Overfitting and Regularization

Improving Model Generalization

Sina Moghimi, Autumn 2024

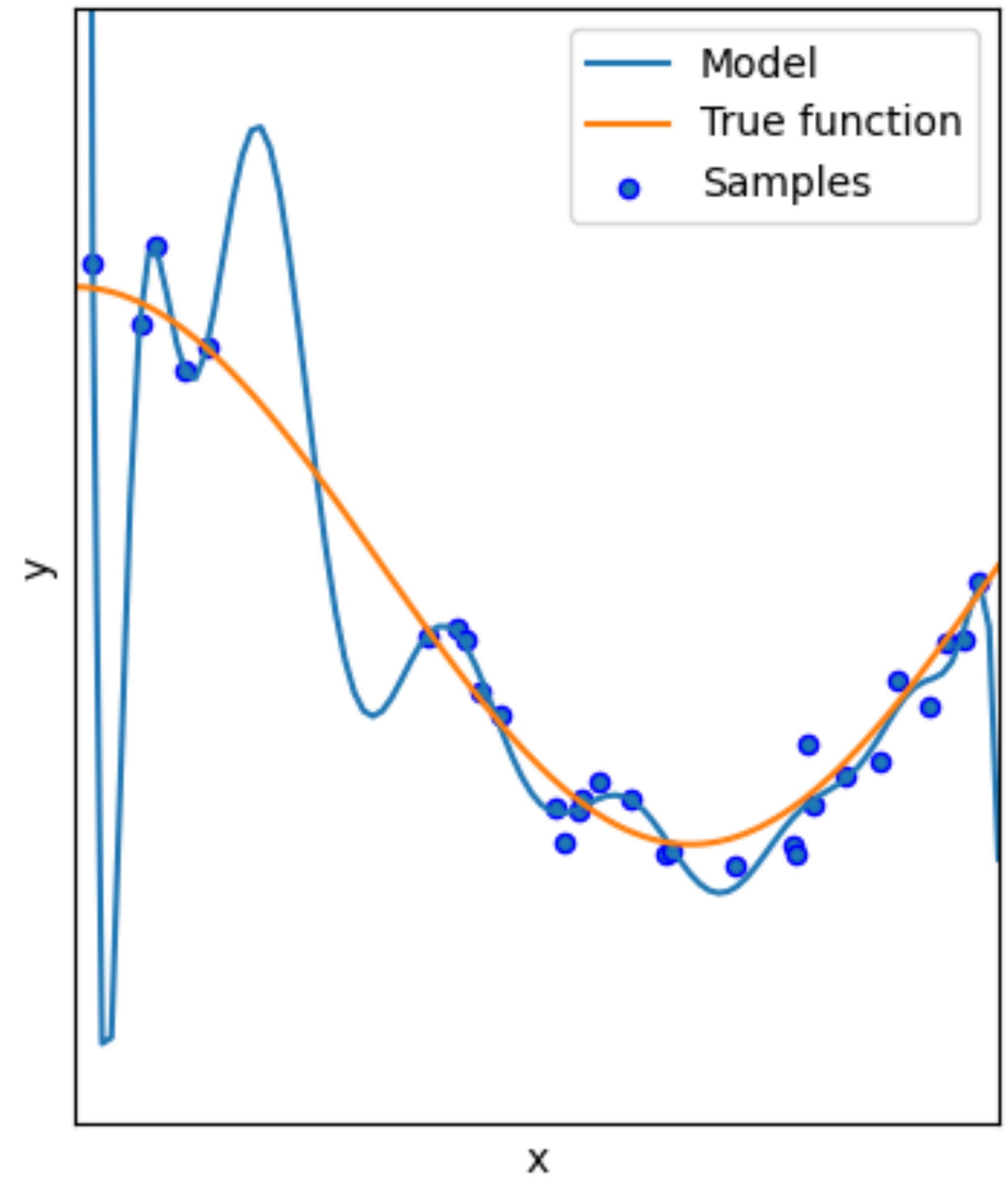
МФТИ

Outline

- What is Overfitting
- Dropout Technique
- L1 Regularization
- L2 Regularization
- Cross-Validation Methods
- Advanced Regularization

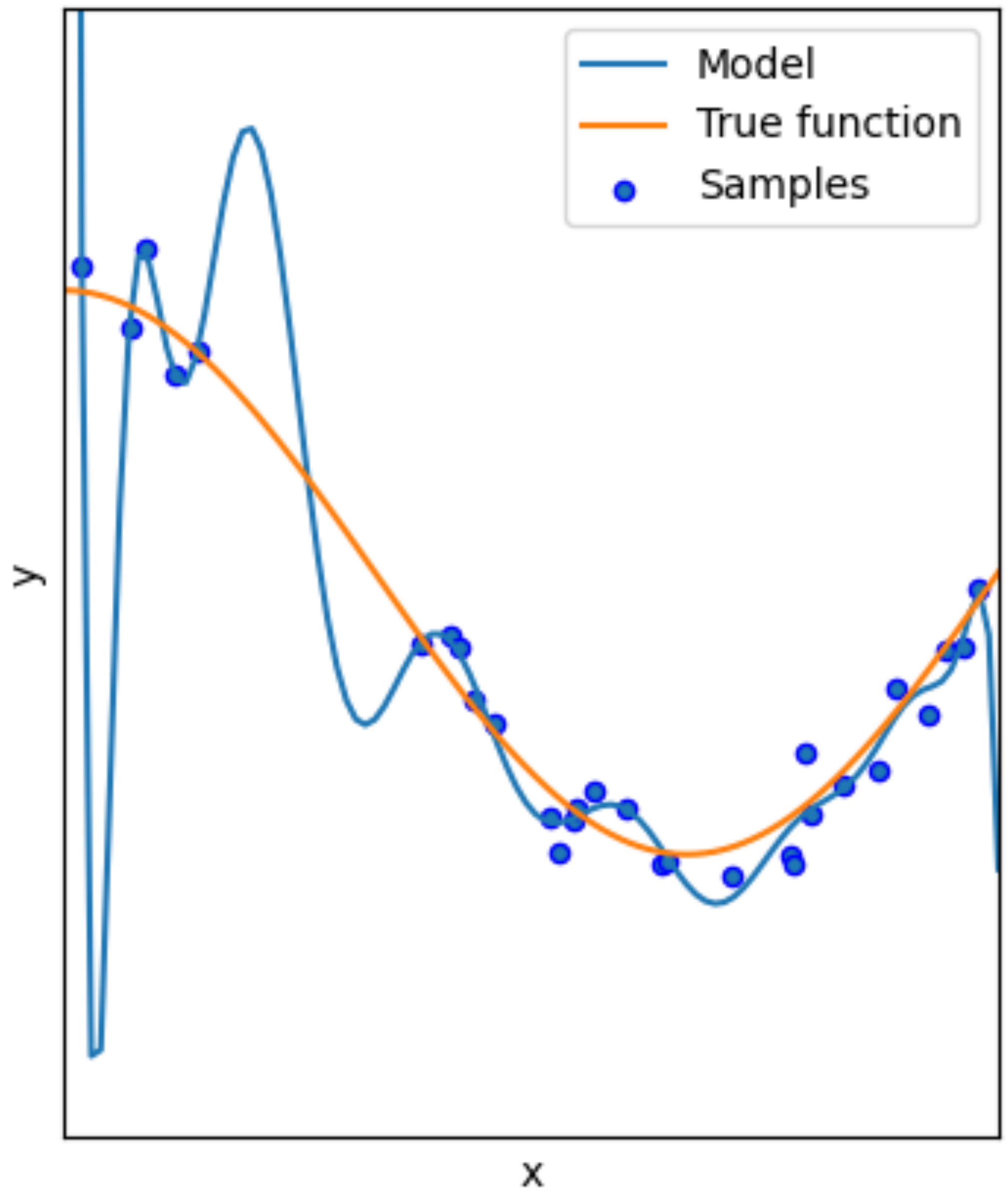
What is Overfitting?

- It happens when the model learns the **noise** in the data **instead of the pattern,** leading to **poor generalization**
- **Symptoms**
  - **High Training Accuracy vs. Low Validation/Test Accuracy**
  - **Low Generalization**
  - **Increased Complexity**
  - **Erratic Validation Loss**
  - **Unreliable Predictions**



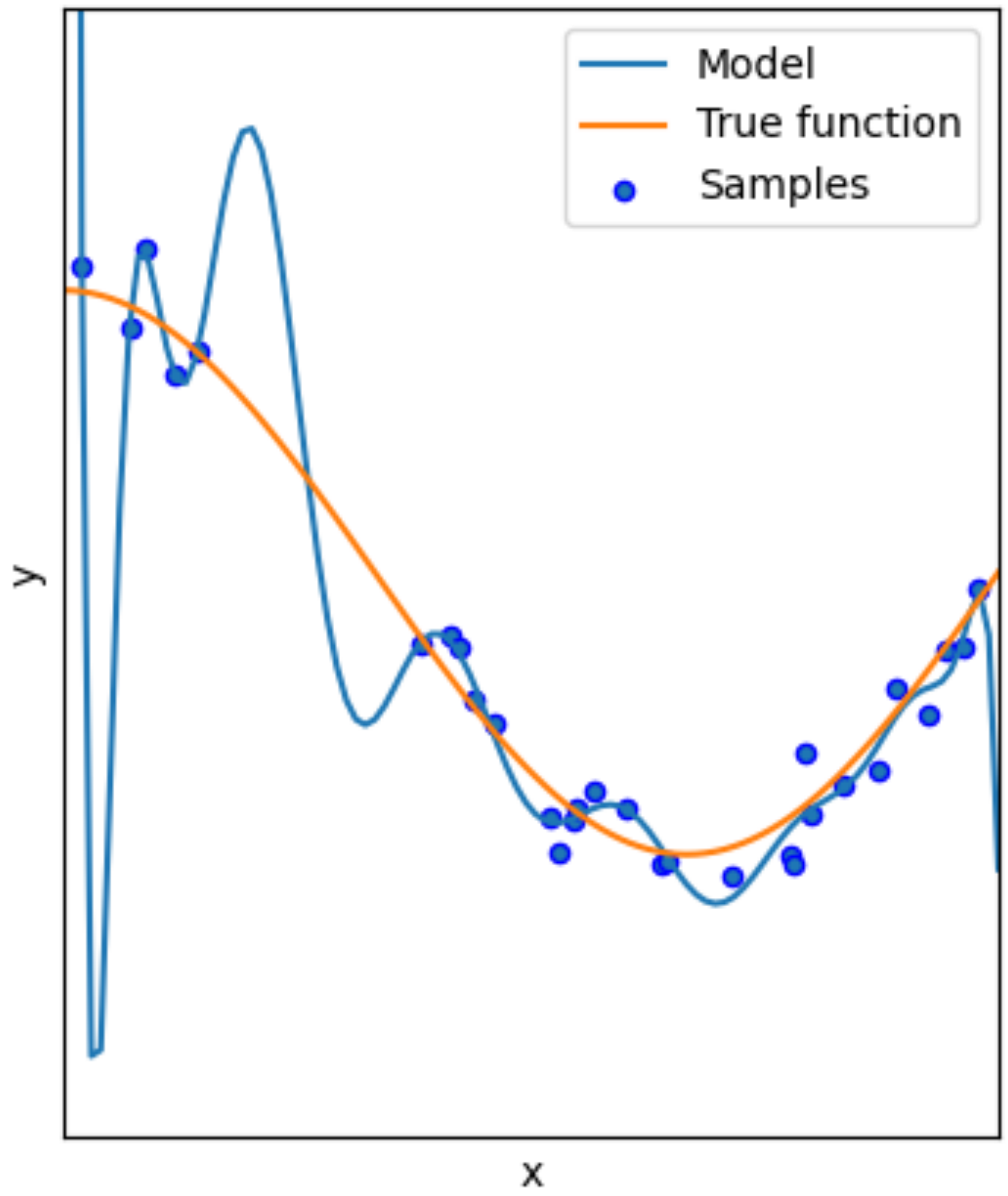
Impact on Performance

- **Training Performance**
  - Perform **Well on Training** Data
- **Validation/Test Performance**
  - Poor generalization, leads to significant drop in metrics such as accuracy, precision, recall, F1-Score on Validation/Test datasets
- **Real-World Applications**
  - Overfitted models are unreliable as they fail adapt to new or slightly different data
- **Loss of Robustness**
  - Overfitting, makes the model sensitive to small changes in input (memorized the training data instead of learning the pattern)



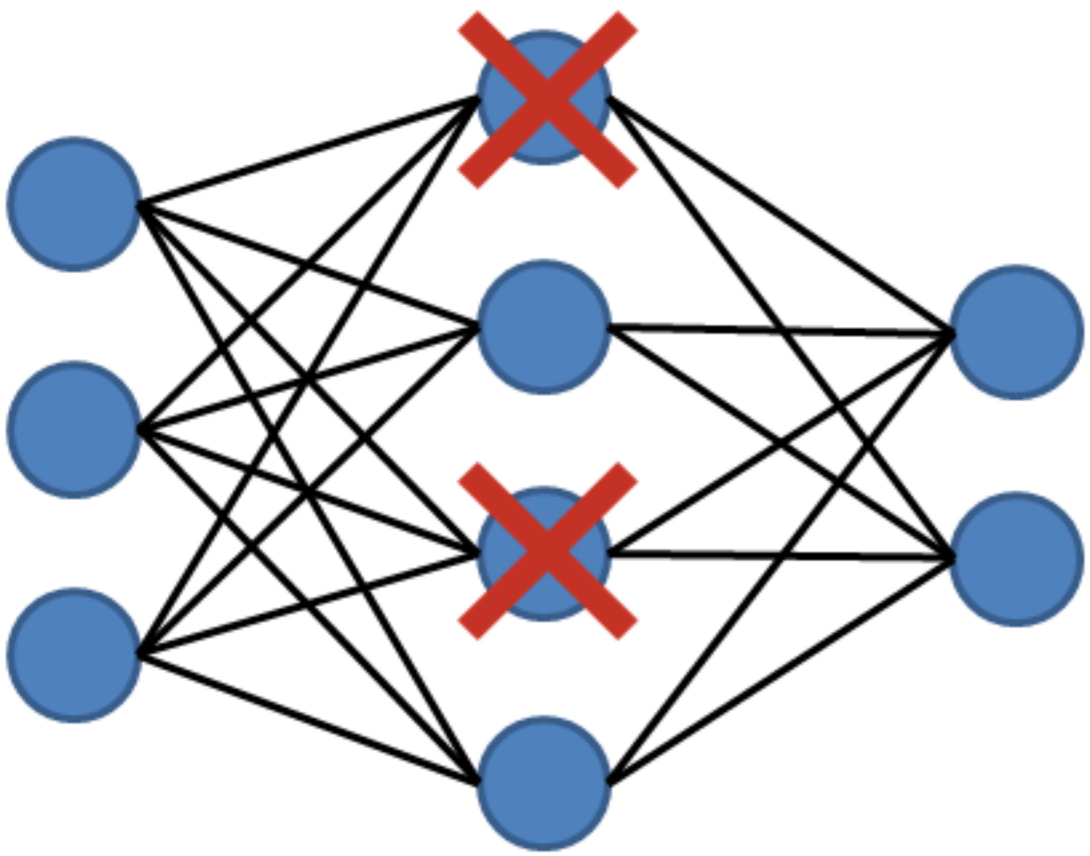
How to Overcome?

- Increase Training data
- Data Augmentation
- Regularization
- Dropout
- Simplifying the model
- Early Stopping
- Cross-Validation
- Ensemble Methods



DropOut

- **Dropout** is a regularization Technique used in training neural networks to prevent over fitting
    - **Simply saying**, We freeze(drop out) some neurons, and do not involve them in backpropagation, thus do not update their parameters
    - For example we choose to freeze 20% of the neurons in one layer. That subset of neurons are selected randomly in each round of training.
- **Prevents Over-Reliance on Specific Features**
    - Encourages the network to learn more generalized features
- **Advantages**
    - Reduces the risk of overfitting
    - Improves model robustness
    - Easy to implement
- **Limitations**
    - More epochs for convergence
    - Not always necessary
    - For shallow networks, can reduce the model ability



https://towardsdatascience.com/coding-neural-network-dropout-3095632d25ce

МФТИ

L1 Regularization

- Also known as **Lasso Regularization (Least Absolute Shrinkage and Selection Operator)**
- **Penalize the absolute value of model weights**

$$L = Loss(y, \hat{y}) + \lambda \sum_{i=1}^{n} |\omega_i|$$

  - λ is regularization parameter, controlling the strength of the penalty
  - **Larger** λ, encourages more weights to become zero, but **smaller** values allow more flexibility
- **Sparse Feature Selection**
  - It automatically selects features that contribute most to the model's prediction while driving irrelevant ones to zero

L1 Regularization

- **Lasso Regression**
  - Is a Linear regression model that incorporates L1 Regularization

$$\min_{\omega}\left(\sum_{i=1}^{m}(y_i - \sum_{j=1}^{n}\omega_j x_{ij})^2 + \lambda \sum_{j=1}^{n}|\omega_j|\right)$$

  - m**:** number of samples
  - n: Number of features
  - x_{ij}: Value of the j-th feature for the i-th sample
  - y_i: i-th sample ground truth

L1 Regularization

Pros	Cons
Feature Selection: Automatically eliminates irrelevant features by setting their weights to zero.	**Model Instability**: Small changes in data can lead to different features being selected.
Simplicity: Produces simpler, interpretable models with fewer non-zero coefficients.	**Bias**: Adds bias to the model by shrinking coefficients, which can hurt performance on complex datasets.
Effective for Sparse Data: Works well when many features are irrelevant or redundant.	**Limitations with Correlated Features**: Among highly correlated features, Lasso tends to pick one and ignore others, which may not be optimal.
Efficient: Helps reduce overfitting by preventing large coefficients.	**Not Always Optimal**: May underperform compared to L2 or ElasticNet in scenarios where feature selection isn't needed.

L2 Regularization

- Also known as **Ridge Regularization**
- Penalize the large weights

$$L = Loss(y, \hat{y}) + \lambda \sum_{i=1}^{n} \omega_i^2$$

  - λ is regularization parameter, controlling the strength of the penalty
- **Ridge Regression**
  - Is a Linear regression model that incorporates L2 Regularization

$$\min_{\omega}\left(\sum_{i=1}^{m} (y_i - \sum_{j=1}^{n} \omega_j x_{ij})^2 + \lambda \sum_{j=1}^{n} \omega_j^2 \right)$$

  - m**:** number of samples
  - n: Number of features
  - x_{ij}: Value of the j-th feature for the i-th sample
  - y_i: i-th sample ground truth

L2 Regularization

- **Weight Decay Mechanism**
  - Refers to gradual reduction of model weights during training to prevent overfitting
- **Mechanism**
  - The penalty term $\lambda \sum \omega_i^2$, discourages large weight values by adding a cost for their magnitude
  - During gradient descent, weight are updated as

$$\omega \leftarrow \omega - \eta \frac{\partial}{\partial \omega}(Loss + \lambda \sum \omega_i^2)$$

L2 Regularization

Pros	Cons
Prevents overfitting by discouraging large weights.	Does not perform feature selection.
Retains all features, useful for correlated data.	May retain irrelevant features with reduced weights.
Works well with small datasets and noisy data.	Not optimal for sparse models.

L2 vs. L1 Regularization

Aspect	L2 (Ridge)	L1 (Lasso)
Penalty	$\lambda \sum \omega_i^2$	$\lambda \sum \lvert\omega_i\rvert$
Effect on weights	Shrinks all weights but does not set them to zero	Drives some weights to exactly zero (sparse solution)
Feature Selection	Retains all features, reduces their impact	Performs automatic feature selection by eliminating irrelevant features
Use case	Suitable for datasets with highly correlated features	Suitable for sparse models or high-dimensional data
Computational cost	Slightly lower due to smooth penalty	Slightly higher due to non-smooth penalty
Gradient Behaviour	Continuous gradients, making optimization easier	Discontinuous gradients, which can cause instability in optimization

МФТИ

Cross-Validation

- Is a statistical technique to improve the performance by splitting the dataset into training and testing sets multiple times.

- It ensures that the model generalizes well to unseen data and avoids overfitting/underfitting

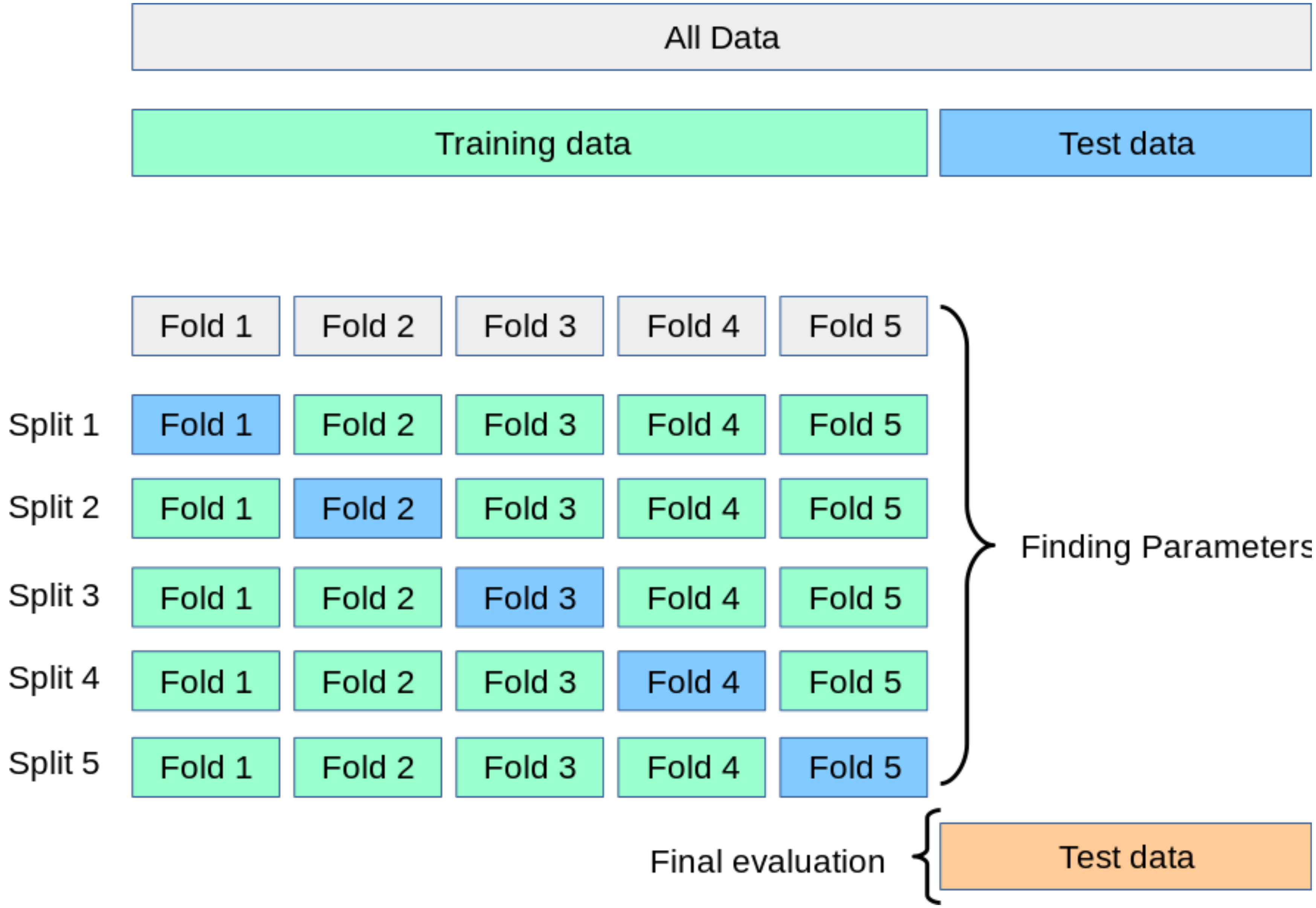
- **Famous Methods**

  - **K-Fold Cross-Validation**

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

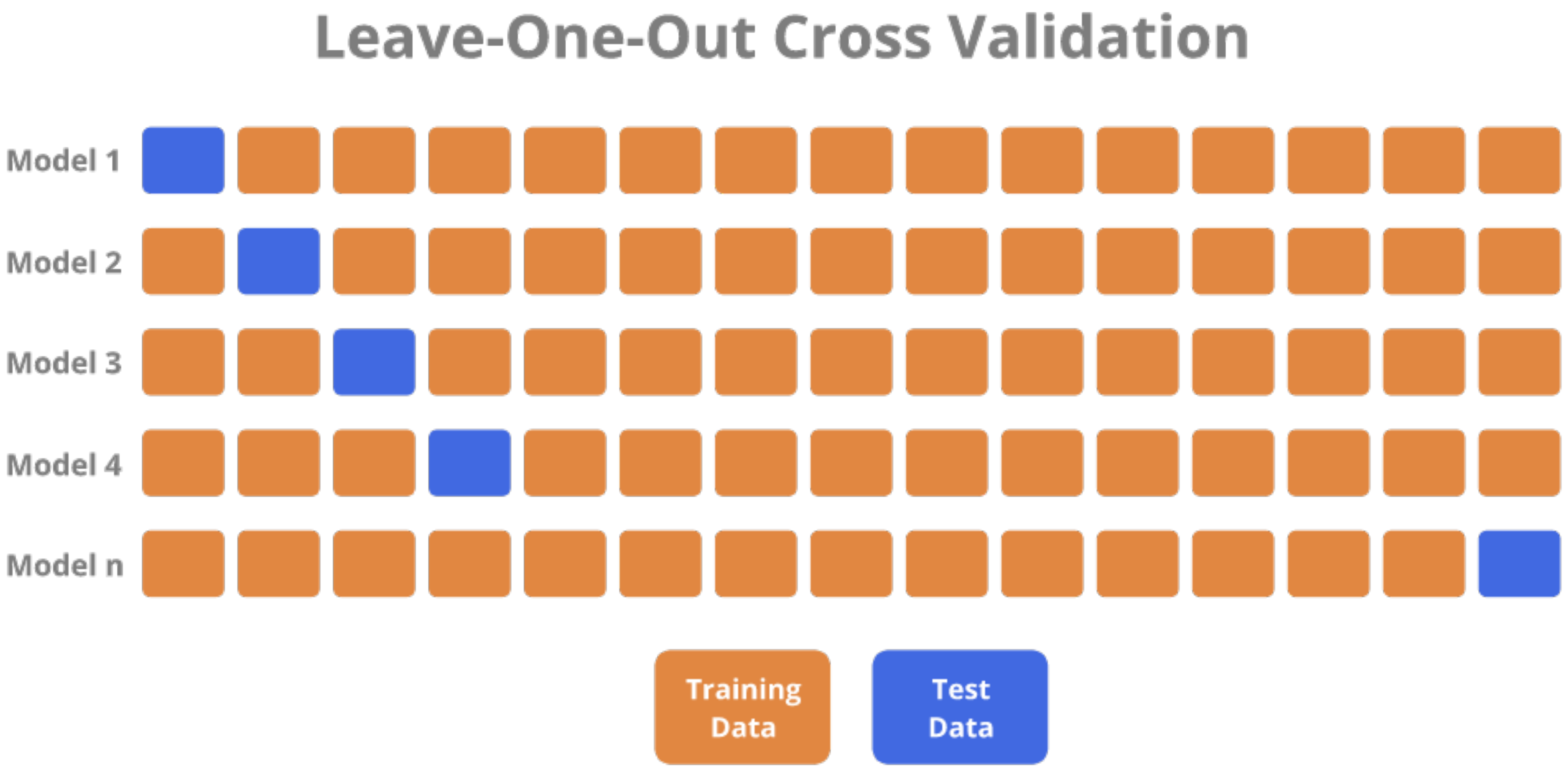
  - **Stratified Cross-Validation**

K-Fold Cross-Validation



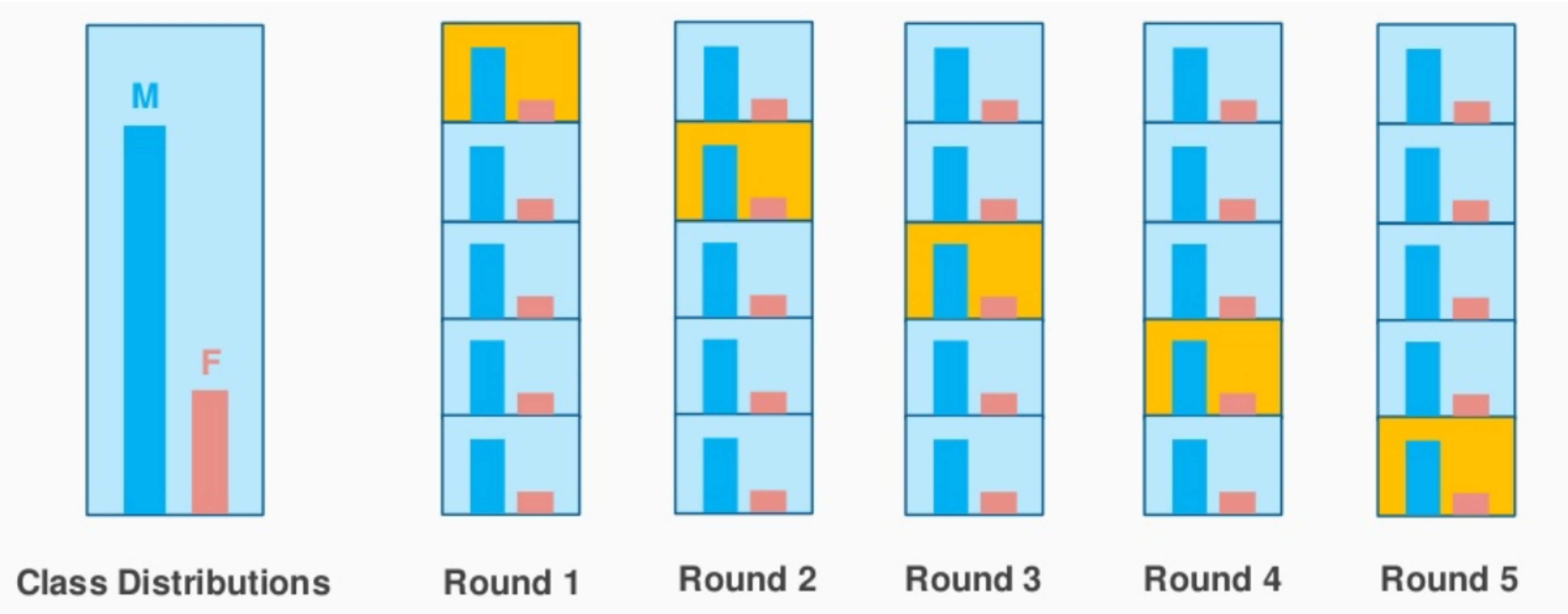
https://scikit-learn.org/1.5/modules/cross_validation.html

Leave-One-Out Cross-Validation (LOOCV)



https://www.bijenpatel.com/guide/islr/resampling-methods/

Stratified Cross-Validation



https://medium.com/@ompramod9921/cross-validation-623620ff84c2

МФТИ

Cross-Validation

Method	Best For	Advantages	Disadvantages
K-Fold	General-purpose, moderately large datasets	Efficient use of data; reduces variance in evaluation	Computationally expensive for large K
Leave-One-Out (LOOCV)	Very small datasets	Almost unbiased estimate of performance	Very high computational cost; high variance in results
Stratified	Imbalanced classification problems	Maintains class distribution across folds	Requires more setup; same computational cost as K-fold

МФТИ

Advanced Regularization Techniques

- **Ensemble Methods**
  - **Bagging (Bootstrap Aggregating)**
    - Train multiple models on different subsets of the data
    - Average or majority-vote the predictions (i.e., Random Forest)
  - **Boosting**
    - Train models sequentially, with each model focusing on correcting errors made by the previous ones (i.e., Gradient Boosting, AdaBoost)
  - **Stacking**
    - Combine multiple models by training a meta-model on their predictions

Advanced Regularization Techniques

- **Early Stopping**
  - Split the dataset into training and validation sets
  - During training, track the validation error (i.e., loss or accuracy)
  - Stop training when validation error starts increasing or stabilizes for a defined number of epochs
- **Data Augmentation**
  - It helps generating more data based on the current dataset, by
    - rotation, flipping, cropping, color transforms, noise addition, etc.

МФТИ

Advanced Regularization Techniques

- **Noise injection**
  - It involves adding randomness to the data during training
  - **Types of Noise Injection**
    - **Input Noise**
    - **Weight Noise**
    - **Gradient Noise**

Advanced Regularization Techniques

Technique	Best For	Advantages	Disadvantages
Ensemble Methods	Complex tasks with sufficient resources	Reduces variance and bias, improves accuracy	High computational cost and complexity
Early Stopping	Models with lengthy training processes	Prevents overfitting, simple to implement	Requires a proper validation set
Data Augmentation	Small or imbalanced datasets	Increases data diversity; improves robustness	May introduce unrealistic data, costly
Noise Injection	Improving robustness to noisy inputs	Encourages generalization, reduces overfitting	Requires careful noise level tuning

МФТИ