Lecture: Model Selection and Evaluation Metrics

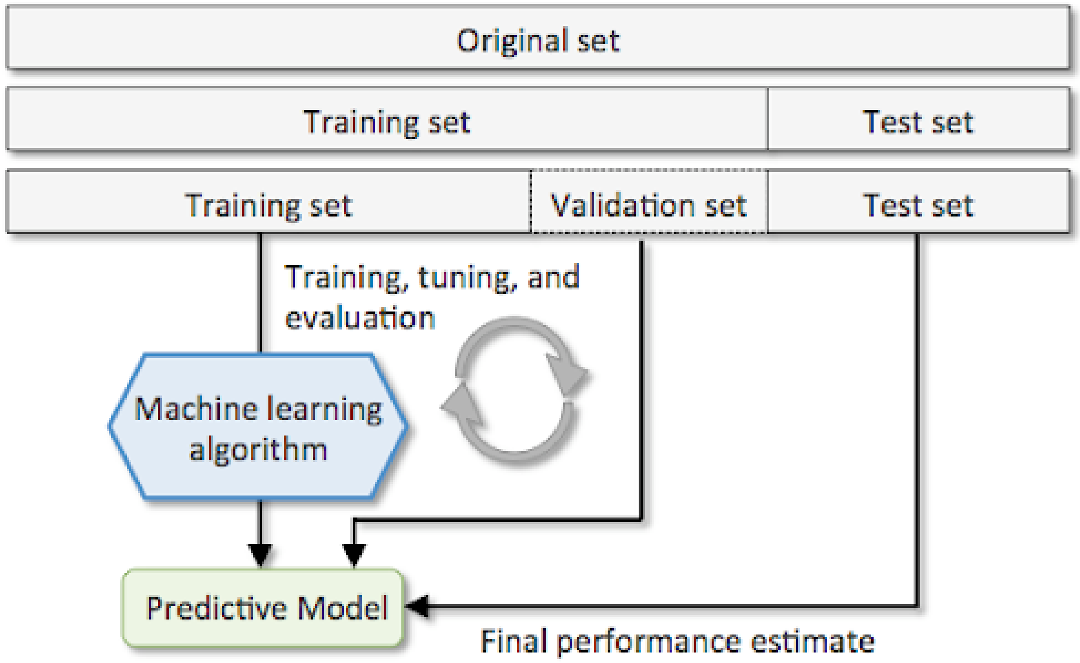
Dr. Amit Ranjan

School of Computer Science

Model Selection Process

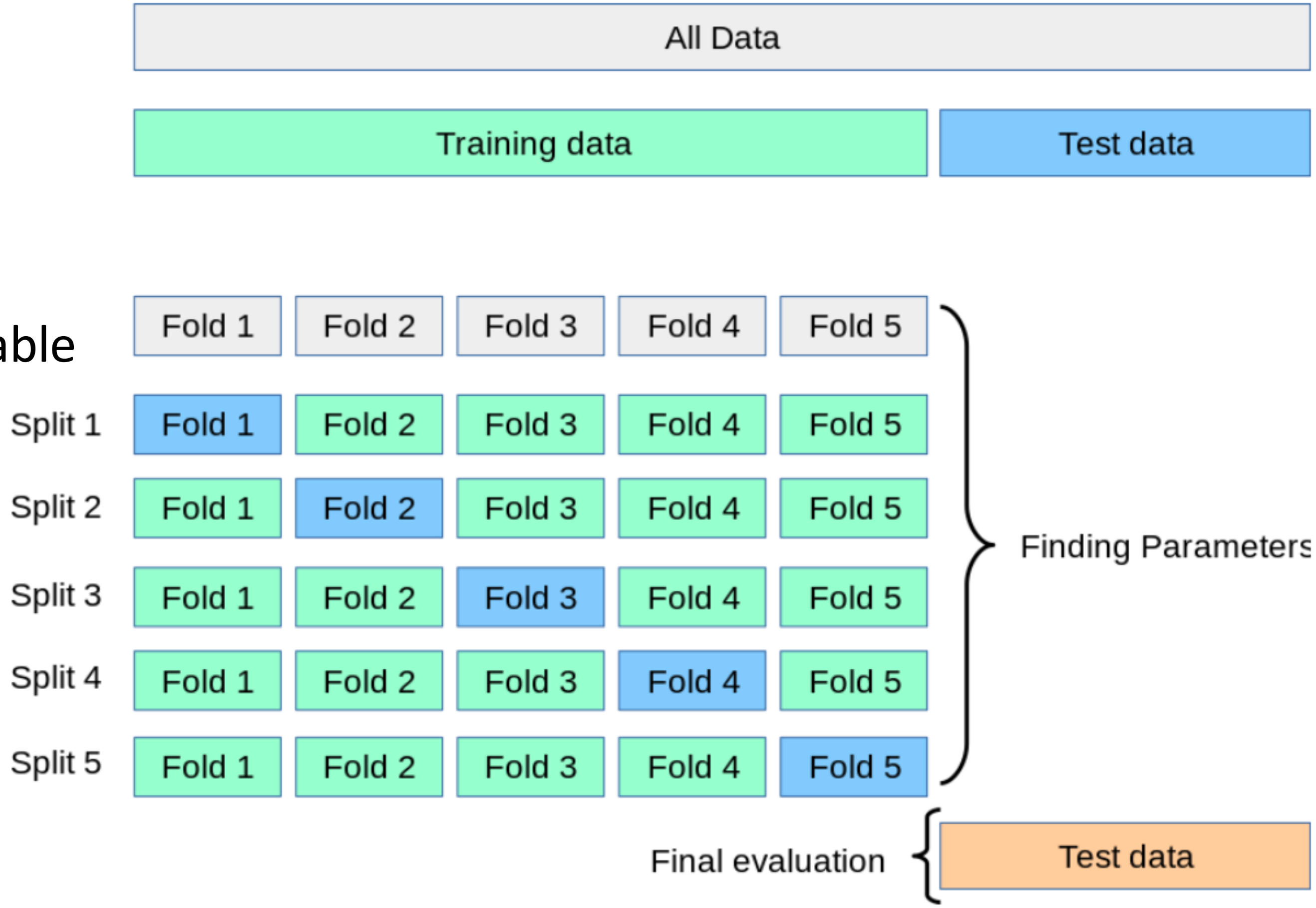
Hold-out Method:

- The dataset is split into two parts: a training set and a testing set.
- Common split ratio: 80% training, 20% testing.
- **Advantages:** Simple and computationally efficient.
- **Disadvantages:** High variance in performance estimation due to dependence on a single train-test split. Not ideal for small datasets.



Model Selection Process

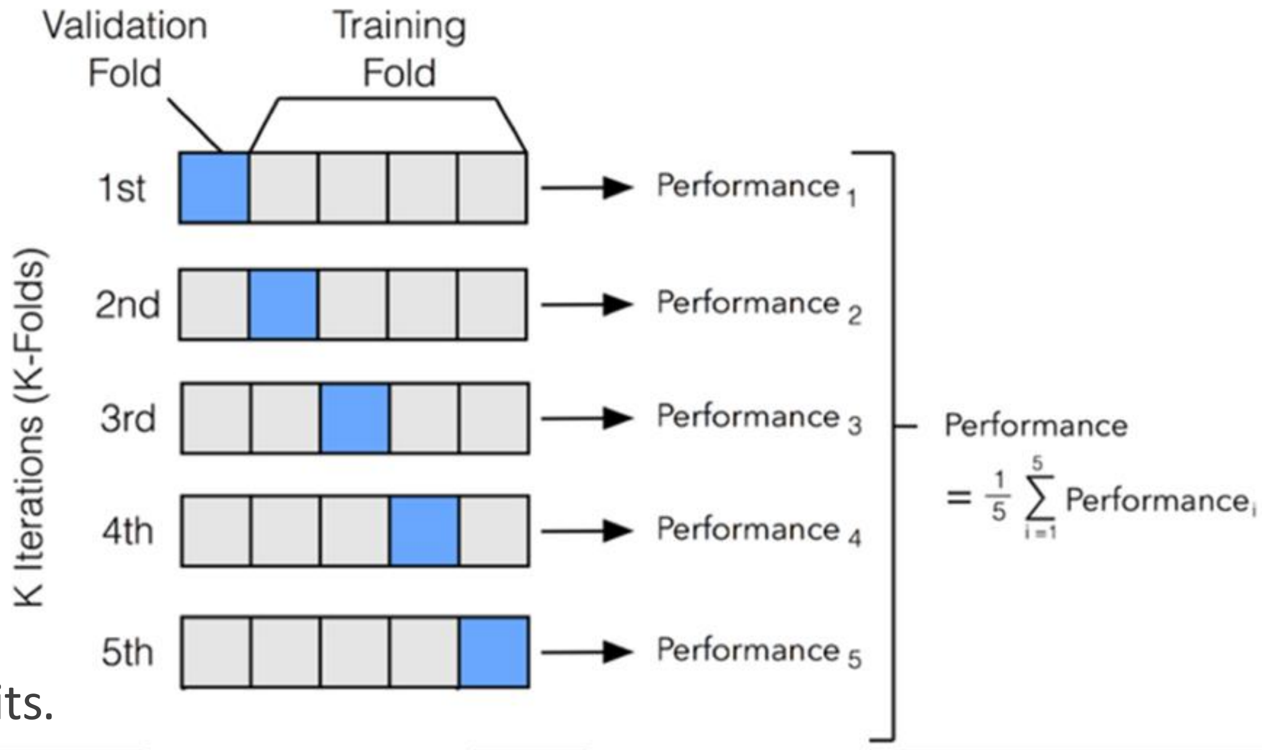
- **Cross-Validation**: Cross-validation is a statistical method used to estimate the performance of machine learning models.
- It helps prevent overfitting and ensures that the model generalizes well to unseen data.
- Cross-validation is particularly useful when the dataset is small, as it maximizes the use of available data for both training and testing.
- Types of Cross-Validation:
    - **K-Fold Cross-Validation**
    - **Stratified K-Fold Cross-Validation**
    - **Leave-One-Out Cross-Validation (LOOCV)**



Types of Cross-Validation

K-Fold Cross-Validation:

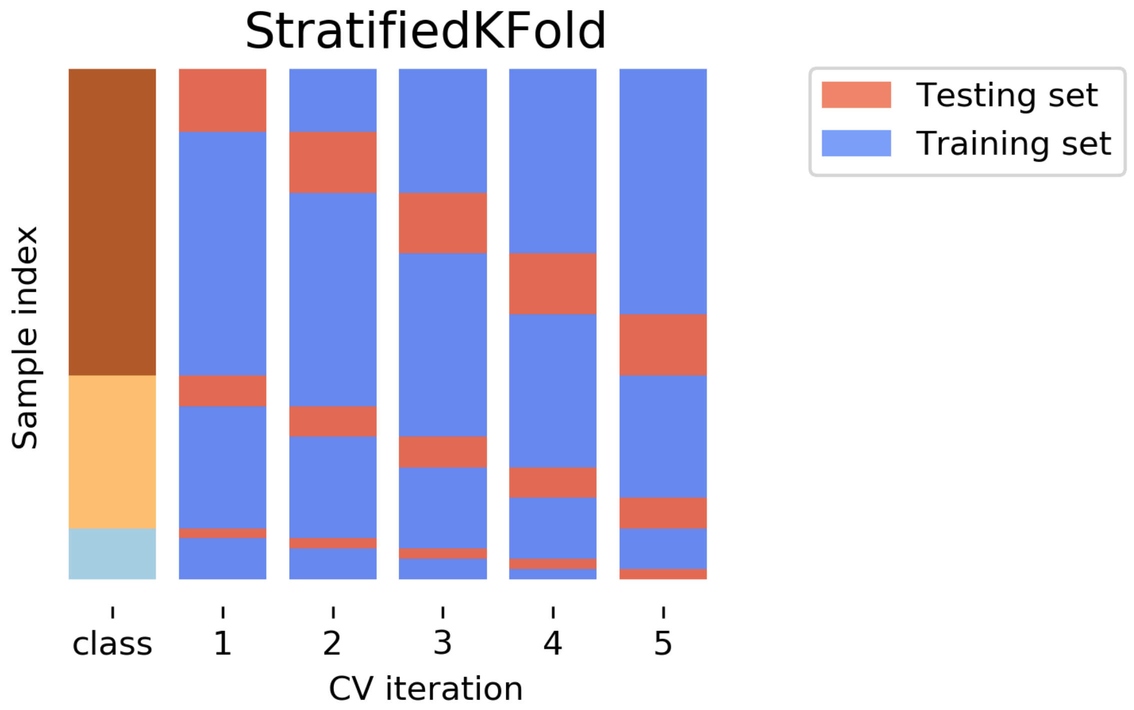
- The dataset is divided into K subsets (folds).
- The model is trained on K-1 folds and tested on the remaining fold.
- This process is repeated K times, each time with a different fold as the test set.
- The final performance is the average of the K performance metrics.
- Common choices: K=5 or K=10.
- **Advantages:** Provides a more reliable estimate of model performance by using multiple train-test splits.
- **Disadvantages:** Computationally expensive for large datasets or complex models.



Types of Cross-Validation

Stratified K-Fold Cross-Validation:

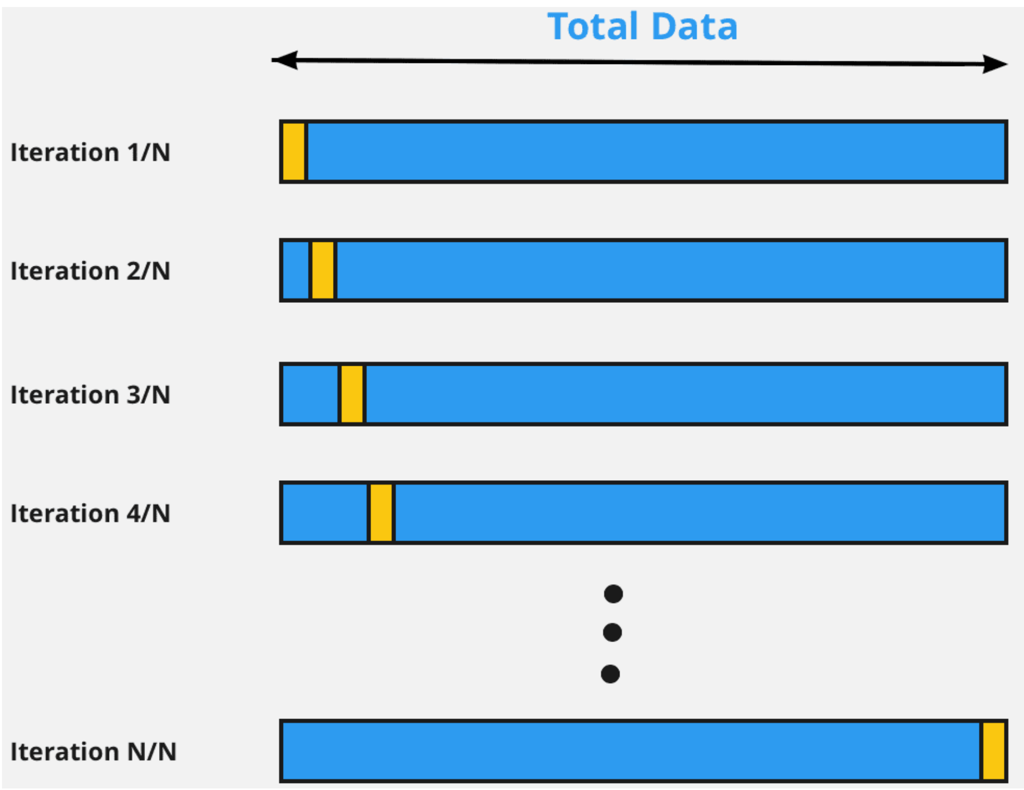
- Similar to K-Fold but ensures that each fold has the same proportion of classes as the original dataset.
- Particularly useful for imbalanced datasets where one class is significantly underrepresented.
- **Advantages:** Preserves class distribution, leading to more reliable performance estimates for imbalanced datasets.



Types of Cross-Validation

Leave-One-Out Cross-Validation (LOOCV):

- Each sample is used as a test set, while the rest are used for training.
- This process is repeated N times (where N is the number of samples).
- **Advantages:** Provides an almost unbiased estimate of model performance.
- **Disadvantages:** Computationally expensive, especially for large datasets.



Types of Cross-Validation

- **Mathematical Representation:**

For K-Fold cross-validation, the CV score is calculated as:

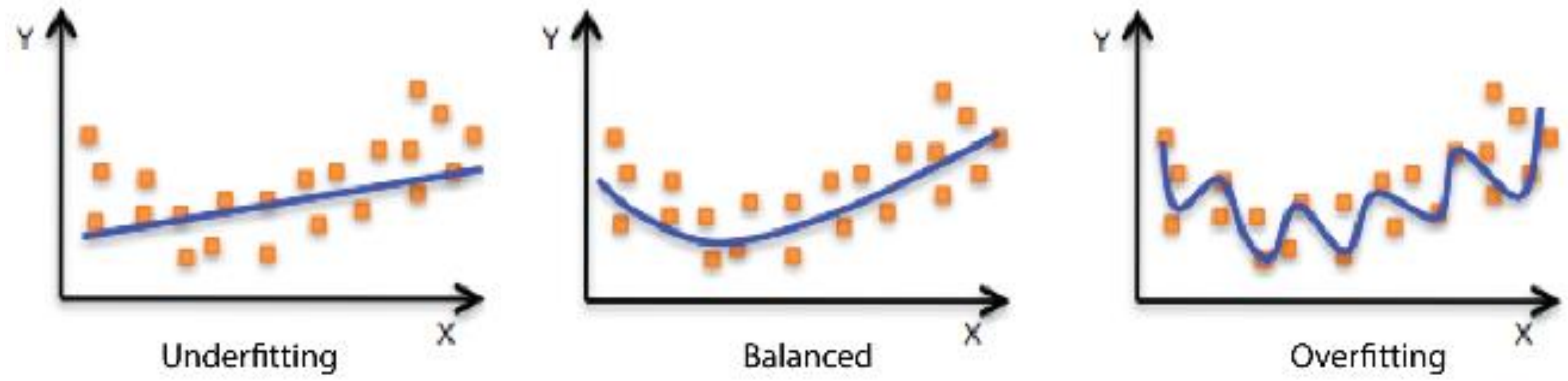
$$\text{CV Score} = \frac{1}{K} \sum_{i=1}^{K} \text{Performance Metric}_i$$

- Where the Performance Metric could be accuracy, precision, recall, F1 score, etc.

Model Selection Process

Bias-Variance Tradeoff

- The bias-variance tradeoff is a fundamental concept in machine learning that balances model complexity and performance.
- **High Bias (Underfitting):**
  - The model is too simple and fails to capture the underlying patterns in the data.
  - **Symptoms:** Poor performance on both training and test data.
  - **Example:** Linear regression applied to a non-linear dataset.
- **High Variance (Overfitting):**
  - The model is too complex and captures noise in the training data.
  - **Symptoms:** Excellent performance on training data but poor performance on test data.
  - **Example:** A deep neural network with too many layers or parameters.



Model Selection Process

- **Mathematical Explanation:**
  The total error of a model is given by:

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

- **Bias:** Error due to overly simplistic assumptions in the model.
- **Variance:** Error due to sensitivity to small fluctuations in the training data.
- **Irreducible Error:** Noise inherent in the data that cannot be reduced by any model.
- **Example:**
- **Linear regression (High Bias):** Too simple, fails to capture data trends.
- **Deep neural network (High Variance):** Memorizes training data but fails on new data.

Evaluation Metrics

Confusion Matrix: A confusion matrix is a table used to evaluate the performance of a classification model by comparing the predicted labels against the actual labels.
It is particularly useful for understanding the types of errors a model makes.

	Predicted Positive (P)	**Predicted Negative (N)**
Actual Positive (P)	True Positive (TP)	False Negative (FN)
Actual Negative (N)	False Positive (FP)	True Negative (TN)

True Positive (TP):

- The model correctly predicted the positive class.
- Example: A spam email is correctly classified as spam.

False Negative (FN):

- The model incorrectly predicted the negative class when the actual class was positive.
- Example: A spam email is incorrectly classified as not spam.

False Positive (FP):

- The model incorrectly predicted the positive class when the actual class was negative.
- Example: A non-spam email is incorrectly classified as spam.

True Negative (TN):

- The model correctly predicted the negative class.
- Example: A non-spam email is correctly classified as not spam.

Evaluation Metrics

Example:

Consider a spam detection model predicting whether an email is spam (1) or not (0):

Actual / Predicted	Spam (1)	Not Spam (0)
Spam (1)	TP = 50	FN = 10
Not Spam (0)	FP = 5	TN = 100

- **True Positive (TP) = 50:** Spam correctly identified as spam.
- **False Negative (FN) = 10:** Spam wrongly classified as not spam.
- **False Positive (FP) = 5:** Non-spam wrongly classified as spam.
- **True Negative (TN) = 100:** Non-spam correctly identified as not spam.

Evaluation Metrics

Logarithmic Loss (Log Loss)

Log Loss is used to evaluate classification models by penalizing incorrect predictions with high confidence. It is particularly useful for probabilistic models.

Mathematical Formula:

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

Where:

- y_i is the actual label (0 or 1).
- $\hat{y}_i$ is the predicted probability of class 1.
- N is the total number of samples.

Evaluation Metrics

Example:

Consider a binary classification problem where we have three predictions:

Sample	Actual y	Predicted $\hat{y}$
1	1	0.9
2	0	0.2
3	1	0.7

$$\text{Log Loss} = -\frac{1}{3}\left[(1 \times \log 0.9) + (0 \times \log 0.2) + (1 \times \log 0.7)\right]$$

Additional Insight:

- Log Loss is sensitive to the predicted probabilities. A model with high confidence in incorrect predictions will have a high Log Loss.
- Lower Log Loss indicates better model performance.

Evaluation Metrics

Precision and Recall

- **Precision (Positive Predictive Value - PPV):** Measures the accuracy of positive predictions.

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

- **Recall (Sensitivity or True Positive Rate - TPR):** Measures the proportion of actual positives correctly identified.

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

Example:

Using the previous spam detection confusion matrix:

$$\text{Precision} = \frac{50}{50 + 5} = \frac{50}{55} = 0.91$$

$$\text{Recall} = \frac{50}{50 + 10} = \frac{50}{60} = 0.83$$

Evaluation Metrics

Other Evaluation Metrics

- **Accuracy:**

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

  Using our example:

$$\text{Accuracy} = \frac{50 + 100}{50 + 100 + 5 + 10} = \frac{150}{165} = 0.91$$

- **ROC-AUC (Receiver Operating Characteristic - Area Under Curve):**
  - **ROC Curve:** Plots the True Positive Rate (TPR) vs. False Positive Rate (FPR) at various thresholds.
  - **AUC (Area Under Curve):** Measures the model's ability to distinguish between classes. Higher AUC indicates better performance.
  - **Additional Insight:** ROC-AUC is robust to imbalanced datasets and provides a comprehensive evaluation of model performance across all classification thresholds.