Machine Learning in Python

Ensemble Methods

Cristian A. Marocico, A. Emin Tatar

Center for Information Technology
University of Groningen

Monday, July 7th 2025

Outline

Introduction to Ensemble Methods

What are Ensemble Methods?

Definition

Introduction to Ensemble Methods

What are Ensemble Methods? Definition

Ensemble methods combine multiple models to improve predictive performance. They leverage the strengths of individual models while mitigating their weaknesses.

Introduction to Ensemble Methods

Why Use Ensemble Methods? Example

- Improved Accuracy: Ensemble methods often outperform individual models by reducing variance and bias.

Introduction to Ensemble Methods

Why Use Ensemble Methods? Example

- Improved Accuracy: Ensemble methods often outperform individual models by reducing variance and bias.
- Robustness: Less sensitive to noise and outliers, leading to more stable predictions.

Introduction to Ensemble Methods

Why Use Ensemble Methods? Example

- Improved Accuracy: Ensemble methods often outperform individual models by reducing variance and bias.
- Robustness: Less sensitive to noise and outliers, leading to more stable predictions.
- Reduced Overfitting: Reduce overfitting by averaging predictions from multiple models.

Introduction to Ensemble Methods

Why Use Ensemble Methods? — Example

- Improved Accuracy: Ensemble methods often outperform individual models by reducing variance and bias.
- Robustness: Less sensitive to noise and outliers, leading to more stable predictions.
- Reduced Overfitting: Reduce overfitting by averaging predictions from multiple models.
- Flexibility: Ensemble methods can be applied to various base models, including decision trees, linear models, and neural networks.

Introduction to Ensemble Methods

Why Use Ensemble Methods? Example

- Improved Accuracy: Ensemble methods often outperform individual models by reducing variance and bias.
- Robustness: Less sensitive to noise and outliers, leading to more stable predictions.
- Reduced Overfitting: Reduce overfitting by averaging predictions from multiple models.
- Flexibility: Ensemble methods can be applied to various base models, including decision trees, linear models, and neural networks.
- Interpretability: Some ensemble methods, like Random Forests, provide insights into feature importance.

Introduction to Ensemble Methods

Why Use Ensemble Methods? Example

- Improved Accuracy: Ensemble methods often outperform individual models by reducing variance and bias.
- Robustness: Less sensitive to noise and outliers, leading to more stable predictions.
- Reduced Overfitting: Reduce overfitting by averaging predictions from multiple models.
- Flexibility: Ensemble methods can be applied to various base models, including decision trees, linear models, and neural networks.
- Interpretability: Some ensemble methods, like Random Forests, provide insights into feature importance.
- Scalability: Many ensemble methods can be parallelized, useful for large datasets.

Introduction to Ensemble Methods

Why Use Ensemble Methods? Example

- Improved Accuracy: Ensemble methods often outperform individual models by reducing variance and bias.
- Robustness: Less sensitive to noise and outliers, leading to more stable predictions.
- Reduced Overfitting: Reduce overfitting by averaging predictions from multiple models.
- Flexibility: Ensemble methods can be applied to various base models, including decision trees, linear models, and neural networks.
- Interpretability: Some ensemble methods, like Random Forests, provide insights into feature importance.
- Scalability: Many ensemble methods can be parallelized, useful for large datasets.
- Combining Different Algorithms: Ensembles can combine predictions from different types of models, enhancing overall performance.

Types of Ensemble Methods

Ensamble Methods

Example

Types of Ensemble Methods

Ensamble Methods

Example

Ensemble methods can be broadly categorized into several types, each with its own approach to combining models:

- Bagging (Bootstrap Aggregating): Combines predictions from multiple models trained on different subsets of the training data. The most common example is Random Forests.

Types of Ensemble Methods

Ensamble Methods — Example

Ensemble methods can be broadly categorized into several types, each with its own approach to combining models:

- Bagging (Bootstrap Aggregating): Combines predictions from multiple models trained on different subsets of the training data. The most common example is Random Forests.
- Boosting: Sequentially builds models, where each new model focuses on correcting the errors made by the previous models. Examples include AdaBoost, Gradient Boosting, and XGBoost.

Types of Ensemble Methods

Ensamble Methods

Example

Ensemble methods can be broadly categorized into several types, each with its own approach to combining models:

- Bagging (Bootstrap Aggregating): Combines predictions from multiple models trained on different subsets of the training data. The most common example is Random Forests.
- Boosting: Sequentially builds models, where each new model focuses on correcting the errors made by the previous models. Examples include AdaBoost, Gradient Boosting, and XGBoost.
- Stacking: Combines multiple models by training a meta-model on their predictions. The base models can be of different types, and the meta-model learns how to best combine their outputs.

Types of Ensemble Methods

Ensamble Methods — Example

Ensemble methods can be broadly categorized into several types, each with its own approach to combining models:

- Bagging (Bootstrap Aggregating): Combines predictions from multiple models trained on different subsets of the training data. The most common example is Random Forests.
- Boosting: Sequentially builds models, where each new model focuses on correcting the errors made by the previous models. Examples include AdaBoost, Gradient Boosting, and XGBoost.
- Stacking: Combines multiple models by training a meta-model on their predictions. The base models can be of different types, and the meta-model learns how to best combine their outputs.
- Other methods include Pasting, Bayesian Model Averaging, Blending, and Voting.

Bootstrap Aggregating (Bagging) Overview

Bootstrap Aggregating (Bagging) Definition

Bootstrap Aggregating (Bagging) Overview

Bootstrap Aggregating (Bagging) Definition

Bagging, or Bootstrap Aggregating, is an ensemble method that combins the predictions of multiple models trained on different subsets of the training data. The key steps in bagging are:

Bootstrap Aggregating (Bagging) Overview

Bootstrap Aggregating (Bagging) — Definition

Bagging, or Bootstrap Aggregating, is an ensemble method that combins the predictions of multiple models trained on different subsets of the training data. The key steps in bagging are:

- Bootstrap Sampling: Create multiple bootstrap samples from the training data. Each sample is created by randomly selecting instances with replacement, meaning some instances may appear multiple times while others may not appear at all.

Bootstrap Aggregating (Bagging) Overview

Bootstrap Aggregating (Bagging) — Definition

Bagging, or Bootstrap Aggregating, is an ensemble method that combins the predictions of multiple models trained on different subsets of the training data. The key steps in bagging are:

- Bootstrap Sampling: Create multiple bootstrap samples from the training data. Each sample is created by randomly selecting instances with replacement, meaning some instances may appear multiple times while others may not appear at all.
- Model Training: Train a separate model on each bootstrap sample. These models can be of the same type (e.g., decision trees) or different types.

Bootstrap Aggregating (Bagging) Overview

Bootstrap Aggregating (Bagging) — Definition

Bagging, or Bootstrap Aggregating, is an ensemble method that combins the predictions of multiple models trained on different subsets of the training data. The key steps in bagging are:

- Bootstrap Sampling: Create multiple bootstrap samples from the training data. Each sample is created by randomly selecting instances with replacement, meaning some instances may appear multiple times while others may not appear at all.
- Model Training: Train a separate model on each bootstrap sample. These models can be of the same type (e.g., decision trees) or different types.
- Aggregation: Combine the predictions of the individual models. For regression tasks, this is typically done by averaging the predictions, while for classification tasks, a majority vote is used to determine the final class label.

Bagging and Random Forests

Bagging and Random Forests

Example

Bagging and Random Forests

Bagging and Random Forests

Example

Random Forests is a popular ensemble method that builds multiple decision trees using bagging. It introduces additional randomness by selecting a random subset of features for each tree, which helps to decorrelate the trees and improve overall model performance.

Feature Importance in Random Forests

Feature Importance in Random Forests

Feature Importance in Random Forests

Feature Importance in Random Forests

Random Forests can also provide insights into feature importance, which helps in understanding which features contribute most to the predictions. This is done by measuring the decrease in model performance when a feature is permuted or removed.

Boosting Overview

Boosting Overview

Definition

Boosting Overview

Boosting Overview

Definition

Boosting is an ensemble method that builds models sequentially, where each new model focuses on correcting the errors made by the previous models. The key steps in boosting are:

Boosting Overview

Boosting Overview — Definition

Boosting is an ensemble method that builds models sequentially, where each new model focuses on correcting the errors made by the previous models. The key steps in boosting are:

- Sequential Model Training: Train a series of models, where each model is trained on the residuals (errors) of the previous model. This allows each new model to learn from the mistakes of its predecessor.

Boosting Overview

Boosting Overview — Definition

Boosting is an ensemble method that builds models sequentially, where each new model focuses on correcting the errors made by the previous models. The key steps in boosting are:

- Sequential Model Training: Train a series of models, where each model is trained on the residuals (errors) of the previous model. This allows each new model to learn from the mistakes of its predecessor.
- Weighted Predictions: Each model's predictions are weighted based on its performance. Models that perform well have a higher weight in the final prediction.

Boosting Overview

Boosting Overview

Definition

Boosting is an ensemble method that builds models sequentially, where each new model focuses on correcting the errors made by the previous models. The key steps in boosting are:

- Sequential Model Training: Train a series of models, where each model is trained on the residuals (errors) of the previous model. This allows each new model to learn from the mistakes of its predecessor.
- Weighted Predictions: Each model's predictions are weighted based on its performance. Models that perform well have a higher weight in the final prediction.
- Final Prediction: Combine the predictions of all models, typically by summing their weighted predictions. For classification tasks, a threshold is applied to determine the final class label.

Boosting Algorithms

Boosting Algorithms

Example

Boosting Algorithms

Boosting Algorithms — Example

Some popular boosting algorithms include:

- AdaBoost: Assigns weights to instances based on their classification errors, focusing more on misclassified instances in subsequent iterations.

Boosting Algorithms

Boosting Algorithms — Example

Some popular boosting algorithms include:

- AdaBoost: Assigns weights to instances based on their classification errors, focusing more on misclassified instances in subsequent iterations.
- Gradient Boosting: Builds models that minimize a loss function by fitting to the residuals of the previous model. It can handle various loss functions, making it versatile for regression and classification tasks.

Boosting Algorithms

Boosting Algorithms

Example

Some popular boosting algorithms include:

- AdaBoost: Assigns weights to instances based on their classification errors, focusing more on misclassified instances in subsequent iterations.
- Gradient Boosting: Builds models that minimize a loss function by fitting to the residuals of the previous model. It can handle various loss functions, making it versatile for regression and classification tasks.
- XGBoost: An optimized implementation of gradient boosting that includes regularization, handling missing values, and parallel processing for faster training.

Ensembles of Ensemble Methods

Ensembles of Ensemble Methods Example

Ensembles of Ensemble Methods

Ensembles of Ensemble Methods

Example

Ensembles of ensemble methods combine multiple ensemble techniques to further enhance predictive performance. This approach can lead to even more robust models by leveraging the strengths of different ensemble methods.

Ensembles of Ensemble Methods

Ensembles of Ensemble Methods

Ensembles of Ensemble Methods

Ensembles of Ensemble Methods

Ensembles of ensemble methods can be created by combining different types of ensembles, such as bagging and boosting, or by stacking multiple ensemble models. This approach allows for greater diversity in the predictions and can lead to improved generalization performance.