ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

Machine Learning

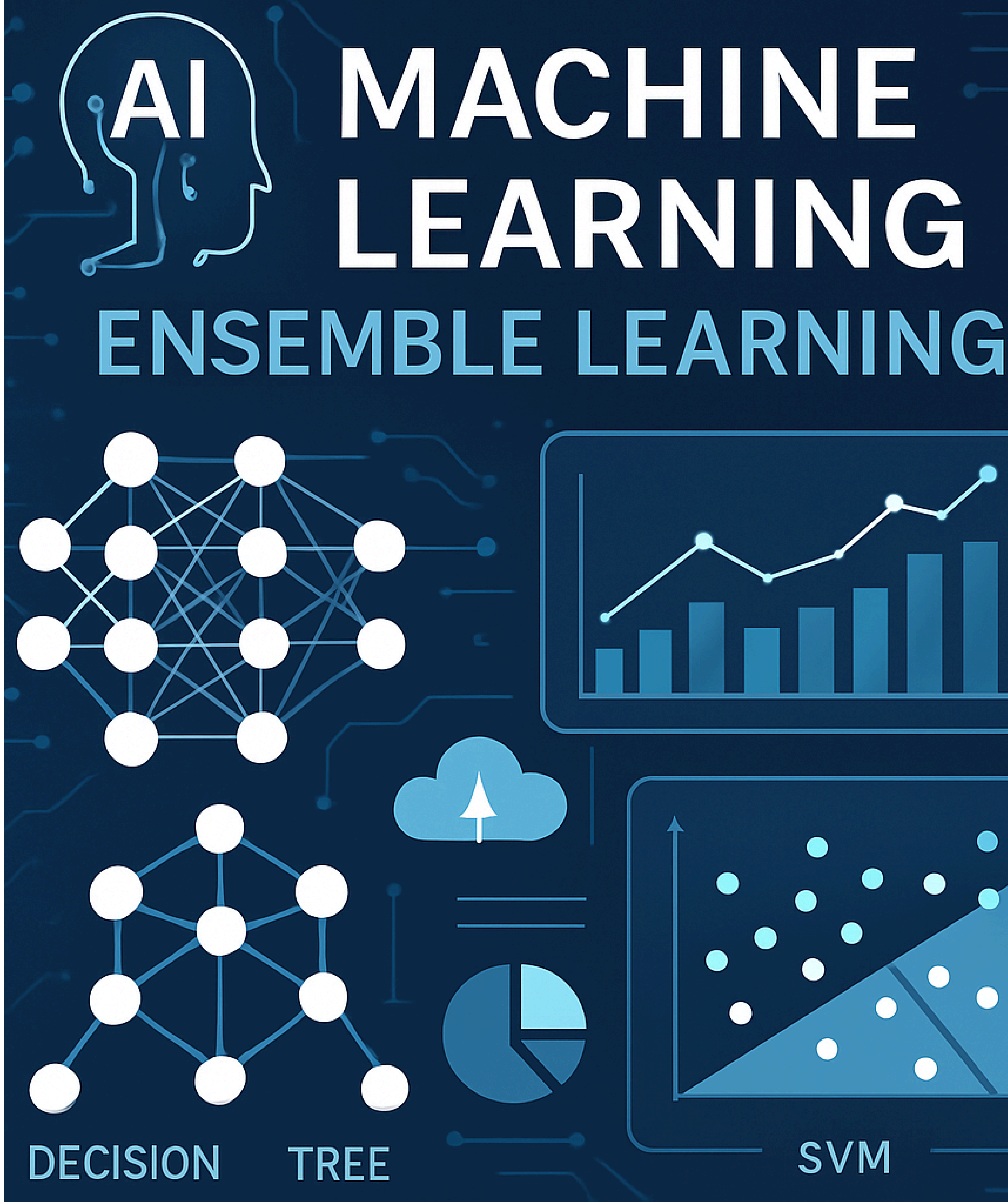
Ensemble Learning

Lecturer: PhD Trần Lương Quốc Đại

Members:

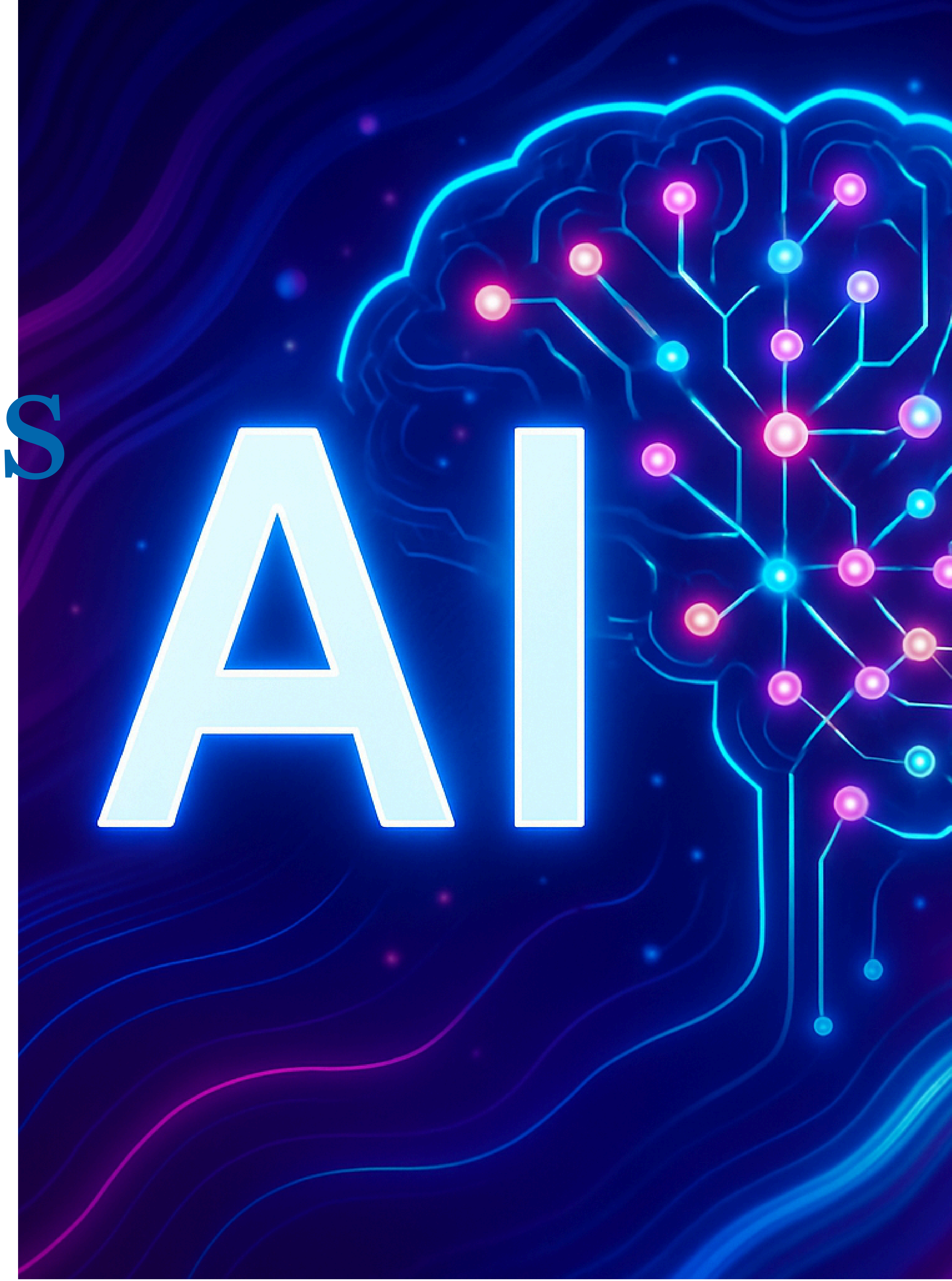
Mai Hoàng Thái - 523H0177

Nguyễn Trần Hoàng Nhân - 523H0164



ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

Table of Contents



Introduction

Introduction

The Problem: Bias-Variance Tradeoff

Concept: In supervised learning, the prediction error consists of three parts: **Bias**, **Variance**, and **Irreducible Error**.

- High Bias (Underfitting): The model is too simple to capture the underlying structure of the data.
- High Variance (Overfitting): The model is too sensitive to the noise in the training data.

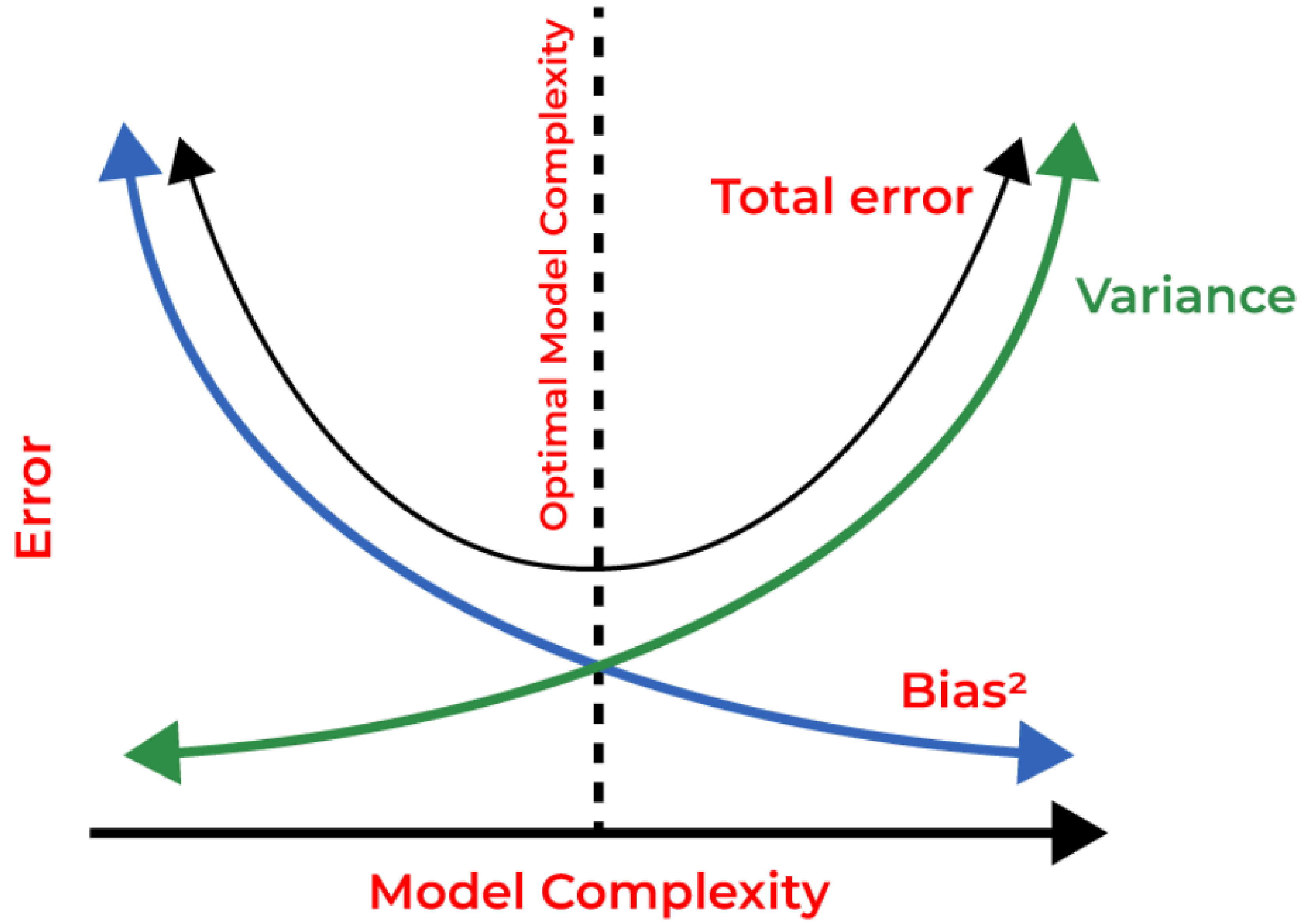
The Tradeoff: Typically, minimizing bias increases variance and vice versa.

Equation:

$$Total\ Error = Bias^2 + Variance + Irreducible\ Error$$

Introduction

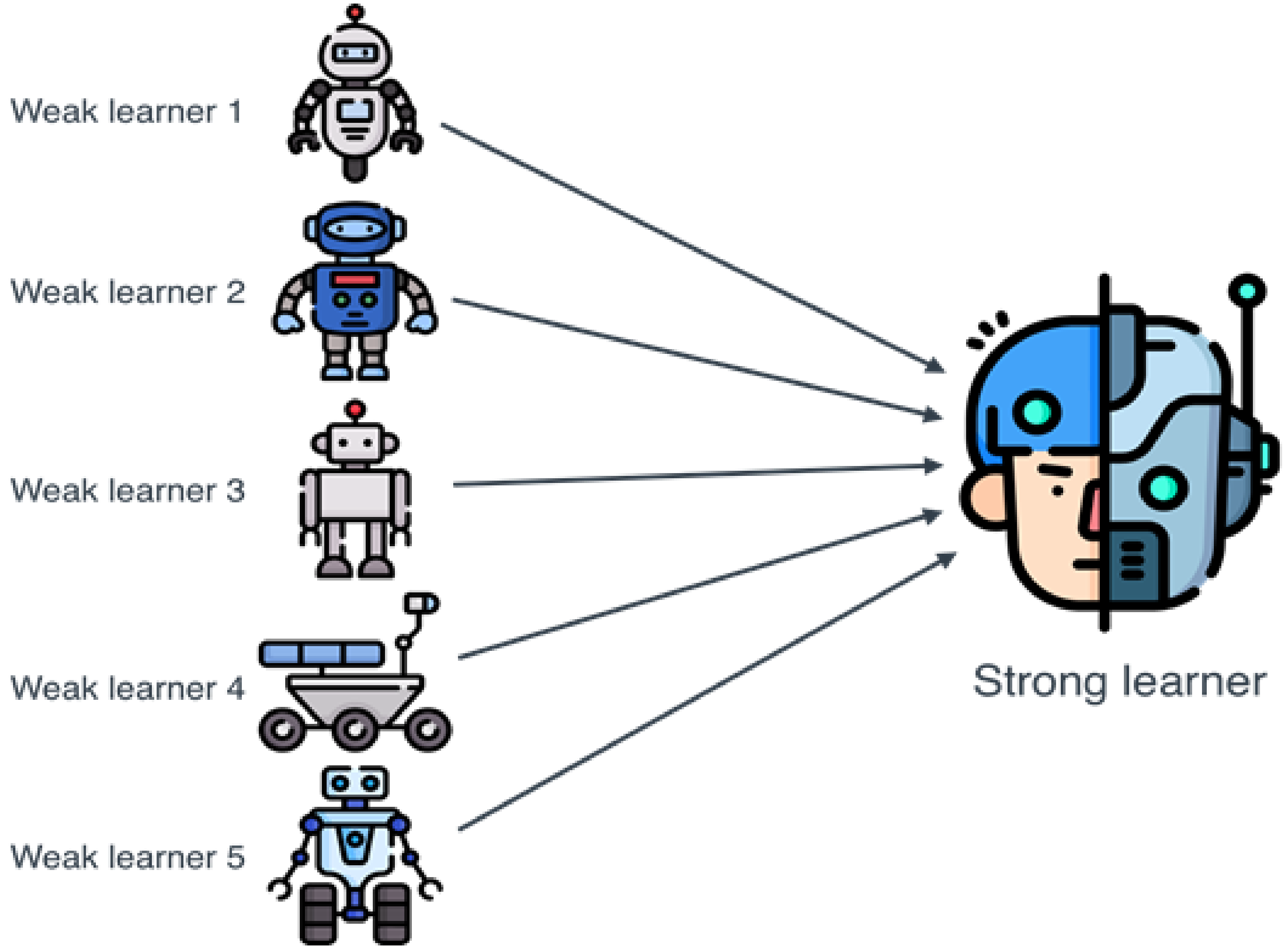
The Problem: Bias-Variance Tradeoff



Introduction

The Solution: Using Ensemble Learning

- **Bagging:** Primarily focuses on reducing Variance (effective for high-variance models like Decision Trees).
- **Boosting:** Primarily focuses on reducing Bias (effective for weak learners with high bias).
- **Stacking:** Primarily focuses on improving predictive accuracy by combine boosting bagging.



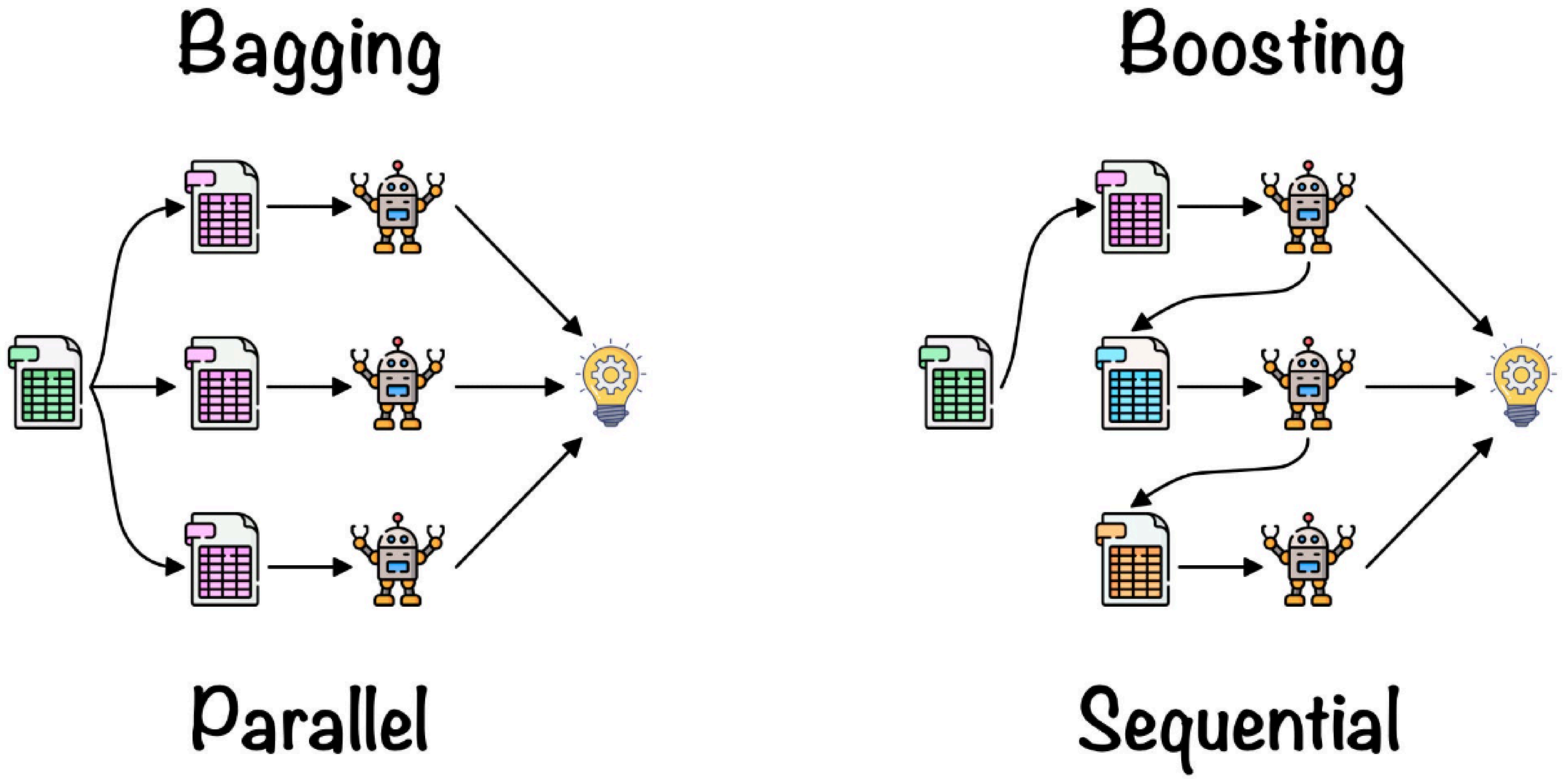
Introduction

What is Ensemble Learning?

- **Definition:** A machine learning paradigm where multiple models (called "base learners" or "weak learners") are strategically combined to create a single, more powerful predictive model
- **Philosophy:** Based on the "Wisdom of Crowds" or "Collective Intelligence"-a committee of experts often makes better decisions than any single expert.

Introduction

What is Ensemble Learning?



Introduction

Goal of Ensemble Learning

- To create a Strong Learner from multiple Weak Learners.
- To improve generalization capability and robustness compared to individual models.

How Ensemble Helps

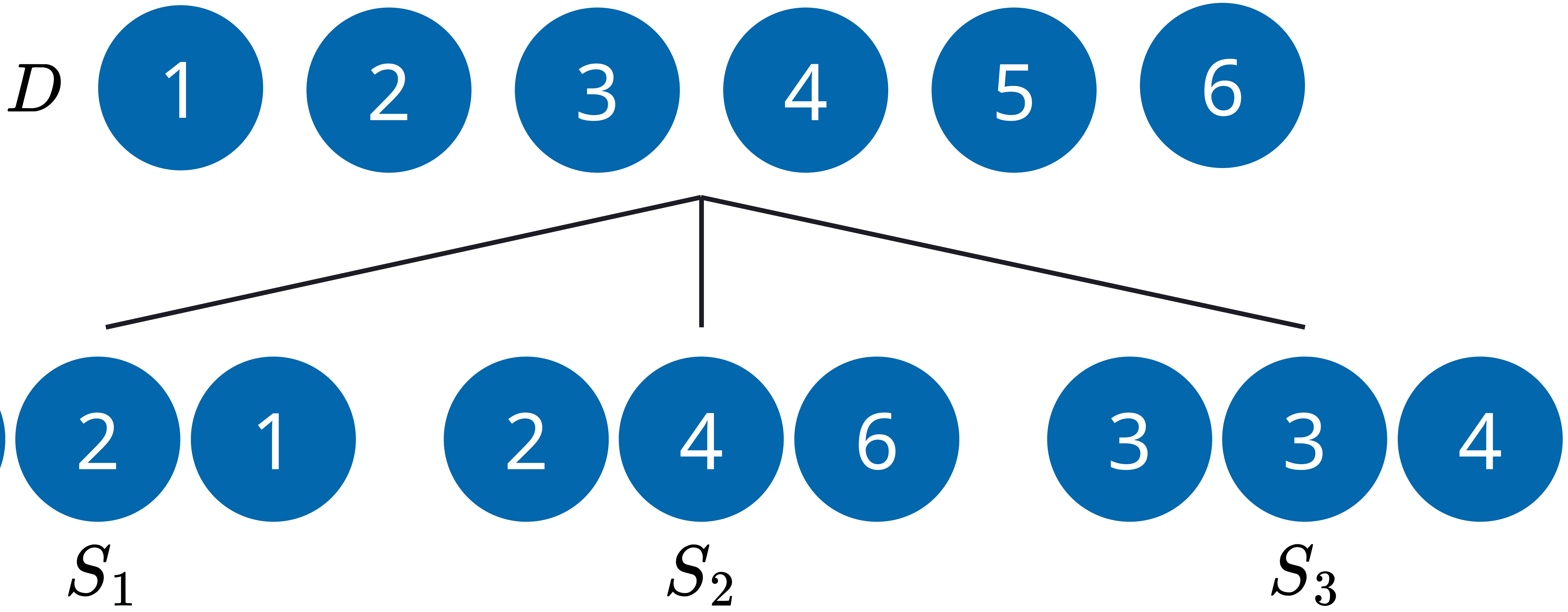
- **Error Reduction:** Random errors cancel out while systematic correct predictions reinforce each other.
- **Stability:** Reduces the risk of choosing a single poor model.

Bagging

Bagging

Theoretical Framework

Bootstrap Sampling (The "Bag" part):



Bagging

Theoretical Framework

Bootstrap Sampling (The "Bag" part):

Out-of-Bag (OOB):

- When we create a Bootstrap sample (the "bag"), we pick data randomly with replacement.
- Because of this, approximately 36.8% of the original data is never selected for a specific model.
- These "leftover" samples are called Out-of-Bag (OOB) samples. We use them to test the model immediately.
- It acts as a built-in validation set. You get an unbiased error estimate "for free" without needing a separate cross-validation step.

Bagging

Theoretical Framework

Parallel Training:

This is the biggest technical advantage of Bagging over other methods.

- Independence: In Bagging, sub-models (base learners) are trained on different subsets of data. Model A does not need the output of Model B. They are completely independent.
- Models are trained in parallel.

→ Helps to significantly reduce training time, especially when the data is large and the model is complex.

Bagging

Theoretical Framework

Aggregation:

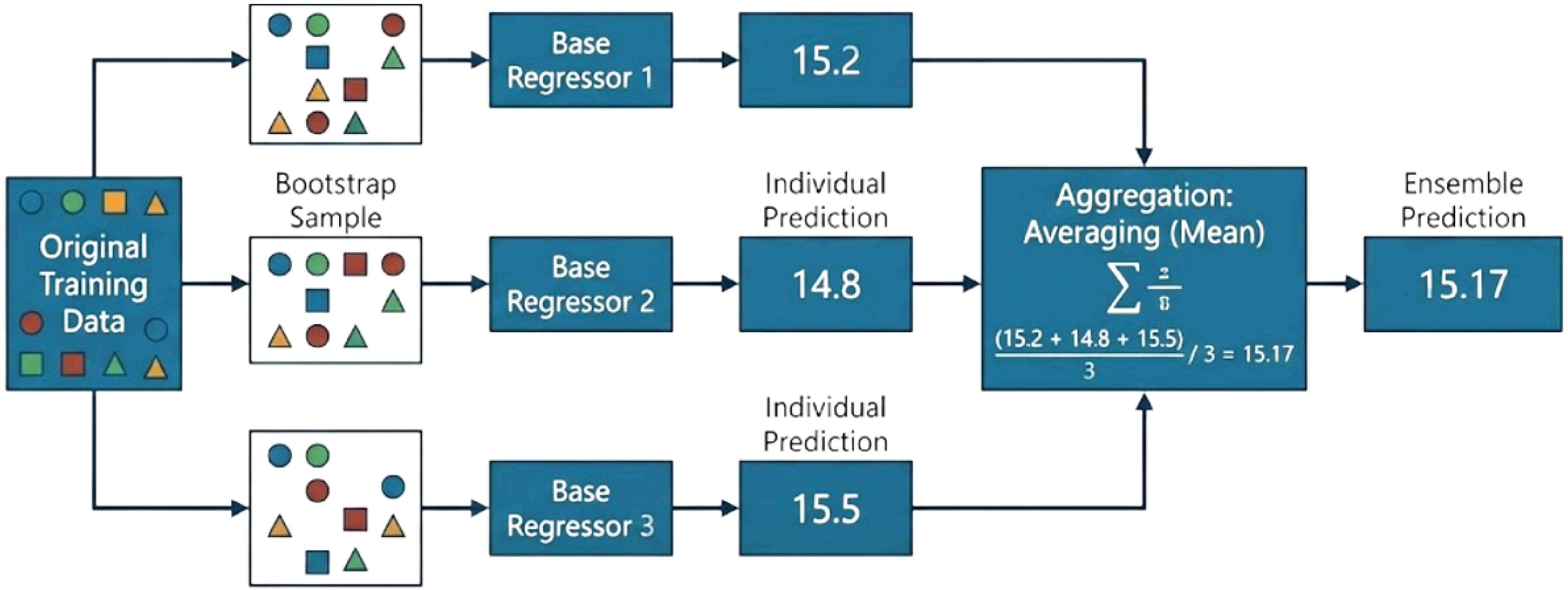
Aggregation is the step where we consult to make the final decision:

- For Regression
- For Classification

Bagging

Theoretical Framework

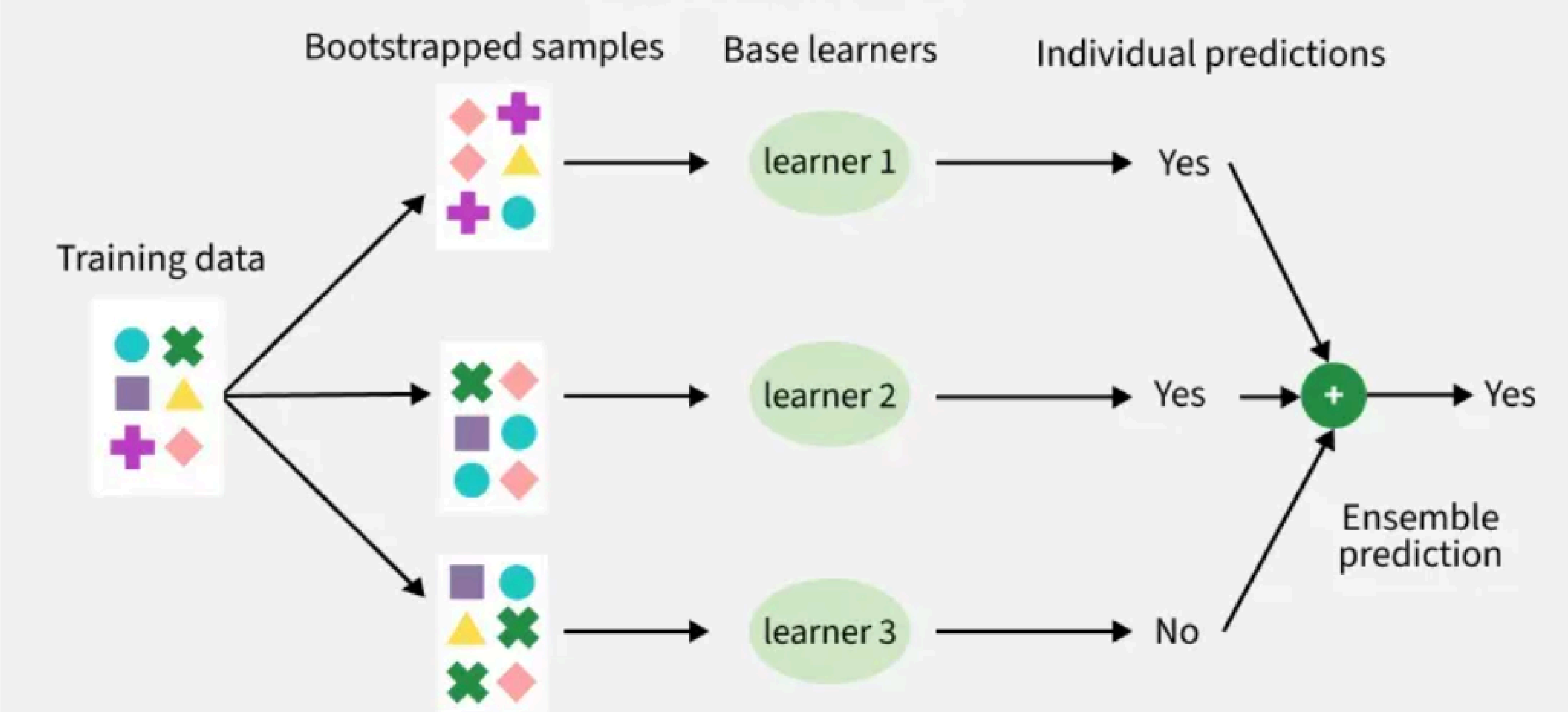
- **For Regression:** The final result is the average of all the predictions from the sub-models.



Bagging

Theoretical Framework

- **For Classification:** The most common method is Voting:
  - **Hard Voting:** Each model votes for 1 class label.
  - **Soft Voting:** The models give probabilities for each class. The system will average those probabilities and choose the class with the highest average probability.



Bagging

Why Bagging Reduces Variance?

Bagging reduces Variance because it follows the statistical principle: "Random errors cancel each other out when added together."

- **Independent:** Due to Bootstrapping, individual base models produce random and distinct errors.
- **Cancellation:** When combined, the positive errors from one model are compensated for by the negative errors of another.
- **Result:** The final output captures the most accurate "central tendency," remaining stable and less susceptible to noise.

Bagging

Why Bagging Reduces Variance?

Variance Reduction Formula:

$$\text{Ensemble Variance} \approx \frac{\text{Variance of One Model}}{\text{Total Number of Models}}$$

(Note: This assumes models are perfectly independent)

Ensemble Learning

Bagging

Visualization

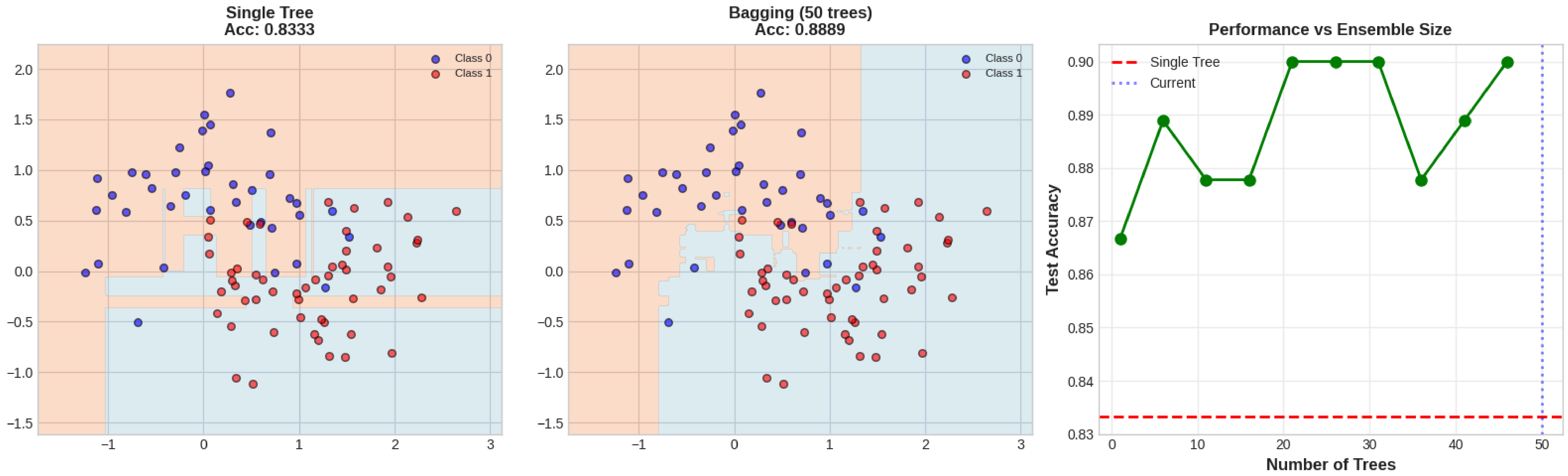
Current configuration: 50 trees

Single tree accuracy: 0.8333

Bagging accuracy: 0.8889

Improvement: +5.56%

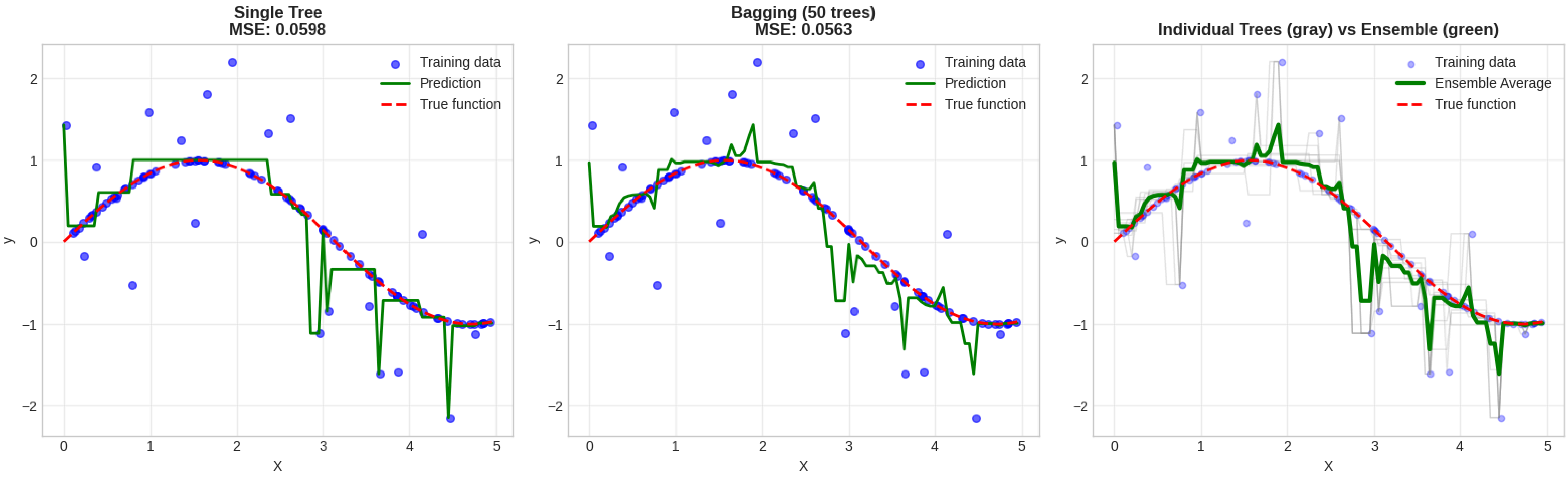
Boundary is much smoother than single tree!



Bagging

Visualization

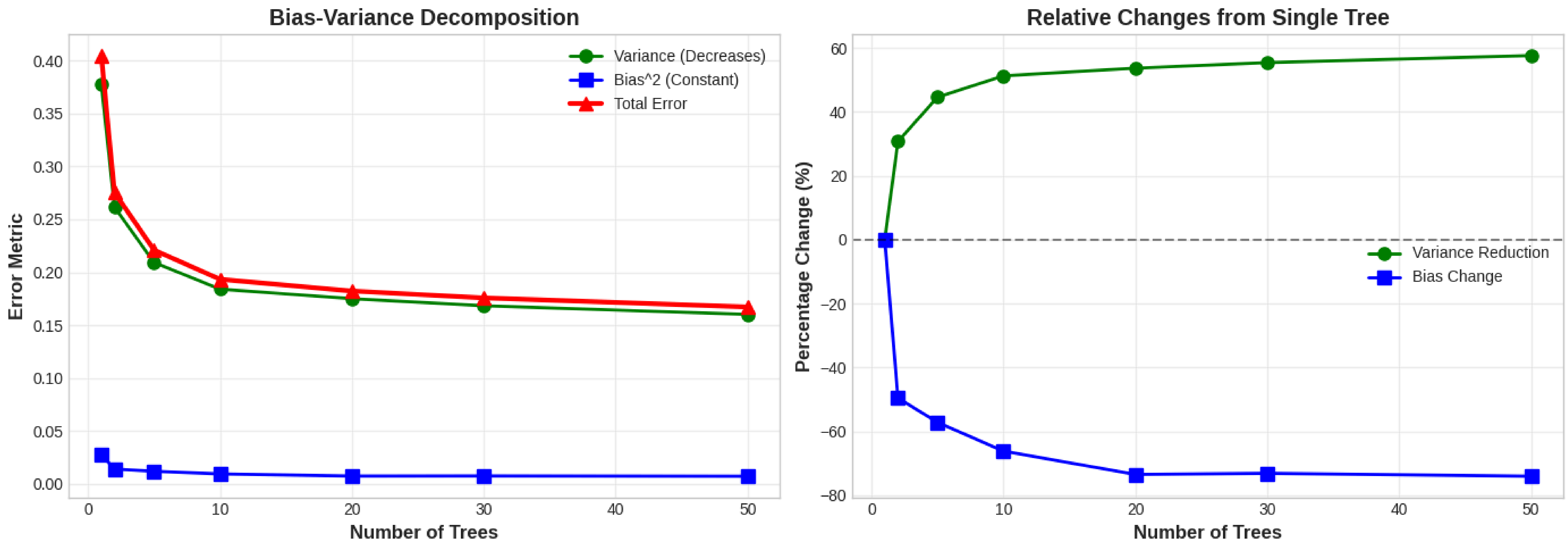
Current configuration: 50 trees
Single tree MSE: 0.0598
Bagging MSE: 0.0563
MSE reduction: 5.85%
Notice: Bagging moderately smooths the prediction!



Bagging

Visualization

Variance reduced by: 57.58%
Bias changed by: -74.03%
Total error reduced by: 58.70%
CONCLUSION: Bagging reduces VARIANCE without increasing BIAS!



Bagging

Random Forest

The Foundation: Random Forest is Bagging applied to Decision Trees.

The Problem of Bagging:

- **Tree Correlation:** If the data has strong features, all trees will select them at the root.
- **Result:** The trees become structurally identical, limiting the reduction of variance.

The Random Forest "Twist":

- **Random Feature Selection:** At each split, the tree is forced to consider only a random subset of features.
- **Effect:** This forces trees to be different, further reducing variance.

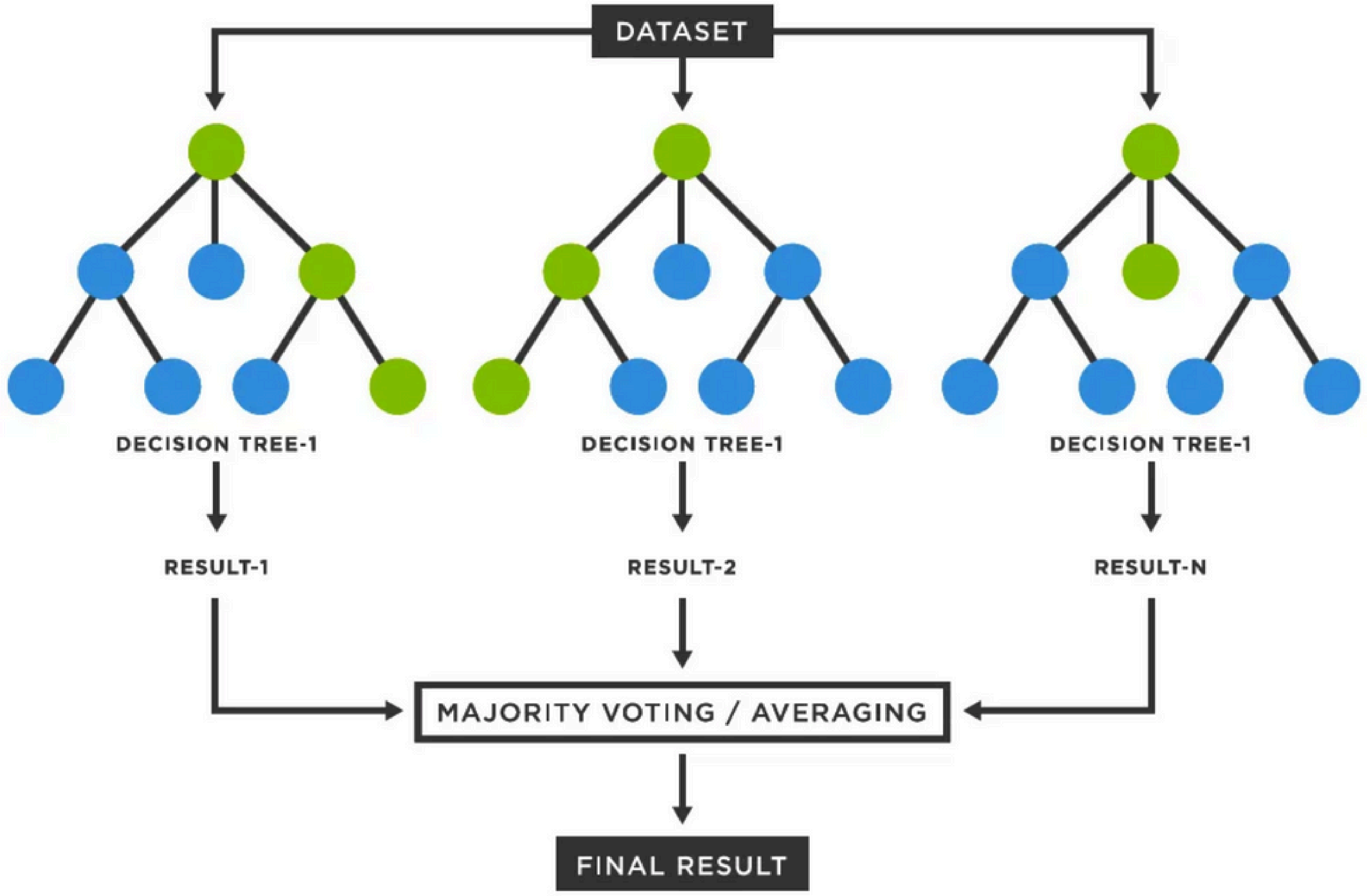
Bagging

Random Forest

The Formula:

$$Random\ Forest\ =\ Bagging\ +\ Feature\ Randomness\ (at\ each\ split)$$

Result: The trees are forced to be more diverse, which reduces Variance more strongly than Normal Bagging.

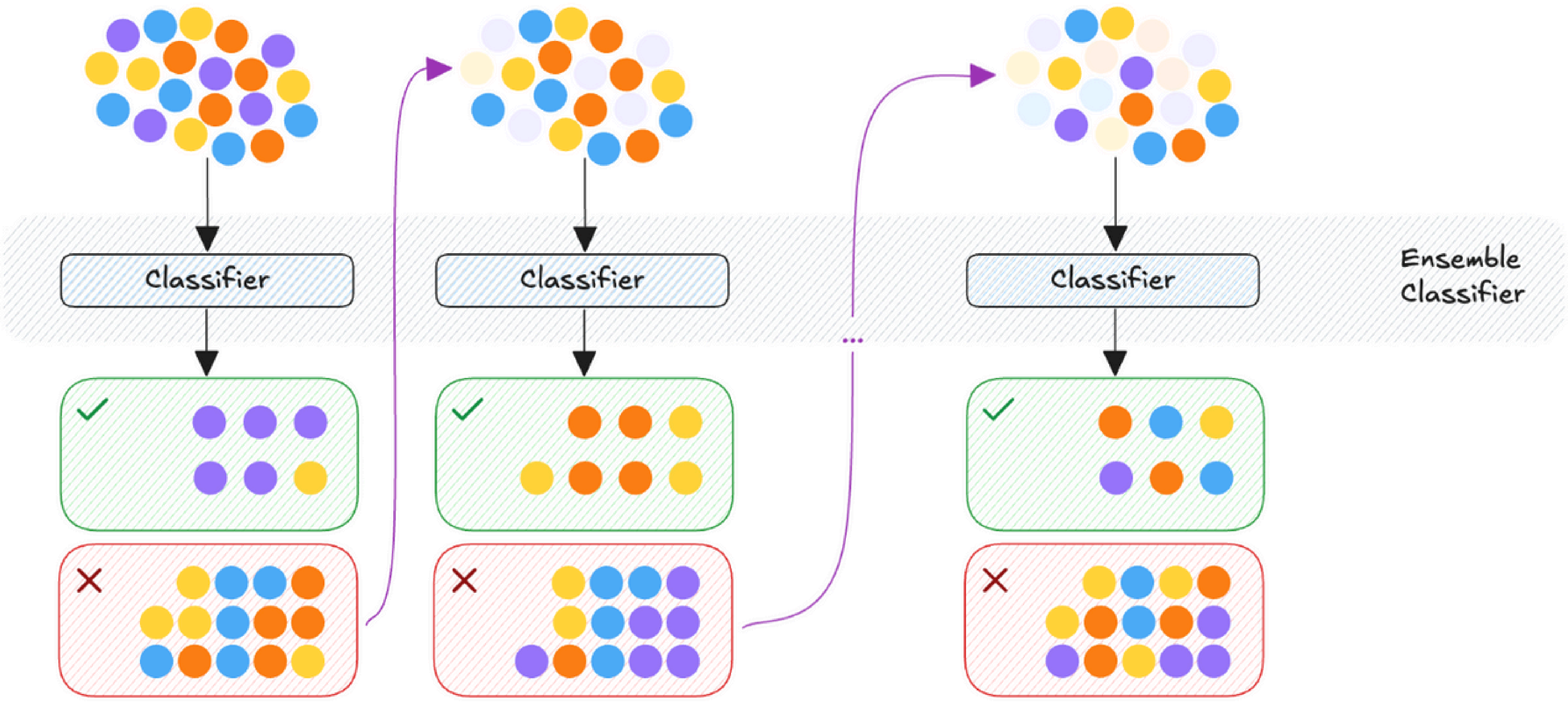


Boosting

Boosting

Theoretical Framework

Boosting is a **sequential** technique. Unlike Bagging (parallel), Boosting builds models one after another, where each new model tries to **fix the mistakes** of the previous one.



Boosting

Why Boosting Reduces Bias?

Sequential Error Correction:

- Boosting does not fit a single complex model at once.
- Instead, it fits a sequence of weak models iteratively.
- Key Mechanism: Every new model is trained specifically to fix the bias of the previous combined model.

$$New\ Prediction = Old\ Prediction + Correction$$

Boosting

Why Boosting Reduces Bias?

Increasing Model Complexity:

- **Base Models:** We start with weak Learners which have High Bias.
- **The Ensemble:** By summing up many simple corrections, the final ensemble becomes a highly complex function capable of fitting very non-linear data boundaries.
- This transforms a High Bias system into a Low Bias system.

Boosting

AdaBoost

Mechanism: "Learning via Weights"

- Initially, all data points have equal weights.
- After training Model 1, the weights of misclassified examples are increased.
- Model 2 is forced to focus on these "hard" examples (high-weight points).

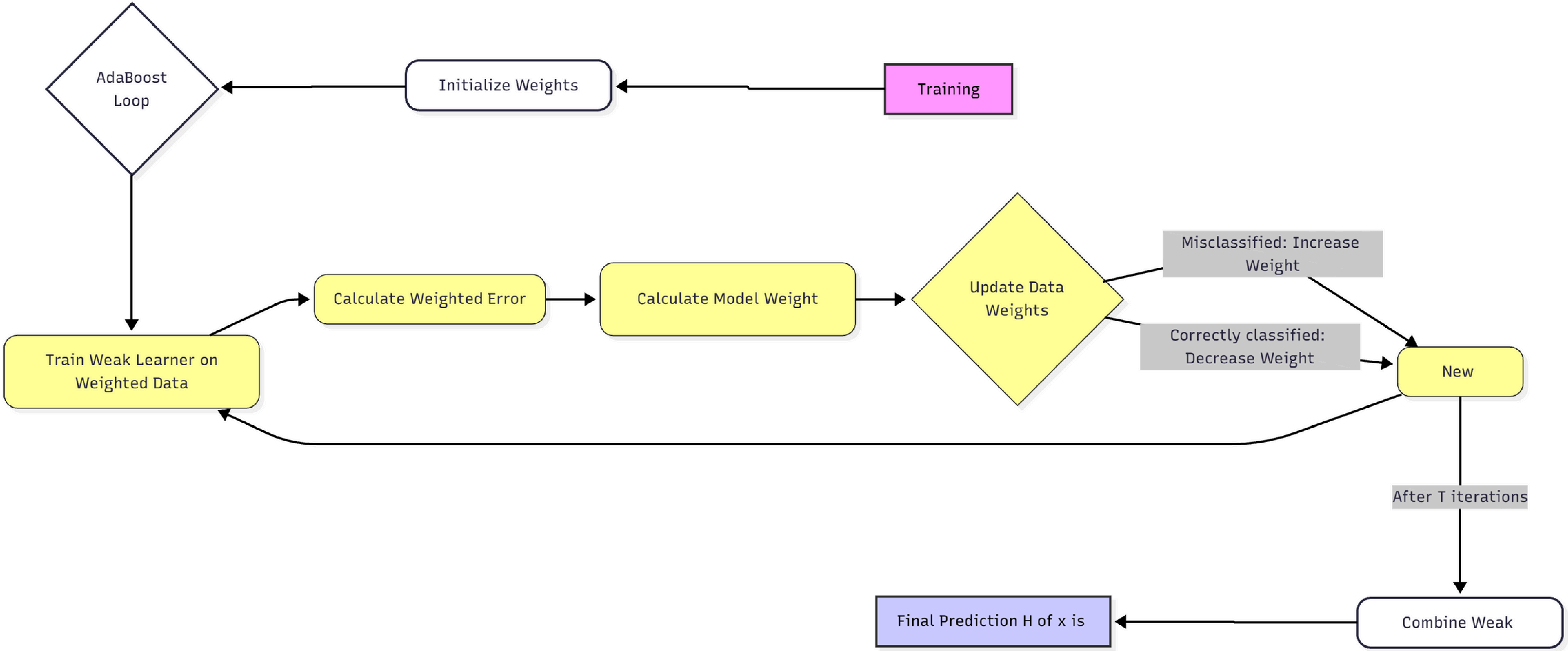
Aggregation:

- Final prediction is a weighted vote.
- More accurate models get higher voting power.

Ensemble Learning

Boosting

AdaBoost

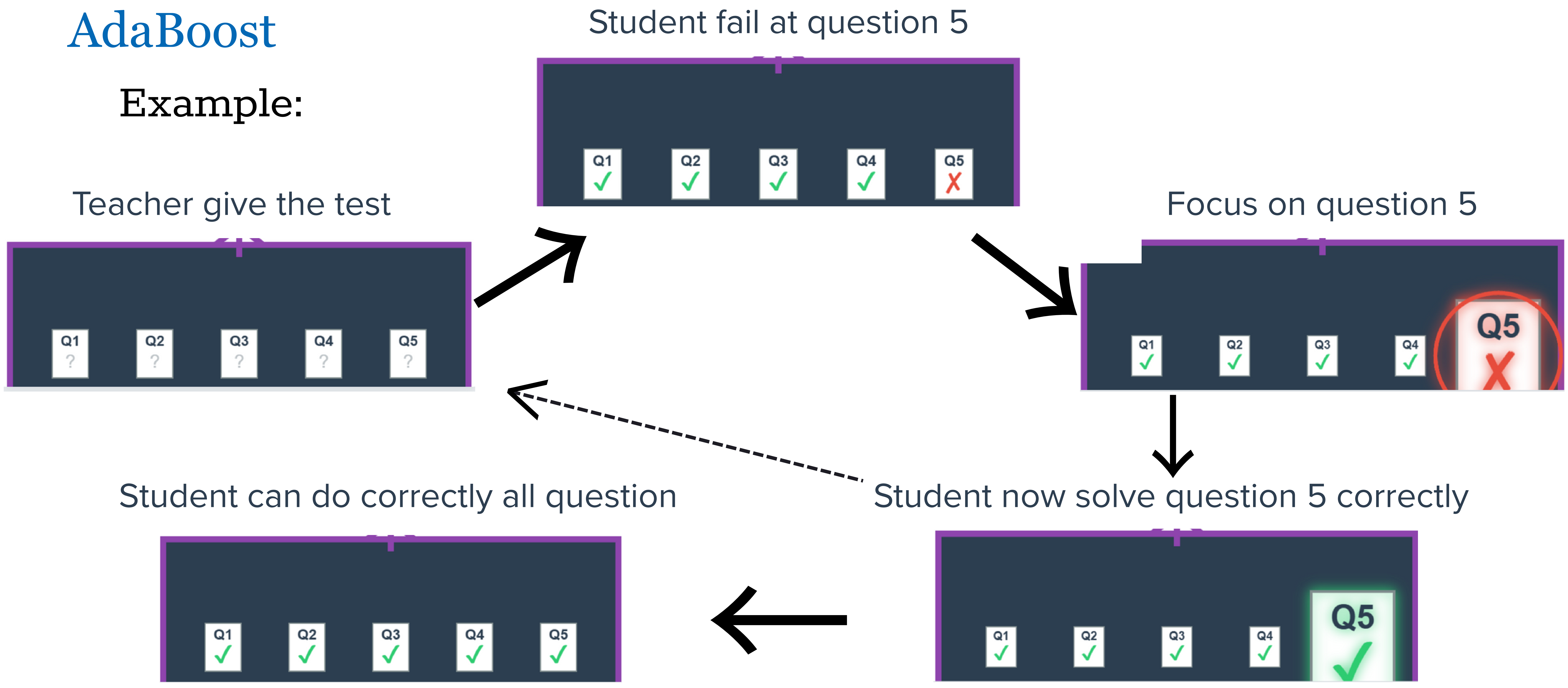


Ensemble Learning

Boosting

AdaBoost

Example:



Boosting

AdaBoost

Mathematical Foundation

AdaBoost Algorithm Step by Step:

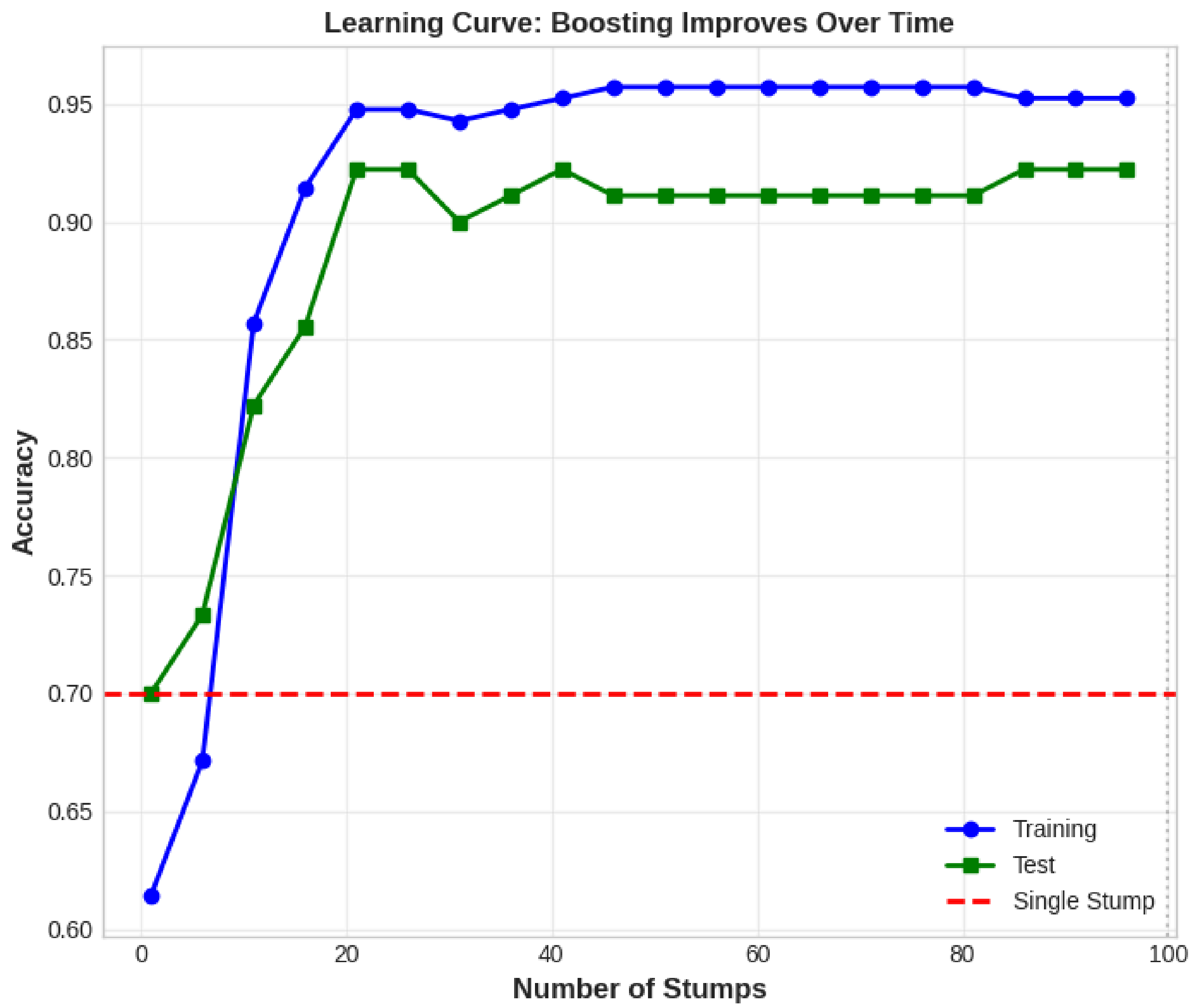
- Step 1: Initialize weights, all N samples initially have equal weights
- Step 2: For each iteration:
  - Train weak learner on data with weights
  - Calculate weighted error rate
  - Calculate model weight
  - Update sample weights
  - Normalize weights
- Step 3: Final Prediction

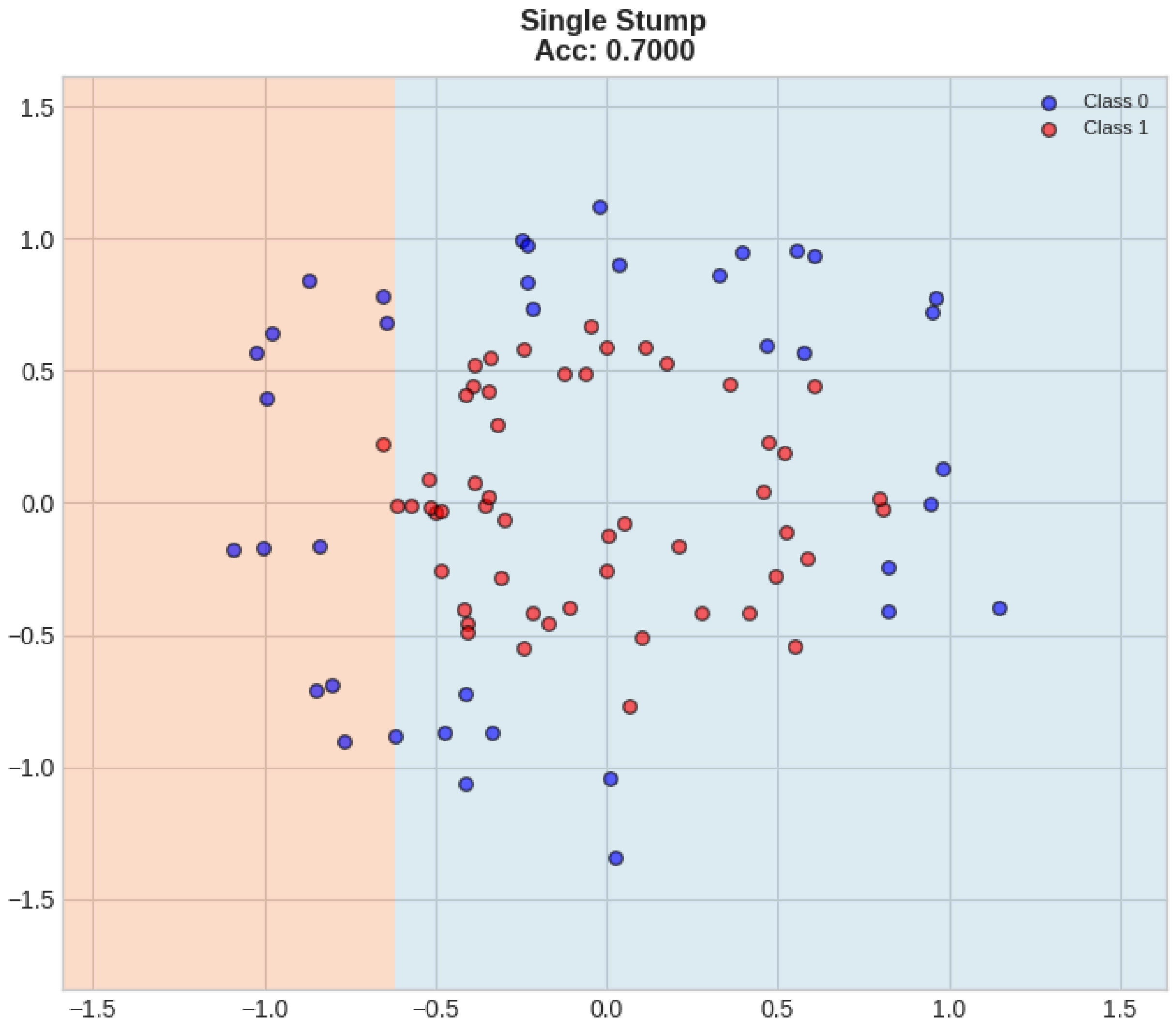
Ensemble Learning

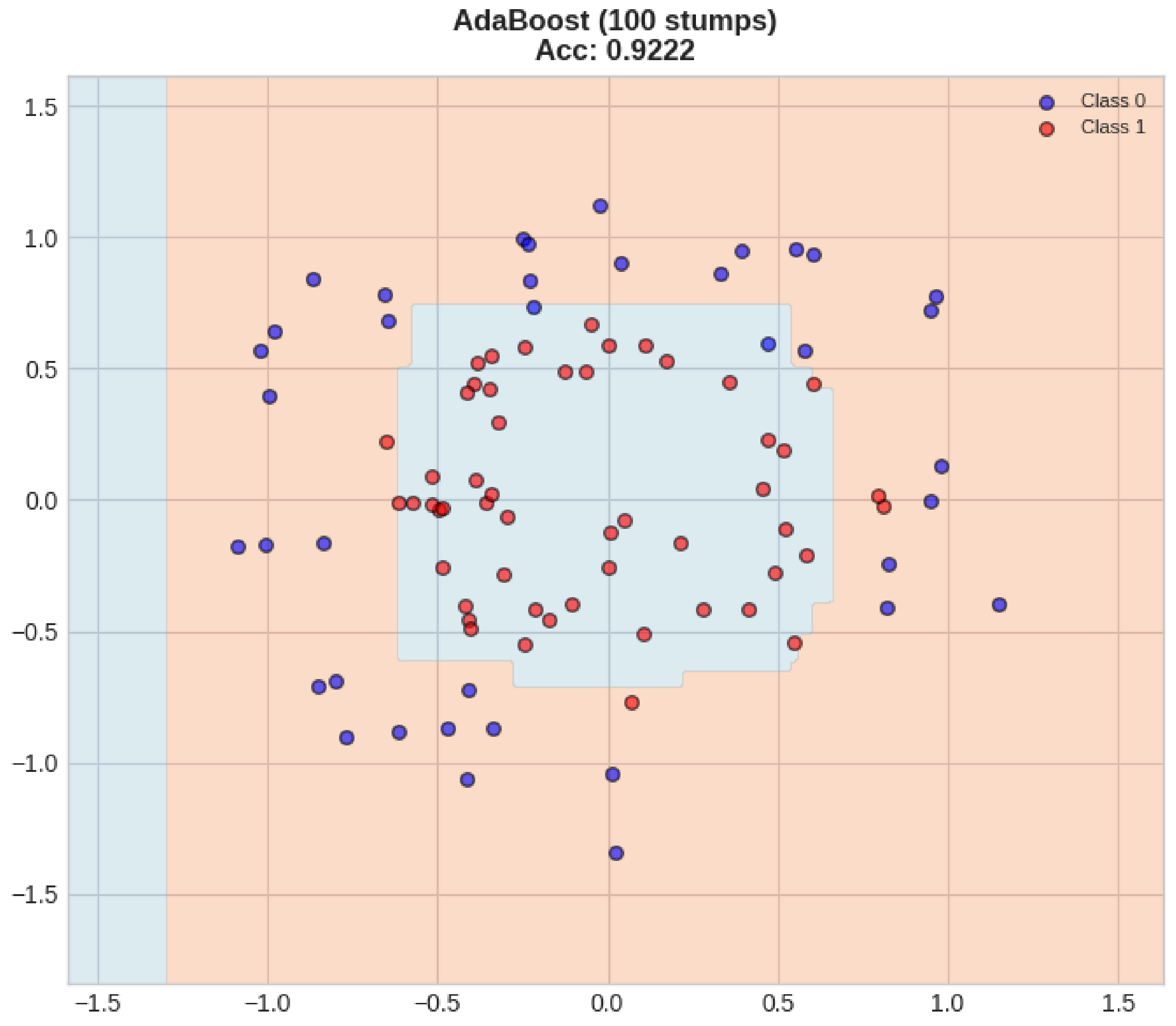
Boosting

AdaBoost

Visualization







Current: 100 stumps
Single stump: 0.7000
AdaBoost: 0.9222
Improvement: +22.22%

Boosting

Gradient Boosting

Mechanism: "Learning via Errors"

- Instead of changing sample weights, the new model tries to predict the error of the previous model.
- It uses Gradient Descent to minimize a Loss Function.

The Process:

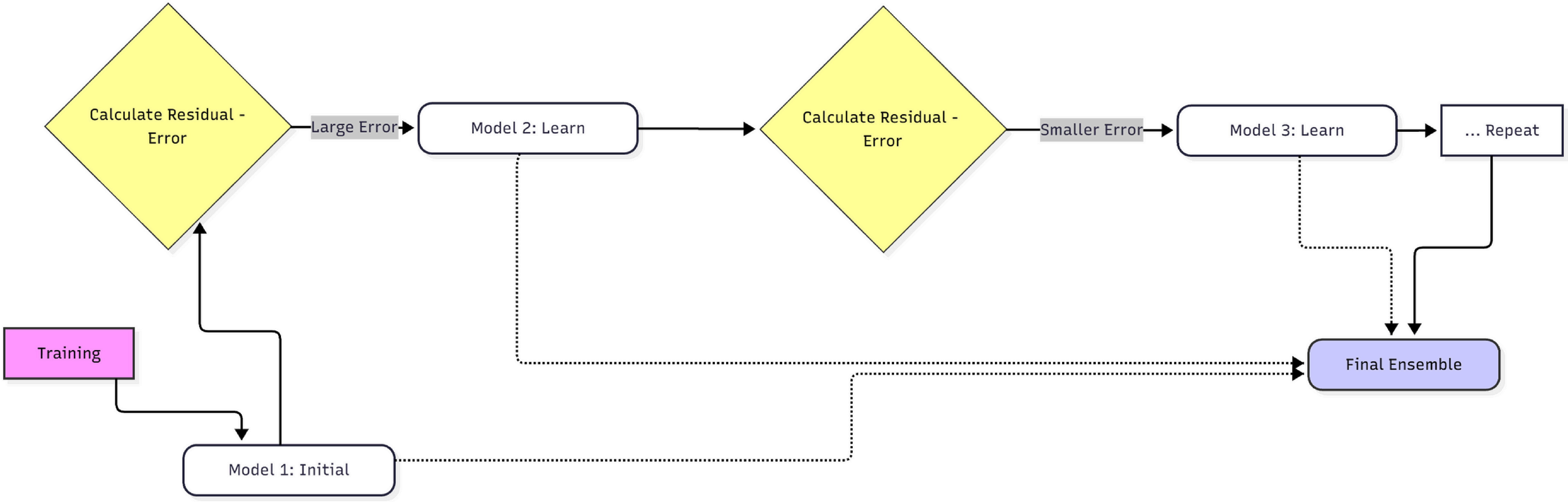
- Model 1 predicts the average.
- Calculate Residual: R = y_true - y_pred.
- Model 2 predicts R.
- Update:

$$y_{new} = y_{old} + Learning\ Rate \times Model_2(x)$$

Ensemble Learning

Boosting

Gradient Boosting



Ensemble Learning

Boosting

Gradient Boosting

Visualization

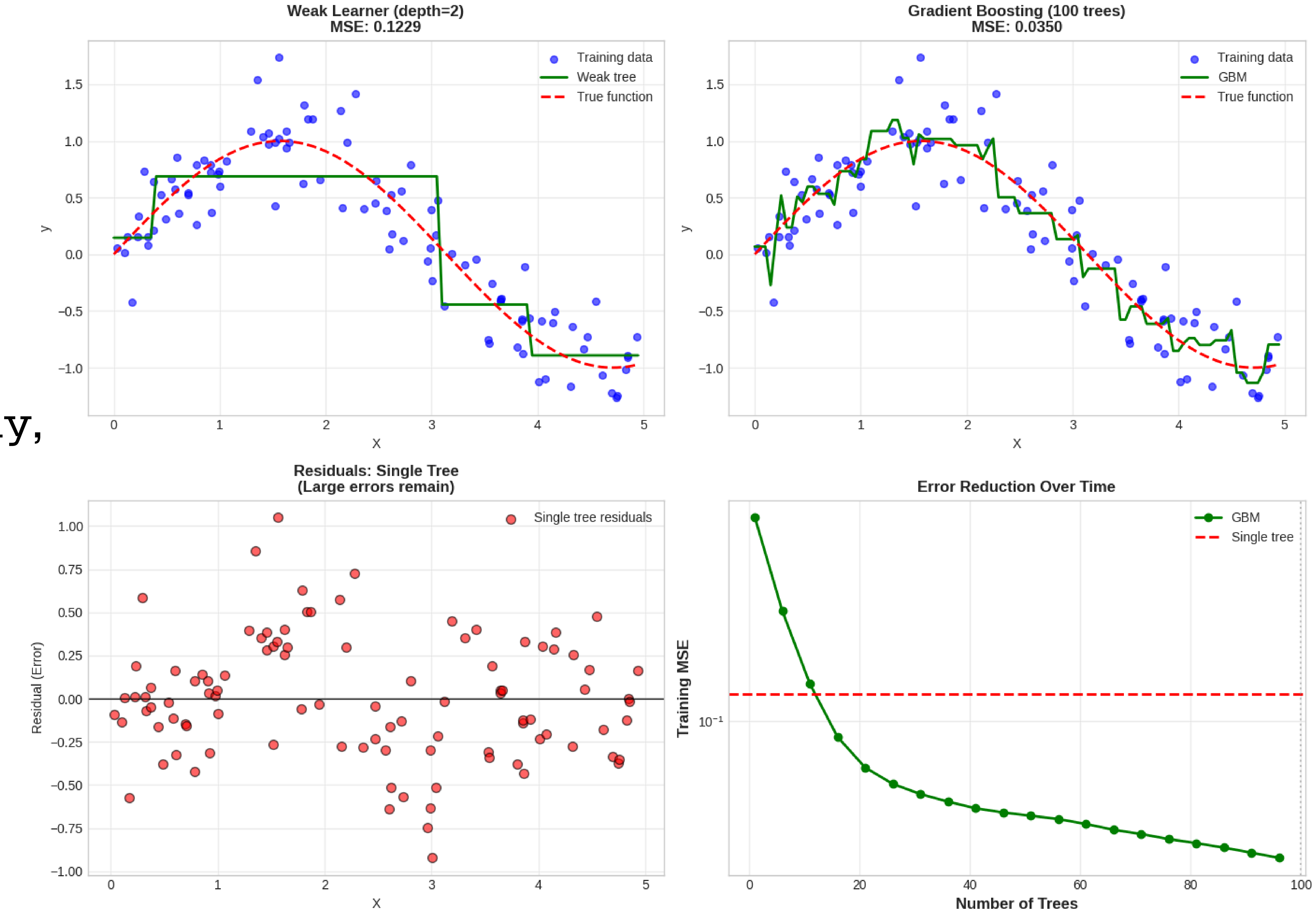
Current: 100 trees

Single tree MSE: 0.1229

GBM MSE: 0.0350

MSE reduction: 71.54%

GBM fits residuals iteratively, reducing bias each step!



Boosting

Modern Implementations

XGBoost (Extreme Gradient Boosting):

- Optimized for speed and performance.
- Includes regularization (L1/L2) to prevent overfitting.

LightGBM:

- Developed by Microsoft. Uses leaf-wise tree growth.
- Extremely fast and memory-efficient for large datasets.

CatBoost:

- Developed by Yandex.
- Handles Categorical Features automatically without complex preprocessing.

Boosting

Visualization

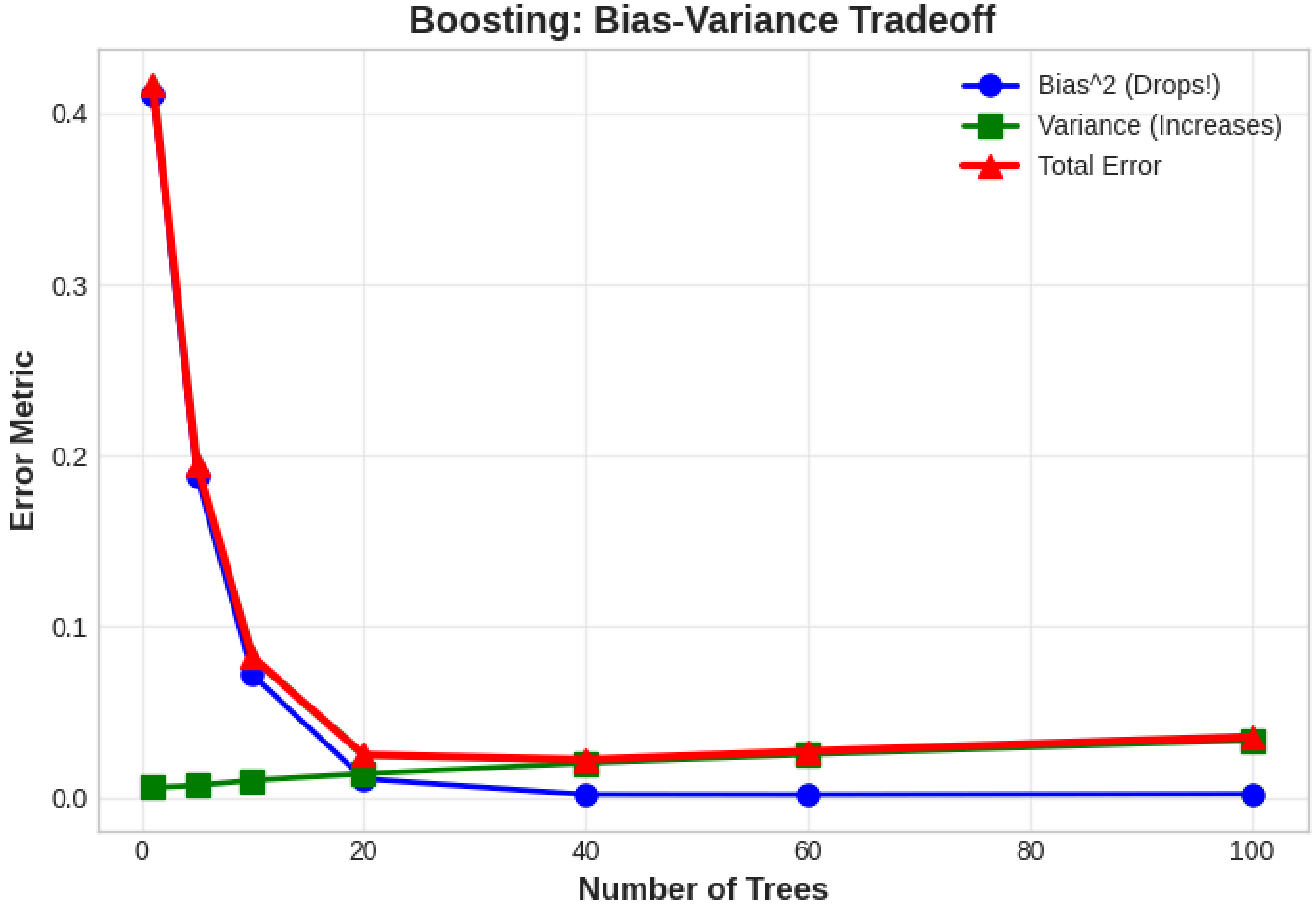
Bias reduced by: 99.49%

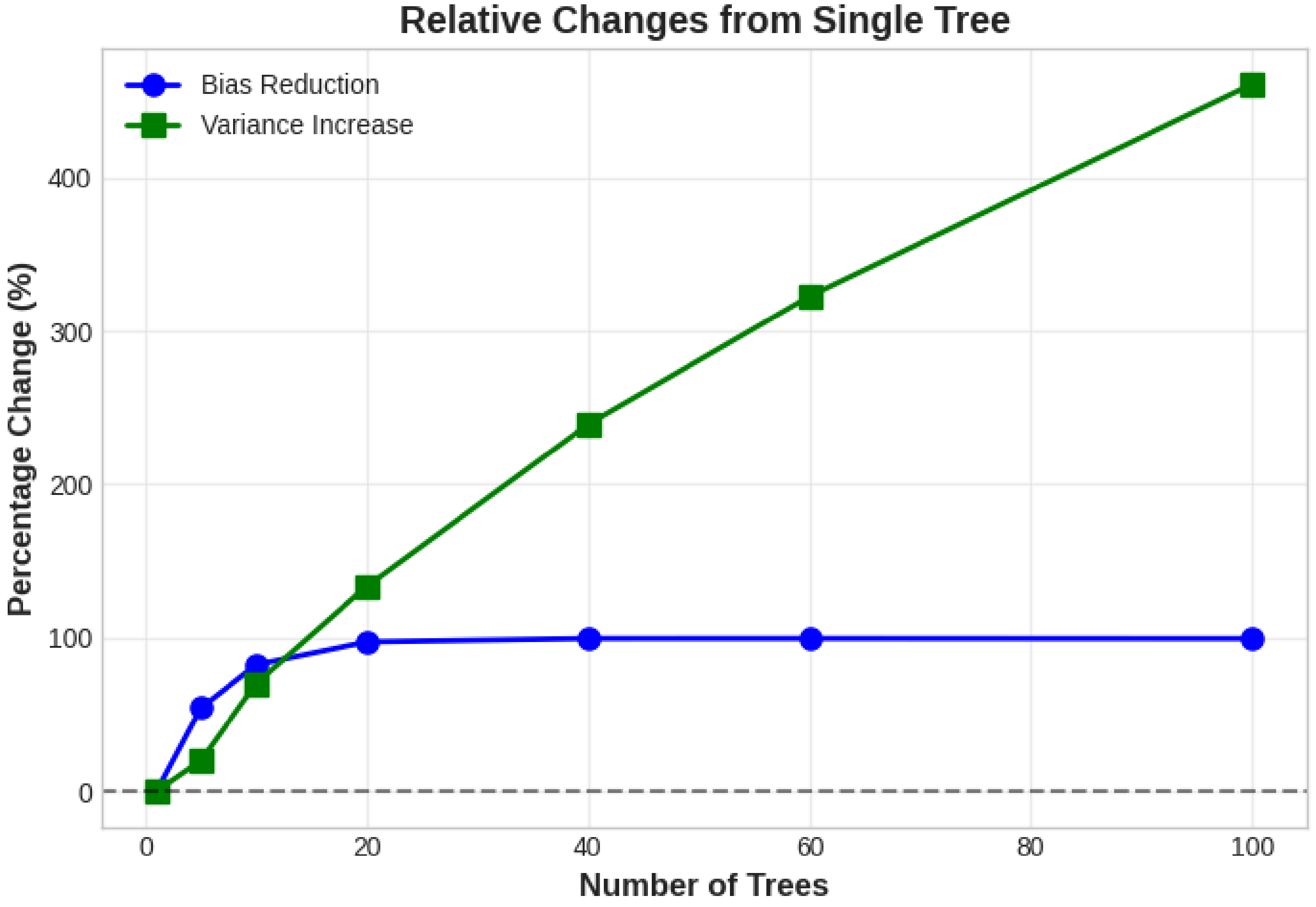
Variance increased by: 461.13%

Net effect: Total error reduced by 91.48%

CONCLUSION: Boosting reduces BIAS but increases VARIANCE!

This is opposite to Bagging.





Boosting

Conclusion & Trade-offs

When to use:

- When you need maximum accuracy on structured/tabular data.
- When the base model has High Bias (Underfitting).

Limitations:

- **Sequential:** Harder to parallelize than Bagging.
- **Noise Sensitivity:** Because it tries to fix every error, it can overfit if the data has many outliers.
- **Tuning:** Requires careful tuning of hyperparameters (Learning rate, tree depth).

Stacking

Stacking

What is Stacking?

Definition: An ensemble learning technique that combines multiple classification or regression models via a meta-classifier or a meta-regressor.

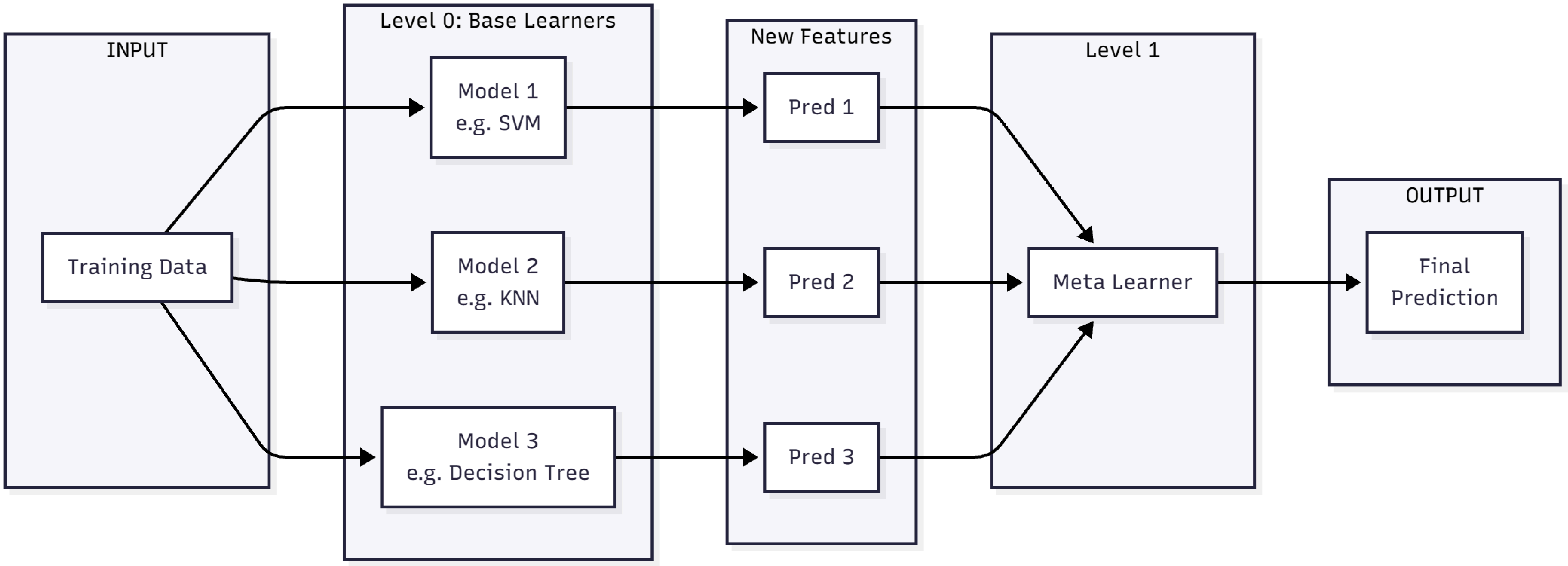
Core Philosophy:

- Bagging uses Voting (Democracy).
- Boosting uses Correction (Improvement).
- Stacking uses Learning (Management).

Analogy: Imagine a team of experts (Base Models) giving their opinions. Instead of just voting, there is a Manager (Meta Model) who knows which expert is reliable in which situation and makes the final decision.

Stacking

What is Stacking?



Stacking

The Architecture

The Two-Level Structure:

Level-0 (Base Learners):

- **Consists** of diverse models.
- **Key Requirement:** Diversity. The models should make different types of errors.

Level-1 (Meta Learner):

- **Input:** The predictions output by Level-0 models.
- **Target:** The original ground truth labels.
- **Goal:** To learn the optimal combination of base model predictions.
- **Common Choice:** Logistic Regression or Linear Regression to keep it simple and avoid overfitting.

Stacking

The Workflow

How to Train Stacking (Preventing Data Leakage)

The Problem: If we train base models and predict on the same data to train the meta-learner, it will overfit the data.

The Solution: K-Fold Cross-Validation Strategy

1. Split training data into K folds.
2. For each fold, train base models on (K-1) folds and predict on the remaining hold-out fold.
3. Collect these "Out-of-Fold" predictions to create a new training set for the Meta Learner.
4. Train the Meta Learner on these new features.

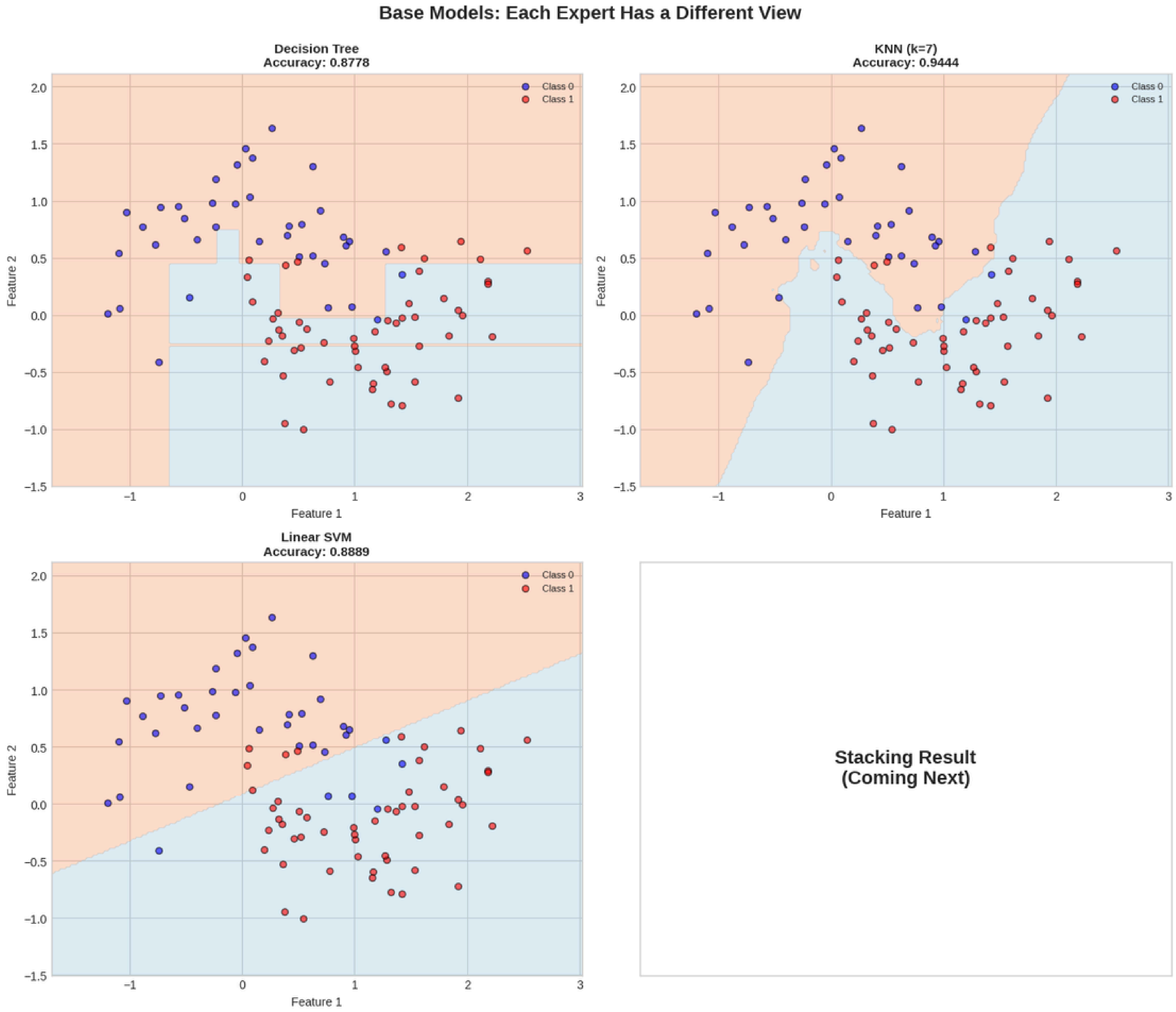
Ensemble Learning

Stacking

Visualization

Observation:

- Each model creates a different decision boundary.
- Stacking will learn how to combine these diverse perspectives!



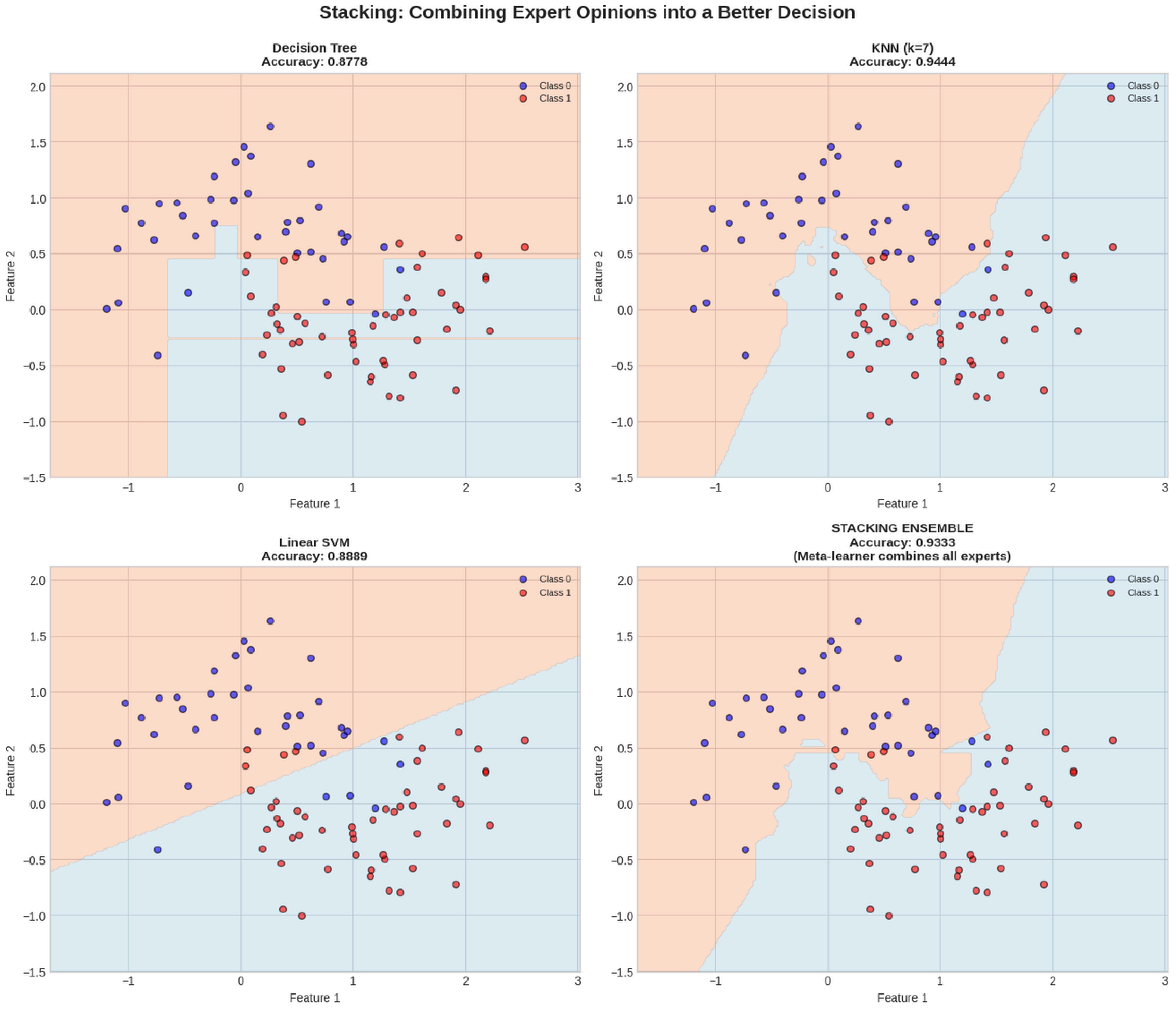
Ensemble Learning

Stacking

Visualization

NOTICE:

- The stacking boundary is smoother and more accurate than individual models.
- The meta-learner learned which expert to trust in different regions!



Ensemble Learning

Stacking

Visualization

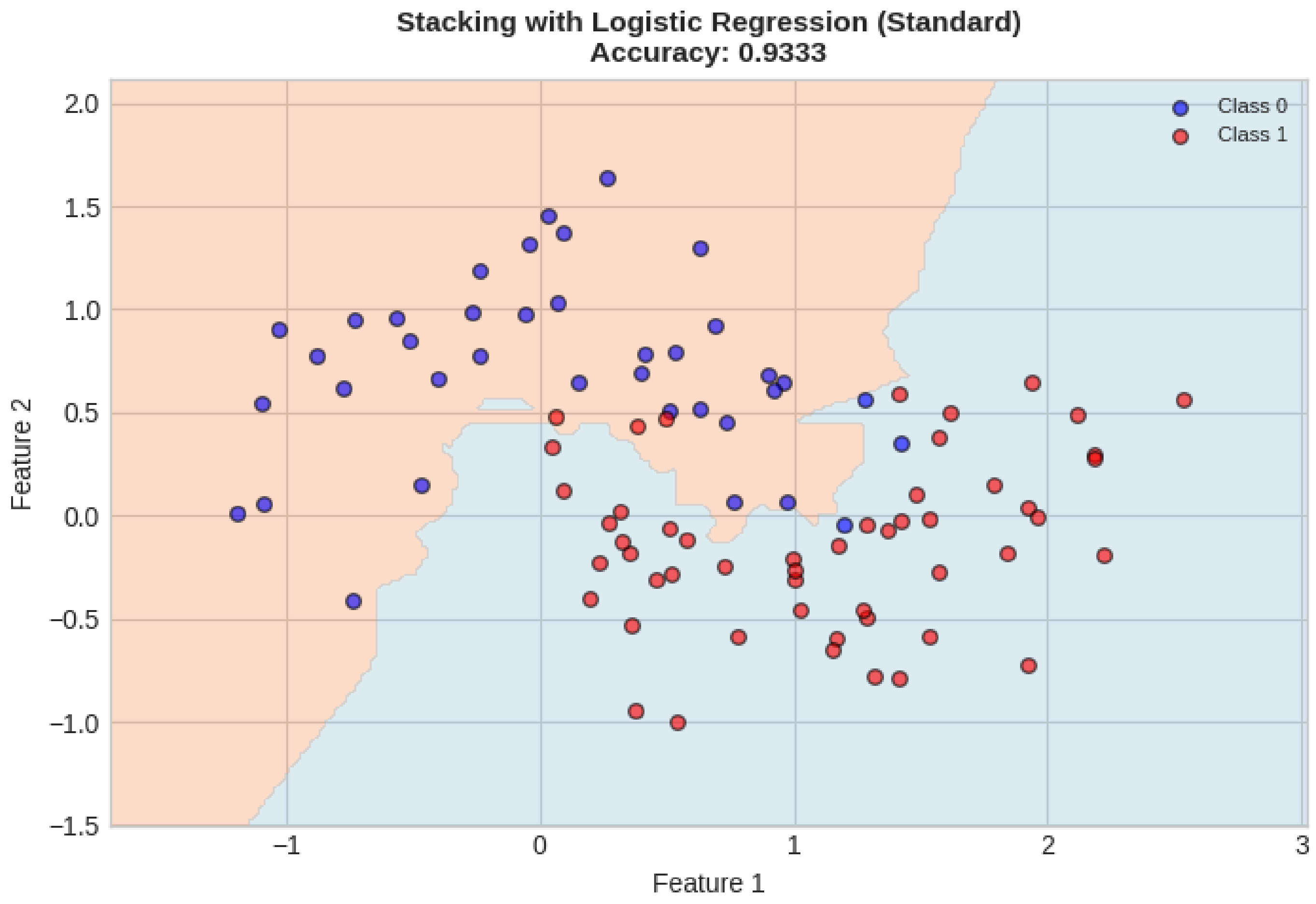
Meta-Learner: Logistic Regression (Standard)

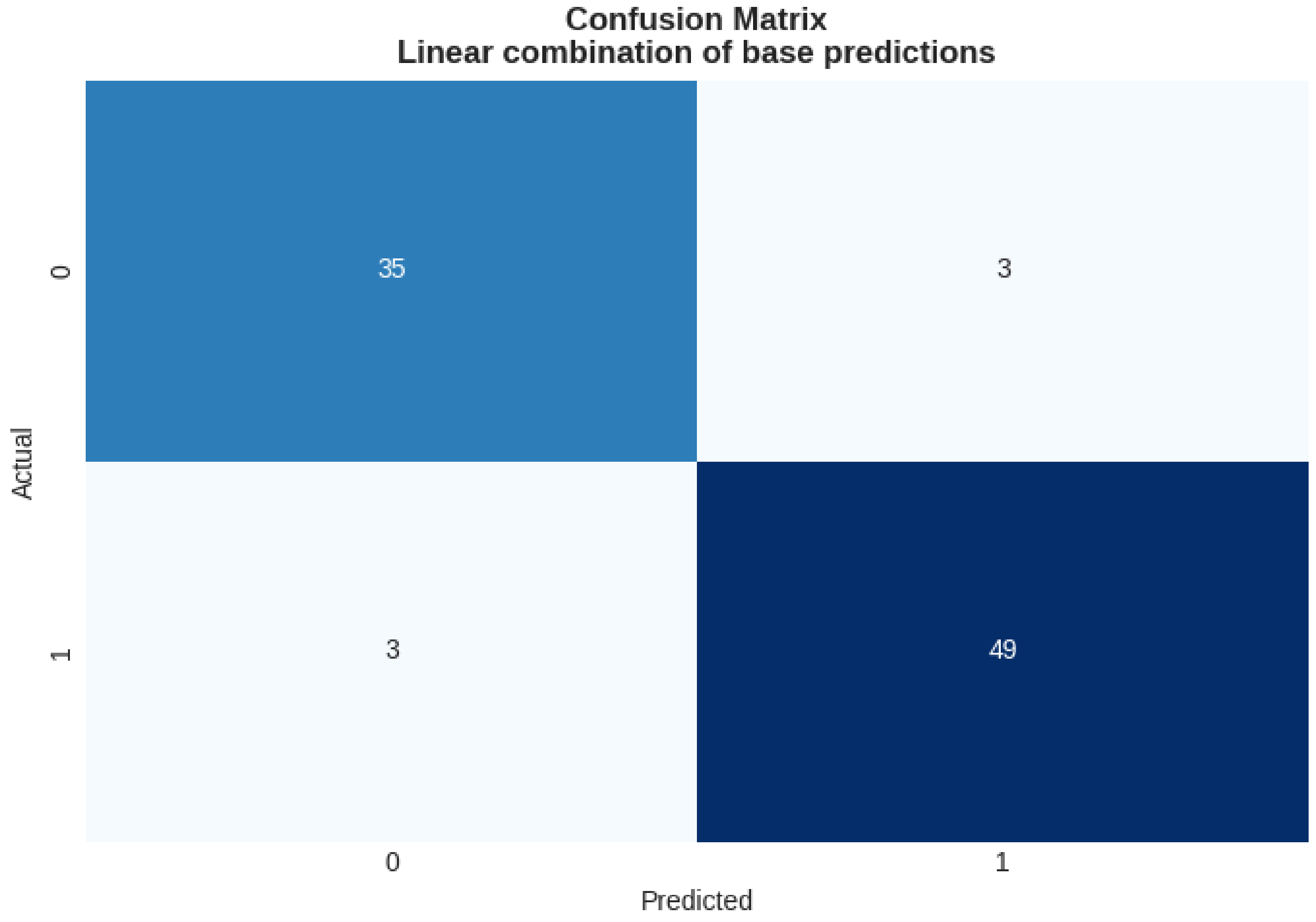
Description: Linear combination of base predictions

Accuracy: 0.9333

Comparison with base models:

- vs Decision Tree: +5.56%
- vs KNN (k=7): -1.11%
- vs Linear SVM: +4.44%





Stacking

Conclusion

Pros:

- Often achieves higher accuracy than any single model or simple voting.
- Standard winning strategy in competitions (Kaggle).

Cons:

- High Complexity: Hard to implement and explain.
- Computationally Expensive: Requires training many models multiple times.

Comparison

Ensemble Learning

Comparison

Criteria	Bagging	Boosting	Stacking
Training	Parallel (independent)	Sequential (dependent)	Multi-level
Main Goal	Reduce Variance	Reduce Bias	Maximize Accuracy
Ideal Base Learner	High Variance (Deep Trees)	High Bias (Stumps)	Diverse models
Overfitting Risk	Low	High (needs regularization)	Medium
Training Speed	Fast (parallelizable)	Slow (sequential)	Slow (many models)
Interpretability	Medium	Low	Very Low
Noise Sensitivity	Robust	Sensitive	Medium
Typical Algorithm	Random Forest	XGBoost, LightGBM	Custom ensemble

Conclusion

Conclusion

1. **When to use Bagging:**
    - You have high-variance models (like deep decision trees)
    - Training time is a concern (can parallelize)
    - Want to reduce overfitting without much tuning
    - Example: Random Forest for general-purpose classification
2. **When to use Boosting:**
    - You have high-bias models (weak learners)
    - Accuracy is more important than training time
    - Have clean data (less sensitive to outliers with proper tuning)
    - Example: XGBoost/LightGBM for Kaggle competitions
3. **When to use Stacking:**
    - You have diverse models with different strengths
    - Want to squeeze out maximum performance
    - Have enough data to properly train meta-learner
    - Computational cost is not a primary concern
    - Example: Final ensemble in competitions

ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

Thanks for watching