McMaster University ENGINEERING

W BOOTH SCHOOL OF ENGINEERING PRACTICE AND TECHNOLOGY

SEP 785: Machine Learning

Lecture 9: Ensembles

Instructor: Dr. Dalia Mahmoud, PhD
(Mechanical Engineering, McMaster University)
Email: mahmoudd@mcmaster.ca

Project and Lecture timeline updated

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Week	lecture		Project
25th of Feb	Logistic Regression + SVM + Naïve Bayes	In person	Data Collection and Pre-processing
4th of March	Models Evaluation + Python hands on	In person	
11th of March	Python hands on --- Logistic Regression	Online	
18th of March	Ensembles	Online	Choosing and Applying ML Algorithm(s)
25th of March	Unsupervised Learning	Online	
1st of April	No lecture --- Eid Vacation		
8th of April	No lecture !! Just assignment (Last Assignment)	Online	Video Presentation Deliverable
15th of April	Projects discussion	Online	
22nd of April	Introduction to Neural Networks	In person	Peer Review and Discussion

Recap

- Supervised ML Models
- Model Evaluation
- Python hands on scripts

Intended Learning Outcomes

- **Understand** why ensemble learning improves model performance.
- **Describe** the role of bootstrapping in ensemble methods.
- **Explain** how Bagging reduces variance and Boosting reduces bias.
- **Differentiate** between key ensemble methods: Bagging, Boosting, Stacking, and Voting.

Contents

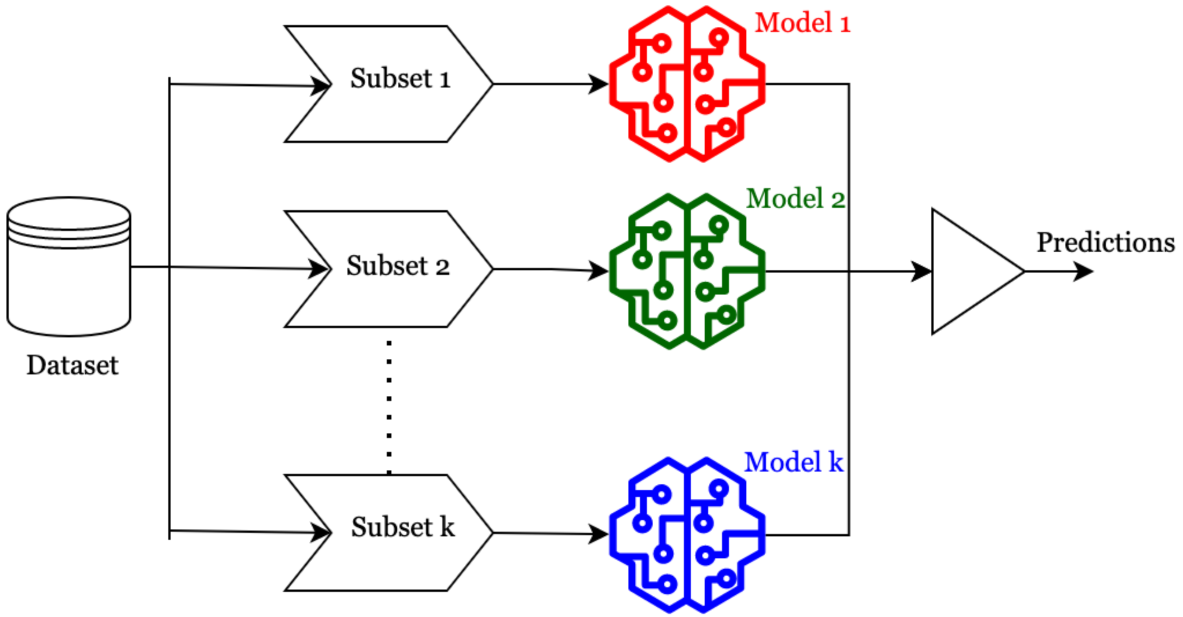
- Decision Tress review
- Random Forest
- Gradient Boosting
- Ada Boosting
- Summary

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Ensemble Learning: What is ?

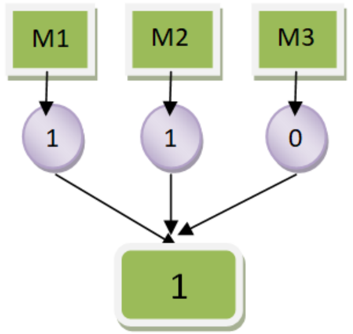
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Ensemble learning combines multiple models to improve overall accuracy and robustness.
- Different models make different mistakes—combining them reduces errors.
- Common ensemble methods: Bagging, Boosting, Stacking, and Voting

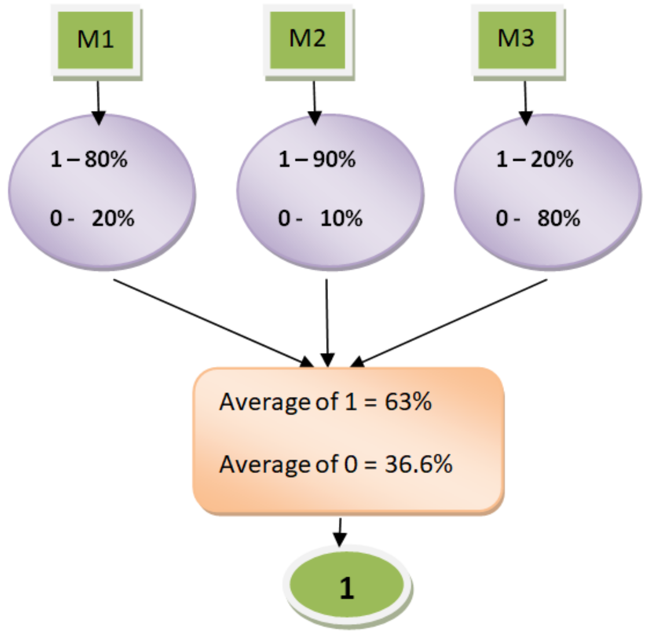


Ensemble Learning: Why Combine Models ?

- Different models make different mistakes. Can averaging their predictions help?
- Ensemble methods improve accuracy if models have diverse errors.
- Two main techniques: Hard Voting (majority rule) & Soft Voting (probability-based).
- Classes can get different weights to influence the final decision.



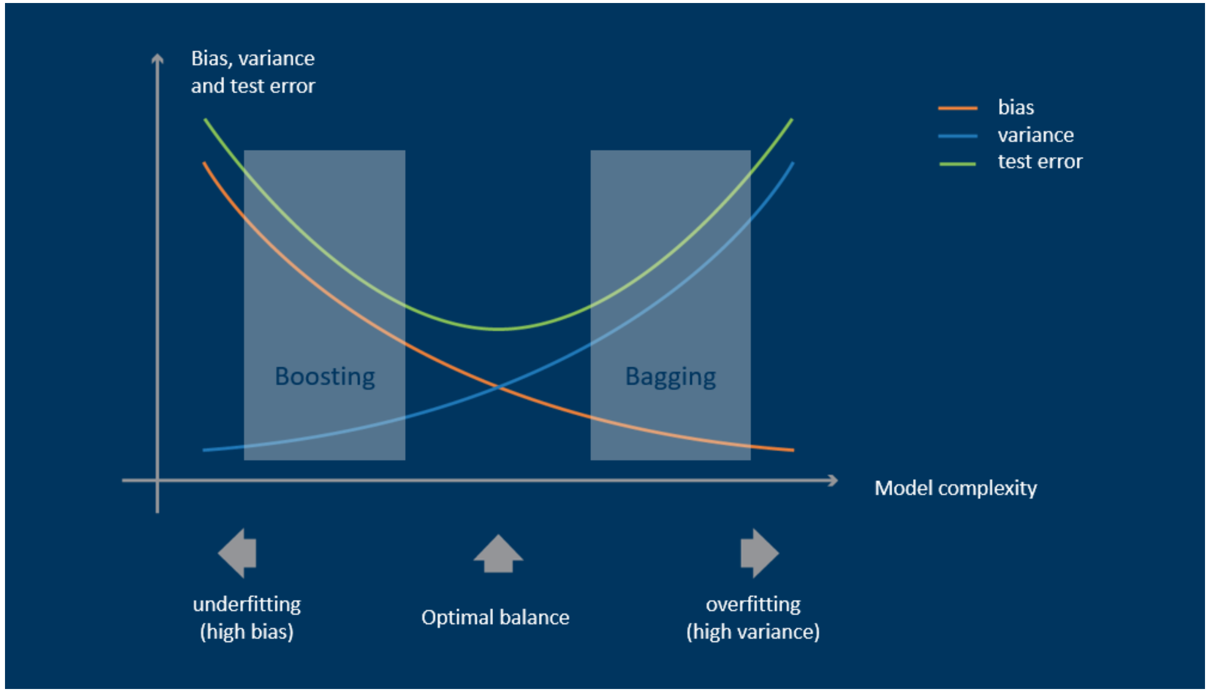
Hard Voting



Soft Voting

Bias-Variance Tradeoff in Ensembles

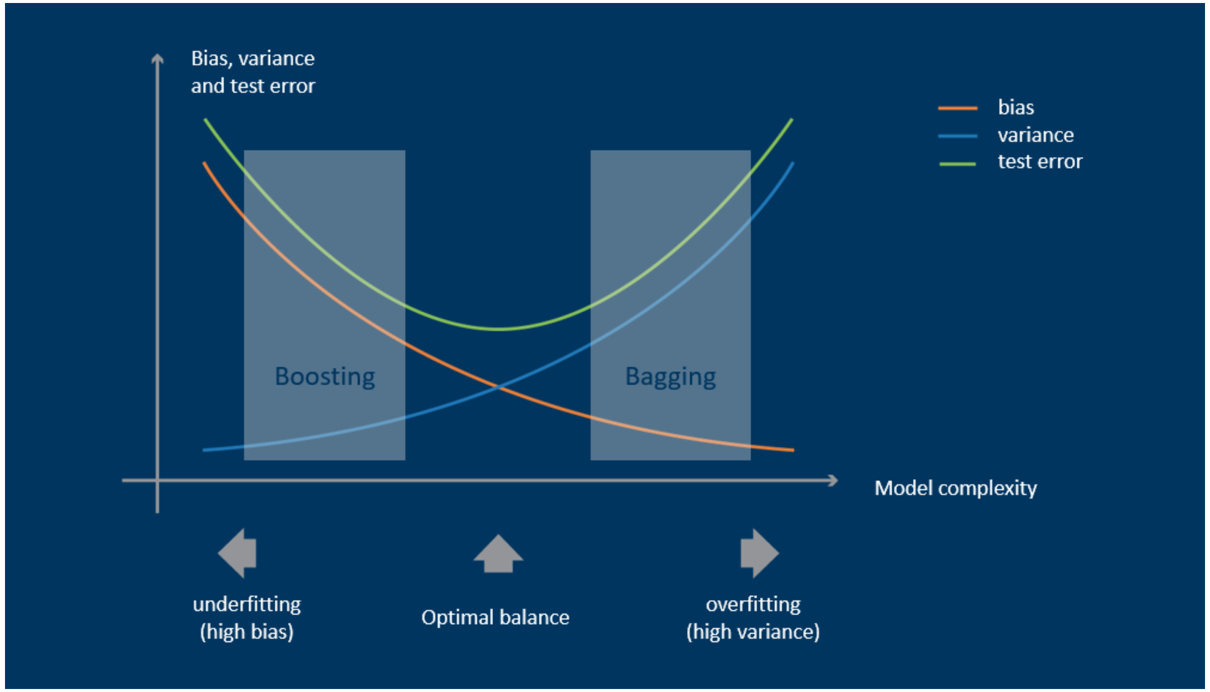
- **High bias (underfitting):** Combine with other low-variance models to reduce bias.
  - models should specialize in different parts of the data.
  - Example: Boosting helps reduce bias.



Bias-Variance Tradeoff in Ensembles

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- **High variance (overfitting):** Combine with other low-bias models to reduce variance.
  - Individual mistakes should be different.
  - Example: Bagging helps reduce variance.

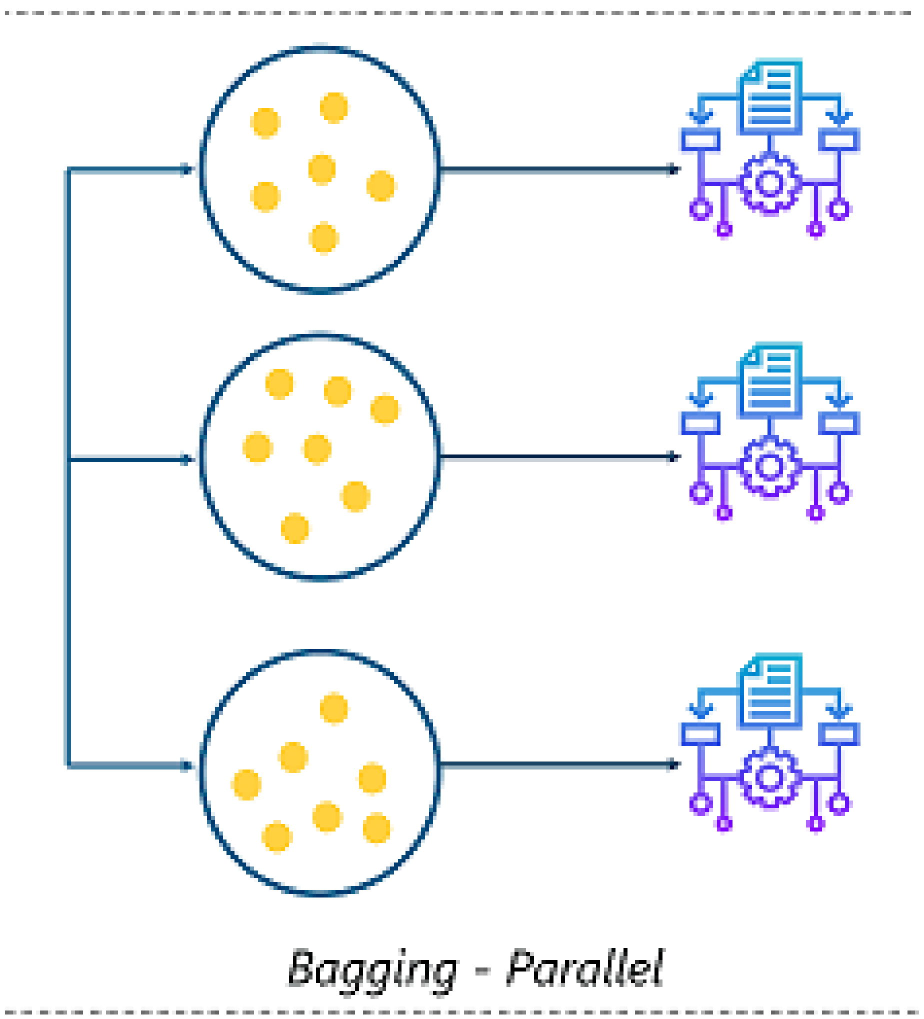


Contents

- Introduction to Ensembles
- Bagging
- Gradient Boosting
- Stacking
- Summary

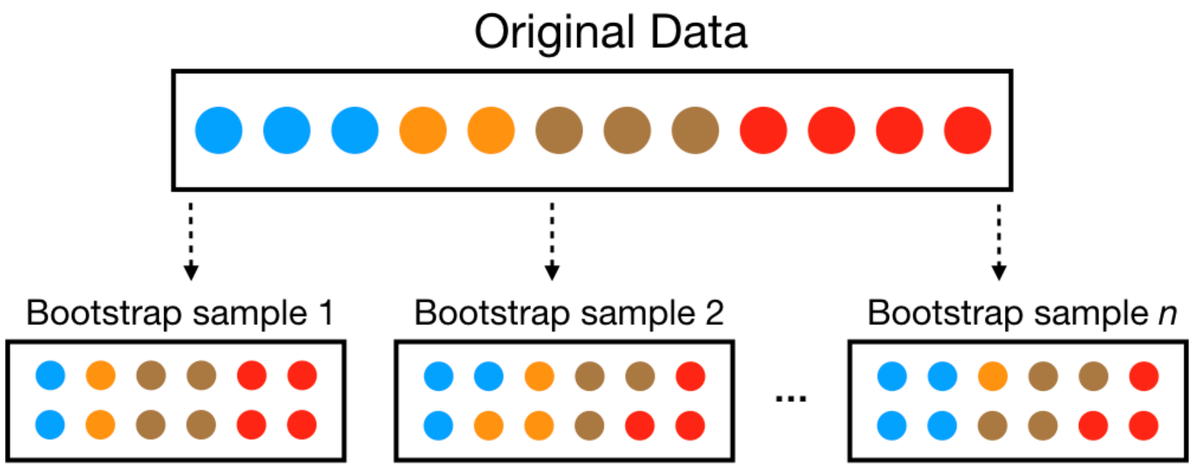
Bagging

- The Bagging (Bootstraps Aggregation) is an ensemble technique that improves accuracy by training multiple models on different random subsets of data and averaging their predictions.
- **Key Idea:** Reduces variance by **averaging multiple weak learners**.



Bootstrapping Sampling

- Bootstrapping is a resampling technique used to create multiple training datasets.
- Each dataset is sampled **with replacement** from the original data.
- Used in **Bagging** (Bootstrap Aggregating) to train multiple models on different subsets.
- Helps reduce variance and improve stability in ensemble models.

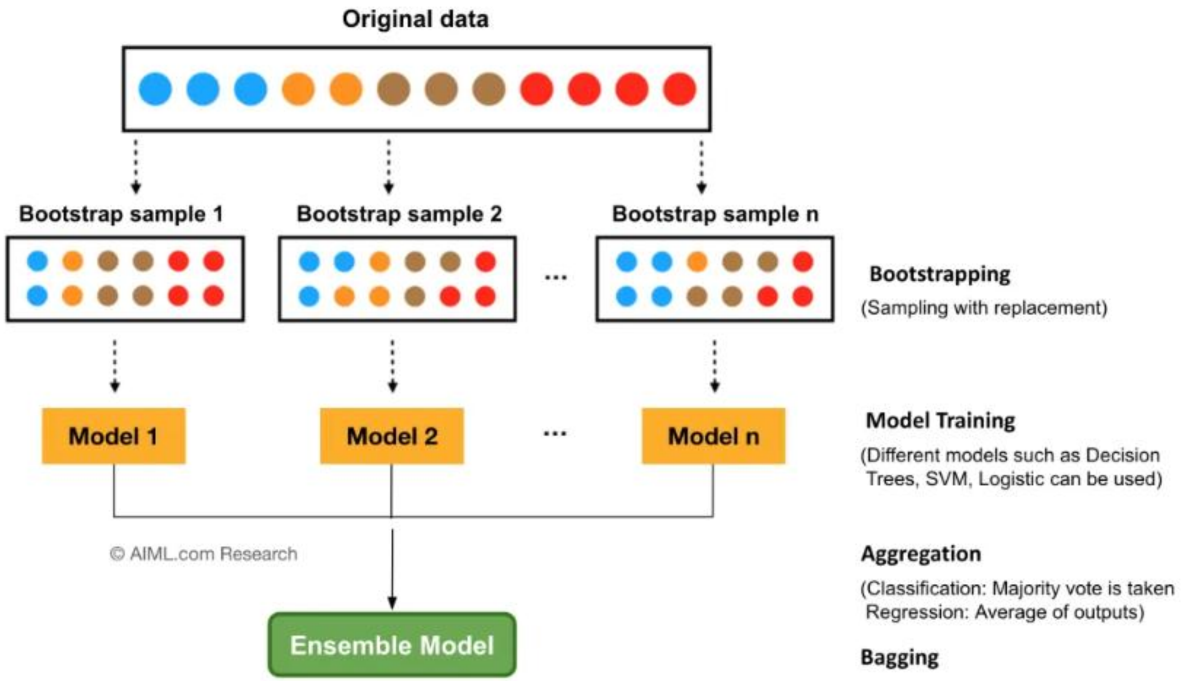


Bagging

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

1. **Bagging (Bootstrap Aggregating)** improves model stability by reducing variance. Steps: Create multiple training sets using **bootstrapping** (sampling with replacement).
2. Train separate models (e.g., Decision Trees) on each dataset.
3. Combine predictions using **averaging** (for regression) or **voting** (for classification).

Works best with **high-variance models** (e.g., Decision Trees, Neural Networks)

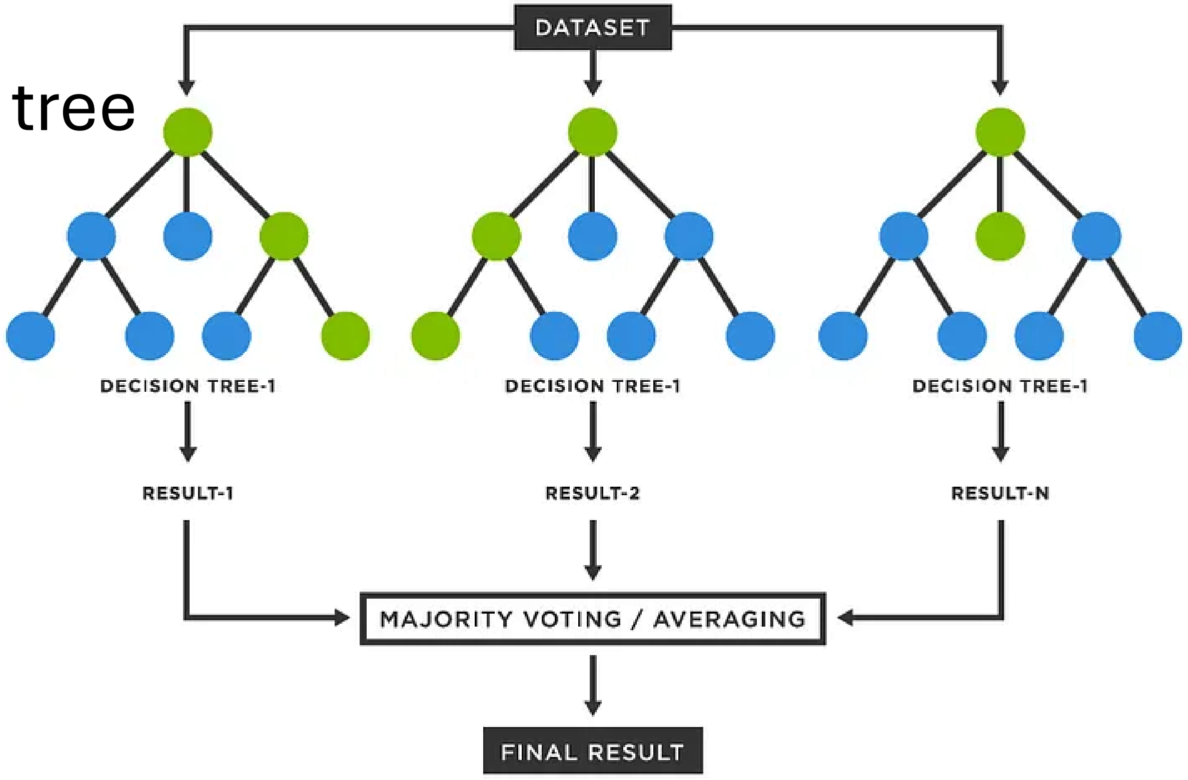


Bagging (Source: AIML.com Research)

Random Forest

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

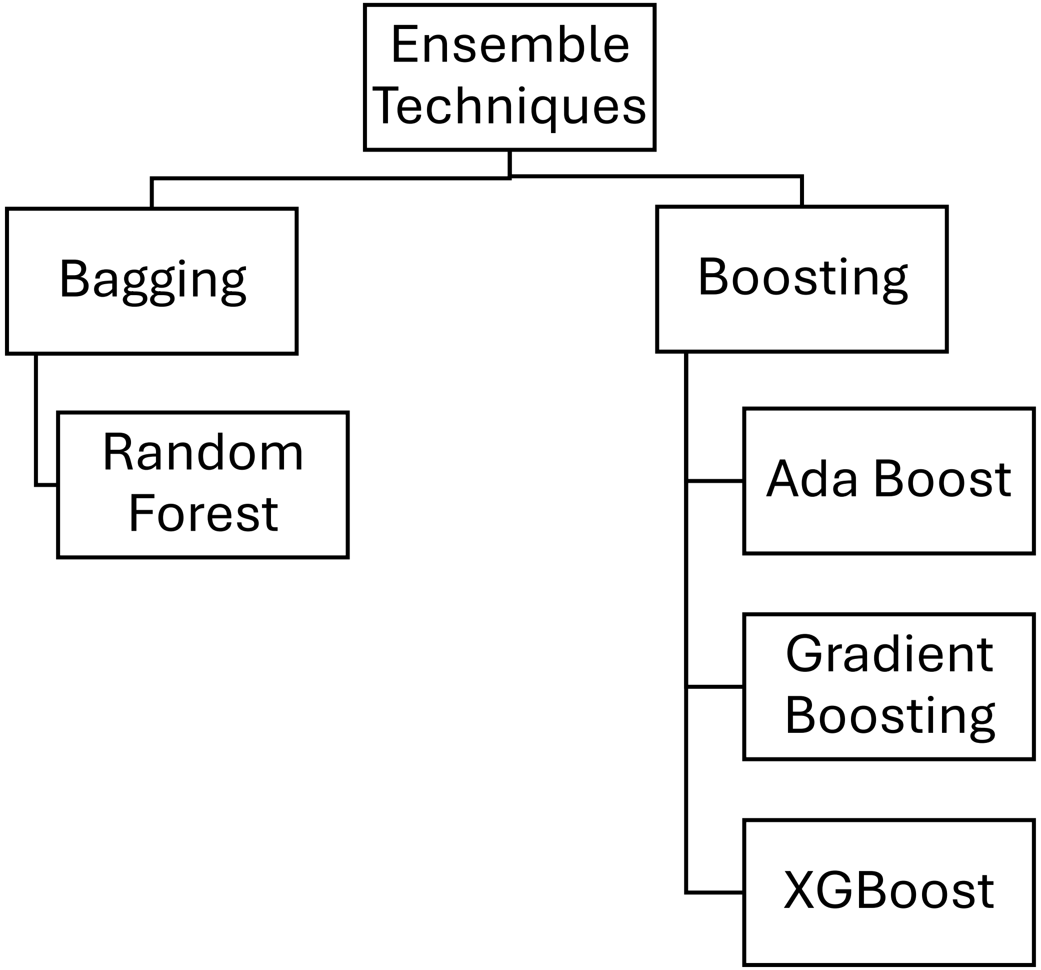
- **Random Forest = Bagging + Decision Trees.**
- Instead of a single Decision Tree, multiple trees are trained on **bootstrapped** datasets. Additional randomness: Each tree only considers a **random subset of features** at each split.
- Predictions are combined using:
    - **Averaging** (for regression)
- **Majority voting** (for classification)



What are Ensembles

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Machine learning models often suffer from **high variance** or **high bias**.
- **Ensemble Learning**: Combines multiple models to improve performance.
- Two main types:
  - **Bagging (Bootstrap Aggregating)** → Reduces variance.
  - **Boosting** → Reduces bias.



Decision Tree Vs Random Forest

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Single Decision Tree

- can become too **complex** and **overfit** to the training data, meaning it performs well on training but poorly on unseen data.
- **highly sensitive** to small changes in the data.
- may be **misled** by noise in the data.
- might focus too much on **specific features**, causing bias. (If there are **many features)**
- can be fast but might **struggle with large datasets**.

Random Forest

- **less prone to overfitting**.
- **averages multiple trees**, reducing variance and ensuring more **consistent** results.
- aggregates predictions, it is **less affected by noise** and handles missing values better.
- **randomly selects a subset of features** for each tree, ensuring more **diverse decision boundaries**.
- can **parallelize computation**, making it scalable for large datasets.

Contents

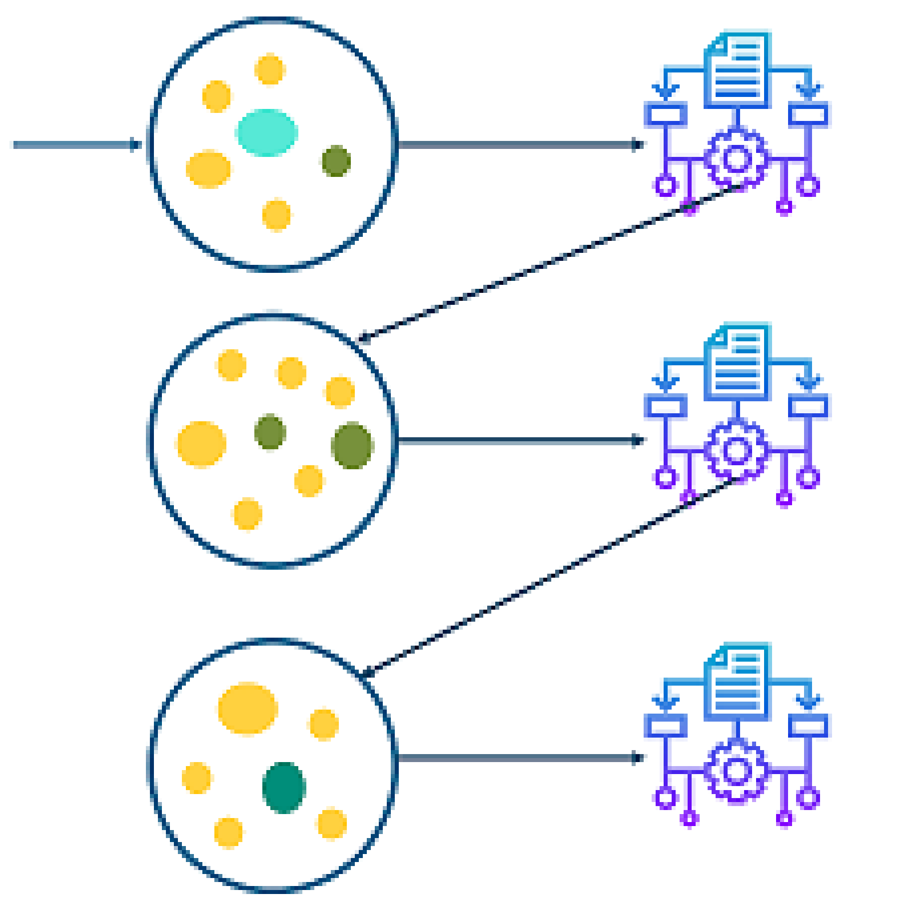
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Introduction to Ensembles
- Bagging
- Gradient Boosting
- Stacking
- Summary

Boosting

1. **Boosting** focuses on reducing **bias** by training models sequentially. Steps: Train a weak model (e.g., shallow Decision Tree).
2. Identify misclassified samples and give them **higher weights**.
3. Train the next model to correct previous mistakes.
4. Repeat and combine models using a weighted sum.

Works best with **high-bias models** (e.g., Decision Stumps).



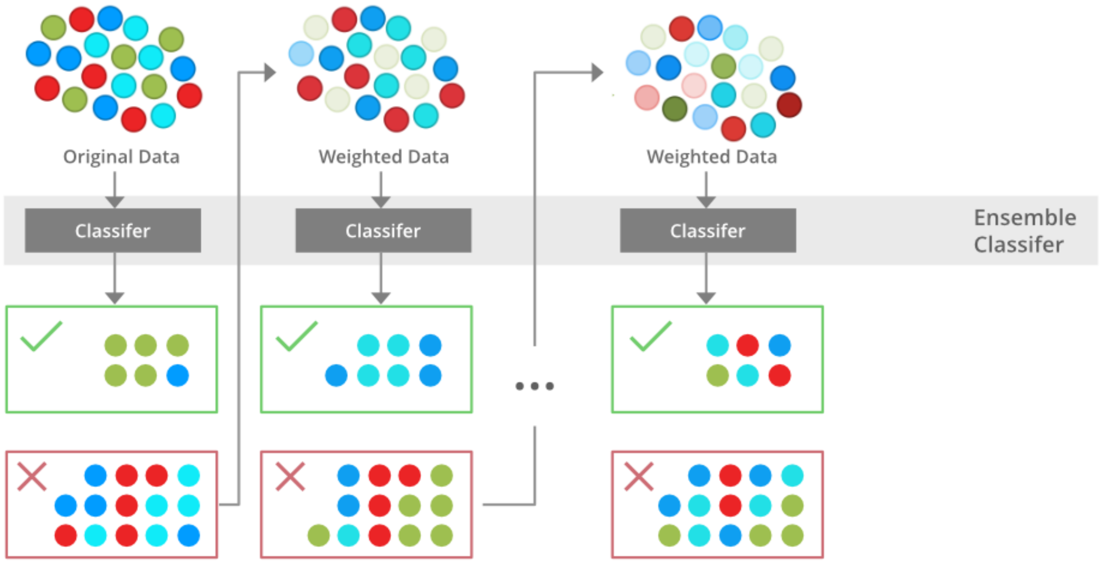
Boosting - Sequential

Boosting Algorithms

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

1. **Adaptive Boosting (AdaBoost)** is a boosting algorithm that focuses on **hard-to-classify** examples. Steps: Train a weak model (e.g., a shallow Decision Tree).
2. Increase the weights of misclassified samples so the next model focuses on them.
3. Train a new model on the updated dataset.
4. Repeat, combining models using a weighted vote.

Key Idea: Weak models (learners) combine into a **strong classifier**. Works well with **high-bias models** like Decision Stumps (1-level trees).

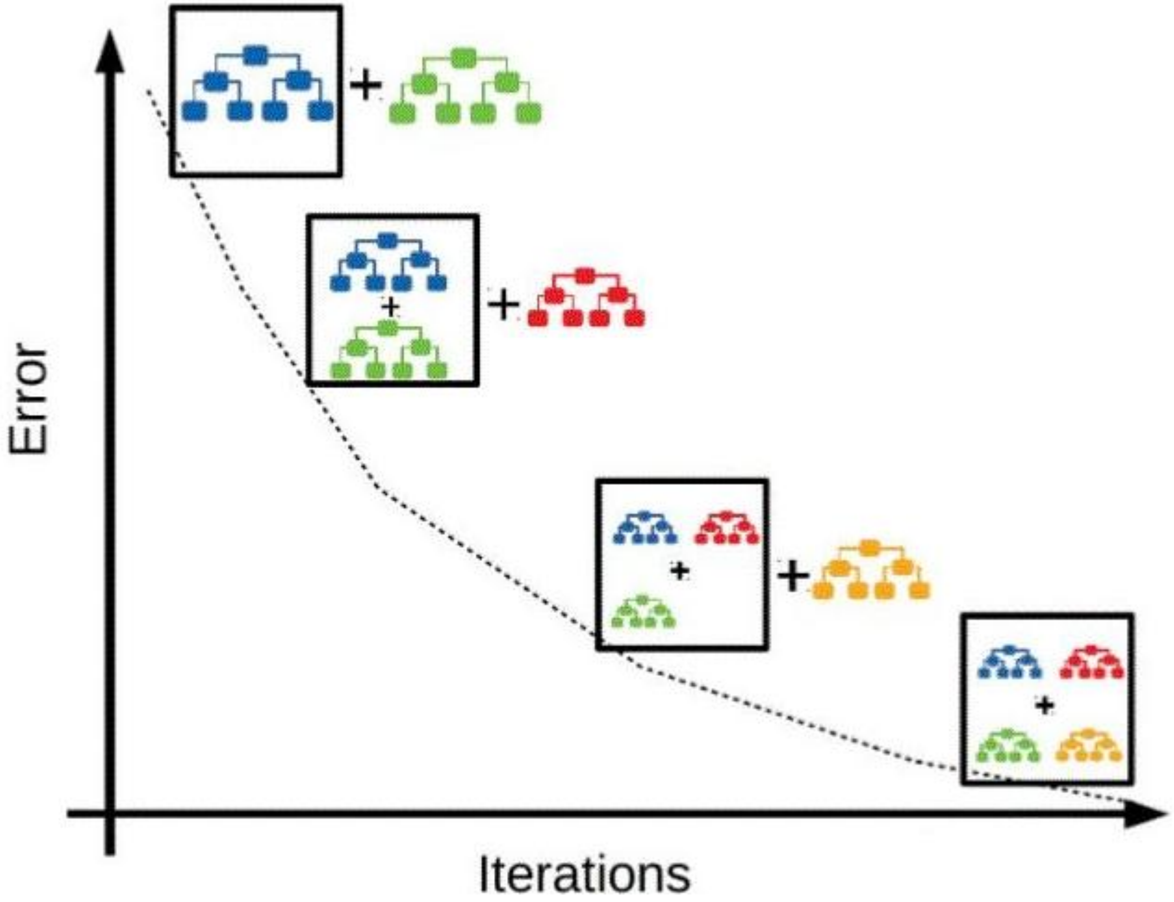


Gradient Boosting

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

1. **Gradient Boosting** is a boosting technique that builds models **sequentially** to minimize errors. Steps: Train a weak model (e.g., Decision Tree).
2. Compute the **error (residuals)** between predictions and actual values.
3. Train the next model to predict these residuals (gradually correcting mistakes).
4. Repeat the process, updating predictions at each step.

- Uses **Gradient Descent** to optimize the model. More flexible than AdaBoost; works well for both **classification** and **regression**. Example algorithms: **XGBoost** (Extreme Gradient Boosting)
- **LightGBM** (Light Gradient Boosting Machine)



Contents

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

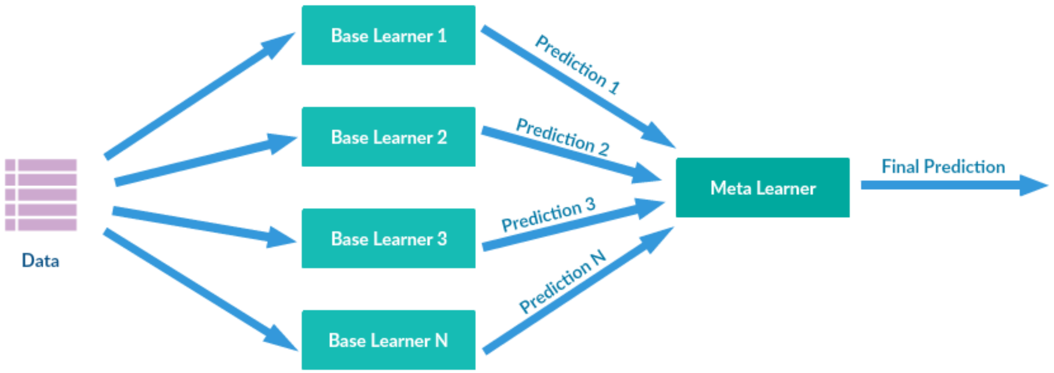
- Introduction to Ensembles
- Bagging
- Gradient Boosting
- Stacking
- Summary

Stacking

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

1. **Stacking (Stacked Generalization)** combines multiple models by **learning how to best combine their predictions**. Steps: Train multiple **base models** (e.g., Decision Trees, SVMs, Neural Networks).
2. Collect their predictions as **features** for a new model.
3. Train a **meta-model** (e.g., Logistic Regression) to make the final prediction.

Key Difference from Voting: Instead of simple averaging, stacking **learns** the best way to combine predictions. Works well when base models are **diverse and complementary**.



Voting Vs Stacking

Feature	Voting	Stacking
Definition	Combines predictions using majority vote (classification) or average (regression).	Uses a meta-model to learn how to best combine multiple model predictions.
Model Training	Models trained independently in parallel.	Models trained first, then meta-model learns from their outputs.
Combination Method	Hard Voting (majority) or Soft Voting (weighted average).	Meta-model (e.g., logistic regression, neural network) decides the best combination.
Complexity	Low (simple aggregation).	High (requires extra training for meta-model).
Bias-Variance Tradeoff	Reduces variance but may not significantly lower bias.	Reduces both bias and variance by learning optimal model combinations.
Performance	Can improve accuracy, but limited compared to stacking.	Typically outperforms voting when tuned properly.
Interpretability	Easy to understand and implement.	Harder to interpret due to added complexity.
Example	Random Forest (built-in voting), simple ensemble classifiers.	Stacked Generalization (e.g., combining SVM, Decision Trees, Neural Networks).

Contents

- Introduction to Ensembles
- Bagging
- Gradient Boosting
- Stacking
- Summary

Bagging Vs Boosting

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Feature	Bagging	Boosting
Purpose	Reduces variance, stabilizes models.	Reduces bias & variance, improves accuracy.
Sampling	Random sampling **with replacement**.	Iterative reweighting to focus on errors.
Model Training	Independent models (parallel).	Sequential models (each corrects previous errors).
Examples	Random Forest.	AdaBoost, Gradient Boosting (XGBoost, LightGBM).
Computational Cost	Lower (parallel training).	Higher (sequential training).
Handling Noisy Data	More robust to noise.	Sensitive to noise (can overfit).