_{X_2} = 40, \sigma_{X_2} = 5$$

A student studied 4 hours and scored 55 marks. Using Gaussian NB, compute posterior probabilities (assume $P(\text{Pass}) = 0.7, P(\text{Fail}) = 0.3$).
Solution:

$$P(X_1 = 4 \mid \text{Pass}) = \frac{1}{\sqrt{2\pi(1.5)^2}} e^{-(4-5)^2/(2 \cdot 1.5^2)} \approx 0.219$$

$$P(X_2 = 55 \mid \text{Pass}) = \frac{1}{\sqrt{2\pi 10^2}} e^{-(55-60)^2/(2 \cdot 10^2)} \approx 0.038$$

$$P(\text{Pass} \mid X_1 = 4, X_2 = 55) \propto 0.7 \cdot 0.219 \cdot 0.038 \approx 0.00583$$

Similarly for Fail:

$$P(X_1 = 4 \mid \text{Fail}) \approx 0.054, \quad P(X_2 = 55 \mid \text{Fail}) \approx 0.00027$$

$$P(\text{Fail} \mid X_1, X_2) \propto 0.3 \cdot 0.054 \cdot 0.00027 \approx 0.0000044$$

Decision: Classify as Pass because posterior is higher.

Example 31: Multinomial Naive Bayes for Text Classification

Suppose we have 2 classes: Spam and Ham, and vocabulary: "win", "free", "offer". Training counts:

$$\text{Spam: win} = 3, \text{free} = 4, \text{offer} = 2, \text{total words} = 9$$

$$\text{Ham: win} = 1, \text{free} = 0, \text{offer} = 1, \text{total words} = 4$$

Compute probability that a new message containing "win free" is Spam (assume $P(\text{Spam}) = 0.6, P(\text{Ham}) = 0.4$).
Solution:

$$P(\text{win free} \mid \text{Spam}) = \frac{3}{9} \cdot \frac{4}{9} = 0.148$$

$$P(\text{win free} \mid \text{Ham}) = \frac{1}{4} \cdot \frac{0}{4} = 0$$

$$P(\text{Spam} \mid \text{message}) \propto 0.6 \cdot 0.148 = 0.0888$$

Decision: Message classified as Spam.

Example 32: Bernoulli Naive Bayes with Binary Features

Email classification with 2 features: X_1 = contains "offer", X_2 = contains "free" (binary). Training statistics:

$$P(X_1 = 1 \mid \text{Spam}) = 0.7, P(X_2 = 1 \mid \text{Spam}) = 0.6$$
$$P(X_1 = 1 \mid \text{Ham}) = 0.1, P(X_2 = 1 \mid \text{Ham}) = 0.05$$
$$P(\text{Spam}) = 0.4, P(\text{Ham}) = 0.6$$

Test email: $X_1 = 1, X_2 = 0$

Solution:

$$P(\text{Spam} \mid X) \propto 0.4 \cdot 0.7 \cdot (1 - 0.6) = 0.112$$

$$P(\text{Ham} \mid X) \propto 0.6 \cdot 0.1 \cdot (1 - 0.05) = 0.057$$

Decision: Classify as Spam.

Example 33: MAP Decision Rule

Weather dataset with features: Outlook (Sunny, Rainy) and Temperature (Hot, Mild). Counts:

$$P(\text{PlayTennis=Yes}) = 0.7, \quad P(\text{PlayTennis=No}) = 0.3$$

$$P(\text{Sunny} \mid \text{Yes}) = 0.5, P(\text{Hot} \mid \text{Yes}) = 0.6$$

$$P(\text{Sunny} \mid \text{No}) = 0.2, P(\text{Hot} \mid \text{No}) = 0.5$$

Test instance: $X = (\text{Sunny, Hot})$

Solution:

$$P(\text{Yes} \mid X) \propto 0.7 \cdot 0.5 \cdot 0.6 = 0.21$$

$$P(\text{No} \mid X) \propto 0.3 \cdot 0.2 \cdot 0.5 = 0.03$$

Decision: Predict PlayTennis = Yes (MAP estimate).

Naive Bayes Variant	Data Type	Advantages	Limitations	Bias / Variance / Sample Size + Independence
Gaussian	Continuous	Small data works, simple	Assumes normal distribution	High bias, low variance: model assumes Gaussian (reduces variance but may misfit non-Gaussian data). Performs well with small samples. Independence assumption may be violated if features correlate, which can bias results.
Multinomial	Word counts	Great for text	Zero count issue (requires smoothing)	High bias, low variance: counts assumed independent (simplifies model → low variance) but words are not truly independent (bias). More data improves estimates; independence violation introduces bias but often classification still works.
Bernoulli	Binary	Good for short text	Ignores word frequency	High bias, low variance: simplifies features to binary (low variance, high bias). Works well with small data. Independence assumption may be violated if features co-occur (e.g., related words), which increases bias.

Naive Bayes variants with reasons for bias/variance/sample-size behavior and independence effects.

3.4 Problems

Problem 65 *Consider a logistic regression model:*

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

If $\beta_0 = -2$ *and* $\beta_1 = 0.5$*, find the probability* $P(Y = 1|X = 6)$.

Problem 66 *In a binary classification problem, the predicted log-odds is given by:*

$$\log\left(\frac{p}{1-p}\right) = 1.5 + 0.3X_1 - 0.2X_2$$

For $(X_1, X_2) = (2, 5)$, compute the odds and the predicted class (using threshold 0.5).

Problem 67 *The sigmoid function is defined as $\sigma(z) = \frac{1}{1+e^{-z}}$. If $z = 0$, what is the derivative $\frac{d\sigma}{dz}$? (Choose the correct numerical value.)*

Problem 68 *Which of the following statements about logistic regression are TRUE?*

(A) It assumes the response variable follows a Bernoulli distribution.

(B) The parameters are estimated by minimizing squared error.

(C) The link function used is the logit function.

(D) The model predicts class probabilities rather than class labels.

Problem 69 *For a logistic regression model with one predictor x, the decision boundary occurs when $p = 0.5$. If $\beta_0 = -1$ and $\beta_1 = 0.25$, find the corresponding x value.*

Problem 70 *Suppose the estimated logistic model for heart disease is:*

$$\log\left(\frac{p}{1-p}\right) = -4 + 0.03(\textit{Age}) + 1.5(\textit{weight})$$

Interpret the coefficient of weight.

Problem 71 *The following confusion matrix corresponds to a logistic classifier:*

	Predicted 0	*Predicted 1*
Actual 0	40	10
Actual 1	5	45

Compute the accuracy and precision of the model.

Problem 72 *The decision boundary in logistic regression is:*

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$$

If $\beta_0 = -4, \beta_1 = 2, \beta_2 = 1$, what is the slope of the boundary line in the x_1–x_2 plane?

Problem 73 *For the same model, what happens to the decision boundary if we multiply all coefficients by 5?*

Problem 74 *If $\beta_0 = 0$, $\beta_1 = 1$, and $x = 0$, what is $P(Y = 1|x)$? Then compute its log-odds value.*

Problem 75 *The derivative of the sigmoid $\sigma(z)$ is maximum at which value of z?*

(A) $z \to -\infty$

(B) $z = 0$

(C) $z = 1$

(D) $z \to \infty$

Problem 76 *In a two-class problem, logistic regression outputs probabilities* $p_1 = 0.45$ *and* $p_2 = 0.55$. *If the decision threshold is changed from 0.5 to 0.6, how will the predicted class for this observation change?*

Problem 77 *The output of logistic regression can be interpreted as:*

(A) Probability estimate

(B) Margin

(C) Distance from hyperplane

(D) Signed distance in feature space

Problem 78 *Suppose logistic regression gives the following parameter estimates:*

$$\beta_0 = 1.2, \quad \beta_1 = -0.6$$

Interpret the meaning of β_1 *in terms of the odds ratio.*

Problem 79 *If the decision threshold is increased from* 0.5 *to* 0.7 *in a logistic regression classifier, and for a sample* $p = 0.65$, *what is the new predicted class label? Represent* $Y = 1$ *as* 1 *and* $Y = 0$ *as* 0.

Problem 80 *In a 2D dataset, class 0 points are at (1,1), (2,2) and class 1 points are at (5,5), (6,6). A test point is at (3,3). Using 1-NN with Euclidean distance, the predicted class is:*

(A) 0

(B) 1

(C) Cannot determine

(D) Depends on scaling

Problem 81 *For a dataset with 100 points, what is the effect of choosing a very large* k *in k-NN classification?*

(A) Overfitting

(B) Underfitting

(C) Optimal classification

(D) No effect

Problem 82 *Which distance metric is more sensitive to outliers in high-dimensional space?*

(A) *Manhattan*

(B) *Euclidean*

(C) *Minkowski with* $p = 3$

(D) *Cosine similarity*

Problem 83 *In k-NN regression, increasing* k *generally:*

(A) *Reduces variance*

(B) *Increases variance*

(C) *Reduces bias*

(D) *Increases training error*

Problem 84 *Feature scaling is important in k-NN because:*

(A) *It changes class labels*

(B) *Distances are sensitive to scale*

(C) *It affects variance but not bias*

(D) *k-NN is scale-invariant*

Problem 85 *Which of the following statements about k-NN are correct?*

(A) *High-dimensional data can reduce accuracy (curse of dimensionality)*

(B) *Increasing* k *reduces overfitting*

(C) *Euclidean distance is always better than Manhattan*

(D) *Feature scaling affects k-NN performance*

(E) *Decision boundary becomes more jagged for higher* k

Problem 86 *In k-NN, the choice of* k *affects:*

(A) *Model bias*

(B) *Model variance*

(C) *Training error only*

(D) *Testing error*

(E) *Distance metric sensitivity*

Problem 87 *Which of the following are true regarding k-NN in high dimensions?*

(A) *Distances between points become more uniform*

(B) *Curse of dimensionality increases model bias*

(C) *Dimensionality affects Euclidean and Manhattan distances differently*

(D) *Increasing k completely removes the curse of dimensionality*

(E) *k-NN is robust to irrelevant features*

Problem 88 *Which statements are correct about k-NN regression?*

(A) *k-NN regression predicts the mean of k nearest neighbors*

(B) *Larger k reduces variance*

(C) *Smaller k increases bias*

(D) *Requires distance metric*

(E) *Can produce discontinuous predictions*

Problem 89 *Regarding k-NN classification, which are true?*

(A) *Non-parametric model*

(B) *Training complexity is low*

(C) *Testing complexity increases with dataset size*

(D) *Decision boundary depends on k*

(E) *Sensitive to noisy labels*

Problem 90 *A dataset has 5 points on x-axis: 0,1,2,3,4. Using 3-NN, predict the average y-value if corresponding y-values are 2,4,6,8,10 and test point at x=2.*

Problem 91 *Dataset points: (0,0),(2,2),(4,4),(6,6). Using 1-NN, find predicted class for test point (3,3).*

Problem 92 *Points: (1,2),(3,2),(2,4). Using Minkowski distance with $p = 3$, compute distance between (1,2) and (2,4).*

Problem 93 *A dataset has two classes, C_1 and C_2, with prior probabilities $P(C_1) = 0.6, P(C_2) = 0.4$. A new sample has likelihoods $P(X \mid C_1) = 0.5, P(X \mid C_2) = 0.8$. According to MAP rule, the sample belongs to:*

(A) C_1

(B) C_2

(C) *Cannot decide*

(D) *Equally likely*

Problem 94 *In Gaussian Naive Bayes, increasing the variance σ^2 of a feature:*

(A) *Makes likelihood narrower*

(B) *Makes likelihood wider*

(C) *Has no effect on likelihood*

(D) *Increases posterior probability always*

Problem 95 *Which of the following is true about Multinomial Naive Bayes for text classification?*

(A) *Assumes word order is important*

(B) *Assumes word counts matter*

(C) *Assumes conditional independence of words given class*

(D) *Cannot handle binary features*

Problem 96 *Bernoulli Naive Bayes differs from Multinomial Naive Bayes in that:*

(A) *It uses binary word occurrence*

(B) *It uses continuous features*

(C) *It assumes Gaussian distribution*

(D) *It ignores absent words*

Problem 97 *If features X_1 and X_2 are perfectly correlated, Gaussian Naive Bayes:*

(A) *Works fine without change*

(B) *Assumes independence and may overcount information*

(C) *Fails to compute probabilities*

(D) *Automatically decorrelates features*

Problem 98 *Which statements are correct regarding Naive Bayes classifiers?*

(A) *Conditional independence assumption may be violated in real data*

(B) *Works surprisingly well even if independence assumption fails*

(C) *Always produces calibrated probability estimates*

(D) *Computationally efficient for large datasets*

Problem 99 *In Gaussian Naive Bayes:*

(A) *Posterior is proportional to product of individual feature likelihoods*

(B) *Likelihood of a feature is modeled as Gaussian*

(C) *Requires covariance matrix inversion*

(D) *Works for continuous and discrete features*

Problem 100 *Multinomial Naive Bayes is suitable when:*

(A) *Feature counts are non-negative integers*

(B) *Feature occurrences are binary*

(C) *Feature distribution is Gaussian*

(D) *Vocabulary size is large*

Problem 101 *Bernoulli Naive Bayes assumes:*

(A) *Features are continuous*

(B) *Features are binary*

(C) *Conditional independence of features given class*

(D) *Cannot handle sparse data*

Problem 102 *MAP decision rule in Naive Bayes:*

(A) *Maximizes posterior probability*

(B) *Minimizes likelihood*

(C) *Ignores prior probability*

(D) *Reduces to ML estimate if priors are equal*

Problem 103 *Gaussian NB: Large feature variance can:*

(A) *Decrease posterior probability influence*

(B) *Increase posterior probability influence*

(C) *Not affect posterior probability*

(D) *Make distribution narrower*

Problem 104 *Which statements are correct regarding prior probability in Naive Bayes?*

(A) *Helps MAP classification*

(B) *Ignored in ML estimate*

(C) *Must sum to 1 across classes*

(D) *Cannot handle class imbalance*

Problem 105 *In text classification using Multinomial NB, zero frequency problem can be solved by:*

(A) *Laplace smoothing*

(B) *Removing words with zero count*

(C) *Using Gaussian NB*

(D) *Ignoring the feature*

Problem 106 *Conditional independence assumption implies:*

(A) $P(X_1, X_2|C) = P(X_1|C)P(X_2|C)$

(B) Features are independent marginally

(C) Features are uncorrelated

(D) Posterior probability factorizes

Problem 107 *Bernoulli NB: For sparse data, advantage is:*

(A) Only non-zero features contribute to likelihood

(B) Works with binary features

(C) Ignores missing values

(D) Assumes Gaussian distribution

Problem 108 *Given* $P(C_1) = 0.5, P(C_2) = 0.5$, $P(X_1 = 1 \mid C_1) = 0.8, P(X_1 = 1 \mid C_2) = 0.3$, $P(X_2 = 1 \mid C_1) = 0.6, P(X_2 = 1 \mid C_2) = 0.5$. *Compute* $P(C_1 \mid X_1 = 1, X_2 = 1)$ *(round to 3 decimal places).*

Problem 109 *A Gaussian NB feature has* $\mu = 10, \sigma = 2$. *Compute* $P(X = 12 \mid C)$ *using Gaussian density formula (round to 4 decimals).*

Problem 110 *In a Multinomial NB with 3 classes, a word appears 5 times. Counts in each class: [3,1,6]. Compute likelihood for the word in each class (ignore smoothing, round to 3 decimals).*

Problem 111 *A dataset contains 5 students with two continuous features: Marks* (X_1) *and Attendance* (X_2).

Student	**Class**	**Marks** X_1	**Attendance** X_2
S1	Pass	85	92
S2	Pass	78	88
S3	Fail	45	60
S4	Fail	55	65
S5	Fail	52	58

Compute MAP classification for a new student with: $X_1 = 80, \quad X_2 = 90$ *using Gaussian Naive Bayes. Assume equal class priors: P(Pass) = P(Fail) = 0.5*

Message	**Class**	free	win	click
M1	Spam	2	1	0
M2	Spam	1	2	1
M3	Ham	0	0	1
M4	Ham	0	1	0
M5	Ham	1	0	0

Problem 112 *We classify an SMS as either **Spam (C1)** or **Ham (C2)** based on word frequencies.*
Vocabulary: {*free, win, click*}
For a new message containing the word counts: $(free = 1,\ win = 1,\ click = 0)$
Compute MAP classification using Multinomial Naive Bayes:
$P(Spam) = 0.4,\ P(Ham) = 0.6$
Use Laplace correction (add 1 smoothing).

Problem 113 *Binary feature presence (1 = word present, 0 = not present)* ***Features:*** X_1 *= machine,* X_2 *= learning,* X_3 *= statistics*

Sample	X_1	X_2	X_3	*Class*
S1	*1*	*1*	*0*	*C1*
S2	*1*	*0*	*0*	*C1*
S3	*1*	*1*	*1*	*C1*
S4	*0*	*0*	*1*	*C2*
S5	*0*	*1*	*1*	*C2*

Given new document: $(1, 0, 1)$ *classify using MAP rule.*

3.5 Try it Yourself

Exercise 41 *A logistic regression model is defined as*

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

where $\beta_0 = -3$ *and* $\beta_1 = 0.75$. *Compute* $P(Y = 1|X = 4)$. *(Round off to three decimal places.)*

Exercise 42 *Given* $\log\left(\frac{p}{1-p}\right) = -1 + 0.5x_1 + 0.2x_2$, *find the probability* p *when* $x_1 = 2$ *and* $x_2 = 3$. *(Round off to three decimal places.)*

Exercise 43 *In a logistic regression, $\hat{p} = 0.8$ and the true label $y = 0$. Compute the cross-entropy (log-loss) value for this observation. (Round off to three decimal places.)*

Exercise 44 *For a logistic model $\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x$, the decision boundary occurs at $p = 0.5$. If $\beta_0 = -2.5$ and $\beta_1 = 0.5$, compute the x value at which the model predicts $p = 0.5$.*

Exercise 45 *A logistic regression model predicts the following probabilities for five data points: $[0.9, 0.8, 0.7, 0.2, 0.1]$ The true labels are $[1, 1, 0, 0, 1]$. Compute the total cross-entropy loss. (Round off to three decimal places.)*

Exercise 46 *For the logistic model $P(Y = 1|x) = \frac{1}{1+e^{-(1.5+0.5x)}}$, compute the log-odds and the odds for $x = 4$.*

Exercise 47 *If the logistic regression parameters are estimated as $\beta_0 = 0.8$, $\beta_1 = -0.4$, what is the probability that $Y = 1$ for $x = 3$? (Round off to three decimal places.)*

Exercise 48 *A dataset gives $\hat{p} = 0.6$ for $y = 1$ and $\hat{p} = 0.3$ for $y = 0$. Compute the average log-loss for these two samples. (Round off to three decimal places.)*

Exercise 49 *Given a logistic regression model $\log\left(\frac{p}{1-p}\right) = -2+0.5x_1+0.25x_2$, for $x_1 = 4$ and $x_2 = 6$, find p. (Round off to three decimal places.)*

Exercise 50 *Suppose the model is $\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ with $\beta_0 = -3$, $\beta_1 = 0.8$, and $\beta_2 = -0.4$. If $x_1 = 2.5$ and $x_2 = 1.5$, compute both the odds and probability of $Y = 1$. (Round off to three decimal places.)*

Exercise 51 *For a logistic regression, $\hat{p}_i = [0.9, 0.1, 0.6, 0.4]$ and $y_i = [1, 0, 0, 1]$. Compute the mean cross-entropy loss across all samples. (Round off to three decimal places.)*

Exercise 52 *A model outputs $\hat{p} = 0.85$ for $y = 1$ and $\hat{p} = 0.05$ for $y = 0$. If both are weighted equally, what is the average binary cross-entropy loss? (Round off to three decimal places.)*

Exercise 53 *If the decision threshold is increased from 0.5 to 0.7 in a logistic regression classifier, and for a sample $p = 0.65$, what is the new predicted class label? Represent $Y = 1$ as 1 and $Y = 0$ as 0.*

Exercise 54 *Given that the sigmoid derivative $\sigma'(z) = \sigma(z)[1-\sigma(z)]$, compute $\sigma'(z)$ when $\sigma(z) = 0.8$. (Round off to three decimal places.)*

Exercise 55 *A logistic regression model is defined as $\log\left(\frac{p}{1-p}\right) = -4 + 0.5x$. Find x when $p = 0.9$. (Round off to two decimal places.)*

Exercise 56 *For a dataset with 6 points along line y=x, (0,0),(1,1),..., (5,5), compute the 3-NN Euclidean distance of test point (2,2) to its neighbors.*

Exercise 57 *Compute Manhattan distance between (0,1,2) and (3,4,5).*

Exercise 58 *Dataset points: (0,0),(1,0),(0,1),(1,1). Compute average distance from (0.5,0.5) to its 2 nearest neighbors using Euclidean distance.*

Exercise 59 *Compute distance between points (2,3) and (5,7) using Minkowski distance with $p = 4$.*

Exercise 60 *Test point (1,1) has neighbors (0,0),(2,2),(3,3). Compute predicted value for k-NN regression using $k = 2$ as average of nearest neighbors.*

Exercise 61 *Given points (0,0),(1,1),(4,5),(6,8). Find 1-NN predicted class of point (3,3) using Euclidean distance.*

Exercise 62 *Gaussian NB with 2 features: X1 N(0,1), X2 N(2,4) for class C. Observed X1=1, X2=4. Compute likelihood.*

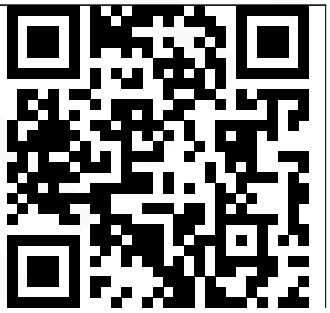
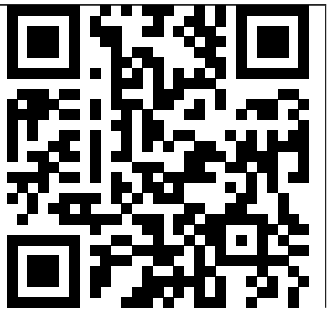
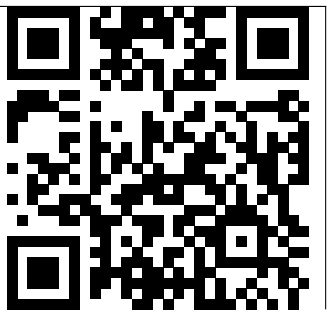
Exercise 63 *MAP prediction: Class priors 0.2,0.3,0.5; likelihoods 0.6,0.4,0.5. Which class predicted?*

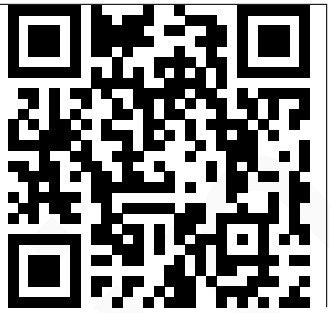
Exercise 64 *In Multinomial NB, smoothing adds 1 to counts. If class word counts are [0,3,2], total words=5, compute smoothed probability for word 1.*

Exercise 65 *Explain why Naive Bayes can perform well even when conditional independence assumption is violated.*

3.6 YouTube Links and QR Codes

Lecture	Details	YouTube Link	QR Code
13	Logistic Regression Explained: Odds, Logit & Classification	https://youtu.be/FKaaKk4OCbI	
14	Problem Solving – Logistic Regression (Solutions 65 - 79)	https://youtu.be/e7BAHUZ7eMs	
15	k-Nearest Neighbors (k-NN): Concepts, Intuition & Examples	https://youtu.be/yANl1SBQB8w	
16	Problem Solving – k-NN Algorithm (Solutions 80 - 92)	https://youtu.be/Uhs6QLruEKA	

17	Naive Bayes Explained in 3 Simple Steps with Example	https://youtu.be/v2F-EWutEA4	
18	MAP Decision Rule (Maximum A Posteriori) Explained with Example	https://youtu.be/S6rGZ45fwzA	
19	Gaussian Naive Bayes Explained with Example	https://youtu.be/7R8gCR4d3XI	
20	Multinomial Naive Bayes Explained: Laplace Smoothing & Example	https://youtu.be/M9YViebrb1w	
21	Bernoulli Naive Bayes Explained with Example	https://youtu.be/1Z305KMo_Ko	

22	Problem Solving – Naive Bayes Classifiers (GNB, MNB, BNB) (Solutions 93 - 113)	https://youtu.be/3w7FO4h34RQ	

Chapter 4

Classification Models - II

4.1 Linear Discrimination Analysis (LDA)

What is LDA?

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction technique used in machine learning, statistics, and pattern recognition. It seeks to find a **linear projection** (a direction or axis) that maximizes the separation between multiple classes while minimizing the spread within each class.
LDA is primarily used for:

- **Feature reduction:** projecting high-dimensional data onto a lower-dimensional subspace.
- **Classification:** improving class separability before applying classifiers such as Logistic Regression or SVM.

Two Perspectives of Linear Discriminant Analysis (LDA)

- **Generative Perspective (Bayesian LDA):**
  This approach treats LDA as a *generative model*, meaning it attempts to learn how each class generates the data. It models the probability distribution of the features for each class, $p(x|C_k)$, and combines it with the prior probability $P(C_k)$ to compute the posterior probability $P(C_k|x)$ using Bayes' theorem. In simple terms, the model first learns the underlying data distribution of each class and then predicts which class is most likely to have produced a new observation.
- **Discriminative Perspective (Fisher's LDA):**
  This approach treats LDA as a *discriminative model*, meaning it focuses on finding a boundary that best separates the classes rather than modeling how the data was generated. Fisher's method seeks a projection direction in which the classes are most distinct — that is, the dis-

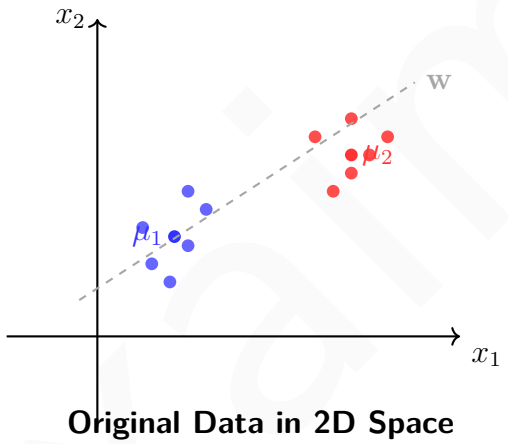
tance between class means is large while the variation within each class is small. It does not model class probabilities explicitly but directly aims to achieve maximum class separability.

Summary:
In essence, the generative form (Bayesian LDA) focuses on how data is produced by each class, while the discriminative form (Fisher's LDA) focuses on how to best separate the classes.

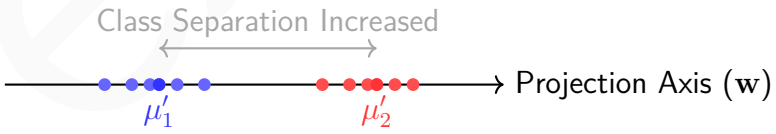
4.2 Fisher's Linear Discriminant Analysis (LDA)

LDA aims to find a direction $\mathbf{w}$ such that, when data points $\mathbf{x}$ are projected onto it, the *between-class variance* is maximized, and the *within-class variance* is minimized.

The following figures illustrate how LDA transforms data by projecting it onto a line that best separates the classes.



Original Data in 2D Space
(Two Overlapping Classes)



Projection of Data onto 1D Subspace (After LDA)

LDA finds a projection direction $\mathbf{w}$ that maximizes class separability. In the top figure, data from two classes overlap in the 2D space. After projection (bottom figure), the classes are well separated in 1D.

Concept of Fisher's LDA

Fisher's Linear Discriminant Analysis (LDA) finds a projection direction (or matrix) that maximizes the separation between classes in a lower-dimensional space.
Goal: Find a projection vector w that projects high-dimensional data $X = (X_1, X_2, \ldots, X_p)^T$ onto a one-dimensional line $y = w^T X$ such that samples from different classes are well-separated while samples within the same class remain close.
Intuition:

- LDA is not just a classifier — it's a dimensionality reduction technique that preserves class separability.
- For two classes, it finds a single direction w maximizing the ratio of between-class variance to within-class variance.
- For K classes, it finds $(K-1)$ optimal directions (columns of W).

4.2.1 Two-Class Fisher LDA: Step-by-Step Derivation

Step 1: Define means and scatter matrices

Let two classes C_1 and C_2 have:

$$\mu_1 = \frac{1}{N_1} \sum_{i \in C_1} x^{(i)}, \quad \mu_2 = \frac{1}{N_2} \sum_{i \in C_2} x^{(i)}$$

The global mean:

$$\mu = \frac{N_1 \mu_1 + N_2 \mu_2}{N_1 + N_2}$$

Within-class scatter:

$$S_W = S_1 + S_2, \quad S_k = \sum_{i \in C_k} (x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T$$

represents how samples of the same class are scattered around their mean.

Between-class scatter:

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

represents how far apart the class means are.

Step 2: Define the Fisher criterion

We project each sample x onto a line:

$$y = w^T x$$

The projected means:

$$m_1 = w^T \mu_1, \quad m_2 = w^T \mu_2$$

The projected within-class variance:

$$s_w^2 = w^T S_W w$$

Fisher's criterion:

$$J(w) = \frac{(m_1 - m_2)^2}{s_w^2} = \frac{w^T S_B w}{w^T S_W w}$$

Goal: Maximize $J(w)$ to find the best discriminant direction.

Step 3: Optimization using generalized eigenvalue problem

Maximizing $J(w)$:

$$\max_w \frac{w^T S_B w}{w^T S_W w}$$

This leads to:

$$S_B w = \lambda S_W w$$

Substitute $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$:

$$(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = \lambda S_W w$$

This gives the solution:

$$w \propto S_W^{-1}(\mu_1 - \mu_2)$$

Interpretation:

- S_W^{-1} "whitens" the data (removes within-class correlation).
- $(\mu_1 - \mu_2)$ gives the direction connecting class means.
- Hence w points in the direction of maximal class separability.

4.2.2 Numerical Example

Example: Two-Class Case

Given:

$$\mu_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad S_W = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Compute:

$$w = S_W^{-1}(\mu_1 - \mu_2)$$

$$S_W^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \Rightarrow w = \frac{1}{3}\begin{bmatrix} -2 \\ -2 \end{bmatrix} \propto \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Decision rule: Project x and classify based on threshold:

$$y = w^T x, \quad \text{assign to } C_1 \text{ if } y < t, \text{ else } C_2$$

where

$$t = \frac{1}{2} w^T (\mu_1 + \mu_2)$$

4.2.3 Multiclass Fisher LDA Derivation

Multiclass Case $(K > 2)$

For K classes, we define:

$$S_W = \sum_{k=1}^{K} \sum_{i \in C_k} (x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T, \quad S_B = \sum_{k=1}^{K} N_k(\mu_k - \mu)(\mu_k - \mu)^T$$

Objective:

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

Optimization leads to the generalized eigenvalue problem:

$$S_B w_i = \lambda_i S_W w_i, \quad i = 1, 2, \ldots, K-1$$

The top $(K-1)$ eigenvectors form the projection matrix:

$$W = [w_1, w_2, \ldots, w_{K-1}]$$

Interpretation:

- Each column of W defines one discriminant axis.
- The data is projected onto the $(K-1)$-dimensional subspace for maximum separability.

4.2.4 Assumptions of Fisher LDA

- Each class is normally distributed (Gaussian).
- All classes share the same covariance matrix Σ.
- Classes are linearly separable in some projection space.
- S_W is nonsingular or regularized.

4.3 Bayesian View of Linear Discriminant Analysis

Step-by-step Bayesian derivation of LDA

Assumption: For each class C_k the class-conditional density is Gaussian with common covariance Σ and mean μ_k:

$$p(x \mid C_k) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\Big(-\tfrac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k) \Big).$$

Step 1: write the log-likelihood (log of class-conditional density).

$$\log p(x \mid C_k) = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| \; - \; \frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k).$$

The first two terms are constants w.r.t. x and k (if Σ is common).
Step 2: expand the quadratic form. Use

$$(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k) = x^\top \Sigma^{-1} x \; - \; 2x^\top \Sigma^{-1}\mu_k \; + \; \mu_k^\top \Sigma^{-1}\mu_k.$$

Substitute into the log-likelihood:

$$\log p(x \mid C_k) = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| \; - \; \frac{1}{2}x^\top\Sigma^{-1}x \; + \; x^\top\Sigma^{-1}\mu_k \; - \; \frac{1}{2}\mu_k^\top\Sigma^{-1}\mu_k.$$

Step 3: form the log-posterior (discriminant) by adding the log prior. Let the class prior be $P(C_k) = \pi_k$. Define

$$\delta_k(x) \; = \; \log p(x \mid C_k) + \log \pi_k.$$

Hence

$$\delta_k(x) = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}x^\top\Sigma^{-1}x + x^\top\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^\top\Sigma^{-1}\mu_k + \log\pi_k.$$

Step 4: drop terms that are common to all classes. The terms $-\frac{p}{2}\log(2\pi)$, $-\frac{1}{2}\log|\Sigma|$ and $-\frac{1}{2}x^\top\Sigma^{-1}x$ do not depend on k, so they cancel when comparing $\delta_i(x)$ vs $\delta_j(x)$. Removing those common terms (or equivalently adding a constant) yields the simplified discriminant:

$$\boxed{\delta_k(x) \; = \; x^\top\Sigma^{-1}\mu_k \; - \; \tfrac{1}{2}\mu_k^\top\Sigma^{-1}\mu_k \; + \; \log\pi_k}$$

This is an affine (linear + constant) function of x.
Step 5: write the pairwise decision boundary. To find the boundary between class C_i and C_j set $\delta_i(x) = \delta_j(x)$. Rearranging:

$$x^\top\Sigma^{-1}(\mu_i - \mu_j) \; = \; \tfrac{1}{2}\left(\mu_i^\top\Sigma^{-1}\mu_i - \mu_j^\top\Sigma^{-1}\mu_j\right) \; - \; \log\frac{\pi_i}{\pi_j}.$$

This is a linear equation in x (a hyperplane), so the decision boundary is linear.
Remark: If covariances are *not* equal (Σ_k differ across classes), the quadratic term $-\frac{1}{2}x^\top\Sigma_k^{-1}x$ will *not* cancel across classes and the resulting boundary is generally *quadratic* — this gives Quadratic Discriminant Analysis (QDA) instead of LDA.
Decision rule: Assign x to the class C_k with the largest $\delta_k(x)$.

4.3.1 Assumptions in Bayesian LDA

- $p(x|C_k)$ is Gaussian for all k.
- All classes share identical covariance Σ.
- Priors π_k are known or estimated from sample proportions.
- The decision boundaries are linear in x.

4.3.2 Example: Bayesian LDA (Two Classes)

Numerical Example

Let:

$$\mu_1 = (1, 2), \quad \mu_2 = (3, 4), \quad \Sigma = I, \quad \pi_1 = \pi_2 = 0.5$$

Then:

$$\delta_k(x) = x^T\mu_k - \frac{1}{2}\mu_k^T\mu_k + \log\pi_k$$

Boundary:

$$x^T(\mu_1 - \mu_2) = \frac{1}{2}(\mu_1^T\mu_1 - \mu_2^T\mu_2)$$

Substitute:

$$x^T[-2, -2] = \frac{1}{2}(5 - 25) \Rightarrow x_1 + x_2 = 5$$

Hence, samples with $x_1 + x_2 > 5$ belong to C_2; otherwise C_1.

4.4 Problems

Problem 114 *Fisher's Linear Discriminant is derived by maximizing which ratio?*

(A) Within-class variance to between-class variance

(B) Between-class variance to within-class variance

(C) Total variance to between-class variance

(D) Total variance to within-class variance

Problem 115 *The optimal projection vector* $\mathbf{w}$ *in Fisher's LDA for two classes is:*

(A) $\mathbf{w} = \Sigma_W^{-1}(\mu_1 + \mu_2)$

(B) $\mathbf{w} = \Sigma_W^{-1}(\mu_1 - \mu_2)$

(C) $\mathbf{w} = (\mu_1 - \mu_2)$

(D) $\mathbf{w} = \Sigma_B^{-1}(\mu_1 - \mu_2)$

Problem 116 *Which of the following statements about Fisher's criterion is true?*

(A) It maximizes both within-class and between-class variance.

(B) It minimizes between-class variance and maximizes within-class variance.

(C) It maximizes the separation between projected class means and minimizes within-class spread.

(D) It depends only on prior probabilities.

Problem 117 *In Fisher's LDA, the between-class scatter matrix* S_B *for two classes is defined as:*

(A) $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$

(B) $S_B = (\mu_1 + \mu_2)(\mu_1 + \mu_2)^T$

(C) $S_B = \Sigma_1 + \Sigma_2$

(D) $S_B = (\mu_1 - \mu)(\mu_1 - \mu)^T + (\mu_2 - \mu)(\mu_2 - \mu)^T$

Problem 118 *If within-class scatter* S_W *is large, what does it imply?*

(A) Classes are well separated.

(B) *There is large overlap between classes.*

(C) *Between-class distance is high.*

(D) *LDA performs better.*

Problem 119 *Which of the following is true for Fisher's LDA with two Gaussian classes having equal covariance?*

(A) *The decision boundary is linear.*

(B) *The decision boundary is quadratic.*

(C) *The projection direction depends on class priors.*

(D) *The within-class scatter matrix becomes singular.*

Problem 120 *Fisher's criterion* $J(\mathbf{w}) = \dfrac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$ *is invariant to:*

(A) *Scaling of* $\mathbf{w}$

(B) *Rotation of* $\mathbf{w}$

(C) *Addition of a bias term*

(D) *Translation of data*

Problem 121 *Which of the following assumptions is essential for Fisher's LDA to coincide with the Bayes optimal classifier?*

(A) *Classes are Gaussian with identical covariance matrices.*

(B) *Classes are Gaussian with different covariances.*

(C) *Data are linearly separable.*

(D) *Covariance matrices are diagonal.*

Problem 122 *The rank of* S_B *for* K *classes is at most:*

(A) K

(B) $K - 1$

(C) $K + 1$

(D) *Depends on data dimension*

Problem 123 *In multi-class Fisher LDA, the number of discriminant vectors obtained is:*

(A) K

(B) $K - 1$

(C) *Equal to the number of features*

(D) $\min(K-1, D)$ *where* D *is feature dimension*

Problem 124 *Select all correct statements about Fisher's Linear Discriminant.*

(A) It performs supervised dimensionality reduction.

(B) It can be derived by maximizing Fisher's ratio.

(C) It always guarantees perfect class separation.

(D) It leads to a linear decision boundary when class covariances are equal.

Problem 125 *Which statements are true about the Fisher projection direction* $\mathbf{w}$*?*

(A) $\mathbf{w}$ *lies in the span of class means.*

(B) $\mathbf{w}$ *maximizes the distance between projected class means.*

(C) $\mathbf{w}$ *minimizes projected within-class variance.*

(D) $\mathbf{w}$ *depends only on* μ_1, μ_2 *and* Σ_W*.*

Problem 126 *In two-class Fisher LDA, changing class priors affects:*

(A) The projection direction $\mathbf{w}$

(B) The threshold of decision boundary

(C) The slope of discriminant line

(D) The location (intercept) of boundary

Problem 127 *Given* $\mu_1 = (1, 2)$*,* $\mu_2 = (4, 3)$*, and* $S_W = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$*, compute* $\mathbf{w}$ *up to a scaling constant.*

Problem 128 *For two classes,* $\mu_1 = (2, 1)$*,* $\mu_2 = (5, 4)$*, and* $S_W = I_2$*, find the unit vector* $\mathbf{w}$ *used for Fisher's projection.*

Problem 129 *If projected class means are* $m_1 = 1.0$ *and* $m_2 = 3.5$ *and the within-class variance on projection is* 0.25*, compute Fisher's criterion* J*.*

Problem 130 *Two classes have means* $\mu_1 = (0, 0)$*,* $\mu_2 = (2, 0)$*, and shared covariance* $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$*.*

Compute $\mathbf{w}$*.*

Problem 131 *Two classes have means* $\mu_1 = (2, 3)$ *and* $\mu_2 = (6, 7)$ *and equal priors. Find the midpoint of the decision boundary if covariance is identity.*

Problem 132 *For 3 classes, the total scatter matrix* $S_T = S_W + S_B$*. If trace*$(S_W) = 10$ *and trace*$(S_B) = 5$*, find trace*(S_T)*.*

Problem 133 *Two-class Fisher LDA projects 2D data onto 1D. If the projected variance is 1.5 and mean difference is 3.0, compute $J(\mathbf{w})$.*

Problem 134 *A two-class LDA model has features X_1 and X_2. The class means are $\mu_1 = (2, 3)$, $\mu_2 = (5, 7)$ and the shared covariance matrix is $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}$. Which of the following statements is true about the decision boundary?*

(A) It is a linear function of X_1 and X_2.

(B) It is a quadratic function of X_1 and X_2.

(C) It passes through the midpoint of μ_1 and μ_2 if priors are equal.

(D) It is always orthogonal to the line connecting μ_1 and μ_2.

Problem 135 *In LDA, if the covariance matrices of two classes are different, then:*

(A) LDA still yields the optimal linear boundary.

(B) The decision boundary becomes quadratic (QDA).

(C) The classes cannot be separated.

(D) Maximum likelihood estimation fails.

Problem 136 *A dataset has three classes with equal priors. Which of the following statements about LDA is correct?*

(A) LDA projects the data to a space of dimension 2 or less.

(B) LDA can produce more than two decision boundaries.

(C) LDA always maximizes overall accuracy.

(D) The number of linear discriminants is at most $K - 1$ for K classes.

Problem 137 *For a two-class LDA, the discriminant function for class 1 is $\delta_1(x) = x^T\Sigma^{-1}\mu_1 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \log \pi_1$. If $\pi_1 = \pi_2$, which term affects only the intercept of the decision boundary?*

(A) $x^T\Sigma^{-1}\mu_1$

(B) $-\frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1$

(C) $\log \pi_1$

(D) Both (b) and (c)

Problem 138 *Which of the following assumptions is NOT required for standard LDA?*

(A) Features are Gaussian within each class.

(B) Covariance matrices are equal across classes.

(C) *Classes have equal prior probabilities.*

(D) *Observations are independent.*

Problem 139 *Consider two features X_1 and X_2. LDA finds a projection vector w for classification. Which of the following statements is correct?*

(A) *w is proportional to $\Sigma^{-1}(\mu_1 - \mu_2)$.*

(B) *w is proportional to $(\mu_1 - \mu_2)$.*

(C) *w maximizes the within-class variance.*

(D) *w is orthogonal to the vector connecting class means.*

Problem 140 *If two classes have identical means $\mu_1 = \mu_2$ but different covariance matrices, then:*

(A) *LDA cannot separate the classes.*

(B) *LDA decision boundary is still linear.*

(C) *QDA should be used.*

(D) *Linear regression will outperform LDA.*

Problem 141 *In LDA, increasing the number of features without increasing sample size may lead to:*

(A) *Improved generalization.*

(B) *Singular covariance matrix and unstable estimates.*

(C) *Decreased within-class variance.*

(D) *Guaranteed increase in R^2.*

Problem 142 *For a two-class LDA with class means $\mu_1 = (2, 1)$ and $\mu_2 = (4, 3)$, and shared covariance matrix $\Sigma = I_2$, compute the linear discriminant score $\delta_1(x)$ for the point $x = (3, 2)$ assuming equal prior probabilities.*

Problem 143 *Three-class LDA: class means $\mu_1 = (0, 0), \mu_2 = (1, 0), \mu_3 = (0, 1)$, $\Sigma = I_2$. Compute the discriminant function value for $x = (0.5, 0.5)$ for class 1.*

4.5 Try it Yourself

Exercise 66 *Two-class LDA: $\mu_1 = (1, 2), \mu_2 = (3, 1)$, $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, prior $\pi_1 = 0.6, \pi_2 = 0.4$. Compute the threshold for the linear boundary.*

Exercise 67 *Two-class LDA with features (X_1, X_2): class means $(1,1)$ and $(4,2)$, $\Sigma = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}$.*

Compute the orientation (slope) of the linear boundary.

Exercise 68 *Two-class LDA: $\mu_1 = (0,0)$, $\mu_2 = (2,2)$, $\Sigma = I_2$, equal priors. Compute the classification of $x = (1, 0.5)$.*

Exercise 69 *Three-class LDA: $\mu_1 = (0,0)$, $\mu_2 = (2,0)$, $\mu_3 = (1,1)$, $\Sigma = I_2$, equal priors. Compute which class $x = (1, 0.5)$ is assigned to.*

4.6 YouTube Links and QR Codes

Lecture	Details	YouTube Link	QR Code
23	Linear Discriminant Analysis Explained with Example	https://youtu.be/8ee6-HSKwQY	
24	Problem Solving – Fisher Linear Discriminant Analysis (LDA) (Solutions 114 - 133)	https://youtu.be/q7ouYk2Sk5c	
25	Linear Discriminant Analysis (Bayesian Overview) Explained	https://youtu.be/z9lJVmtuo0Y	
26	Problem Solving – Bayesian Linear Discriminant Analysis (Solutions 134 - 143)	https://youtu.be/oYJAywqVvnE	

Chapter 5

Model Evaluation and Generalization

5.1 Cross-Validation Methods

Cross-validation is a statistical method used to estimate the performance of machine learning models on unseen data. It is widely used for model evaluation and hyperparameter tuning, especially when the dataset is limited.

5.1.1 Hold-Out Validation

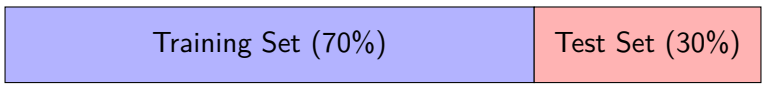
Concept: Hold-out validation splits the dataset into two disjoint subsets:

- **Training set:** used to fit the model.
- **Test set:** used to evaluate the model performance.

Procedure:

1. Randomly split the data, e.g., 70% training and 30% test.
2. Train the model on the training set.
3. Compute evaluation metrics (MSE, accuracy, etc.) on the test set.

Advantages: Simple, fast. **Disadvantages:** Sensitive to the random split, may not utilize all data for training.



Dataset split in Hold-Out Validation.

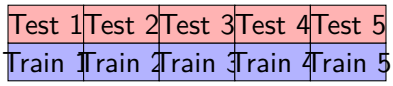
5.1.2 Leave-One-Out (LOO) Cross-Validation

Concept: LOO is an extreme case of k-fold cross-validation where $k = n$ (number of samples). Each sample is used once as a test set, and the model is trained on the remaining $n - 1$ samples.

Procedure:

1. For each sample $i = 1, \ldots, n$, train the model on all other samples except i.
2. Test the model on the excluded sample.
3. Average the evaluation metrics across all iterations.

Advantages: Uses maximum data for training, low bias. **Disadvantages:** Computationally expensive for large datasets.



LOO Cross-Validation: Each sample is tested once.

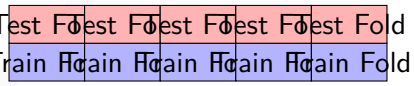
5.1.3 k-Fold Cross-Validation

Concept: In k-fold cross-validation, the dataset is divided into k roughly equal folds. Each fold is used once as a test set, and the remaining $k - 1$ folds are used for training.

Procedure:

1. Split the dataset into k folds.
2. For $i = 1$ to k, train on $k - 1$ folds and test on the i-th fold.
3. Average the evaluation metrics over all folds.

Advantages: Reduces variance of the performance estimate, uses all data for training and testing. **Choice of k:** $k = 5$ or 10 are commonly used.



5-Fold Cross-Validation: Each fold is tested once.

5.1.4 Model Selection and Hyperparameter Tuning

Concept: Cross-validation is extensively used to select the best model or hyperparameters, e.g., regularization strength, number of neighbors (k in k-NN), or depth of decision trees.

Procedure:

1. Define a grid of hyperparameter values.
2. For each combination, perform k-fold CV.
3. Select the combination with the best average performance.

Reasoning: Using cross-validation prevents overfitting on a single validation set and provides a more robust estimate of model performance.

Examples: Cross-Validation Methods

Example 34:

Hold-Out Validation Example: House Prices
A dataset contains 50 samples of house prices with features: area, bedrooms, age. We split it 70% training and 30% test. Train a linear regression model and evaluate MSE on the test set: MSE $= 25$.
Interpretation: Hold-out validation gives an estimate of test error. The result may vary if the random split changes.

Example 35:

Leave-One-Out Cross-Validation Example
Dataset: $X = [1, 2, 3, 4]$, $Y = [2, 3, 2.5, 5]$.
Perform LOO CV for linear regression:

1. Iteration 1: Train on $\{2, 3, 4\}$, test on 1 $\rightarrow$ Predicted $\hat{Y}_1 = 2.1$, squared error $= 0.01$
2. Iteration 2: Train on $\{1, 3, 4\}$, test on 2 $\rightarrow$ Predicted $\hat{Y}_2 = 2.9$, squared error $= 0.01$
3. Iteration 3: Train on $\{1, 2, 4\}$, test on 3 $\rightarrow$ Predicted $\hat{Y}_3 = 2.7$, squared error $= 0.04$
4. Iteration 4: Train on $\{1, 2, 3\}$, test on 4 $\rightarrow$ Predicted $\hat{Y}_4 = 4.8$, squared error $= 0.04$

LOO CV error: $CV_{LOO} = \frac{0.01+0.01+0.04+0.04}{4} = 0.025$

Example 36:

k-Fold Cross-Validation Example
Dataset: 12 samples. Perform 4-fold CV for a k-NN model ($k = 3$):

1. Split dataset into 4 folds of 3 samples each.
2. Iteration 1: Train on folds 2-4, test fold 1 $\rightarrow$ accuracy $= 0.67$
3. Iteration 2: Train on folds 1,3,4, test fold 2 $\rightarrow$ accuracy $= 0.67$
4. Iteration 3: Train on folds 1,2,4, test fold 3 $\rightarrow$ accuracy $= 1.0$
5. Iteration 4: Train on folds 1-3, test fold 4 $\rightarrow$ accuracy $= 0.67$

4-fold CV accuracy: $\frac{0.67+0.67+1.0+0.67}{4} = 0.7525$

Example 37:

Hyperparameter Tuning using k-Fold CV
We want to choose k in k-NN for a dataset of 100 samples. Test $k = 1, 3, 5, 7$ using 5-fold CV:

- $k = 1$: average accuracy $= 0.78$
- $k = 3$: average accuracy $= 0.82$
- $k = 5$: average accuracy $= 0.81$
- $k = 7$: average accuracy $= 0.79$

Best choice: $k = 3$ because it gives the highest cross-validation accuracy.

Interpretation: k-fold CV helps select hyperparameters while avoiding overfitting.

Example 38:

LOO vs Hold-Out Comparison
Dataset: 10 samples for linear regression.

- Hold-Out: Train 7, Test 3 $\rightarrow$ test MSE $= 10.5$
- LOO: Compute prediction for each sample leaving it out $\rightarrow$ average MSE $= 9.8$

Observation: LOO provides a more stable estimate because it uses almost all data for training, whereas hold-out can vary depending on the random split. However, LOO is more computationally expensive.

5.2 ROC & AUC

The **ROC curve** (Receiver Operating Characteristic curve) is a standard way to visualize and evaluate the performance of a binary classifier across all possible decision thresholds. The curve plots the *True Positive Rate* (TPR) on the vertical axis against the *False Positive Rate* (FPR) on the horizontal axis. The **AUC** (Area Under the ROC Curve) is a single scalar summarizing the ROC; it ranges from 0 to 1. Higher AUC indicates better ability to separate positive and negative classes.

Confusion matrix and rates

A confusion matrix for a binary classifier:

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

From this we define:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{(True Positive Rate, a.k.a. Recall)}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \qquad \text{(False Positive Rate, a.k.a. False alarm rate)}$$

What is a decision threshold?

A probabilistic classifier outputs a score $s(x) \in [0, 1]$. To obtain labels we pick a threshold τ and predict positive if $s(x) \geq \tau$. Different choices of τ produce different confusion matrices and thus different (TPR,FPR) pairs. The ROC curve collects those pairs for all thresholds.

ROC curve and AUC (intuition)

- Each point on the ROC corresponds to a threshold and shows the trade-off between detection rate (TPR) and false-alarm rate (FPR).
- The **AUC** is the area under the ROC curve. It has the following useful interpretation:

  $\mathrm{AUC} = P\big(s(X_+) > s(X_-)\big)$ (probability a random positive ranks above a random negative).

- Values: AUC $=$ 1.0 (perfect ranking), AUC $=$ 0.5 (random), AUC ¡ 0.5 (means ranking reversed).

Worked numeric example (spam detector)

We reuse the concrete threshold outcomes described earlier (counts are not all given; we present the rates you provided):

At three decision thresholds we observed:

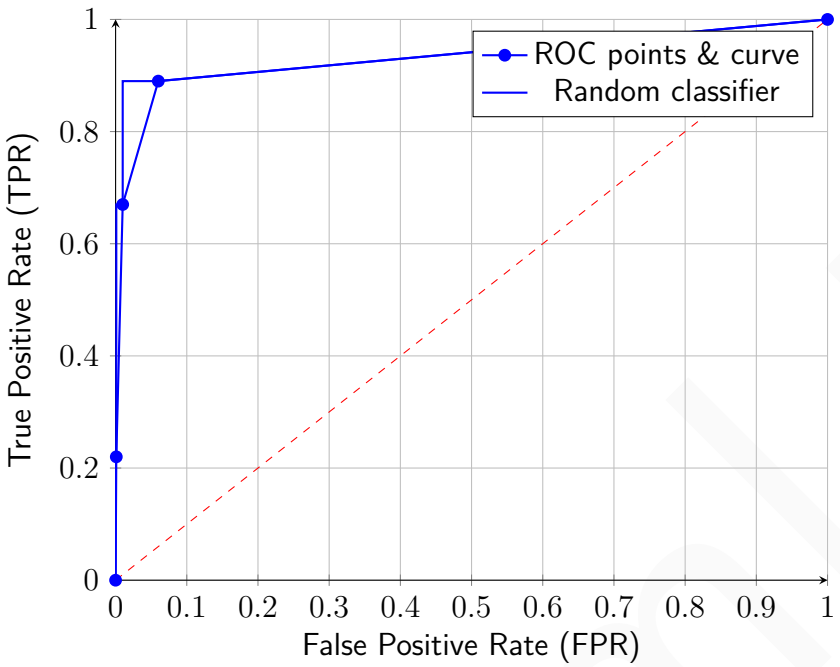
Threshold	$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$	$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$
0.50	$800/(800+100) = 0.89$	$500/(500+8600) = 0.06$
0.80	$600/(600+300) = 0.67$	$100/(100+9000) = 0.01$
0.95	$200/(200+700) = 0.22$	$10/(10+9090) = 0.001$

Interpretation:

- At $\tau = 0.50$ the detector is lenient: high recall (0.89) but more false alarms (FPR=0.06).
- At $\tau = 0.95$ the detector is strict: low recall (0.22) and almost no false alarms (FPR=0.001).

Plotting the ROC curve (from these points)

To draw the ROC we plot the FPR on the x-axis and TPR on the y-axis for each threshold and connect the points (also include $(0,0)$ and $(1,1)$).



Note: the curve above is drawn from the listed (FPR,TPR) pairs; with more threshold values the curve becomes smoother.

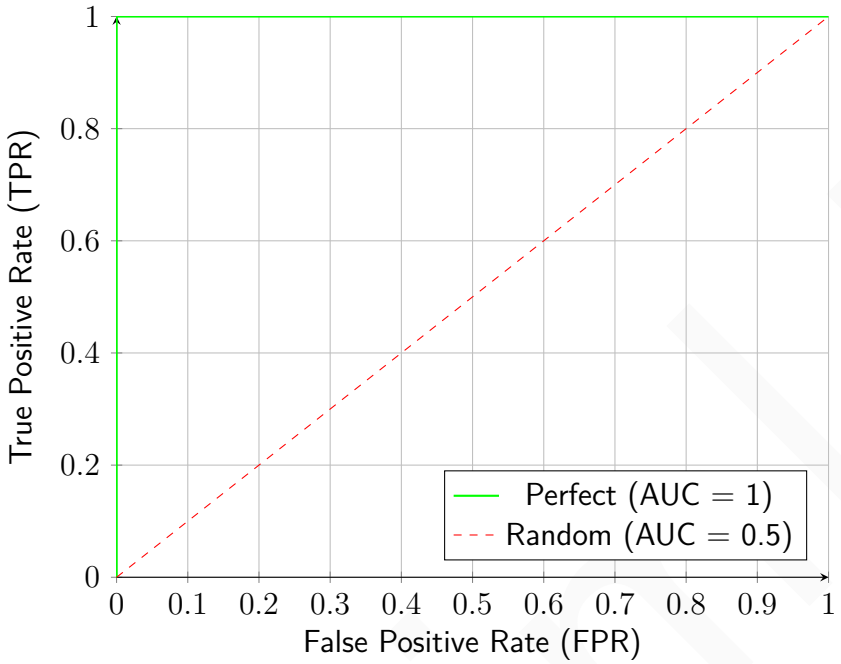
Computing AUC (brief)

A common numerical method is the trapezoidal rule on the ROC points ordered by FPR:

$$\mathrm{AUC} \approx \sum_i (x_{i+1} - x_i)\,\frac{y_i + y_{i+1}}{2},$$

where x_i are FPRs and y_i are the corresponding TPRs. Use sufficiently many thresholds for a good estimate.

Perfect and random classifiers (illustration)



Practical notes

- **AUC is threshold-independent:** it summarizes ranking quality for all thresholds.
- **Use-case sensitivity:** if the positive class is rare, Precision–Recall curves may be more informative than ROC.
- **Interpretation:** AUC $= 0.5$ means no discrimination (random), AUC close to 1.0 means excellent discrimination.
- **Choice of threshold:** AUC does not choose a threshold — select τ based on application-specific trade-offs (costs of FP vs FN).

Conclusion

ROC visualizes the trade-off between detecting positives and producing false alarms as the decision threshold varies. AUC condenses this visualization into a single number expressing the classifier's overall ranking ability.

5.3 Bias–Variance Trade-off

5.3.1 Understanding the Trade-off

In supervised learning, our goal is to build a model that generalizes well to unseen data. The performance of a model on unseen data can be decomposed into three components: bias, variance, and irreducible error (noise).

$$\text{Expected Test Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Each term has an intuitive meaning:

- **Bias:** Error due to simplifying assumptions made by the model.
- **Variance:** Error due to sensitivity of the model to small fluctuations in training data.
- **Irreducible Error:** Noise inherent in the data that no model can explain.

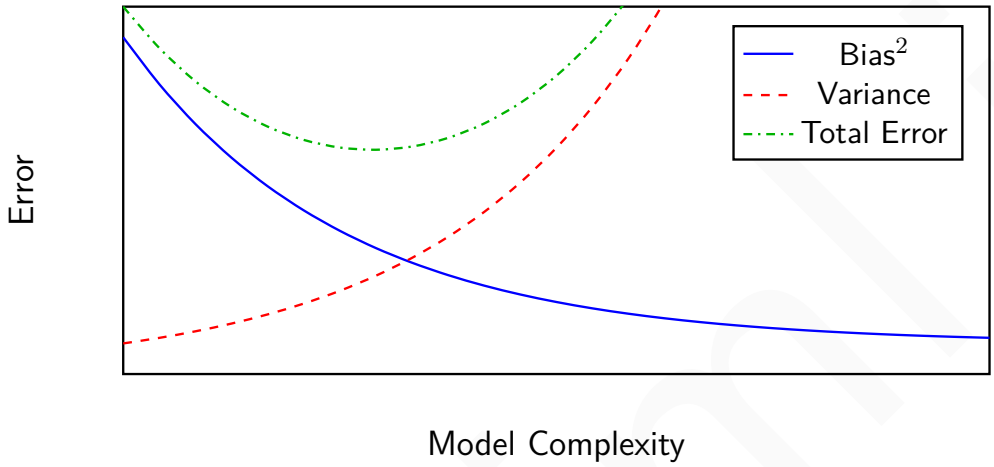


Figure: The bias–variance trade-off. As model complexity increases, bias decreases but variance increases. The total error is minimized at an optimal complexity.

Bias

Bias measures the error introduced by approximating a real-world problem (which may be extremely complex) by a simplified model. A high-bias model makes strong assumptions about the data and fails to capture important regularities.

$$\text{Bias}^2 = [E(\hat{f}(x)) - f(x)]^2$$

Interpretation: It represents how far the average prediction of the model is from the true function $f(x)$.

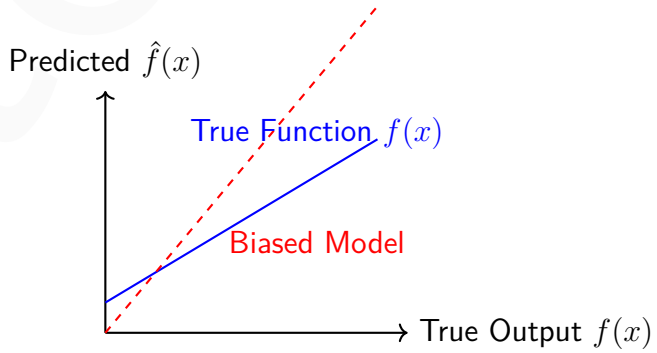


Figure: A biased model (red) systematically underestimates the true relationship (blue).

Variance

Variance measures how much the model's predictions would vary if we trained it on different samples from the same population.

$$\text{Var}[\hat{f}(x)] = E[(\hat{f}(x) - E[\hat{f}(x)])^2]$$

Interpretation: High-variance models fit training data too closely and perform poorly on unseen data. They have high sensitivity to random noise in the training set.

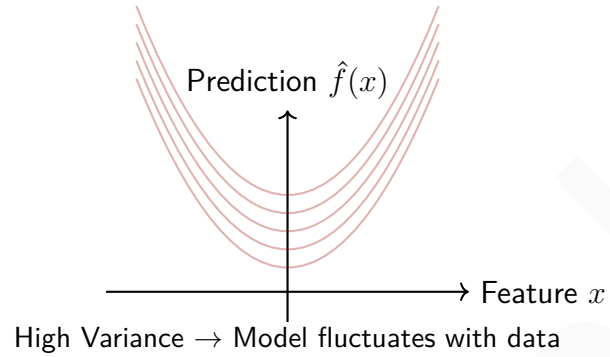


Figure: High-variance models change drastically with different data samples.

Bias–Variance Trade-off Summary

- Increasing model complexity (e.g., higher polynomial degree) reduces bias but increases variance.
- Decreasing model complexity increases bias but reduces variance.
- The goal is to find the sweet spot minimizing total expected error.

5.3.2 Mathematical Derivation

Suppose $Y = f(X) + \varepsilon$, where $E[\varepsilon] = 0$, $\text{Var}(\varepsilon) = \sigma^2$. We predict Y using an estimator $\hat{f}(X)$ obtained from data.

$$E[(Y - \hat{f}(X))^2] = [E[\hat{f}(X)] - f(X)]^2 + E[(\hat{f}(X) - E[\hat{f}(X)])^2] + \sigma^2$$

$$\Rightarrow \text{Expected Test MSE} = \underbrace{\text{Bias}^2}_{\text{Systematic Error}} + \underbrace{\text{Variance}}_{\text{Sensitivity to Data}} + \underbrace{\sigma^2}_{\text{Noise}}$$

—

Example 39:

Comparing Linear vs Polynomial Regression Models
Consider data generated from $Y = X^2 + \varepsilon$.

- A **linear regression model** $Y = \beta_0 + \beta_1 X$ will have *high bias* (it cannot capture curvature).
- A **degree-10 polynomial** will have *low bias* but *very high variance*.

Cross-validation typically identifies an intermediate degree (say, 3 or 4) that minimizes total test error.

—

Example 40:

Effect of Sample Size
When training a model on a small dataset:

- Variance increases — the model depends heavily on limited data.
- Bias may remain unchanged if model class is fixed.

Thus, increasing dataset size reduces variance but not bias.

—

Key Takeaways

- High bias $\rightarrow$ underfitting (model too simple).
- High variance $\rightarrow$ overfitting (model too complex).
- Bias–variance trade-off can be visualized as an U-shaped total error curve.
- Techniques like **regularization** (Ridge/Lasso) and **cross-validation** help control this trade-off.

5.4 Problems

Problem 144 *A dataset of 120 samples is split into 80% training and 20% test. The model trained achieves 90% accuracy on training and 75% on test. Which statement is most likely correct?*

(A) Model is underfitting

(B) Model is overfitting

(C) Training set is too small

(D) Test set is too large

Problem 145 *In leave-one-out cross-validation (LOO-CV) for n samples:*

(A) Each iteration uses $n - 1$ samples for training

(B) Each iteration tests on a single sample

(C) Total number of iterations $= n$

(D) Total number of iterations $= n/2$

Problem 146 *Which of the following are true about k-fold cross-validation?*

(A) Reduces variance of performance estimate compared to hold-out

(B) Can be used for hyperparameter tuning

(C) Training data in each fold is disjoint from test fold

(D) Always results in lower bias than LOO

Problem 147 *A 5-fold CV is applied to a dataset of 50 samples for a linear regression model. If one fold has unusually large errors, what is the likely effect on the average CV error?*

(A) Average CV error increases

(B) Average CV error decreases

(C) CV error remains unaffected

(D) Training error decreases

Problem 148 *In hyperparameter tuning using k-fold CV, which statements are correct?*

(A) All combinations of hyperparameters are evaluated

(B) CV error is computed for each combination

(C) Best combination minimizes training error

(D) Best combination maximizes CV accuracy

Problem 149 *LOO CV is preferred over hold-out when:*

(A) Dataset is very small

(B) Computational resources are limited

(C) Dataset is very large

(D) We want low-bias error estimate

Problem 150 *Which of the following statements are true?*

(A) k-fold CV uses all samples for training at some point

(B) LOO CV uses less data for training in each iteration than hold-out

(C) Hold-out validation can give high variance estimates

(D) k-fold CV can never overestimate model performance

Problem 151 *A dataset has 200 samples. 10-fold CV is applied to a k-NN model. Each fold contains how many test samples?*

(A) 20

(B) 10

(C) 15

(D) 25

Problem 152 *Which statement about bias-variance tradeoff in cross-validation is correct?*

(A) LOO CV has higher variance but lower bias than 5-fold CV

(B) LOO CV has lower variance and higher bias than 5-fold CV

(C) Hold-out validation always has lower variance than k-fold

(D) 10-fold CV gives higher bias than hold-out

Problem 153 *In 4-fold CV, which of the following are true?*

(A) Each sample is used exactly once for testing

(B) Each sample is used exactly three times for training

(C) Number of iterations = 4

(D) Average training set size per iteration = 75% of total

Problem 154 *Which of the following statements about ROC curves is/are TRUE? (MSQ)*

(A) ROC curve is threshold-independent.
(B) ROC curve plots Precision vs Recall.
(C) AUC = 0.5 means classifier is no better than random.
(D) ROC is affected by class imbalance.

Problem 155 *A classifier produces the following scores for 3 positive and 3 negative samples:*

$$\textit{Positives: } 0.9,\ 0.7,\ 0.6 \textit{Negatives: } 0.8,\ 0.4,\ 0.1$$

Compute AUC using the "probability of ranking" interpretation:

$$\textit{AUC} = P(\textit{score(pos)} > \textit{score(neg)})$$

(NAT)

Problem 156 *Consider two classifiers A and B. Their AUCs are:*

$$AUC_A = 0.92, \qquad AUC_B = 0.78$$

Which statements are TRUE? (MSQ)

(A) Classifier A has better separability.
(B) Classifier B has better TPR at all thresholds.
(C) Classifier A may still have worse accuracy at a given threshold.
(D) AUC alone is sufficient to judge classifier performance.

Problem 157 *The ROC curve of a classifier lies completely below the diagonal line (random classifier). Which of the following is TRUE? (MCQ)*

(A) $AUC < 0.5$
(B)$AUC = 0.5$
(C)$AUC > 0.5$
(D) ROC curve cannot lie below diagonal

Problem 158 *Given the following threshold results:*

t	*TPR*	*FPR*
0.9	0.10	0.00
0.7	0.40	0.10
0.5	0.70	0.20
0.3	0.90	0.40

Compute AUC using trapezoidal rule. (NAT)

Problem 159 *Which of the following plots is used in ROC analysis? (MCQ)*

(A) TPR vs FPR
(B) Precision vs Recall
(C) Accuracy vs Threshold
(D) Loss vs Epoch

Problem 160 *A classifier produces the following:*

$$TPR = 1, \qquad FPR = 1$$

Which statement describes the classifier? (MCQ)
(A) Perfect classifier (B) Always predicts positive (C) Always predicts negative (D) AUC = 1

Problem 161 *In supervised learning, which expression correctly decomposes the expected test MSE?*

(A) $E[(Y - \hat{f}(X))^2] = \text{Bias} + \text{Variance} + \text{Noise}$

(B) $E[(Y - \hat{f}(X))^2] = \text{Bias}^2 + \text{Variance} + \text{Noise}$

(C) $E[(Y - \hat{f}(X))^2] = \text{Bias}^2 + \text{Variance}$

(D) $E[(Y - \hat{f}(X))^2] = \text{Bias} + \text{Variance}$

Problem 162 *A regression model that is too simple for the data is most likely to exhibit:*

(A) High bias and low variance

(B) Low bias and high variance

(C) Low bias and low variance

(D) High bias and high variance

Problem 163 *Which of the following statements about bias and variance is correct?*

(A) Increasing model complexity increases bias

(B) *Increasing model complexity increases variance*

(C) *Reducing regularization increases bias*

(D) *Bias and variance are independent of model complexity*

Problem 164 *A polynomial regression model has* $(\text{Bias}^2, \text{Variance})$ *values as follows: Degree 1* $\rightarrow$ *(25, 4), Degree 3* $\rightarrow$ *(9, 9), Degree 5* $\rightarrow$ *(4, 25). For which degree is the total error minimum?*

(A) *Degree 1*

(B) *Degree 3*

(C) *Degree 5*

(D) *All have equal error*

Problem 165 *A model achieves perfect accuracy on the training data but performs poorly on test data. This model has:*

(A) *High bias*

(B) *High variance*

(C) *Both high bias and variance*

(D) *Low bias and low variance*

Problem 166 *When Ridge Regression is applied instead of OLS, the general effect on bias and variance is:*

(A) *Bias increases, variance decreases*

(B) *Bias decreases, variance increases*

(C) *Both bias and variance decrease*

(D) *Bias decreases, variance constant*

Problem 167 *Which of the following combinations correctly describes overfitting and underfitting?*

(A) *Overfitting* $\rightarrow$ *High bias, Underfitting* $\rightarrow$ *Low variance*

(B) *Overfitting* $\rightarrow$ *High variance, Underfitting* $\rightarrow$ *High bias*

(C) *Overfitting* $\rightarrow$ *Low variance, Underfitting* $\rightarrow$ *High variance*

(D) *Overfitting* $\rightarrow$ *High bias, Underfitting* $\rightarrow$ *High bias*

Problem 168 *Increasing training data while keeping model complexity constant generally:*

(A) *Increases variance*

(B) *Increases bias*

(C) *Reduces variance*

(D) *Has no effect on variance*

Problem 169 *Which of the following is true about the bias–variance trade-off?*

(A) *Bias decreases and variance increases with complexity*

(B) *Variance decreases and bias increases with complexity*

(C) *Both bias and variance increase with complexity*

(D) *Bias and variance are independent of complexity*

Problem 170 *Which of the following statements are true?*

(A) *Regularization helps control overfitting by increasing bias slightly*

(B) *Bias and variance can be minimized simultaneously*

(C) *Ensemble methods like bagging reduce variance*

(D) *Increasing model complexity always reduces test error*

Problem 171 *A high-variance model can be improved by:*

(A) *Reducing model complexity*

(B) *Collecting more training samples*

(C) *Using L2 regularization*

(D) *All of the above*

Problem 172 *Under which condition is the training error equal to the test error?*

(A) *Model perfectly generalizes*

(B) *Model is overfitting*

(C) *Model is underfitting*

(D) *Model has high variance*

Problem 173 *Which model adjustment primarily reduces bias?*

(A) *Adding higher-order polynomial features*

(B) *Increasing regularization parameter*

(C) *Removing features*

(D) *Reducing dataset size*

Problem 174 *If the model fits every training point exactly but generalizes poorly, it suffers from:*

(A) *High bias*

(B) *High variance*

(C) *Low bias, low variance*

(D) *Noise dominance*

5.5 Try it Yourself

Exercise 70 *Dataset: 15 samples. 5-fold CV squared errors: [0.9,1.1,0.8,0.7,1.0]. Compute average CV error.*

Exercise 71 *4-fold CV. Fold test errors: [0.05, 0.1, 0.07, 0.08]. Compute average CV error.*

Exercise 72 *A dataset of 30 samples. 3-fold CV. Fold test errors: [0.12,0.15,0.1]. Compute CV error.*

Exercise 73 *Dataset: 9 samples. LOO CV squared errors: [0.4,0.5,0.3,0.6,0.5,0.7,0.4,0.5,0.6]. Compute CV MSE.*

Exercise 74 *A regression model trained on a dataset of 100 points shows the following:*

$$\textit{Training MSE} = 4, \quad \textit{Test MSE} = 16$$

If the irreducible error (noise) is 3, estimate the variance component of the test error.

Exercise 75 *For a linear regression model with 10 parameters trained on 50 samples, the bias is found to be 2.5. If the expected total error is 11 and irreducible noise is 1.5, compute the variance term.*

Exercise 76 *Suppose a decision tree with depth = 10 gives training error = 0.05 and test error = 0.45. When depth = 3, training error = 0.25 and test error = 0.30. Find the approximate bias and variance trend (in terms of test error difference) between the two cases numerically.*

Exercise 77 *In a bias–variance decomposition experiment, the model gives:*

$$E[(Y - \hat{f}(X))^2] = 25, \quad \textit{Bias}^2 = 9, \quad \textit{Noise} = 4$$

Find the variance of the estimator.

Exercise 78 *A dataset is used to train polynomial regression models of degree 1, 3, 5, and 9. Their corresponding test MSEs are 18, 10, 12, and 35 respectively. Estimate the polynomial degree where the bias–variance trade-off is optimal.*

5.6 YouTube Links and QR Codes

Lecture	Details	YouTube Link	QR Code
27	Evaluation & Cross-Validation Explained: Hold-Out, K-Fold & LOOCV	https://youtu.be/5zbXeq-ce6M	
28	ROC Curve & AUC Explained: Bias–Variance Trade-off with Examples	https://youtu.be/RbAMKScb8Ms	
29	Problem Solving – Cross Validation & ROC AUC (Solutions 144 - 174)	https://youtu.be/UlZL0ncHj74	

Chapter 6

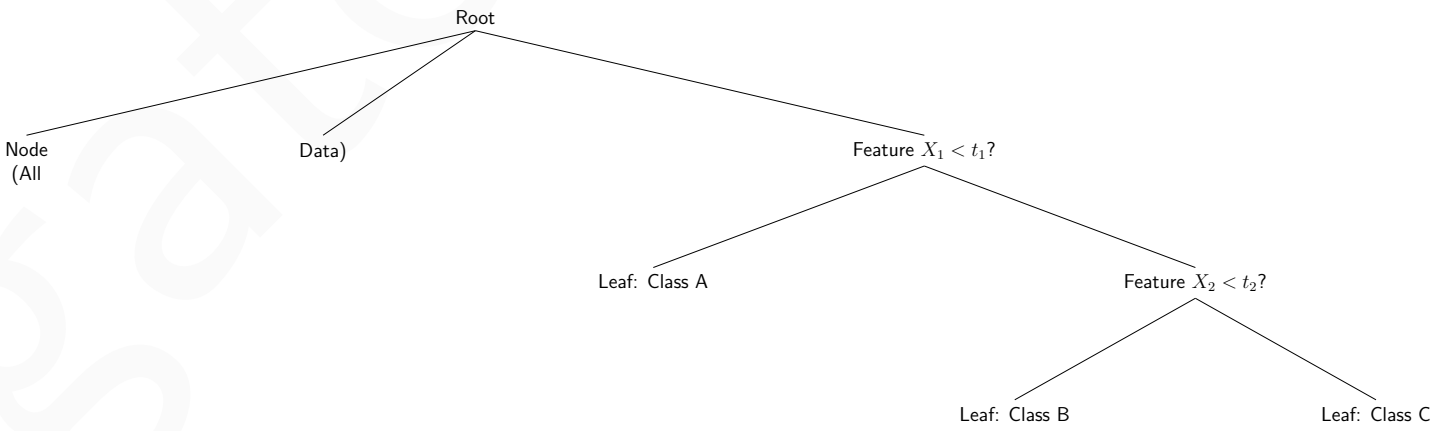
Regression & Classification Models

6.1 Decision Trees

Decision Trees are non-parametric supervised learning models used for both classification and regression tasks. They work by recursively splitting the data space into homogeneous subsets based on feature values, forming a tree-like structure.

A Decision Tree consists of:

- **Root Node:** Represents the entire dataset.
- **Internal Nodes:** Represent decisions based on feature tests.
- **Leaf Nodes:** Represent class labels (in classification) or continuous values (in regression).



6.1.1 Entropy, Gini Index, and Information Gain

Decision Trees divide data into smaller, purer subsets. To decide the best attribute for splitting, the tree evaluates impurity measures such as **Shannon Entropy** (ID3, C4.5) and **Gini Index** (CART). The attribute giving the maximum reduction in impurity is chosen.

Shannon Entropy

Entropy (from information theory) quantifies the uncertainty or disorder in a dataset.

$$H(S) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

where

- k = number of classes
- p_i = proportion of samples belonging to class i

Interpretation:

- $H(S) = 0$ when the dataset is pure (all samples belong to one class).
- Entropy is maximum when classes are equally likely.

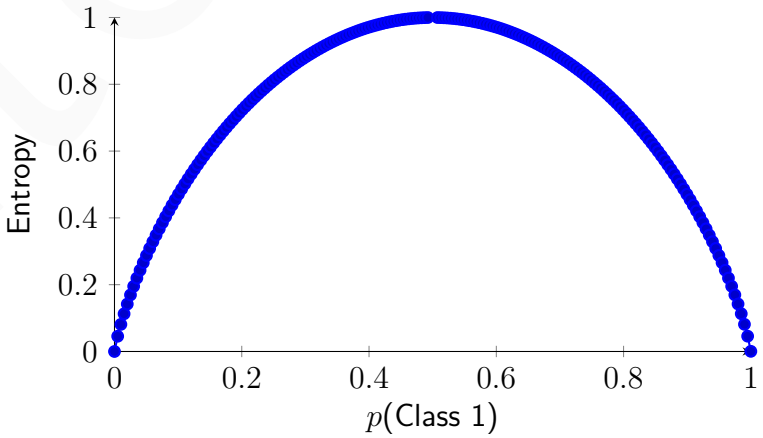
Example 41:

A dataset has 8 samples: 5 belong to Class A and 3 belong to Class B.

$$H(S) = -\left(\frac{5}{8}\log_2\frac{5}{8} + \frac{3}{8}\log_2\frac{3}{8}\right)$$

$$H(S) = -(0.625\log_2 0.625 + 0.375\log_2 0.375) = 0.954$$

This indicates moderate impurity.



Gini Index

Gini Index (used in CART) measures the probability of a random sample being incorrectly classified.

$$Gini(S) = 1 - \sum_{i=1}^{k} p_i^2$$

Interpretation:

- $Gini = 0$ means pure node.
- Lower Gini indicates a better split.

Example 42:

Using the same dataset: 5 in Class A, 3 in Class B,

$$Gini(S) = 1 - \left(\left(\frac{5}{8}\right)^2 + \left(\frac{3}{8}\right)^2\right)$$

$$Gini = 1 - (0.3906 + 0.1406) = 0.4688$$

The node has moderate impurity.

Entropy vs Gini (Key Differences)

- Entropy uses logarithmic measure, Gini uses squared probabilities.
- Both rank splits similarly in most cases.
- Gini is computationally simpler (preferred in CART).

Information Gain

Information Gain quantifies the reduction in impurity after splitting on a feature.

$$IG(S, A) = Impurity(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Impurity(S_v)$$

Notes:

- For ID3/C4.5: use Entropy in the formula.
- For CART: use Gini Index.

Example 43:

A dataset has 10 samples: 6 in Class 1, 4 in Class 2.

$$H(S) = -\left(\frac{6}{10}\log_2\frac{6}{10} + \frac{4}{10}\log_2\frac{4}{10}\right) = 0.971$$

A feature splits the dataset into two subsets:

$$S_1 : 4 \text{ samples}, \quad H_1 = 0.5$$
$$S_2 : 6 \text{ samples}, \quad H_2 = 0.811$$

Weighted entropy:

$$H_{\text{split}} = \frac{4}{10} \times 0.5 + \frac{6}{10} \times 0.811 = 0.847$$

Information Gain:

$$IG = 0.971 - 0.847 = 0.124$$

Thus, this feature gives an improvement of 0.124 in purity.

Match No.	Pitch	Format	Winner
1	S	T	Green
2	S	T	Blue
3	F	O	Blue
4	S	O	Blue
5	F	T	Green
6	F	O	Blue
7	S	O	Green
8	F	T	Blue
9	F	O	Blue
10	S	O	Green

Table C: Details of cricket games between Green and Blue

Example 44:

Details of ten international cricket games between two teams "Green" and "Blue" are given in Table C. The attribute **Pitch** can take one of two values: spin-friendly (S) or pace-friendly (F). The attribute **Format** can take values: one-day (O) or test (T). The target variable is the **Winner**. Compute the Information Gain IG(C, Pitch) with respect to the Target. Give the answer rounded to two decimal places.

Total matches = 10

Green wins = 4 Blue wins = 6

$$H(\text{Winner}) = -\frac{4}{10}\log_2\left(\frac{4}{10}\right) - \frac{6}{10}\log_2\left(\frac{6}{10}\right)$$

$$H(\text{Winner}) = -0.4\log_2(0.4) - 0.6\log_2(0.6) = 0.97095 \text{ bits}$$

Step 2: Entropy for each value of Pitch

Pitch = S (Spin-friendly)
Matches: 1,2,4,7,10 Green = 3, Blue = 2

$$H(\text{Winner}|\text{Pitch} = S) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)$$

$$= -0.6\log_2(0.6) - 0.4\log_2(0.4) = 0.97095$$

Pitch = F (Pace-friendly)
Matches: 3,5,6,8,9 Green = 1, Blue = 4

$$H(\text{Winner}|\text{Pitch} = F) = -\frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\left(\frac{4}{5}\right)$$

$$= -0.2\log_2(0.2) - 0.8\log_2(0.8) = 0.72193$$

Step 3: Conditional Entropy

$$H(\text{Winner}|\text{Pitch}) = \frac{5}{10}(0.97095) + \frac{5}{10}(0.72193)$$

$$= 0.84644$$

Step 4: Information Gain

$$\text{IG}(C, \text{Pitch}) = H(\text{Winner}) - H(\text{Winner}|\text{Pitch})$$

$$= 0.97095 - 0.84644 = 0.12451$$

$$\boxed{\text{IG}(C, \text{Pitch}) \approx 0.12}$$

6.1.2 Tree Construction and Stopping Criteria

Algorithm for Building a Decision Tree

1. Start with all training samples at the root.
2. Compute impurity (Entropy or Gini).
3. For each feature, calculate Information Gain.
4. Split on the feature with the highest IG.
5. Repeat recursively until:
   - Node is pure (all samples same class), or
   - Maximum depth reached, or
   - Minimum number of samples per node is met.

Example 45:

Consider features:

Outlook: Sunny, Overcast, Rain, Target: Play Tennis?

Compute IG for each feature; the one with the highest gain (say, Outlook) becomes the root node. Repeat for child nodes recursively.

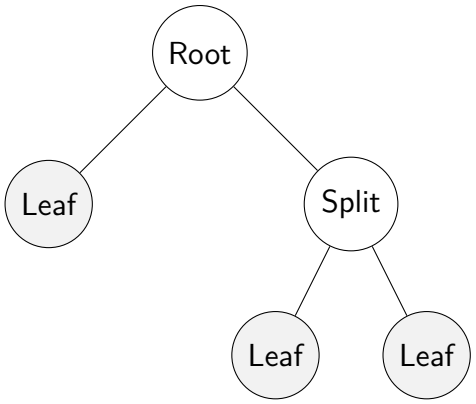
—

6.1.3 Tree Pruning and Overfitting

A large tree can overfit—capturing noise instead of true structure.

Pruning Techniques

- **Pre-pruning:** Stop growing tree early (limit depth, min samples).
- **Post-pruning:** Grow full tree, then remove branches that do not improve validation accuracy.



Post-pruning removes redundant subtrees

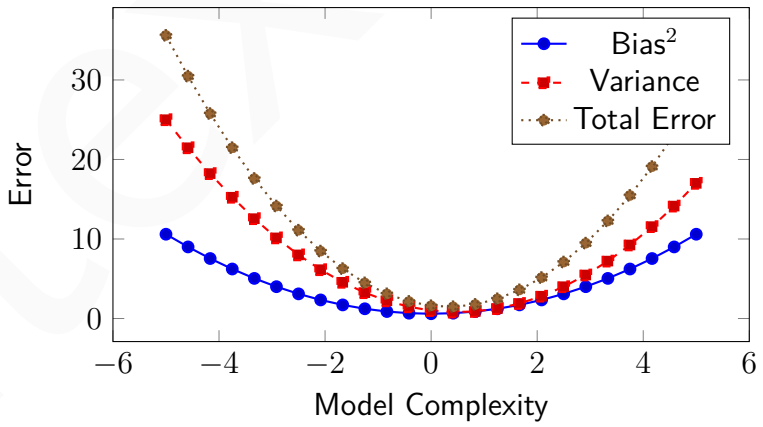
Reasoning: Pruning improves generalization by reducing variance. The goal is to find the simplest model that performs well on unseen data.

—

6.1.4 Bias–Variance Characteristics

Bias–Variance Tradeoff in Decision Trees

- **Small trees:** High bias, low variance (underfitting).
- **Large trees:** Low bias, high variance (overfitting).
- **Ensemble methods (Random Forests):** Reduce variance while maintaining low bias.



Insight: Decision Trees are powerful but unstable learners. Small changes in training data can drastically change splits. Ensembles like Random Forests or Gradient Boosted Trees help stabilize performance.

Example 46:

If a Decision Tree trained on a dataset of 100 samples achieves 100% accuracy on training but only 60% on test data, it exhibits:

- Low bias (fits training perfectly)
- High variance (poor generalization)

Using pruning or Random Forest can reduce variance.

Criteria	ID3	C4.5	CART
Splitting Measure	Entropy and Information Gain	Gain Ratio	Gini Index
Type of Output Tree	Multi-way splits	Multi-way splits	Binary splits only
Handles Continuous Attributes	No	Yes	Yes
Handles Missing Values	No	Yes	Yes
Pruning Method	Pre-pruning only	Post-pruning	Cost-complexity pruning
Target Variable Type	Classification only	Classification only	Classification and Regression
Bias Toward Attributes with Many Values	Yes	Reduced due to Gain Ratio	No
Works with Numeric Output	No	No	Yes (regression trees)

Comparison of ID3, C4.5, and CART Algorithms

6.2 Support Vector Machine (SVM)

6.2.1 Introduction

Support Vector Machine (SVM) is a supervised learning algorithm grounded in statistical learning theory and Vapnik–Chervonenkis (VC) theory, primarily used for classification and regression tasks. In its most fundamental form, SVM is designed to construct an optimal decision boundary that best separates data points belonging to different classes.

SVM emphasizes the concept of **margin maximization**, wherein the separation boundary is chosen not merely to classify correctly, but to be as far as possible from the nearest training points of both classes, thereby enhancing generalization performance.

Throughout this section, we restrict our discussion to:

- Binary classification
- Linear decision boundaries
- Linearly separable or near-separable datasets

The SVM framework evolves through three major conceptual stages:

1. Maximal Margin Classifier (Hard Margin SVM)
2. Support Vector Classifier (Soft Margin SVM)

3. Kernelized SVM (Kernel Trick)

6.2.2 Hyperplane

Definition

A *hyperplane* in $\mathbb{R}^d$ is the affine set

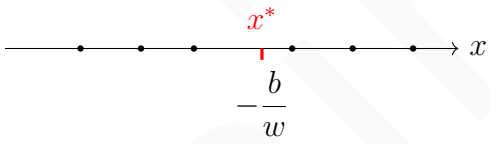
$$H = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{w}^T\mathbf{x} + b = 0\},$$

where $\mathbf{w} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ is the normal vector and $b \in \mathbb{R}$ the bias. The sign of $\mathbf{w}^T\mathbf{x} + b$ determines which side of the hyperplane $\mathbf{x}$ lies on.

1D ($\mathbb{R}$) – a point

In one dimension ($d = 1$) the hyperplane is a single point. With $w \neq 0$ and scalar x,

$$wx + b = 0 \qquad \Longrightarrow \qquad x = -\frac{b}{w}.$$



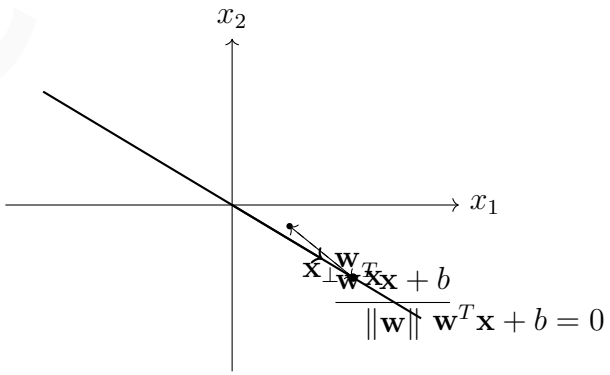
2D ($\mathbb{R}^2$) – a line

In two dimensions the hyperplane is a line. For $\mathbf{x} = (x_1, x_2)^T$ and $\mathbf{w} = (w_1, w_2)^T$:

$$w_1x_1 + w_2x_2 + b = 0.$$

The signed (orthogonal) distance from a point $\mathbf{x}$ to the line is

$$\text{dist}(\mathbf{x}, H) = \frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}.$$



General d ($\mathbb{R}^d$) – derivation (orthogonal projection)

Take any $\mathbf{x} \in \mathbb{R}^d$. Decompose $\mathbf{x}$ into a point on the hyperplane $\mathbf{x}_H \in H$ plus a component parallel to $\mathbf{w}$:

$$\mathbf{x} = \mathbf{x}_H + t\frac{\mathbf{w}}{\|\mathbf{w}\|}, \qquad t \in \mathbb{R}.$$

Because $\mathbf{x}_H$ satisfies $\mathbf{w}^T\mathbf{x}_H + b = 0$, substitute into that relation:

$$0 = \mathbf{w}^T\mathbf{x}_H + b = \mathbf{w}^T\left(\mathbf{x} - t\frac{\mathbf{w}}{\|\mathbf{w}\|}\right) + b = \mathbf{w}^T\mathbf{x} - t\|\mathbf{w}\| + b.$$

Solve for t to obtain the signed orthogonal distance from $\mathbf{x}$ to H:

$$\boxed{t = \frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}}.$$

Hence $\mathbf{w}^T\mathbf{x} + b$ is proportional to the signed distance (scale factor $\|\mathbf{w}\|$).

Decision function (linear classifier)

The linear decision function used in SVM and other linear classifiers is

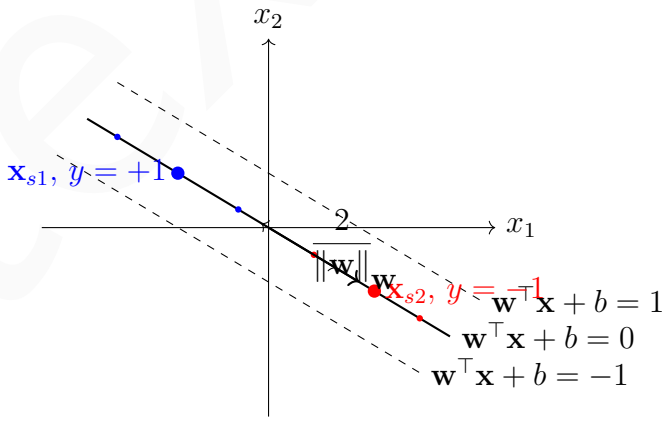
$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b,$$

and the class prediction is $\hat{y} = \text{sign}(f(\mathbf{x}))$.

$$\boxed{y = \mathbf{w}^T\mathbf{x} + b}$$

6.2.3 Hard-Margin SVM: geometry, formulation and solution

Geometric figure and support vectors (2D)



Concept (informal)

The maximal margin classifier chooses the separating hyperplane so that the perpendicular distance between the two parallel margin hyperplanes

$$\mathbf{w}^\top\mathbf{x} + b = +1 \quad \text{and} \quad \mathbf{w}^\top\mathbf{x} + b = -1$$

is maximized. Points lying exactly on these margins are the *support vectors*; they alone determine the classifier.

Derivation of the optimization problem

Distance between margin planes $= 2/\|\mathbf{w}\|$. Maximizing this is equivalent to minimizing $\|\mathbf{w}\|^2$ subject to correct classification with margin:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \qquad i = 1, \ldots, n.$$

Thus the primal problem (hard margin) is

$$\boxed{\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \ (i = 1, \ldots, n).}$$

Lagrangian, KKT and consequence

Form Lagrangian with multipliers $\alpha_i \geq 0$:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \Big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \Big).$$

Stationarity conditions (set partial derivatives to zero) give

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \ \Rightarrow \ \boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i}, \qquad \frac{\partial \mathcal{L}}{\partial b} = 0 \ \Rightarrow \ \sum_{i=1}^{n} \alpha_i y_i = 0.$$

Complementary slackness states that for each i either $\alpha_i = 0$ (non-support) or

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1 \quad \text{(support vector)}.$$

Hence support vectors are exactly those with $\alpha_i > 0$.

Dual statement (brief) and solution strategy

The dual quadratic program is

$$\max_{\boldsymbol{\alpha}} \ \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j,$$

$$\text{subject to} \ \ \alpha_i \geq 0, \ \ \sum_{i=1}^{n} \alpha_i y_i = 0.$$

Solve for α^* (QP). Recover primal weight using the boxed formula above.

Bias b (practical formula)

For any support vector s (with $\alpha_s > 0$) we have equality:

$$y_s(\mathbf{w}^\top \mathbf{x}_s + b) = 1 \ \Rightarrow \ \boxed{b = y_s - \mathbf{w}^\top \mathbf{x}_s}.$$

In practice average over all support vectors to get a numerically stable b:

$$b \ = \ \frac{1}{|S|} \sum_{s \in S} \Big(y_s - \mathbf{w}^\top \mathbf{x}_s \Big), \qquad S = \{ i : \alpha_i > 0 \}.$$

Summary (final formulas)

$$\boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i\,, \qquad b = \text{average}_{s\in S}\left(y_s - \mathbf{w}^\top \mathbf{x}_s\right)}$$

Support vectors are those indices with $\alpha_i^* > 0$ and they alone define the classifier.

Classifier function

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

Example 47:

Training data and Lagrange multipliers:

i	$\mathbf{x}_i$	y_i	α_i
1	$(1,2)$	$+1$	2
2	$(2,1)$	-1	2
3	$(2,2)$	$+1$	0

Classify $\mathbf{x} = (1,3)$.

Step 1 — Compute weight vector $\mathbf{w}$ Use the relation $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$. Compute each term explicitly:

$$\alpha_1 y_1 \mathbf{x}_1 = 2 \cdot (+1) \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix},$$

$$\alpha_2 y_2 \mathbf{x}_2 = 2 \cdot (-1) \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} -4 \\ -2 \end{pmatrix},$$

$$\alpha_3 y_3 \mathbf{x}_3 = 0 \cdot (+1) \cdot \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Sum the terms:

$$\mathbf{w} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} + \begin{pmatrix} -4 \\ -2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}.$$

Step 2 — Compute bias b For any support vector s with $\alpha_s > 0$ we have

$$y_s(\mathbf{w}^\top \mathbf{x}_s + b) = 1 \quad \Longrightarrow \quad b = y_s - \mathbf{w}^\top \mathbf{x}_s.$$

Support vectors here are indices 1 and 2 (since $\alpha_1 = 2,\ \alpha_2 = 2,\ \alpha_3 = 0$).
Compute $\mathbf{w}^\top \mathbf{x}_1$:

$$\mathbf{w}^\top \mathbf{x}_1 = (-2)\cdot 1 + 2\cdot 2 = -2 + 4 = 2. \quad \Rightarrow \quad b_1 = y_1 - \mathbf{w}^\top \mathbf{x}_1 = 1 - 2 = -1.$$

Compute $\mathbf{w}^\top \mathbf{x}_2$:

$$\mathbf{w}^\top \mathbf{x}_2 = (-2)\cdot 2 + 2\cdot 1 = -4 + 2 = -2. \quad \Rightarrow \quad b_2 = y_2 - \mathbf{w}^\top \mathbf{x}_2 = -1 - (-2) = -1 + 2 = 1.$$

Average the two values for numerical stability:

$$b = \frac{b_1 + b_2}{2} = \frac{-1+1}{2} = 0.$$

(Using a single support vector would give $b = -1$ or $b = 1$; averaging is standard practice.)

Step 3 — Decision function for $\mathbf{x} = (1, 3)$ Compute the raw decision value $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$.
First the dot product:

$$\mathbf{w}^\top \mathbf{x} = (-2)\cdot 1 + 2\cdot 3 = -2 + 6 = 4.$$

Add $b = 0$:

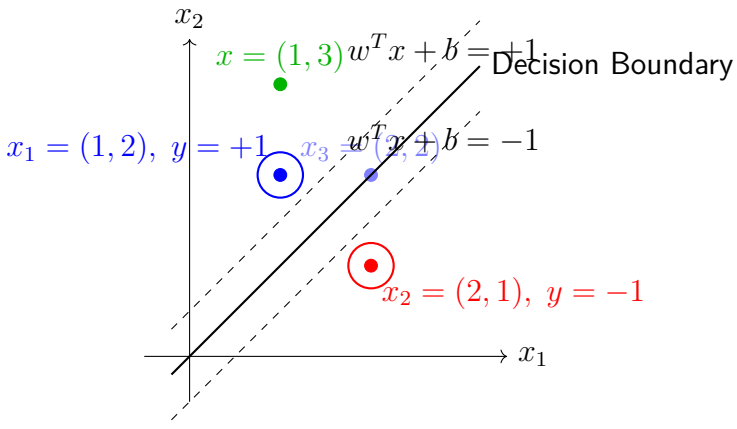
$$f(1, 3) = 4 + 0 = 4.$$

Step 4 — Classification (sign rule) Apply $\hat{y} = \text{sign}(f(\mathbf{x}))$:

$$\text{sign}(4) = +1.$$

Remarks Support vectors are indices 1 and 2 (those with $\alpha_i > 0$). The classifier parameters are

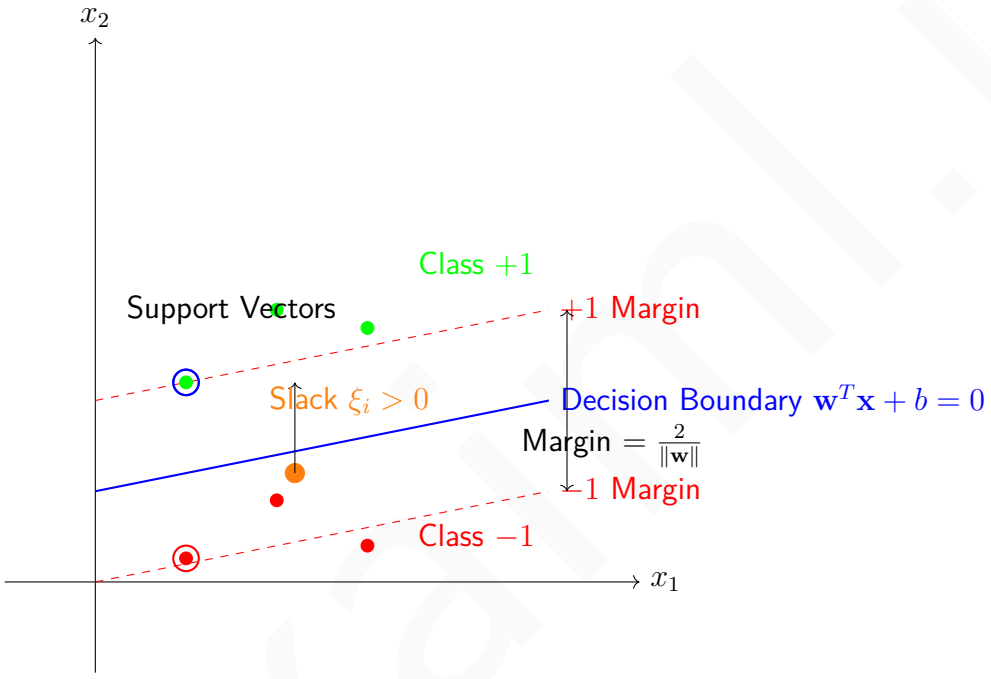
$$\mathbf{w} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \qquad b = 0.$$



6.2.4 Soft Margin - Support Vector Classifier

Motivation: why soft margin

The hard-margin SVM requires exact linear separability of the training data. In practice this is brittle: outliers, mislabels and naturally overlapping classes make the hard-margin program infeasible or lead to poor generalization. The soft-margin SVM remedies this by introducing nonnegative slack variables $\xi_i \geq 0$ which permit controlled violations of the margin constraint. The total violation is penalized in the objective with weight $C > 0$, which controls the trade-off between model complexity (large margin) and training error.



Primal formulation (soft margin)

Given labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $y_i \in \{\pm 1\}$, the soft-margin primal optimization problem is:

$$\min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i,$$

$$\text{subject to} \qquad y_i\Big(\mathbf{w}^\top \mathbf{x}_i + b\Big) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \qquad i = 1, \ldots, n.$$

The hyperparameter $C > 0$ trades off margin size against penalty for margin violations.

Lagrangian and stationarity

Introduce Lagrange multipliers $\alpha_i \geq 0$ for the margin constraints and $\mu_i \geq 0$ for $\xi_i \geq 0$. The Lagrangian is

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i\Big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i\Big) - \sum_{i=1}^{n} \mu_i \xi_i.$$

Stationarity conditions (set partial derivatives to zero) yield:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \;\Rightarrow\; \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i,$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \;\Rightarrow\; \sum_{i=1}^{n} \alpha_i y_i = 0,$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \;\Rightarrow\; C - \alpha_i - \mu_i = 0 \;\Rightarrow\; 0 \le \alpha_i \le C.$$

Dual problem

Eliminating $\mathbf{w}$ and b leads to the dual quadratic program

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n} \alpha_i \;-\; \tfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j,$$

$$\text{subject to} \quad 0 \le \alpha_i \le C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0.$$

This dual is a concave quadratic program in the variables α_i with simple bound constraints and one linear equality.

KKT conditions and practical consequences

From complementary slackness and stationarity we derive the operational rules:

- **Primal-dual relation (weights)**:

$$\boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i} \quad \text{(always)}.$$

- **Dual feasibility:** $0 \le \alpha_i \le C$ for every i, and $\sum_i \alpha_i y_i = 0$.
- **Complementary slackness (margin constraint):**

$$\alpha_i \Big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i \Big) = 0.$$

  Thus:

  - If $\alpha_i = 0$, then the margin constraint may be slack (point lies strictly outside margin) and we have $y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1$ and $\xi_i = 0$.
  - If $0 < \alpha_i < C$, then necessarily $\xi_i = 0$ and $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ (the point lies exactly on the margin boundary) — these are *support vectors on the margin*.
  - If $\alpha_i = C$, then $\mu_i = 0$ and ξ_i may be > 0; complementary slackness for α_i enforces

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1 - \xi_i \le 1,$$

    so such points lie inside the margin (or are misclassified).

- **Stationarity wrt ξ_i:** $C - \alpha_i - \mu_i = 0$ together with $\mu_i \xi_i = 0$ implies:
  - If $\alpha_i < C$ then $\mu_i > 0$ and therefore $\xi_i = 0$.
  - If $\alpha_i = C$ then $\mu_i = 0$ and ξ_i may be > 0.
- **Bias b:** For any i with $0 < \alpha_i < C$ (i.e. a margin support vector) we have the exact identity

$$\boxed{b = y_i - \mathbf{w}^\top \mathbf{x}_i}.$$

  In practice average b over all indices with $0 < \alpha_i < C$ to increase numerical stability.

Summary table: α_i, ξ_i and geometric meaning

Range of α_i	ξ_i	Geometric location	Condition on margin value
$\alpha_i = 0$	$\xi_i = 0$	outside margin, non-support	$y_i(\mathbf{w}^\top x_i + b) > 1$
$0 < \alpha_i < C$	$\xi_i = 0$	on margin (support vector)	$y_i(\mathbf{w}^\top x_i + b) = 1$
$\alpha_i = C$	$\xi_i \geq 0$	inside margin or misclassified	$y_i(\mathbf{w}^\top x_i + b) = 1 - \xi_i \leq 1$

Example 48:

Consider the training set and parameter:

i	$\mathbf{x}_i$	y_i	α_i
1	$(1,\ 0)$	$+1$	0.30
2	$(-1,\ 0)$	-1	0.60
3	$(0,\ 0)$	$+1$	0.10

$C = 0.6.$

Given the supplied vector

$$\boldsymbol{\alpha} = (0.30,\ 0.60,\ 0.10), \qquad y = (+1,\ -1,\ +1),$$

compute the primal weight vector

$$\mathbf{w} = \sum_{i=1}^{3} \alpha_i y_i \mathbf{x}_i.$$

$$\alpha_1 y_1 \mathbf{x}_1 = 0.30 \cdot (+1) \cdot (1, 0) = (0.30,\ 0),$$

$$\alpha_2 y_2 \mathbf{x}_2 = 0.60 \cdot (-1) \cdot (-1, 0) = 0.60 \cdot (1, 0) = (0.60,\ 0),$$

$$\alpha_3 y_3 \mathbf{x}_3 = 0.10 \cdot (+1) \cdot (0, 0) = (0, 0).$$

$$\boxed{\mathbf{w} = (0.30 + 0.60,\ 0) = (0.90,\ 0)}.$$

Per-support bias values (KKT equality for $0 < \alpha_i < C$) For any index with $0 < \alpha_i < C$ the KKT equality gives

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1 \quad \Longrightarrow \quad b = y_i - \mathbf{w}^\top \mathbf{x}_i.$$

Apply to the two non-bound support indices $i = 1$ and $i = 3$:

$$b_1 = 1 - (0.90, 0) \cdot (1, 0) = 1 - 0.90 = 0.10,$$

$$b_3 = 1 - (0.90, 0) \cdot (0, 0) = 1 - 0 = 1.00.$$

Practical (heuristic) averaging of these two values yields

$$\boxed{b = \frac{b_1 + b_3}{2} = \frac{0.10 + 1.00}{2} = 0.55}.$$

Classification of $\mathbf{x} = (1, 3)$ with the chosen $(\mathbf{w}, b)$ Decision function:

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

Compute:

$$\mathbf{w}^\top (1, 3) + b = (0.90, 0) \cdot (1, 3) + 0.55 = 0.90 + 0 + 0.55 = 1.45,$$

hence

$$\boxed{f(1, 3) = \text{sign}(1.45) = +1\,.}$$

6.2.4.1 Hinge Loss

In Soft Margin SVM, not all violations are treated equally. The classifier applies a penalty that increases as a point moves deeper into the margin or crosses the decision boundary. This behaviour is visually described by the hinge loss curve.

Conceptual interpretation:

- Correct and confidently classified points incur **zero penalty**.
- Points inside the margin incur a **gradual linear penalty**.
- Misclassified points incur the **highest penalty**.

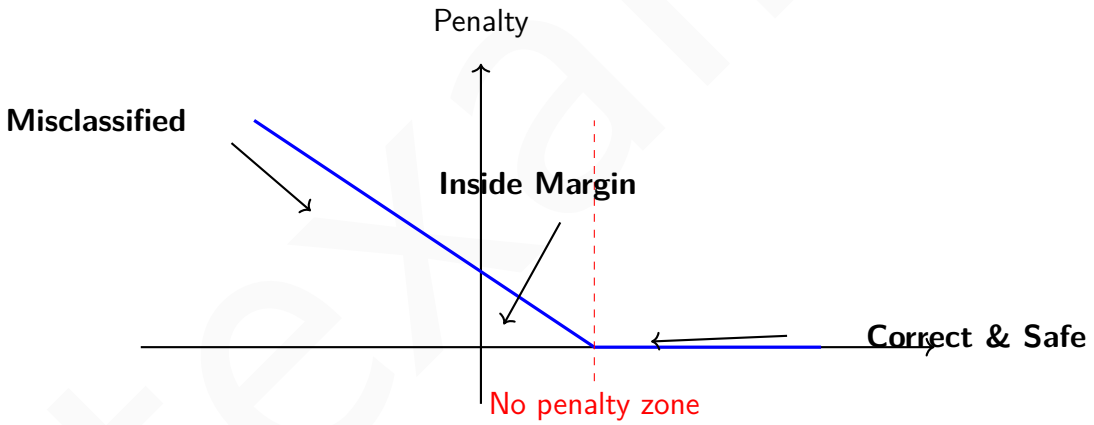


Figure Interpretation:

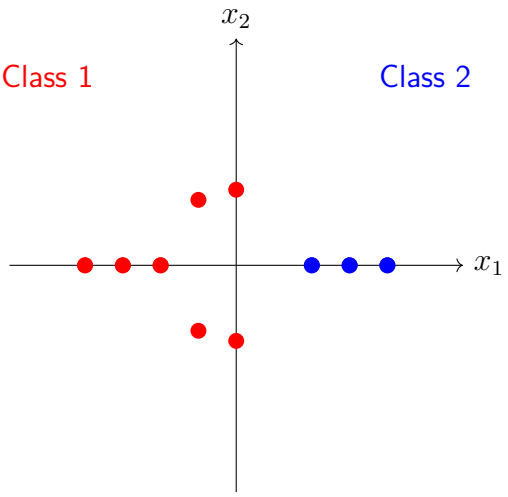
- The slanted segment (slope =1) represents increasing penalty for margin violations.
- The flat segment (slope =0) represents regions where points are sufficiently confident and incur no cost.
- The hinge point marks the transition between penalized and unpenalized zones.

This illustrates how Soft Margin SVM balances classification accuracy and robustness by tolerating controlled violations rather than enforcing strict separability.

6.2.5 Kernel Trick

Linear SVM works by finding a hyperplane that separates classes. However, real-world data is often **non-linear**, as shown below:

Non-linear Data Example



Clearly, no straight line can separate these classes. We need a **non-linear mapping**.

Feature Mapping $\phi(\cdot)$

The idea is to map input features to a higher-dimensional space where data becomes linearly separable:

$$\phi : \mathbb{R}^n \to \mathbb{R}^m, \quad m > n$$

Definition: Feature Mapping

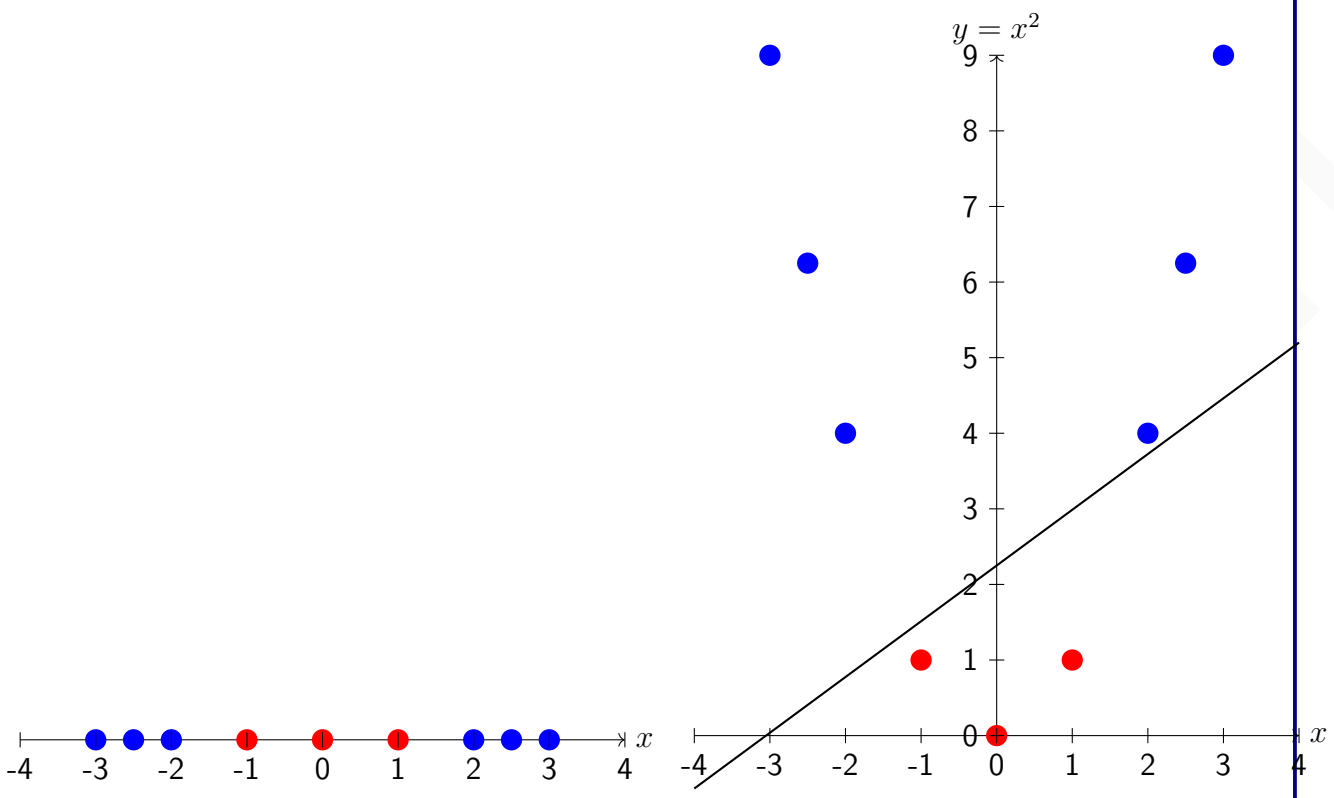
A function $\phi(x)$ transforms input x to a higher-dimensional space to make the data linearly separable.

Example: 1D to 2D Mapping

Consider 1D data $x \in \mathbb{R}$ and map it to 2D using:

$$\phi(x) = (x, x^2)$$

Visualization 1D → 2D



6.2.5.1 Kernel Function

Instead of computing $\phi(x)$ explicitly, we use a kernel function:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Definition: Kernel Function

A kernel function computes the dot product of two vectors in the higher-dimensional space without explicitly computing the mapping $\phi(x)$.

The "Trick" with Dot Products

We can write the SVM classifier as:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b$$

Here, $K(x_i, x)$ implicitly uses the higher-dimensional mapping. This avoids costly computations in high-dimensional space.

Common Kernel Functions

- **Linear Kernel:** $K(x_i, x_j) = x_i \cdot x_j$
- **Polynomial Kernel:** $K(x_i, x_j) = (x_i \cdot x_j + c)^d$

- **RBF / Gaussian Kernel:** $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
- **Sigmoid Kernel:** $K(x_i, x_j) = \tanh(\kappa x_i \cdot x_j + c)$

Summary

The kernel trick allows SVM to perform non-linear classification efficiently by:

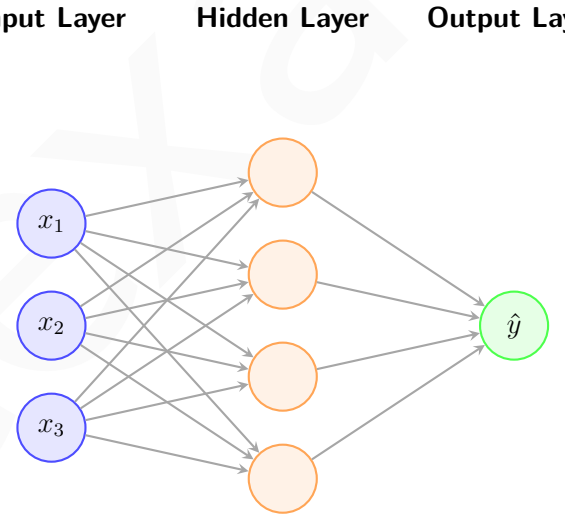
1. Mapping data to higher dimensions via $\phi(x)$
2. Using kernel functions $K(x_i, x_j)$ to compute dot products
3. Avoiding explicit computation in high-dimensional space

6.3 Neural Network Models

Feed-Forward Neural Networks (FFNN)

6.3.1 Network Architecture and Notation

A **Feed-Forward Neural Network (FFNN)** is a computational model inspired by the biological neuron. It consists of multiple layers of interconnected nodes (neurons), where information flows in one direction — from input to output — without any feedback loops.



Notation:

- x_i : Input features $(i = 1, 2, \ldots, d)$
- $w_{ij}^{(l)}$: Weight connecting neuron j in layer $(l-1)$ to neuron i in layer (l)
- $b_i^{(l)}$: Bias term for neuron i in layer (l)
- $a_i^{(l)}$: Activation (output) of neuron i in layer (l)

—

6.3.2 Activation Functions (Sigmoid, ReLU, Tanh)

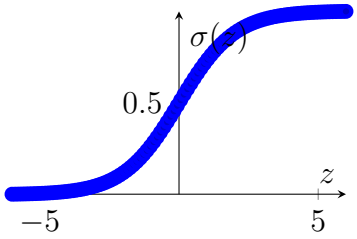
The activation function introduces non-linearity, allowing the network to model complex relationships.

1. Sigmoid Function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

It maps input values to $(0, 1)$, useful for probabilities.

Sigmoid Activation



2. Tanh Function:

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

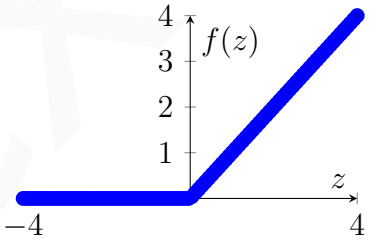
Range: $(-1, 1)$ — centered around zero.

3. ReLU (Rectified Linear Unit):

$$f(z) = \max(0, z)$$

It is widely used due to faster convergence and less vanishing gradient.

ReLU Activation



—

6.3.3 Forward Propagation and Loss Computation

In forward propagation, data moves from the input layer through hidden layers to produce the output.

For a single neuron:

$$z_i^{(l)} = \sum_j w_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)}$$

$$a_i^{(l)} = f(z_i^{(l)})$$

For the output layer in regression:

$$\hat{y} = a^{(L)} = f(z^{(L)})$$

Example: Suppose input $x = [1, 2]$, weights $w = [0.5, -1]$, bias $b = 0.2$ and $f(z) = \text{ReLU}(z)$.

$$z = (1)(0.5) + (2)(-1) + 0.2 = -1.3 \implies a = 0$$

So the neuron output is 0.

Loss Function (Mean Squared Error):

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

—

6.3.4 Backpropagation Algorithm and Gradient Flow

Goal: Adjust weights w and biases b to minimize the loss.

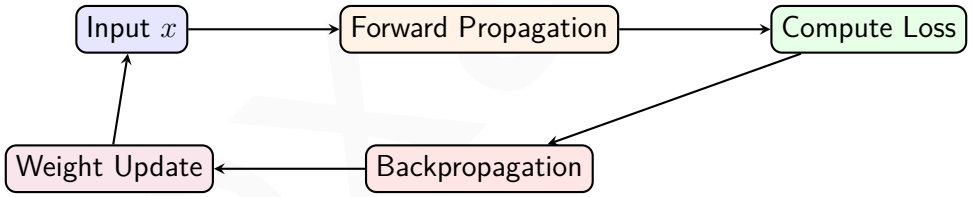
$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial L}{\partial w_{ij}^{(l)}}$$

where η is the learning rate.

Gradient computation:

$$\frac{\partial L}{\partial w_{ij}^{(l)}} = \delta_i^{(l)} a_j^{(l-1)}$$

$$\delta_i^{(l)} = f'(z_i^{(l)}) \sum_k \delta_k^{(l+1)} w_{ki}^{(l+1)}$$



—

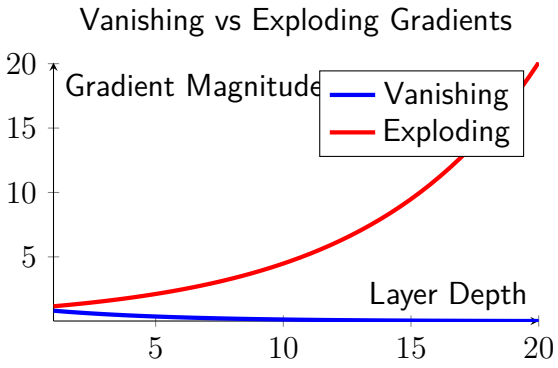
6.3.5 Vanishing/Exploding Gradient Problem

Definition: When gradients become extremely small (vanish) or large (explode) during backpropagation, training becomes unstable.

Why it happens:

$$\delta^{(l)} = f'(z^{(l)}) W^{(l+1)T} \delta^{(l+1)}$$

If $|f'(z^{(l)})| < 1$ repeatedly (e.g., Sigmoid), gradients shrink exponentially $\rightarrow$ vanishing. If weights > 1, gradients may explode.



Remedies:

- Use ReLU instead of Sigmoid or Tanh.
- Normalize inputs and use Batch Normalization.
- Use residual connections (as in ResNet).
- Initialize weights carefully (e.g., Xavier or He initialization).

—

Summary

- FFNN learns a non-linear mapping from inputs to outputs via multiple layers.
- Activation functions enable non-linearity.
- Forward pass computes prediction; backward pass updates weights.
- Deep networks suffer from vanishing/exploding gradients, which can be mitigated by normalization and better initialization.

6.4 Multi-Layer Perceptron (MLP)

6.4.1 Universal Approximation Theorem (Conceptual Insight)

Universal Approximation Theorem

The Universal Approximation Theorem states that a feed-forward neural network with:

- at least one hidden layer,
- finite number of neurons,
- and a non-linear activation function (e.g., sigmoid, ReLU, tanh)

can approximate any continuous function $f : \mathbb{R}^n \to \mathbb{R}$ to an arbitrary degree of accuracy given sufficient neurons.

Reasoning: Hidden layers provide non-linear transformations, allowing the network to capture complex mappings between inputs and outputs that linear models cannot. This theorem gives theoretical assurance of MLP's capability, although practical training may require regularization and careful weight initialization.

Example 49:

Suppose we want to approximate $f(x) = \sin(x)$ over $[0, \pi]$.

- Using a single hidden layer with 10 neurons and a sigmoid activation, an MLP can approximate $\sin(x)$ closely.
- Increasing neurons improves accuracy but may increase overfitting risk.

6.4.2 Training and Regularization Techniques

Training MLPs

Training involves minimizing a loss function (e.g., MSE for regression, cross-entropy for classification) using:

- Forward propagation to compute outputs.
- Backpropagation to compute gradients w.r.t weights.
- Gradient-based optimization (e.g., SGD, Adam) to update weights.

Regularization Techniques

To prevent overfitting:

- **L2 Regularization (Weight Decay):** Adds $\lambda \sum w^2$ to loss.
- **Dropout:** Randomly zeroes a fraction of hidden neurons during training.
- **Early Stopping:** Halt training when validation loss stops improving.

Example 50:

A classification MLP with 3 hidden layers (64, 32, 16 neurons) overfits training data. Applying dropout of 0.3 and L2 regularization $\lambda = 0.01$ improves validation accuracy by 8%.

6.4.3 Comparison with Traditional ML Models

Comparison

- **Linear Models:** Can only capture linear relationships.
- **Decision Trees:** Non-linear, but may struggle with high-dimensional data and interactions.
- **MLPs:** Capture complex non-linear patterns and interactions, scalable with more data.

Example 51:

Predicting housing prices:

- Linear regression underfits the data due to non-linear interactions.
- Random Forest captures some non-linearities but saturates at 90% R-squared.
- MLP with 2 hidden layers achieves 97% R-squared by modeling complex interactions.

6.4.4 Advantages and Limitations of MLPs

Advantages

- Can approximate any continuous function (Universal Approximation).
- Flexible architecture (number of layers/neurons).
- Suitable for regression, classification, and complex pattern recognition.

Limitations

- Requires large amounts of labeled data.
- Computationally intensive, especially with many layers.
- Prone to overfitting without proper regularization.
- Difficult to interpret compared to simple models.

Example 52:

A small dataset (500 samples, 10 features) trained with a deep MLP (5 hidden layers) overfits drastically. Using a shallow MLP (2 hidden layers) with dropout maintains similar accuracy but improves generalization.

6.5 Problems

Problem 175 *Consider the dataset below:*

Weather	*Humidity*	*Wind*	*Play*
Sunny	*High*	*Weak*	*No*
Sunny	*Low*	*Strong*	*Yes*
Rainy	*High*	*Weak*	*Yes*
Rainy	*Low*	*Strong*	*Yes*

*Compute the Information Gain of the attribute **Weather** w.r.t the target **Play**.*

Problem 176 *Given the following dataset:*

Match No.	Pitch	Format	Winner
1	S	T	Green
2	S	T	Blue
3	F	O	Blue
4	S	O	Blue
5	F	T	Green
6	F	O	Blue
7	S	O	Green
8	F	T	Blue
9	F	O	Blue
10	S	O	Green

Table C: Details of cricket games between Green and Blue

Age	*Income*	*Student*	*Class*
Young	*High*	*No*	*No*
Young	*Medium*	*No*	*Yes*
Old	*High*	*No*	*Yes*
Old	*Medium*	*Yes*	*Yes*

*Compute the entropy of the target **Class** and the Information Gain of attribute **Student**.*

Problem 177 *Details of ten international cricket games between two teams "Green" and "Blue" are given in Table C. The attribute **Pitch** can take one of two values: spin-friendly (S) or pace-friendly (F). The attribute **Format** can take values: one-day (O) or test (T). The target variable is the **Winner**.*

Compute the Information Gain IG(C, Pitch) with respect to the Target. Give the answer rounded to two decimal places.

Problem 178 *A Decision Tree classifier uses the Gini Index as the impurity measure. For a node with class proportions $[0.2, 0.3, 0.5]$, the Gini Index is approximately:*

(A) 0.38

(B) 0.58

(C) 0.62

(D) 0.47

Problem 179 *In a binary classification tree, if both child nodes after a split are pure, then the Information Gain is:*

(A) Zero

(B) Equal to parent node entropy

(C) Negative

(D) Depends on number of samples

Problem 180 *If a feature perfectly separates the training data in a Decision Tree, its Information Gain is:*

(A) 0

(B) 0.5

(C) 1 (maximum possible)

(D) Undefined

Problem 181 *Entropy reaches its maximum value when:*

(A) All data belong to one class

(B) Classes are equally likely

(C) Classes are inversely proportional to sample size

(D) The number of classes is one

Problem 182 *For a binary class dataset with 8 positive and 2 negative samples, the entropy is approximately:*

(A) 0.29

(B) 0.72

(C) 0.97

(D) 1.00

Problem 183 *In CART, splits are chosen by minimizing:*

(A) Gini impurity

(B) Entropy

(C) Misclassification rate

(D) Both (a) and (b)

Problem 184 *Which of the following trees is likely to have the lowest variance?*

(A) Very deep tree

(B) Very shallow tree

(C) Tree with no pruning

(D) Tree fit with noisy data

Problem 185 *A Decision Tree model trained with high training accuracy and low test accuracy most likely suffers from:*

(A) High bias

(B) High variance

(C) Low variance

(D) Balanced bias and variance

Problem 186 *If the depth of a tree is restricted, it generally:*

(A) Increases variance

(B) Increases bias

(C) Reduces bias

(D) Has no effect on bias

Problem 187 *Which of the following statements about Decision Trees are correct?*

(A) They can handle both categorical and numerical data.

(B) They are invariant to monotonic transformations of features.

(C) They require feature scaling for optimal performance.

(D) They are prone to overfitting on small datasets.

Problem 188 *Consider two impurity measures: Entropy and Gini. Which statements are true?*

(A) Both achieve maximum value when classes are equally likely.

(B) Gini penalizes impurity more heavily than entropy.

(C) They produce identical feature rankings for most practical datasets.

(D) Gini is always less than entropy for the same class distribution.

Problem 189 *For a dataset with 3 classes, which conditions yield zero entropy?*

(A) All samples belong to one class.

(B) *Classes are equally distributed.*

(C) *One class dominates completely.*

(D) *The dataset is empty.*

Problem 190 *In Decision Tree pruning:*

(A) *Pre-pruning avoids overfitting by stopping early.*

(B) *Post-pruning removes weak branches after full growth.*

(C) *Pruning increases model variance.*

(D) *Cost-complexity pruning uses validation data to prune.*

Problem 191 *Which of the following are valid stopping criteria in tree induction?*

(A) *Node purity exceeds threshold.*

(B) *No remaining features to split.*

(C) *Split gain below minimum threshold.*

(D) *Dataset is perfectly separable.*

Problem 192 *Consider the use of Decision Trees in regression tasks. Which statements are true?*

(A) *The impurity measure used is variance reduction.*

(B) *Predictions are the mean of samples in a leaf.*

(C) *Entropy is used instead of variance.*

(D) *Gini Index is not used for regression trees.*

Problem 193 *Regarding bias–variance trade-off:*

(A) *Shallow trees have high bias and low variance.*

(B) *Deep trees have low bias and high variance.*

(C) *Ensemble methods reduce variance.*

(D) *Overfitting corresponds to high bias.*

Problem 194 *Which are advantages of Decision Trees?*

(A) *They are easily interpretable.*

(B) *They can model nonlinear boundaries.*

(C) *They require feature scaling.*

(D) *They can be used for both regression and classification.*

Problem 195 *For a binary Decision Tree, which are true?*

(A) *Entropy = 0 means pure node.*

(B) *Gini Index = 0.5 means completely mixed.*

(C) *IG can be negative.*

(D) *Pruned trees improve generalization.*

Problem 196 *A Decision Tree overfits when training accuracy is 100% and test accuracy is 60%. If pruning increases test accuracy to 80%, the reduction in variance can be approximated as ____ percentage points.*

Problem 197 *For a linearly separable dataset, the margin width between two classes in an SVM with weight vector w is:*

(A) $\|w\|$

(B) $\frac{1}{\|w\|}$

(C) $\frac{2}{\|w\|}$

(D) $\frac{1}{2\|w\|}$

Problem 198 *Which of the following statements about a hard-margin SVM are true?*

(A) *It assumes the data is perfectly linearly separable.*

(B) *It can tolerate noise points inside the margin.*

(C) *It minimizes $\|w\|^2$ subject to strict separation constraints.*

(D) *It is equivalent to minimizing hinge loss with $C \to \infty$.*

Problem 199 *The decision boundary for a linear SVM classifier is:*

$$?$$

(A) $w^T x + b = 1$

(B) $w^T x + b = -1$

(C) $w^T x + b = 0$

(D) $w^T x = 0$

Problem 200 *Increasing the parameter C in soft-margin SVM leads to:*

(A) *Wider margin and more violations*

(B) *Narrower margin and fewer violations*

(C) *No change in margin*

(D) *Randomly wider or narrower margin*

Problem 201 *Consider two classes with overlapping Gaussian distributions. Which classifier is likely to generalize better?*

(A) *Hard-margin SVM*

(B) *Soft-margin SVM*

(C) *Perceptron*

(D) *Decision tree*

Problem 202 *Which of the following is **true** for a nonlinear SVM with RBF kernel?*

(A) *The data is explicitly transformed into higher-dimensional space.*

(B) *The kernel function computes inner products without explicit transformation.*

(C) *It is equivalent to polynomial regression.*

(D) *It fails when γ is large.*

Problem 203 *The hinge loss function in SVM is defined as:*

$$L(y, f(x)) = ?$$

(A) $\log(1 + e^{-yf(x)})$

(B) $\max(0, 1 - yf(x))$

(C) $(y - f(x))^2$

(D) $|y - f(x)|$

Problem 204 *In SVM dual form, the Lagrange multipliers α_i corresponding to non-support vectors are:*

(A) *Positive*

(B) *Zero*

(C) *Negative*

(D) *Infinite*

Problem 205 *For a 2D dataset with points $(1, 2, +1)$, $(2, 3, +1)$, $(3, 2, -1)$, $(4, 3, -1)$, find the equation of separating hyperplane using SVM if $w = [1, -1], b = 0$. Answer as an equation of the form: $ax_1 + bx_2 + c = 0$.*

Problem 206 *If all training samples are multiplied by a constant scalar $k > 0$, then the margin in SVM will:*

(A) *Increase by k*

(B) *Decrease by k*

(C) *Remain unchanged*

(D) *Increase by* $1/k$

Problem 207 *Which kernel can produce infinite-dimensional feature mapping?*

(A) *Linear*

(B) *Polynomial*

(C) *RBF (Gaussian)*

(D) *Sigmoid*

Problem 208 *Which of the following conditions must be true for the optimal separating hyperplane in a SVM?*

(A) $y_i(w^T x_i + b) \geq 1$ *for all* i

(B) $\alpha_i(y_i(w^T x_i + b) - 1) = 0$

(C) $\sum_i \alpha_i y_i = 0$

(D) $0 < \alpha_i < C$ *only for support vectors*

Problem 209 *If two support vectors are equidistant from the decision boundary, what does it imply?*

(A) *Margin is symmetric*

(B) *Margin is asymmetric*

(C) *Hyperplane will shift*

(D) *Slack variables will become zero*

Problem 210 *Given a dataset perfectly separable by multiple hyperplanes, SVM will choose:*

(A) *The hyperplane that minimizes misclassification error*

(B) *The hyperplane with maximum margin*

(C) *The one passing through class centroids*

(D) *Any arbitrary separating plane*

Problem 211 *Which of the following loss functions is used in soft-margin SVM?*

(A) *Squared error loss*

(B) *Hinge loss*

(C) *Log loss*

(D) *Exponential loss*

Problem 212 *If* C *(penalty regularization hyperparameter) is very small, SVM behaves like:*

(A) *Overfitted classifier*

(B) *Underfitted classifier*

(C) *Perfect separator*

(D) *Ensemble model*

Problem 213 *When using an RBF kernel, which hyperparameter combination is most likely to cause underfitting?*

(A) *Large C, large γ*

(B) *Small C, small γ*

(C) *Large C, small γ*

(D) *Small C, large γ*

Problem 214 *A soft-margin SVM introduces slack variables ξ_i. What does $\xi_i = 2$ imply for a data point?*

(A) *It lies exactly on the margin*

(B) *It is correctly classified but inside the margin*

(C) *It is misclassified and beyond the opposite margin*

(D) *It has no effect on loss*

Problem 215 *If $w = [2, -1]$ and $b = -3$, find the distance of point $(2, 3)$ from the hyperplane. Give a numerical answer up to two decimal places.*

Problem 216 *For a linear SVM, $y_i(w^T x_i + b) = 1$ for support vectors. If $w = [1, 2]$, $b = -4$, and $x_i = [1, 1]$, compute the class label y_i.*

Problem 217 *A dataset in $\mathbb{R}^2$ has 10 points perfectly separable by multiple lines. How many unique maximum-margin hyperplanes exist? (Numeric answer)*

Problem 218 *For a linear SVM, the separating hyperplane is given as $2x_1 + 3x_2 - 6 = 0$. Compute the perpendicular distance from the origin to the hyperplane.*

Problem 219 *If the weight vector of a linear SVM is $\mathbf{w} = (4, 3)$, compute the margin width.*

Problem 220 *For a soft-margin SVM, a point violates the margin by 0.8. Find the slack variable ξ_i value for that point.*

Problem 221 *For a polynomial kernel $K(x, z) = (x^\top z + 1)^2$, find $K(x, z)$ for $x = (1, 2)$ and $z = (2, 3)$.*

Problem 222 *Given the decision function $f(x) = \text{sign}(2x_1 - 3x_2 + 1)$, determine $f(1, 1)$.*

Problem 223 *For an RBF kernel with $\gamma = 0.25$, the squared distance between two samples is $d^2 = 4$. Compute the kernel value $K(x, z)$.*

i	$\mathbf{x_i}$	$\mathbf{y_i}$	$\boldsymbol{\alpha_i}$
1	$(2, 1)$	$+1$	0.50
2	$(0, 1)$	-1	0.70
3	$(1, 0)$	$+1$	0.20

Problem 224 *Consider the following dataset which is linearly separable:*
Compute $f(1, 1)$ and predict the class label for the point $x = (1, 1)$, assume bias $= 0$.

Problem 225 *Consider the following soft-margin SVM dataset with $C = 0.6$:*
The SVM solution has weight vector $w = [1, -1]$ and bias $b = 0.5$.

(A) Check if the point $x = (1, 3)$ violates the margin.

(B) If yes, compute the slack variable ξ_i for this point.

i	$\mathbf{x_i}$	$\mathbf{y_i}$	$\boldsymbol{\alpha_i}$
1	$(1, 1)$	$+1$	0.40
2	$(-1, 1)$	-1	0.50
3	$(0, 0)$	$+1$	0.10

Problem 226 *Which of the following statements is/are correct about the rectified linear unit (ReLU) activation function defined as $R(x) = \max(0, x)$?*

(A) ReLU is continuous everywhere

(B) ReLU is differentiable everywhere

(C) ReLU is not differentiable at $x = 0$

(D) ReLU(x) = ReLU(ax), for all $a \in \mathbb{R}$

Problem 227 *A neuron has input $x = 2$, weight $w = 3$, bias $b = -1$, and activation $f(z) = \text{ReLU}(z)$. Compute the output y of the neuron.*

(A) 0

(B) 5

(C) 4

(D) 6

Problem 228 *In a 2-layer feedforward network with ReLU activations, which of the following can lead to **dead neurons**?*

(A) Large negative bias terms

(B) Very small learning rate

(C) High positive weight initialization

(D) Gradient clipping

Problem 229 *For a neuron with activation $a = \sigma(z)$ where $\sigma(z) = \frac{1}{1+e^{-z}}$, the derivative $\sigma'(z)$ attains its maximum value at:*

(A) $z = 0$

(B) $z \to \infty$

(C) $z \to -\infty$

(D) $z = 1$

Problem 230 *In a single-layer perceptron, which of the following conditions make the model equivalent to a linear classifier?*

(A) The activation function is linear

(B) The weights are constant

(C) The bias term is zero

(D) The inputs are normalized

Problem 231 *Consider a two-layer ReLU network: $h = \text{ReLU}(Wx + b)$, $y = Uh + c$. If $x = [1, -2]^T$, $W = \begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix}$, $b = [0, 1]^T$, $U = [1, 2]$, $c = 0$, find y.*

(A) 8

(B) -4

(C) 4

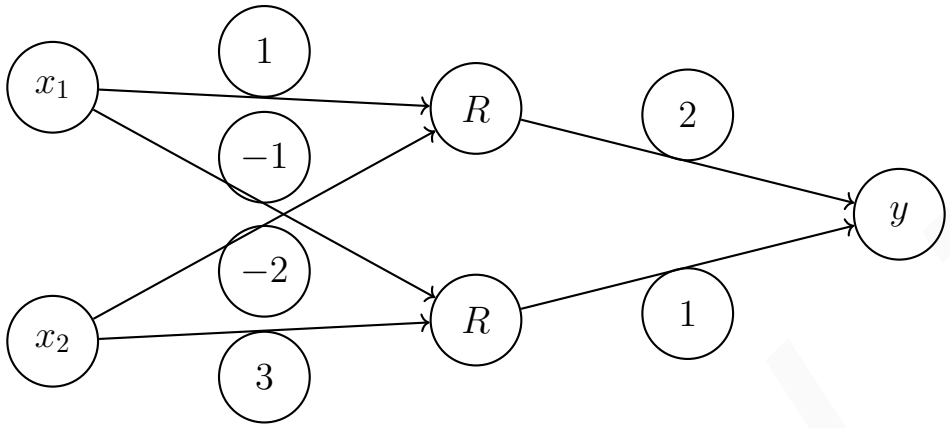
(D) 0

Problem 232 *A neural network with ReLU activation has one hidden layer with three neurons. Each hidden neuron computes $R(x)$, $R(2x)$, and $R(-x)$ respectively. The output is the sum of all hidden neuron outputs.*

The resulting function $f(x)$ is best described as:

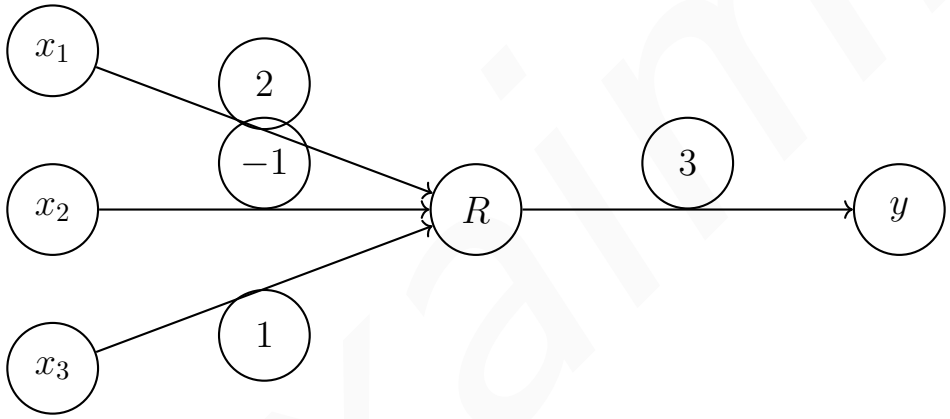
(A) $f(x) = |x|$
(B) $f(x) = 2|x|$
(C) $f(x) = 3|x|$
(D) Piecewise linear but not symmetric

Problem 233 *Consider the neural network shown below with inputs x_1, x_2. Hidden neurons use ReLU activation $R(z) = \max(0, z)$. The output neuron is linear.*



For input $(x_1, x_2) = (2, 1)$*, the value of* $\dfrac{\partial y}{\partial x_1}$ *is* _____.

Problem 234 *Consider the following ReLU neural network. All biases are zero.*

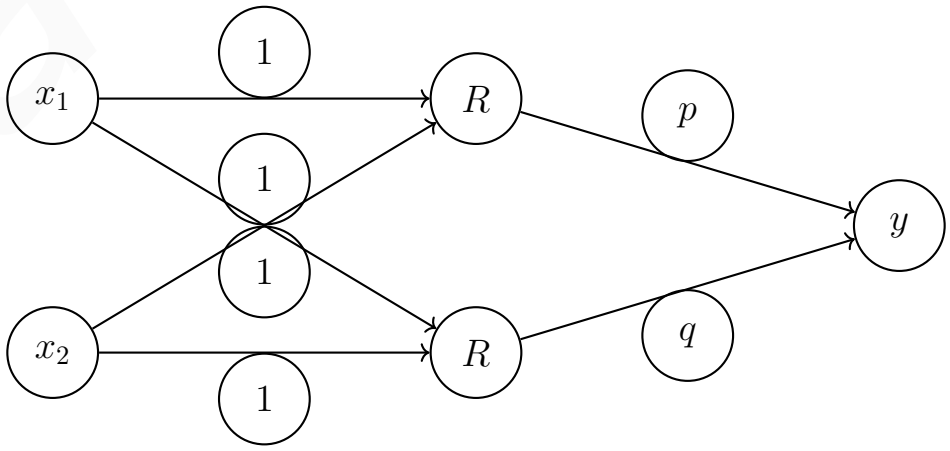


For which of the following regions does the network behave as a linear function of (x_1, x_2, x_3)*?*

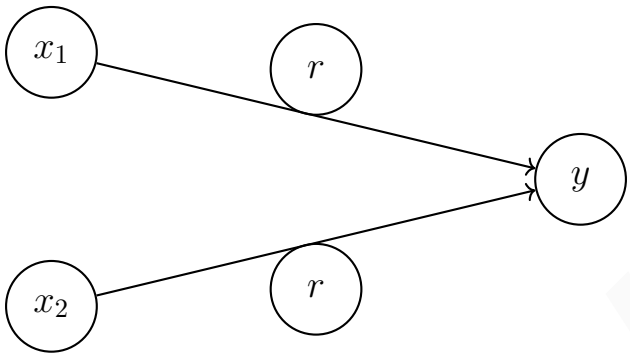
(A) $2x_1 - x_2 + x_3 > 0$
(B) $2x_1 - x_2 + x_3 < 0$
(C) $x_1 + x_2 + x_3 > 0$
(D) It is never linear

Problem 235 *Two neural networks are shown below. All activations are ReLU and all biases are zero. Assume* $x_1, x_2 > 0$.

Network 1

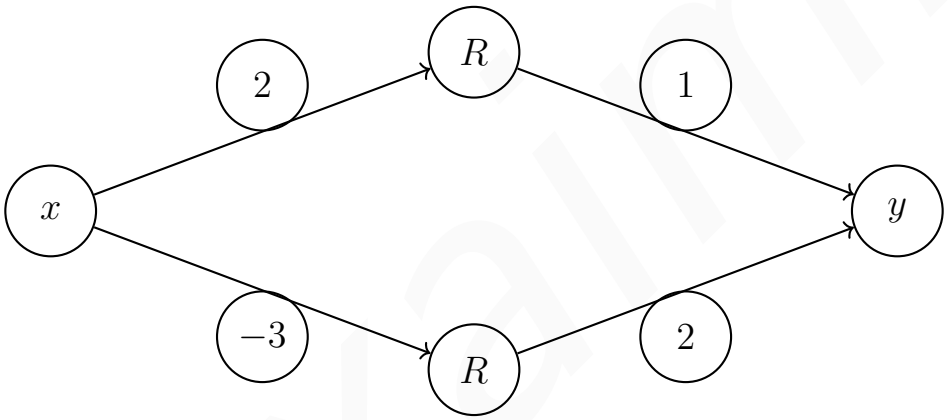


Network 2



For the two networks to be equivalent for all positive inputs, the values of (p, q, r) *are* _____.

Problem 236 *Consider the following one-dimensional ReLU network.*



The function $y(x)$ *represented by the network is:*

(A) Linear
(B) Piecewise linear with one kink
(C) Piecewise linear with two kinks
(D) Non-piecewise nonlinear

Problem 237 *If all activations in a network are linear, stacking multiple layers:*

(A) Increases non-linearity

(B) Has the same effect as a single linear transformation

(C) Always decreases error

(D) Causes vanishing gradients

Problem 238 *Which of the following statements is correct regarding the vanishing gradient problem?*

(A) It is prominent with ReLU activations

(B) It occurs when derivatives of activations are small

(C) It only affects output layers

(D) It can be mitigated using large learning rates

Problem 239 *The output of a sigmoid neuron is 0.99. The magnitude of its gradient with respect to the input z is approximately:*

(A) 0.01

(B) 0.09

(C) 0.0001

(D) 0.99

Problem 240 *In an FFNN, increasing the number of hidden units always:*

(A) Improves generalization

(B) Reduces training error

(C) Prevents overfitting

(D) Causes underfitting

Problem 241 *A feedforward neural network with no hidden layer and sigmoid activation in the output neuron is equivalent to:*

(A) Linear regression

(B) Logistic regression

(C) Polynomial regression

(D) Decision tree

Problem 242 *For a tanh activation function, the range of output values is:*

(A) $[0, 1]$

(B) $[-1, 1]$

(C) $[-\infty, \infty]$

(D) $[0, \infty)$

Problem 243 *The computational complexity of forward propagation in a dense FFNN with n_i inputs, n_h hidden neurons, and n_o outputs is:*

(A) $O(n_i + n_h + n_o)$

(B) $O(n_i n_h + n_h n_o)$

(C) $O(n_i n_o)$

(D) $O(n_h^2)$

Problem 244 *Which of the following is true about bias terms in a neural network?*

(A) *Bias helps shift the activation threshold*

(B) *Bias must always be positive*

(C) *Bias prevents vanishing gradients*

(D) *Removing bias increases network depth*

Problem 245 *Which of the following functions can act as an activation function?*

(A) *Any non-linear differentiable function*

(B) *Any piecewise linear function*

(C) *Constant function*

(D) *Step function*

Problem 246 *In backpropagation, the local gradient at layer l depends on:*

(A) *Derivative of activation at layer l*

(B) *Gradients from layer $l+1$*

(C) *Both (a) and (b)*

(D) *Only input data*

Problem 247 *For ReLU activation $R(x) = \max(0, x)$, the sum of the gradient $\frac{dR}{dx}$ at $x = -3$ and at $x = 5$ is __ (NAT)*

Problem 248 *A neuron has output $y = ReLU(2x - 4)$. For input $x = 3$, compute both y and $\frac{dy}{dx}$.*

Problem 249 *Consider a 3-layer neural network (input $\rightarrow$ hidden $\rightarrow$ output) where W_1 has dimensions (4×3) and W_2 has dimensions (1×4). If the input dimension is 3, calculate the total number of trainable parameters (including biases).*

Problem 250 *During backpropagation, the gradient of loss w.r.t. activation a_i is 0.01. If the neuron uses sigmoid activation with output $a_i = 0.99$, compute the effective gradient $\frac{\partial L}{\partial z_i}$, where z_i is the pre-activation input.*

6.6 Try it Yourself

Exercise 79 *A company wants to predict approval using these features:*

CreditScore	*Debt*	*Collateral*	*Approve*
Low	*High*	*None*	*No*
Low	*Low*	*Yes*	*Yes*
High	*High*	*Yes*	*Yes*
High	*Low*	*None*	*Yes*

*Compute the Gini impurity of the root node and the Gini Gain for a split on **CreditScore**.*

Exercise 80 *A small dataset for disease classification is shown below:*

Fever	*Cough*	*Travel*	*Disease*
High	*Yes*	*Yes*	*Positive*
High	*No*	*No*	*Negative*
Low	*Yes*	*No*	*Negative*
Low	*Yes*	*Yes*	*Positive*

*Compute the Information Gain for attribute **Travel** using Shannon entropy.*

Problem 251 *A dataset for predicting purchase:*

Age	*Gender*	*BrowsingTime*	*Buy*
Young	*M*	*Short*	*No*
Young	*F*	*Long*	*Yes*
Old	*F*	*Short*	*Yes*
Old	*M*	*Long*	*Yes*

*Compute the Gini impurity of the dataset and determine the best split between **Age**, **Gender**, and **BrowsingTime**.*

Exercise 81 *For a linearly separable dataset with margin width 0.4, what is the distance of each support vector from the decision boundary?*

(Assume $\|\mathbf{w}\| = 5$)

Exercise 82 *Given two classes with training points: $x_1 = (1, 1)$ (label $+1$), $x_2 = (2, 2)$ (label $+1$), and $x_3 = (0, 3)$ (label -1). Find the equation of the decision boundary using a linear SVM with soft-margin constant $C \to \infty$.*

Exercise 83 *If the optimal separating hyperplane has $\|\mathbf{w}\| = 4$, compute the geometric margin.*

Exercise 84 *For a 2D dataset, the separating hyperplane is given as $2x_1 + 3x_2 - 6 = 0$. Determine the margin width.*

Exercise 85 *If an SVM classifier uses polynomial kernel $K(x, y) = (x^\top y + 1)^2$, find the feature space dimension after transformation.*

Exercise 86 *In a soft-margin SVM, a point lies within the margin but on the correct side of the decision boundary. What is the likely value of its slack variable ξ_i?*

Exercise 87 *In a dataset, the decision boundary is $x_1 + x_2 - 2 = 0$. Predict the class of point $(3, 1)$ assuming label $+1$ for points above the boundary.*

Exercise 88 *A linear SVM classifier trained on normalized data yields $\mathbf{w} = (0.6, -0.8)$ and $b = 0.1$. Find the signed distance of a point $x = (1, 2)$ from the hyperplane.*

Exercise 89 *A dataset has 12 samples, out of which 8 belong to class A and 4 belong to class B. Compute the entropy (base 2) of the dataset.*

Exercise 90 *For the same dataset, compute the Gini Index.*

Exercise 91 *A node has 10 samples: 6 positives and 4 negatives. After a split, two child nodes contain:*

Child	*Positive*	*Negative*
Left	*4*	*1*
Right	*2*	*3*

Compute the Information Gain using entropy (base 2).

Exercise 92 *Suppose the entropy of a parent node is 0.971 bits and the weighted average entropy after splitting is 0.811 bits. Compute the Information Gain.*

Exercise 93 *At a node, the Gini index before split is 0.5. After splitting into two equal-sized child nodes, their Gini indices are 0.3 and 0.1. Compute the Gini Gain.*

Exercise 94 *A decision tree perfectly classifies the training data. The training accuracy is 100%, and test accuracy is 70%. Compute the difference between training and test error rates.*

Exercise 95 *Pruning a subtree reduces the number of leaves from 8 to 3. Each leaf has a misclassification rate of 0.05 before pruning and 0.12 after pruning. Compute the change in overall error rate.*

Exercise 96 *Dataset with 16 samples (10 positive, 6 negative) is split using attribute A:*

- *Left child: 8 samples (6 positive, 2 negative)*
- *Right child: 8 samples (4 positive, 4 negative)*

Compute Information Gain using entropy (base 2).

Exercise 97 *Binary classification dataset with 80% class A and 20% class B. Compute entropy (base 2).*

Exercise 98 *A decision tree built using ID3 algorithm has the following information gains:*

Attribute	*Information Gain*
A	*0.15*
B	*0.25*
C	*0.19*

Which attribute will be selected as the root node? Enter the attribute number (A=1, B=2, C=3).

Exercise 99 *A single neuron has input* $x = [2, -1]^T$*, weights* $w = [0.5, 1.5]^T$*, and bias* $b = -0.5$*. If the activation function is ReLU, compute the neuron output.*

Exercise 100 *For a sigmoid activation* $\sigma(z) = \frac{1}{1+e^{-z}}$*, compute the derivative* $\sigma'(z)$ *when* $z = 0$*.*

Exercise 101 *A neural network with one hidden layer uses ReLU activation. If the input to a neuron is* -3.2*, what is the output and its gradient value?*

Exercise 102 *A neuron has weights* $w = [1, -2, 1]$*, bias* $b = 2$*, and input* $x = [3, 1, -1]$*. Compute the output after applying the sigmoid activation (up to 3 decimal places).*

Exercise 103 *Consider a 3-layer FFNN: input(2)* $\rightarrow$ *hidden(3)* $\rightarrow$ *output(1). Find the total number of trainable parameters (including all biases).*

Exercise 104 *For a mean squared error (MSE) loss* $L = \frac{1}{2}(y - \hat{y})^2$*, if* $y = 1.0$ *and* $\hat{y} = 0.7$*, compute* $\frac{\partial L}{\partial \hat{y}}$*.*

Exercise 105 *The input to a ReLU neuron is* $z = 4.5$*. If the loss gradient at the neuron's output is* $\frac{\partial L}{\partial a} = 0.3$*, compute* $\frac{\partial L}{\partial z}$*.*

Exercise 106 *A tanh activation neuron receives an input* $z = 2.0$*. Compute the output* $\tanh(z)$ *and its derivative (to 3 decimal places).*

Exercise 107 *A feed-forward network uses sigmoid activation in all neurons. If the weight initialization mean is 0 and variance is 16, will the network experience vanishing or exploding gradients? Give 1 for vanishing, 2 for exploding.*

Exercise 108 *For a neuron with output* $a = ReLU(2x - 5)$*, compute the output and its derivative* $\frac{da}{dx}$ *when* $x = 4$*.*

Exercise 109 *An MLP has 2 input neurons, 2 hidden neurons, and 1 output neuron. Weights from input to hidden:* $W_1 = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}$*, hidden to output:* $W_2 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$*. Biases are zero. Input* $x = [1, 2]^T$ *and ReLU activation is used. Compute the output of the network.*

Exercise 110 *An MLP with 3 hidden neurons receives input* $x = [1, 0, -1]$*, weights* $W = [[1, 0, -1], [0, 1, 1], [1, 1, 0]$ *biases* $b = [0, 1, -1]$*. Activation function: ReLU. Compute the output of hidden layer.*

Exercise 111 *A single-hidden-layer MLP has 4 input neurons and 5 hidden neurons. How many weight parameters are there (excluding biases)? Provide numeric answer.*

Exercise 112 *Consider an MLP for regression with sigmoid output. The target range is [0,1]. Input* $x = [0.5, 0.5]$*, weights* $w = [1, 1]$*, bias=0. Compute the output.*

Exercise 113 *An MLP has 2 hidden layers with 3 and 2 neurons. Compute total number of weights including biases if input layer has 2 neurons and output layer has 1 neuron.*

Exercise 114 *For an MLP with 2 hidden layers (4 and 3 neurons) and ReLU activation, a hidden neuron outputs zero for all inputs. What is the likely cause? (Enter 1 for dead neuron, 2 for underfitting)*

Exercise 115 *Input* $x = [1, 2]$*, hidden layer weights* $W = [[1, 1], [0, 1]]$*, bias=0, activation ReLU. Output layer weight* $v = [1, 2]$*, bias=0. Compute final network output.*

Exercise 116 *An MLP with 1 hidden layer (3 neurons) and input* $x = [-1, 2]$*, weights* $W = [[1, 2], [0, -1], [1, 1]]$*, bias* $b = [0, 1, 0]$*, ReLU activation. Compute hidden layer outputs.*

Exercise 117 *MLP uses softmax output for 3-class classification. Input to output layer* $z = [1, 2, 3]$*. Compute softmax probabilities. Provide numeric answer for class 3 probability rounded to 2 decimal places.*

Exercise 118 *An MLP with 1 hidden layer has ReLU activations. Input* $x = [2, -1]$*, hidden weights* $W = [[1, -1], [0, 2]]$*, output weights* $v = [1, 1]$*, biases=0. Compute network output.*

6.7 YouTube Links and QR Codes

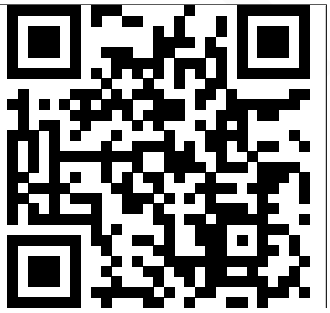
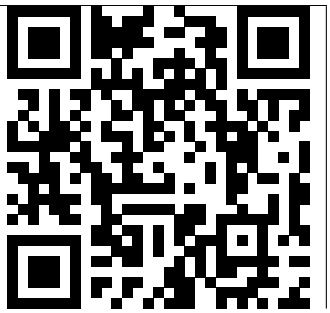
Lecture	Details	YouTube Link	QR Code
30	Decision Tree Classifier Explained: Gini, Entropy & Example	https://youtu.be/oHNJ9axvwmg	
31	Problem Solving – Decision Tree Gini & Entropy (Solutions 175 - 196)	https://youtu.be/TjwD5icVARs	
32	SVM Hard Margin Explained: Geometry, Optimization & Example	https://youtu.be/SY3GFjk98QA	
33	SVM Soft Margin Explained: Geometry, Optimization & Example	https://youtu.be/CjmkOeezUkM	

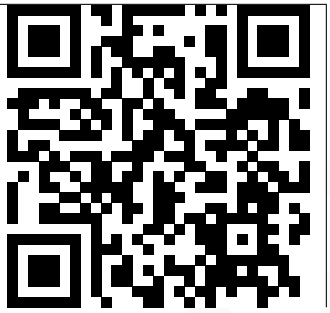
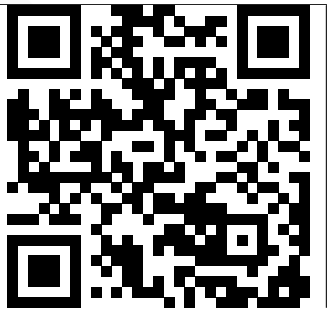
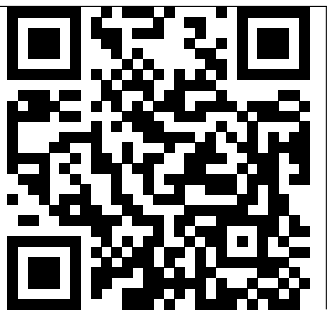
34	Kernel Trick Explained: Types of Kernel Functions with Intuition	https://youtu.be/kzHxkbaZ4jo	
35	Problem Solving – Support Vector Machines & Kernel Trick (Solutions 197 - 225)	https://youtu.be/VIgUxWse-jQ	
36	Neural Networks Explained: Neuron, Feedforward & Backpropagation Basics	https://youtu.be/UOZUAlSbdsU	
37	Problem Solving – ReLU & Neural Networks Basics (Solutions 226 - 250)	https://youtu.be/uSOWgKyjOsY	

Chapter 7

Solutions

Problems Covered	YouTube Link	QR Code
Solutions to Problems 1–11 (Lecture 4)	https://youtu.be/ouXrZEQPM8Q	
Solutions to Problems 12–23 (Lecture 7)	https://youtu.be/KT0gwV4ca7U	
Solutions to Problems 24–41 (Lecture 10)	https://youtu.be/z0ffEOi5TRk	

Solutions to Problems 42–64 (Lecture 12)	https://youtu.be/-NYU26KhVkg	
Solutions to Problems 65–79 (Lecture 14)	https://youtu.be/e7BAHUZ7eMs	
Solutions to Problems 80–92 (Lecture 16)	https://youtu.be/Uhs6QLruEKA	
Solutions to Problems 93–113 (Lecture 22)	https://youtu.be/3w7FO4h34RQ	
Solutions to Problems 114–133 (Lecture 24)	https://youtu.be/q7ouYk2Sk5c	

Solutions to Problems 134–143 (Lecture 26)	https://youtu.be/oYJAywqVvnE	
Solutions to Problems 144–174 (Lecture 29)	https://youtu.be/UlZL0ncHj74	
Solutions to Problems 175–196 (Lecture 31)	https://youtu.be/TjwD5icVARs	
Solutions to Problems 197–225 (Lecture 35)	https://youtu.be/VIgUxWse-jQ	
Solutions to Problems 226–250 (Lecture 37)	https://youtu.be/uSOWgKyjOsY	

Bibliography

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006. Available at: https://link.springer.com/book/9780387310732

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016. Available at: https://www.deeplearningbook.org

[3] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997. Available at: https://www.cs.cmu.edu/~tom/mlbook.html

[4] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed., O'Reilly Media, 2019. Available at: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

[5] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer, 2009. Available at: https://web.stanford.edu/~hastie/ElemStatLearn/

GateXAIML

Free GATE resources for Data Science & AI and CSE

Website: www.gatexaiml.in

Email: contact@gatexaiml.in