Bias-Variance and Cross Validation

Nipun Batra and teaching staff

IIT Gandhinagar

September 11, 2025

Table of Contents

Introduction to Bias-Variance

What is the Bias-Variance Tradeoff?

Important: The Central Challenge in Machine Learning

Every ML model faces a fundamental tension:

- Make simple assumptions $\rightarrow$ Miss important patterns (High Bias)
- Make complex assumptions $\rightarrow$ Overfit to noise (High Variance)

A Real-World Analogy: Weather Prediction

Example: Simple Model: ”Tomorrow = Today”

High Bias: Ignores weather patterns
Low Variance: Always makes same type of prediction

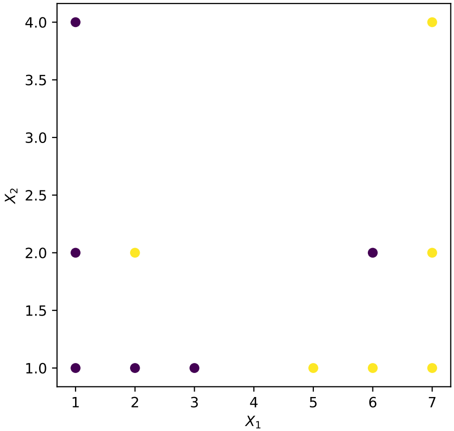
Example: Complex Model: 1000+ Variables

Low Bias: Can capture complex patterns
High Variance: Small errors $\rightarrow$ wildly different forecasts

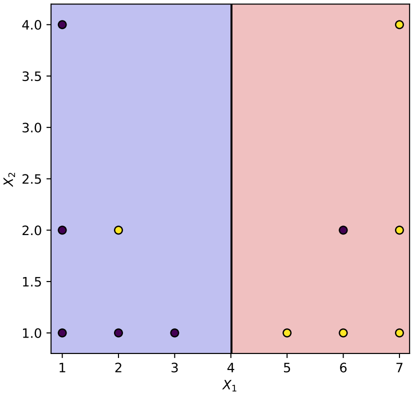
Goal: Find the sweet spot between these extremes

A Question!

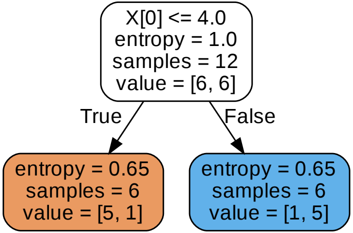
What would be the decision boundary of a decision tree classifier?



Decision Boundary for a tree with depth 1

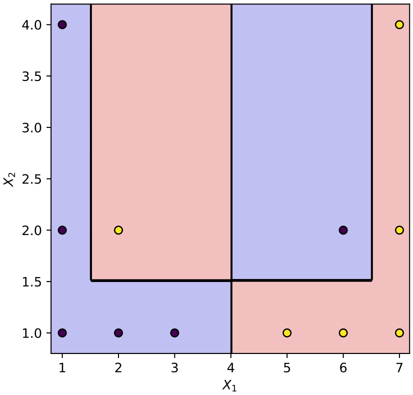


Decision Boundary

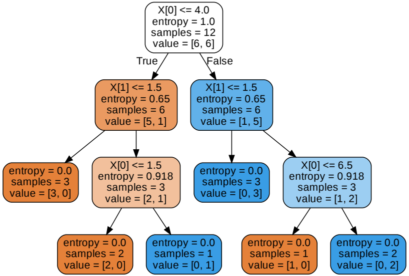


Decision Tree

Decision Boundary for a tree with no depth limit



Decision Boundary



Decision Tree

Are deeper trees always better?

As we saw, deeper trees learn more complex decision boundaries.

Are deeper trees always better?

As we saw, deeper trees learn more complex decision boundaries.

But, sometimes this can lead to poor generalization

The Fundamental Question: Model Complexity

Important: What We Just Observed

- **Depth 1**: Simple boundary, might miss patterns (underfitting)
- **No depth limit**: Complex boundary, might memorize noise (overfitting)

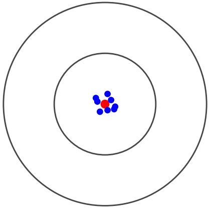
Key Points:

This Leads to Three Key Concepts:

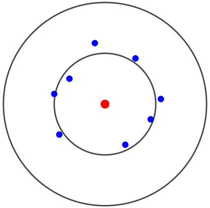
1. **Bias**: How much do our assumptions limit our model's ability to learn?
2. **Variance**: How much does our model change with different training data?
3. **Irreducible Error**: The noise we can never eliminate

The Bias-Variance Tradeoff: We can't minimize both bias and variance simultaneously!

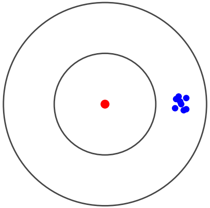
Dartboard Analogy: Four Scenarios



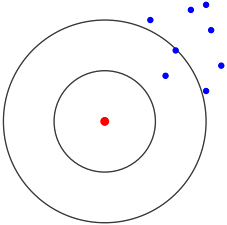
Low Bias, Low Var
Best



Low Bias, High Var
Inconsistent



High Bias, Low Var
Consistent & wrong



High Bias, High Var
Worst

Mathematical Foundation: Bias-Variance Decomposition

Definition: The Fundamental Equation

For any learning algorithm, the expected prediction error can be decomposed as:

$$\text{Expected Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Where:

- $\text{Bias}^2 = (\mathbb{E}[\hat{f}(x)] - f(x))^2$
  Squared difference between average prediction and true function
- $\text{Variance} = \mathbb{E}[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2]$
  Expected squared deviation from average prediction
- $\text{Irreducible Error} = \sigma^2$
  Noise in the data that no model can eliminate

Intuitive Understanding of Each Component

Example: Bias: "Are we systematically wrong?"

- High Bias: Linear model fitting curved data
- Low Bias: Flexible model that can approximate true function
- **Think**: Average error if we could train on infinite datasets

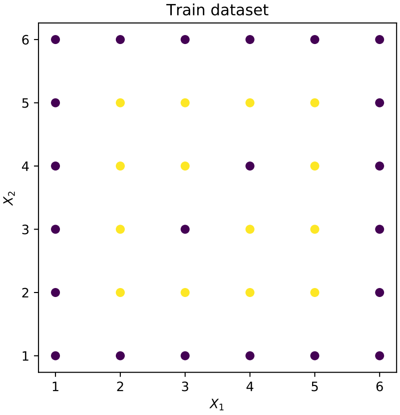
Example: Variance: "Are we consistently wrong?"

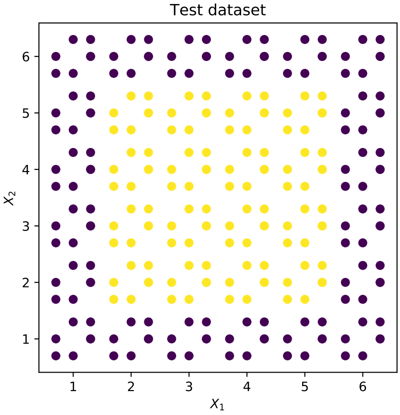
- High Variance: Model predictions change drastically with new training data
- Low Variance: Model predictions remain stable across different datasets
- **Think**: How much do predictions fluctuate between training runs?

Key Insight: Both contribute to total error, but reducing one often increases the other!

An example

Consider the dataset below





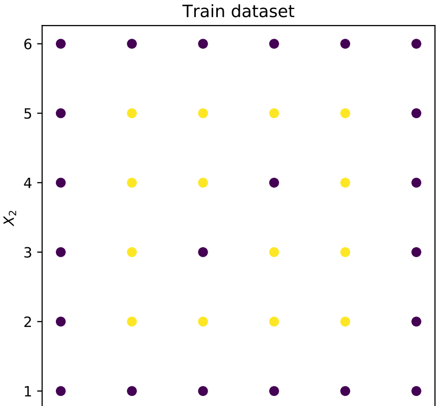
Train Set

Test Set

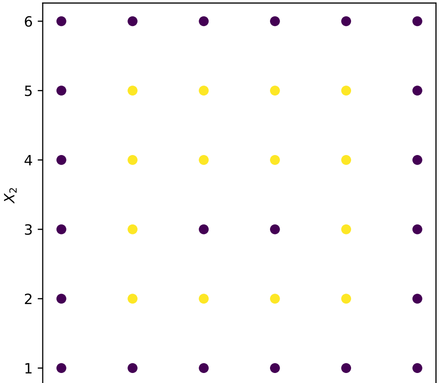
Intuition for Variance

A small change in data can lead to very different models.

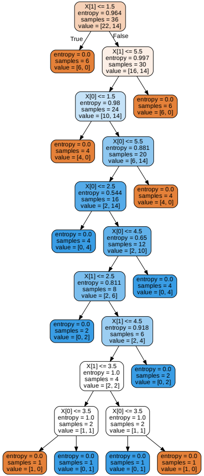
Dataset 1

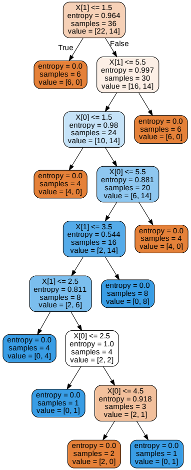


Dataset 2



Intuition for Variance





High Bias Example: Medical Diagnosis

Example: Oversimplified Medical Model

Model: "All patients with fever have flu"

- High Bias: Ignores other symptoms, medical history
- Systematic Error: Misses pneumonia, COVID, etc.
- Poor Performance: Wrong diagnoses even with more data

Key Points:

Solution: Increase model complexity, add features, use more flexible algorithms

High Bias Example: Financial Prediction

Example: Oversimplified Stock Model

Model: "Stock price only depends on previous day's price"

- High Bias: Ignores market conditions, company news, economic indicators
- Systematic Error: Misses major trend changes
- Poor Performance: Consistently wrong about market direction

Key Points:

Key Insight: High bias models have systematic blind spots that more data cannot fix

High Variance Example: Image Recognition

Example: Overcomplicated Vision Model

Model: Deep network with 1000 layers on small dataset

- High Variance: Different training sets $\rightarrow$ completely different filters
- Memorization: Learns specific pixel patterns, not object features
- Poor Generalization: Fails on slightly different images

Key Points:

Solution: Reduce model complexity, regularization, more training data

High Variance Example: Text Classification

Example: Overcomplicated Text Model

Model: Memorizing entire sentences instead of key words

- High Variance: Adding one new training email changes all predictions
- Memorization: Learns exact phrases, not semantic meaning
- Poor Generalization: Fails on paraphrased or slightly different text

Key Points:

Key Insight: High variance models are too sensitive to training data variations

The Bias-Variance Tradeoff Visualized

Example: The Fundamental Tradeoff

- **Increase Model Complexity**:
  - Bias $\downarrow$ (can fit more complex patterns)
  - Variance $\uparrow$ (more sensitive to training data)
- **Decrease Model Complexity**:
  - Bias $\uparrow$ (makes stronger assumptions)
  - Variance $\downarrow$ (more stable predictions)

Key Points:

The Sweet Spot: Find the complexity that minimizes: $\text{Bias}^2 + \text{Variance} + \text{Noise}$

Different algorithms have different bias-variance profiles!

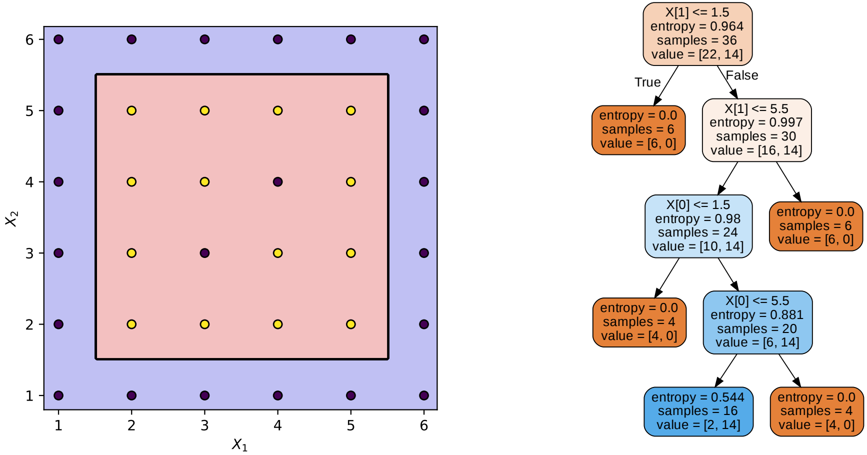
Comparing Algorithms: Bias-Variance Profiles

Definition: Algorithm Characteristics

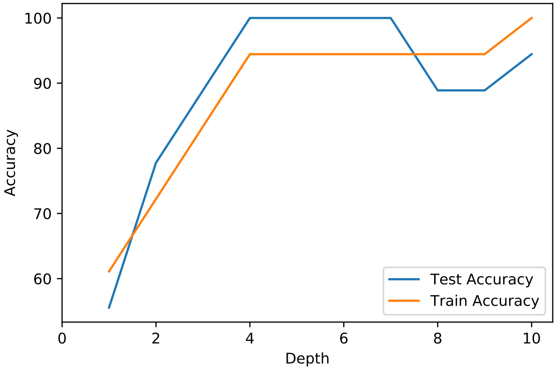
- **k-NN with small k**: Low bias, high variance
- **k-NN with large k**: High bias, low variance
- **Linear Regression**: High bias, low variance
- **Deep Neural Networks**: Low bias, high variance (without regularization)
- **Decision Trees**: Low bias, high variance
- **Random Forest**: Lower variance than single trees

Key Insight: No single algorithm is best for all problems!

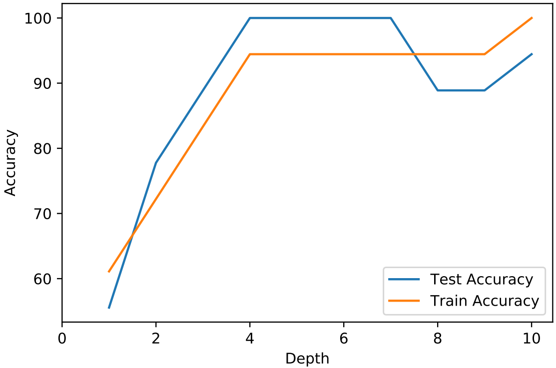
A Good Fit: Finding the Sweet Spot



Accuracy vs Depth Curve

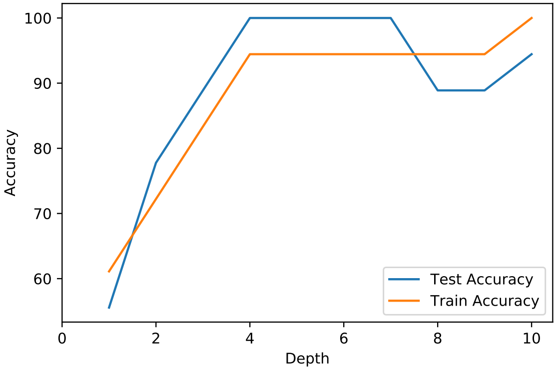


Accuracy vs Depth Curve



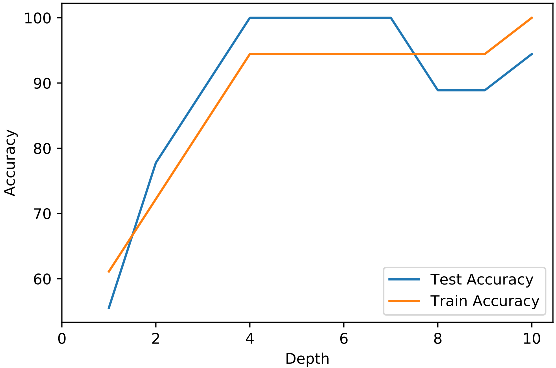
As depth increases, train accuracy improves

Accuracy vs Depth Curve



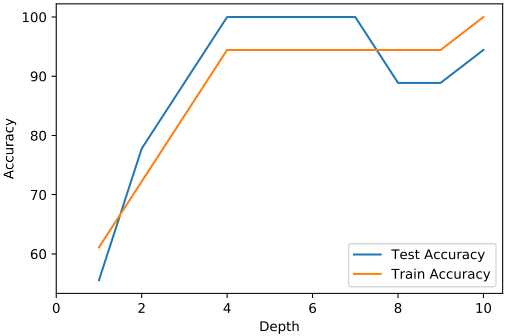
As depth increases, train accuracy improves
As depth increases, test accuracy improves till a point

Accuracy vs Depth Curve



As depth increases, train accuracy improves
As depth increases, test accuracy improves till a point
At very high depths, test accuracy is not good (overfitting).

Accuracy vs Depth: Understanding All Three Regions

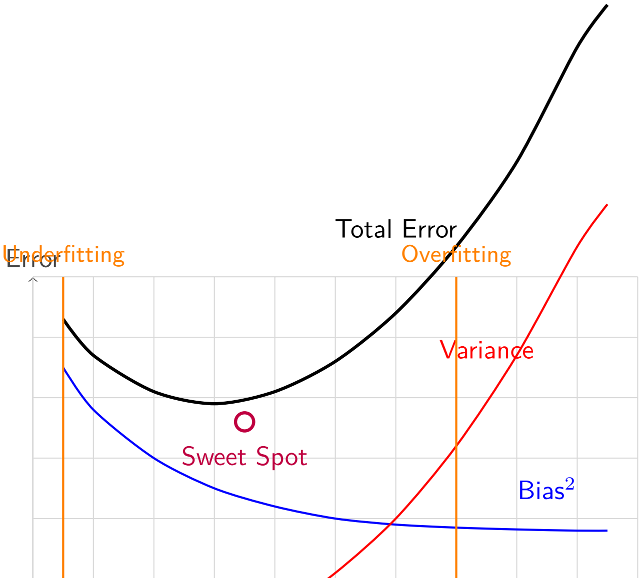


Key Points:

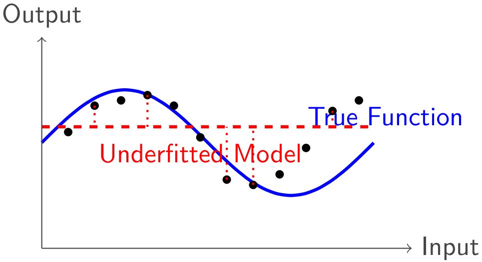
Three Key Regions:

- **Underfitting**: Too simple models, poor performance on both training and test

The Complete Picture: Bias-Variance Through Model Complexity



Underfitting Visualized: When Models Are Too Simple

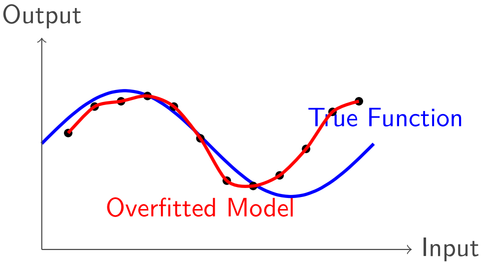


Important: High Bias Problem

- **Model**: Too simple (constant)
- **Assumption**: "Output never changes"
- **Result**: Systematic error
- **Training Error**: High
- **Test Error**: High

Solution: Increase model complexity

Overfitting Visualized: When Models Are Too Complex

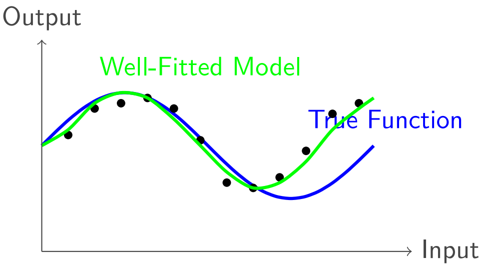


Important: High Variance Problem

- **Model**: Too complex (memorizes)
- **Assumption**: "Fit every data point exactly"
- **Result**: Learns noise
- **Training Error**: Very low
- **Test Error**: High

Solution: Reduce complexity or regularize

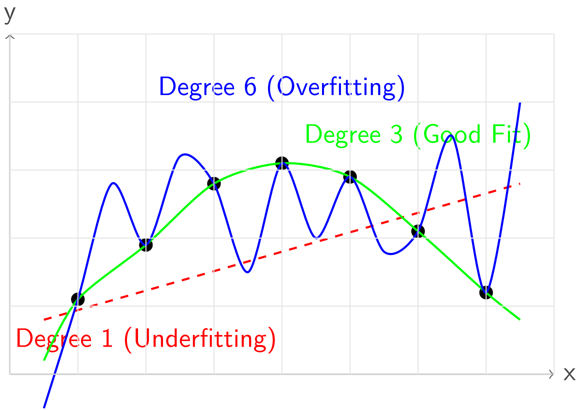
Good Fit: The Sweet Spot



Example: Goldilocks Principle

- **Model**: Just right complexity
- **Result**: Good generalization
- **Errors**: Training moderate, test low

Interactive Example: Polynomial Fitting



Question: Which polynomial would you choose and why?

The big question!?

How to find the optimal depth for a decision tree?

The big question!?

How to find the optimal depth for a decision tree?

Use cross-validation!

The Problem: How Do We Find the Sweet Spot?

Important: The Fundamental Challenge

- Can't use test data to select model complexity (that's cheating!)
- Can't trust training error (always decreases with complexity)
- Need unbiased estimate of generalization performance

Key Points:

Solution: Cross-validation provides honest estimates of generalization!

Why Training Error Fails for Model Selection

Example: Training Error is Optimistically Biased

- Complex models: Training error ≈ 0, but test error is high
- Simple models: Training error is high, test error might be high too
- Training error systematically underestimates true error

Key Points:

Key Insight: Models that fit training data perfectly often fail on new data

Cross-Validation: The Core Idea

Definition: The Philosophy

Simulate having multiple independent test sets by:

1. Split training data into multiple folds
2. Train on some folds, validate on others
3. Rotate which folds are used for validation
4. Average the validation performance

Benefits of Cross-Validation

Key Points:

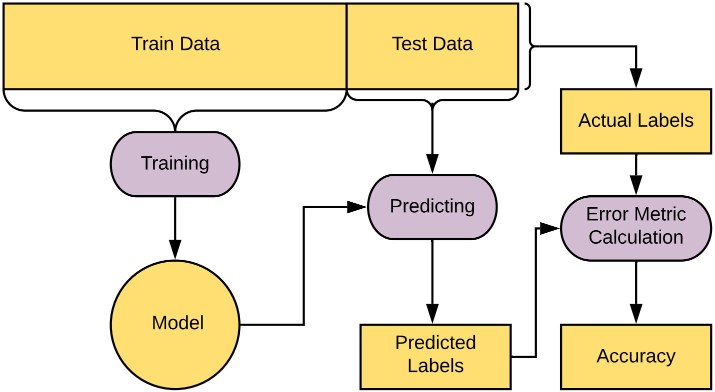
Key Benefits:

- Uses all data for both training and validation
- Provides robust estimates vs single train/validation split
- Reduces dependence on particular data split
- Helps detect overfitting to validation set

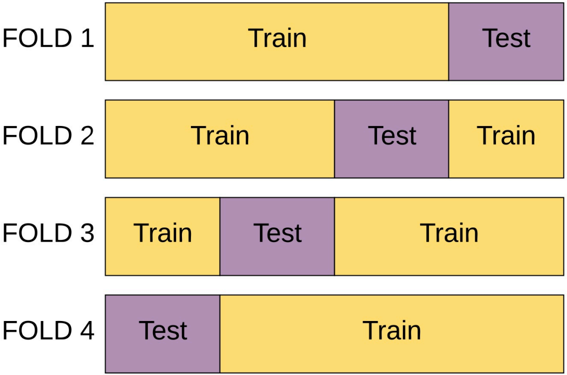
Important: Important

Cross-validation gives us honest estimates for model selection!

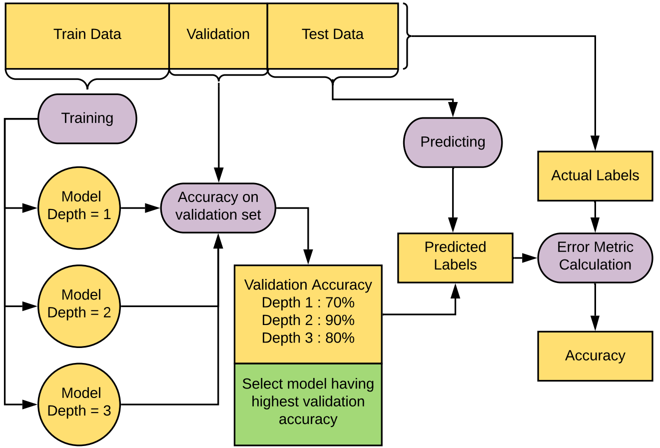
Our General Training Flow



K-Fold cross-validation: Utilise full dataset for testing

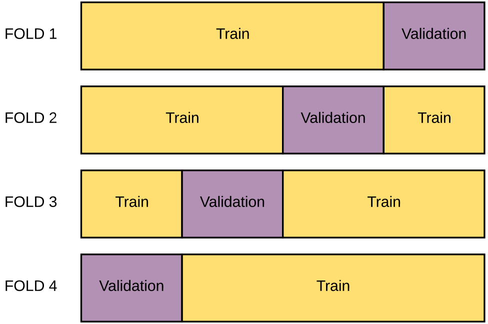


The Validation Set



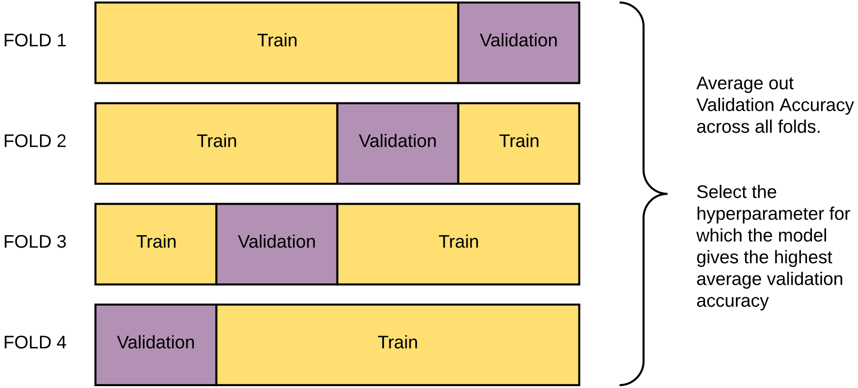
Nested Cross Validation

Divide your training set into k equal parts.
Cyclically use 1 part as "validation set" and the rest for training.
Here $k = 4$



Nested Cross Validation

Average out the validation accuracy across all the folds
Use the model with highest validation accuracy



Next time: Ensemble Learning

- How to combine various models?
- Why to combine multiple models?
- How can we reduce bias?
- How can we reduce variance?