Cross-Validation

Nipun Batra and teaching staff

IIT Gandhinagar

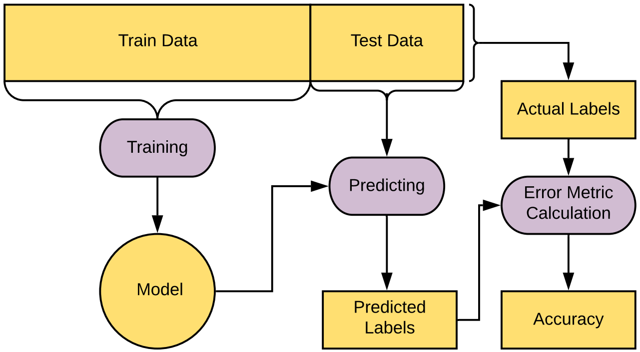
August 29, 2025

Introduction to Cross-Validation

Outline

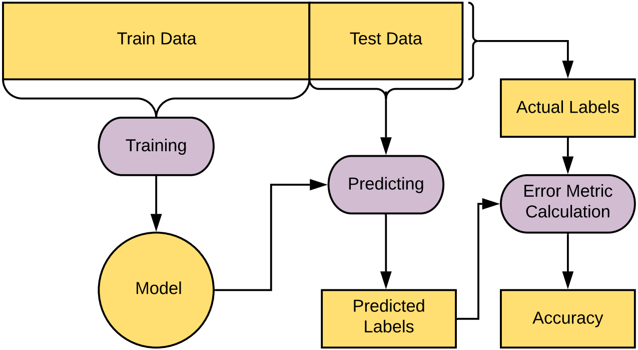
1. Introduction to Cross-Validation
2. Full Dataset Utilization
3. K-Fold Cross-Validation
4. Hyperparameter Optimization
5. Nested Cross-Validation
6. Cross-Validation Variants
7. Time Series Cross-Validation
8. Common Pitfalls and Best Practices
9. Summary and Key Takeaways

Our General Training Flow



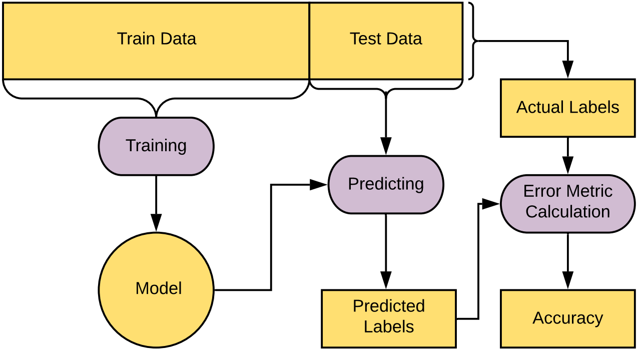
- Does not use the full dataset for training and does not test on the full dataset

Our General Training Flow



- Does not use the full dataset for training and does not test on the full dataset
- No way to optimize hyperparameters

Our General Training Flow



- Does not use the full dataset for training and does not test on the full dataset
- No way to optimize hyperparameters
- This simple train/test split has limitations we need to address

Pop Quiz #1

Answer this!

What are the main limitations of using only a single train/test split?

Pop Quiz #1

Answer this!

What are the main limitations of using only a single train/test split?

Answer:

- Does not utilize the full dataset for training
- Cannot optimize hyperparameters systematically
- Results depend on the particular split chosen
- May not get reliable performance estimates

Full Dataset Utilization

How to use the full dataset for training?

- Over multiple iterations, use different parts of the dataset for training and testing

How to use the full dataset for training?

- Over multiple iterations, use different parts of the dataset for training and testing
- Typically done via different random splits of the dataset

How to use the full dataset for training?

- Over multiple iterations, use different parts of the dataset for training and testing
- Typically done via different random splits of the dataset
- **Challenge:** How to ensure systematic evaluation?

How to use the full dataset for training?

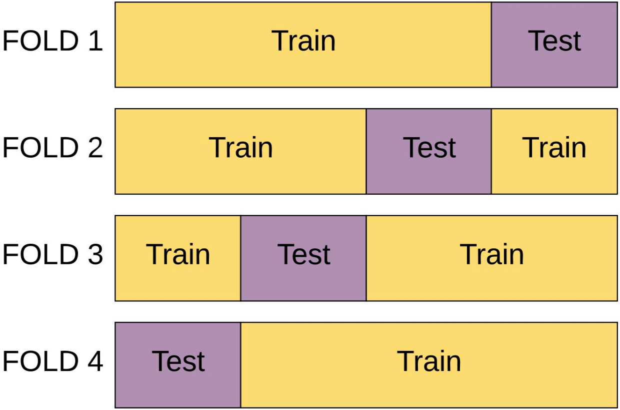
- Over multiple iterations, use different parts of the dataset for training and testing
- Typically done via different random splits of the dataset
- **Challenge:** How to ensure systematic evaluation?
- May not use every data point for training or testing with random splits

How to use the full dataset for training?

- Over multiple iterations, use different parts of the dataset for training and testing
- Typically done via different random splits of the dataset
- **Challenge:** How to ensure systematic evaluation?
- May not use every data point for training or testing with random splits
- May be computationally expensive

K-Fold Cross-Validation

K-Fold Cross-Validation: Utilize Full Dataset for Testing



K-Fold Cross-Validation: Utilize Full Dataset for Testing

- Each data point is used for testing exactly once

K-Fold Cross-Validation: Utilize Full Dataset for Testing

- Each data point is used for testing exactly once
- Each data point is used for training $(k-1)/k$ of the time

K-Fold Cross-Validation: Utilize Full Dataset for Testing

- Each data point is used for testing exactly once
- Each data point is used for training $(k-1)/k$ of the time
- Provides more robust performance estimates

Pop Quiz #2

Answer this!

If you have 100 data points and use 5-fold cross-validation, how many data points are used for training in each fold?

Pop Quiz #2

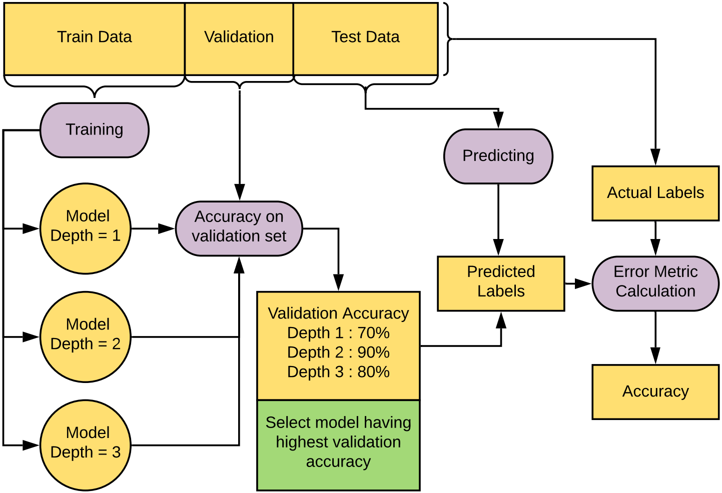
Answer this!

If you have 100 data points and use 5-fold cross-validation, how many data points are used for training in each fold?

Answer: **80 data points** (4 out of 5 folds $= 4/5 \times 100 = 80$)

Hyperparameter Optimization

Optimizing Hyperparameters via the Validation Set



Optimizing Hyperparameters via the Validation Set

- Validation set helps select the best hyperparameters

Optimizing Hyperparameters via the Validation Set

- Validation set helps select the best hyperparameters
- Test set remains untouched until final evaluation

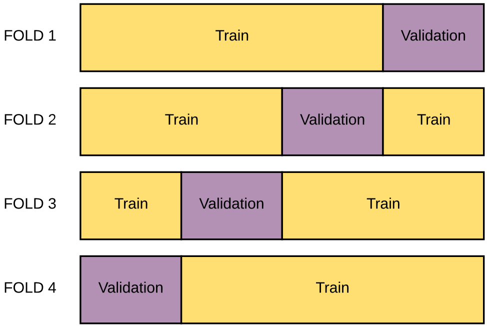
Optimizing Hyperparameters via the Validation Set

- Validation set helps select the best hyperparameters
- Test set remains untouched until final evaluation
- This prevents overfitting to the test set

Nested Cross-Validation

Nested Cross-Validation Process

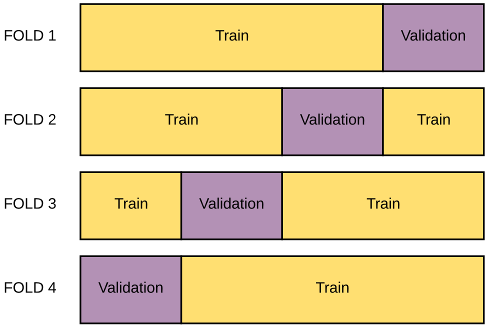
Divide your training set into k equal parts.
Cyclically use 1 part as "validation set" and the rest for training.
Here $k = 4$



- Each fold provides one validation score

Nested Cross-Validation Process

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



- Each fold provides one validation score
- Process is systematic and exhaustive

Pop Quiz #3

Answer this!

What is the difference between simple cross-validation and nested cross-validation?

Pop Quiz #3

Answer this!

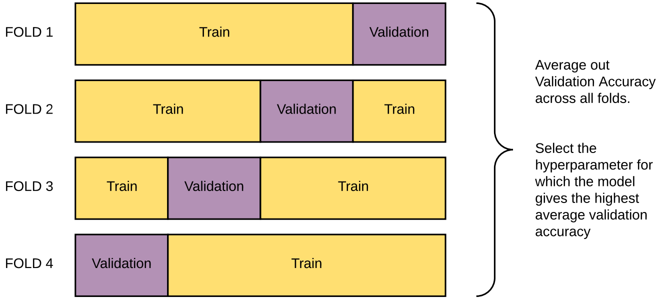
What is the difference between simple cross-validation and nested cross-validation?

Answer:

- **Simple CV**: Used for model evaluation only
- **Nested CV**: Outer loop for model evaluation, inner loop for hyperparameter tuning
- **Nested CV** provides unbiased estimates when doing hyperparameter search

Cross-Validation Results

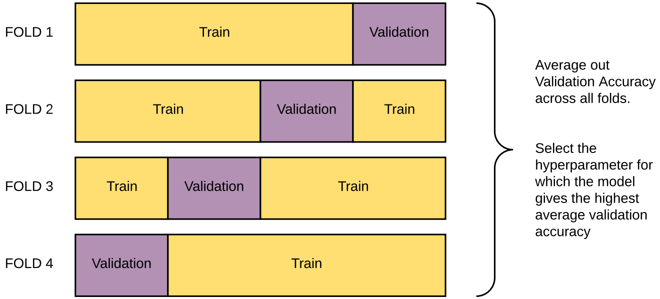
Average out the validation accuracy across all the folds
Use the hyperparameters with highest average validation accuracy



- Final model is trained on entire training set

Cross-Validation Results

Average out the validation accuracy across all the folds
Use the hyperparameters with highest average validation accuracy



- Final model is trained on entire training set
- Standard deviation gives confidence in results

Pop Quiz #4

Answer this!

Why do we average the results across all folds instead of picking the best single fold?

Pop Quiz #4

Answer this!

Why do we average the results across all folds instead of picking the best single fold?

Answer:

- Single fold results can be misleading due to data variance
- Averaging provides more robust performance estimates
- Reduces impact of lucky/unlucky splits
- Standard deviation indicates reliability of the estimate

Cross-Validation Variants

Leave-One-Out Cross-Validation (LOOCV)

- Special case where $k = n$ (number of data points)

Leave-One-Out Cross-Validation (LOOCV)

- Special case where $k = n$ (number of data points)
- Each fold uses exactly one data point for testing

Leave-One-Out Cross-Validation (LOOCV)

- Special case where $k = n$ (number of data points)
- Each fold uses exactly one data point for testing
- **Advantages:**

Leave-One-Out Cross-Validation (LOOCV)

- Special case where $k = n$ (number of data points)
- Each fold uses exactly one data point for testing
- **Advantages:**
  - Maximum use of data for training

Leave-One-Out Cross-Validation (LOOCV)

- Special case where $k = n$ (number of data points)
- Each fold uses exactly one data point for testing
- **Advantages:**
  - Maximum use of data for training
  - Deterministic (no randomness)

Leave-One-Out Cross-Validation (LOOCV)

- Special case where $k = n$ (number of data points)
- Each fold uses exactly one data point for testing
- **Advantages:**
  - Maximum use of data for training
  - Deterministic (no randomness)
- **Disadvantages:**

Leave-One-Out Cross-Validation (LOOCV)

- Special case where $k = n$ (number of data points)
- Each fold uses exactly one data point for testing
- **Advantages:**
  - Maximum use of data for training
  - Deterministic (no randomness)
- **Disadvantages:**
  - Computationally expensive

Leave-One-Out Cross-Validation (LOOCV)

- Special case where $k = n$ (number of data points)
- Each fold uses exactly one data point for testing
- **Advantages:**
  - Maximum use of data for training
  - Deterministic (no randomness)
- **Disadvantages:**
  - Computationally expensive
  - High variance in estimates

Stratified Cross-Validation

- Maintains class distribution in each fold

Stratified Cross-Validation

- Maintains class distribution in each fold
- Important for imbalanced datasets

Stratified Cross-Validation

- Maintains class distribution in each fold
- Important for imbalanced datasets
- Each fold has approximately same proportion of classes

Stratified Cross-Validation

- Maintains class distribution in each fold
- Important for imbalanced datasets
- Each fold has approximately same proportion of classes
- **Example:** If dataset is 70% class A, 30% class B, each fold maintains this ratio

Stratified Cross-Validation

- Maintains class distribution in each fold
- Important for imbalanced datasets
- Each fold has approximately same proportion of classes
- **Example:** If dataset is 70% class A, 30% class B, each fold maintains this ratio
- Reduces variance in performance estimates

Pop Quiz #5

Answer this!

You have a binary classification dataset with 90% negative and 10% positive examples. Why is stratified cross-validation important here?

Pop Quiz #5

Answer this!

You have a binary classification dataset with 90% negative and 10% positive examples. Why is stratified cross-validation important here?

Answer:

- Regular CV might create folds with very few (or zero) positive examples
- This would give misleading performance estimates
- Stratified CV ensures each fold has $\sim$10% positive examples
- Results in more reliable and consistent evaluation

Time Series
Cross-Validation

Time Series Cross-Validation

- Regular CV assumes data points are independent

Time Series Cross-Validation

- Regular CV assumes data points are independent
- Time series data has temporal dependencies

Time Series Cross-Validation

- Regular CV assumes data points are independent
- Time series data has temporal dependencies
- **Forward Chaining:** Train on past, test on future

Time Series Cross-Validation

- Regular CV assumes data points are independent
- Time series data has temporal dependencies
- **Forward Chaining:** Train on past, test on future
- **Rolling Window:** Fixed-size training window

Time Series Cross-Validation

- Regular CV assumes data points are independent
- Time series data has temporal dependencies
- **Forward Chaining:** Train on past, test on future
- **Rolling Window:** Fixed-size training window
- **Expanding Window:** Growing training set over time

Time Series Cross-Validation

- Regular CV assumes data points are independent
- Time series data has temporal dependencies
- **Forward Chaining:** Train on past, test on future
- **Rolling Window:** Fixed-size training window
- **Expanding Window:** Growing training set over time
- Never use future data to predict past!

Common Pitfalls and Best Practices

Common Cross-Validation Mistakes

- **Data Leakage:** Information from test set influences training

Common Cross-Validation Mistakes

- **Data Leakage:** Information from test set influences training
- **Incorrect Splitting:** Not accounting for grouped data

Common Cross-Validation Mistakes

- **Data Leakage:** Information from test set influences training
- **Incorrect Splitting:** Not accounting for grouped data
- **Overfitting to CV:** Too much hyperparameter tuning

Common Cross-Validation Mistakes

- **Data Leakage:** Information from test set influences training
- **Incorrect Splitting:** Not accounting for grouped data
- **Overfitting to CV:** Too much hyperparameter tuning
- **Wrong Preprocessing:** Scaling on entire dataset before splitting

Common Cross-Validation Mistakes

- **Data Leakage:** Information from test set influences training
- **Incorrect Splitting:** Not accounting for grouped data
- **Overfitting to CV:** Too much hyperparameter tuning
- **Wrong Preprocessing:** Scaling on entire dataset before splitting
- **Ignoring Class Imbalance:** Not using stratified CV when needed

Pop Quiz #6

Answer this!

What's wrong with computing mean and standard deviation on the entire dataset before doing cross-validation?

Pop Quiz #6

Answer this!

What's wrong with computing mean and standard deviation on the entire dataset before doing cross-validation?

Answer:

- This causes data leakage!
- Test fold statistics influence the training preprocessing
- Should compute statistics only on training folds
- Apply same transformation to corresponding test fold
- This gives more realistic performance estimates

Summary and Key Takeaways

Cross-Validation: Key Benefits

- **Better Data Utilization:** Every point used for both training and testing

Cross-Validation: Key Benefits

- **Better Data Utilization:** Every point used for both training and testing
- **Robust Evaluation:** Multiple train/test splits reduce variance

Cross-Validation: Key Benefits

- **Better Data Utilization:** Every point used for both training and testing
- **Robust Evaluation:** Multiple train/test splits reduce variance
- **Hyperparameter Tuning:** Systematic way to select best parameters

Cross-Validation: Key Benefits

- **Better Data Utilization:** Every point used for both training and testing
- **Robust Evaluation:** Multiple train/test splits reduce variance
- **Hyperparameter Tuning:** Systematic way to select best parameters
- **Model Comparison:** Fair comparison between different algorithms

Cross-Validation: Key Benefits

- **Better Data Utilization:** Every point used for both training and testing
- **Robust Evaluation:** Multiple train/test splits reduce variance
- **Hyperparameter Tuning:** Systematic way to select best parameters
- **Model Comparison:** Fair comparison between different algorithms
- **Confidence Estimates:** Standard deviation indicates reliability

When to Use Different CV Types

- **K-Fold (k=5,10):** General purpose, most common

When to Use Different CV Types

- **K-Fold (k=5,10):** General purpose, most common
- **Stratified:** Imbalanced classification problems

When to Use Different CV Types

- **K-Fold (k=5,10):** General purpose, most common
- **Stratified:** Imbalanced classification problems
- **LOOCV:** Small datasets, when computational cost is acceptable

When to Use Different CV Types

- **K-Fold (k=5,10):** General purpose, most common
- **Stratified:** Imbalanced classification problems
- **LOOCV:** Small datasets, when computational cost is acceptable
- **Time Series CV:** Temporal data with dependencies

When to Use Different CV Types

- **K-Fold (k=5,10):** General purpose, most common
- **Stratified:** Imbalanced classification problems
- **LOOCV:** Small datasets, when computational cost is acceptable
- **Time Series CV:** Temporal data with dependencies
- **Nested CV:** When doing extensive hyperparameter search

Cross-Validation Best Practices

- Always preprocess within each fold separately

Cross-Validation Best Practices

- Always preprocess within each fold separately
- Use stratification for classification problems

Cross-Validation Best Practices

- Always preprocess within each fold separately
- Use stratification for classification problems
- Report mean $\pm$ standard deviation

Cross-Validation Best Practices

- Always preprocess within each fold separately
- Use stratification for classification problems
- Report mean $\pm$ standard deviation
- Don't overfit to cross-validation results

Cross-Validation Best Practices

- Always preprocess within each fold separately
- Use stratification for classification problems
- Report mean $\pm$ standard deviation
- Don't overfit to cross-validation results
- Consider computational cost vs. benefit trade-off

Cross-Validation Best Practices

- Always preprocess within each fold separately
- Use stratification for classification problems
- Report mean $\pm$ standard deviation
- Don't overfit to cross-validation results
- Consider computational cost vs. benefit trade-off
- Use nested CV for unbiased hyperparameter search

Next time: Ensemble Learning

- How to combine various models?

Next time: Ensemble Learning

- How to combine various models?
- Why combine multiple models?

Next time: Ensemble Learning

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

Next time: Ensemble Learning

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

Next time: Ensemble Learning

- How to combine various models?
- Why combine multiple models?
- How can we reduce bias?
- How can we reduce variance?
- Bootstrap aggregating (Bagging)

Next time: Ensemble Learning

- How to combine various models?
- Why combine multiple models?
- How can we reduce bias?
- How can we reduce variance?
- Bootstrap aggregating (Bagging)
- Boosting methods