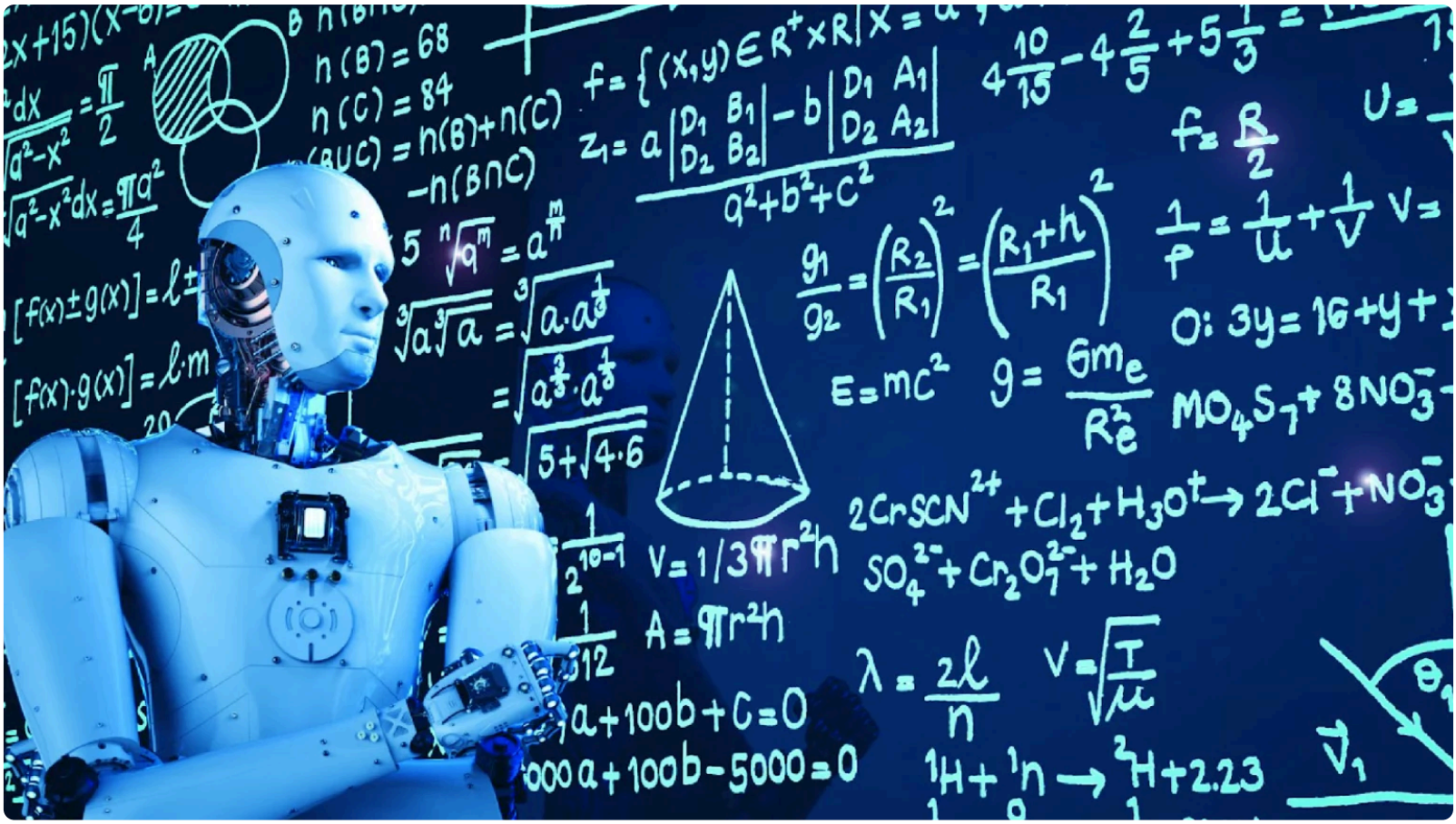
Bagging, Boosting and Stacking: Ensemble Learning in ML Models

Mbali Kalirane
Last Updated : 03 Jan, 2025

13 min read

231

Machine learning is great! But there's one thing that makes it even better: ensemble learning. Ensemble learning in machine learning helps enhance the performance of machine learning models. The concept behind it is simple. Multiple machine learning models are combined to obtain a more accurate model. Stacking, bagging, and boosting are the three most popular ensemble learning techniques. Each of these techniques offers a unique approach to improving predictive accuracy. Each technique is used for a different purpose, with the use of each depending on varying factors. Although each technique differs, many of us struggle to distinguish between them. Knowing when or why we should use each technique is difficult.



In this tutorial, I'll explain the difference between bagging, boosting, and stacking. I'll explain their purposes and processes, as well as their advantages and disadvantages. By the end of this article, you will understand Ensemble learning in Machine Learning, how each technique works, and which technique to use and when. Also, you will get a

deep learning all are cover in this tutorial.

In this article, you will learn about the Augmented Dickey-Fuller (ADF) test, a statistical method used to determine if a time series has a unit root. The ADF test, also called the adfuller test, helps assess the stationarity of a time series.

Learning Objectives:

- Understand the concept of ensemble learning and its purpose of combining multiple models to improve accuracy and performance.
- Learn about the three main ensemble techniques: bagging, boosting, and stacking.
- Understand the differences in the working principles and applications of bagging, boosting, and stacking.
- Know when to apply bagging, boosting or stacking based on the specific requirements and characteristics of the machine learning problem.

This article was published as a part of the Data Science Blogathon.

Table of contents

What is Ensemble Learning in Machine Learning?

Ensemble learning in **machine learning** combines multiple individual models to create a stronger, more accurate predictive model. By leveraging the diverse strengths of different models, ensemble learning aims to mitigate errors,

What is Bagging?

Bagging (Bootstrap Aggregating) is an ensemble learning technique designed to improve the accuracy and stability of machine learning algorithms. It involves the following steps:

1. **Data Sampling**: Creating multiple subsets of the training dataset using bootstrap sampling (random sampling with replacement).
2. **Model Training**: raining a separate model on each subset of the data.
3. **Aggregation**: Combining the predictions from all individual models (averaged for regression or majority voting for classification) to produce the final output.

Key Benefits:

- **Reduces Variance**: By averaging multiple predictions, bagging reduces the variance of the model and helps prevent overfitting.
- **Improves Accuracy**: Combining multiple models usually leads to better performance than individual models.

Example of Bagging Algorithms:

- **Random Forests** (an extension of bagging applied to decision trees)

***Also Read:** Beginner's Guide to Ensemble Learning in Python*

What is Boosting?

Boosting is another ensemble learning technique that focuses on creating a strong model by combining several weak models. It involves the following steps:

1. **Sequential Training**: Training models sequentially, each one trying to correct the errors made by the previous models.
2. **Weight Adjustment**: Each instance in the training set is weighted. Initially, all instances have equal weights. After each model is trained, the weights of misclassified instances are increased so that the next model focuses more on difficult cases.
3. **Model Combination**: Combining the predictions from all models to produce the final output, typically by weighted voting or weighted averaging.

Key Benefits:

- **Produces Strong Predictors**: Combining weak learners leads to a strong **predictive model**.

Example of Boosting Algorithms:

- **AdaBoost**
- **Gradient Boosting Machines (GBM)**
- **XGBoost**
- **LightGBM**

Similarities between Bagging and Boosting

Bagging (Bootstrap Aggregating) and Boosting are both ensemble learning techniques designed to improve the performance of machine learning models by combining the predictions of multiple base models.

Both bagging and boosting in machine learning involve training multiple models on different subsets of the training data and then combining their predictions to make a final prediction. These techniques aim to reduce the variance of the model and improve its overall accuracy and stability.

Additionally, using bagging and boosting in machine learning with various base models, such as decision trees, to create a diverse set of models that capture different aspects of the data.

Differences between Bagging and Boosting

While bagging and boosting share some similarities, their approach and methodology differ.

Bagging trains each base model independently and in parallel, using bootstrap sampling to create multiple subsets of the training data. The final prediction is then made by averaging the predictions of all base models. Bagging focuses on reducing variance and overfitting by creating diverse models.

In contrast, boosting trains models sequentially, with each subsequent model focusing on correcting the errors made by the previous ones. Boosting adjusts the weights of training instances to prioritize difficult-to-classify instances, thus reducing bias and improving predictive accuracy. The final prediction combines the outputs of all models, usually through weighted voting or averaging.

Additionally, while bagging is relatively simple and easy to parallelize, boosting is more complex due to its sequential nature and may be more prone to overfitting if not properly controlled.

Stacking (Stacked Generalization) is an ensemble learning technique that aims to combine multiple models to improve predictive performance. It involves the following steps:

1. **Base Models**: Training multiple models (level-0 models) on the same dataset.
2. **Meta-Model**: Training a new model (level-1 or meta-model) to combine the predictions of the base models. Using the predictions of the base models as input features for the meta-model.

Key Benefits:

- **Leverages Model Diversity**: By combining different types of models, stacking can capture a wide range of patterns in the data.
- **Improves Performance**: The meta-model learns how to best combine the predictions from the base models, often leading to improved performance over individual models.

Example Process:

- Train several base models (e.g., **decision trees**, **neural networks**, **SVMs**) on the training data.
- Use the predictions of these base models to create a new dataset.
- Train a meta-model (e.g., **linear regression**, **logistic regression**) on this new dataset to make the final predictions.

How Did Ensemble Learning Come into Existence?

One of the first uses of ensemble methods in machine learning was the bagging technique. This technique was developed to overcome instability in decision trees. An example of the bagging technique is the **random forest algorithm**. The random forest is an ensemble of multiple decision trees. Decision trees tend to be prone to overfitting. Because of this, a single decision tree doesn't provide reliable predictions. To improve the prediction accuracy of decision trees, bagging is employed to form a random forest. The resulting random forest has a lower variance compared to the individual trees.

The success of bagging led to developing other ensemble techniques such as boosting, stacking, and many others. Today, these developments are an important part of machine learning.

The many real-life machine learning applications show these ensemble methods in machine learning' importance. These applications include many critical systems. These include decision-making systems, spam detection, autonomous vehicles, medical diagnosis, etc. These systems are crucial because they can impact human lives and business revenues. Therefore, ensuring the accuracy of machine learning models is paramount. An inaccurate model can lead to disastrous consequences for many businesses or organizations. At worst, they can lead to the endangerment of human lives.

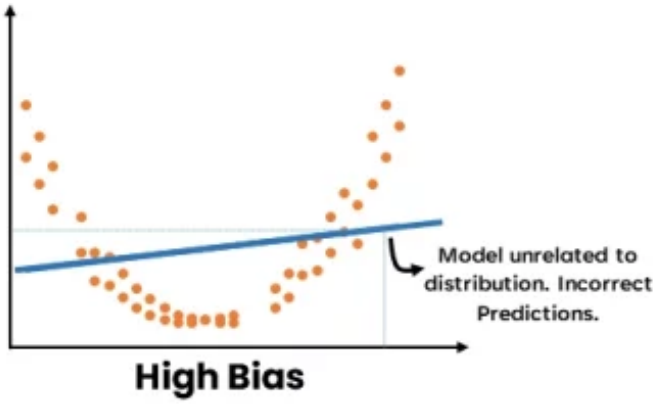
Ensemble learning is a learning method that consists of combining multiple machine learning models.

A problem in machine learning is that individual models tend to perform poorly. In other words, they tend to have low prediction accuracy. To mitigate this problem, we combine multiple models to get one with a better performance.

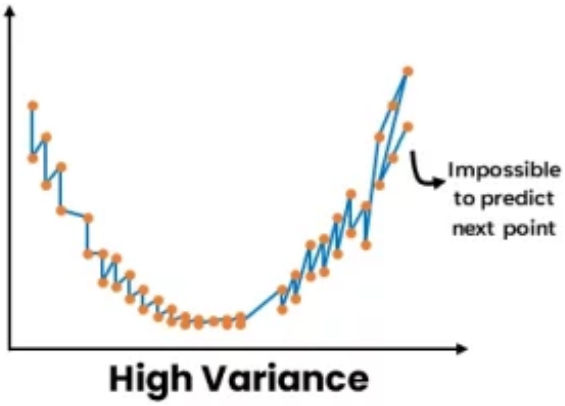
The individual models that we combine are known as weak learners. We call them weak learners because they either have a high bias or high variance. Because they either have high bias or variance, weak learners cannot learn efficiently and perform poorly.

High-bias and High-variance Models

- A high-bias model results from not learning data well enough. It is not related to the distribution of the data. Hence, future predictions will be unrelated to the data and thus incorrect.



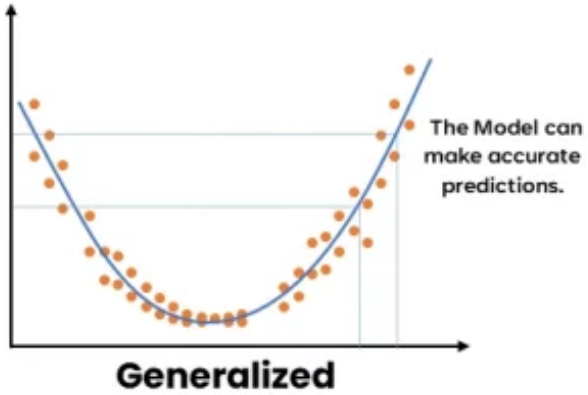
- A high-variance model results from learning the data too well. It varies with each data point, making it impossible to predict the next point accurately.



Thus, both high bias and high variance models cannot be generalized properly. Thus, weak learners will make incorrect generalizations or fail to generalize altogether. Because of this, the predictions of weak learners cannot be relied on by themselves.

balance, both the bias and variance need to be low. Ensemble learning tries to balance this bias-variance trade-off by reducing either the bias or the variance.

Ensemble learning aims to reduce the bias if we have a weak model with high bias and low variance. This way, the resulting model will be much more balanced, with low bias and variance. Thus, the resulting model will be known as a strong learner. This model will be more generalized than the weak learners. It will thus be able to make accurate predictions.



Monitoring Ensemble Learning Models

We use bagging to combine weak learners of high variance. Bagging aims to produce a model with lower variance than the individual weak models. These weak learners are homogenous, meaning they are of the same type.

Bagging is also known as Bootstrap aggregating. It consists of two steps: bootstrapping and aggregation.

Bootstrapping

Involves resampling subsets of data with replacement from an initial dataset. In other words, the initial dataset provides subsets of data. Creating these subsets, bootstrapped datasets or simply bootstraps, by resampling 'with replacement,' which means an individual data point can be sampled multiple times. Each bootstrap dataset trains a weak learner.

Aggregating

Individual weak learners train independently from each other. Each learner makes independent predictions. The system aggregates the results of those predictions to get the overall prediction. The predictions are aggregated using either max voting or averaging.

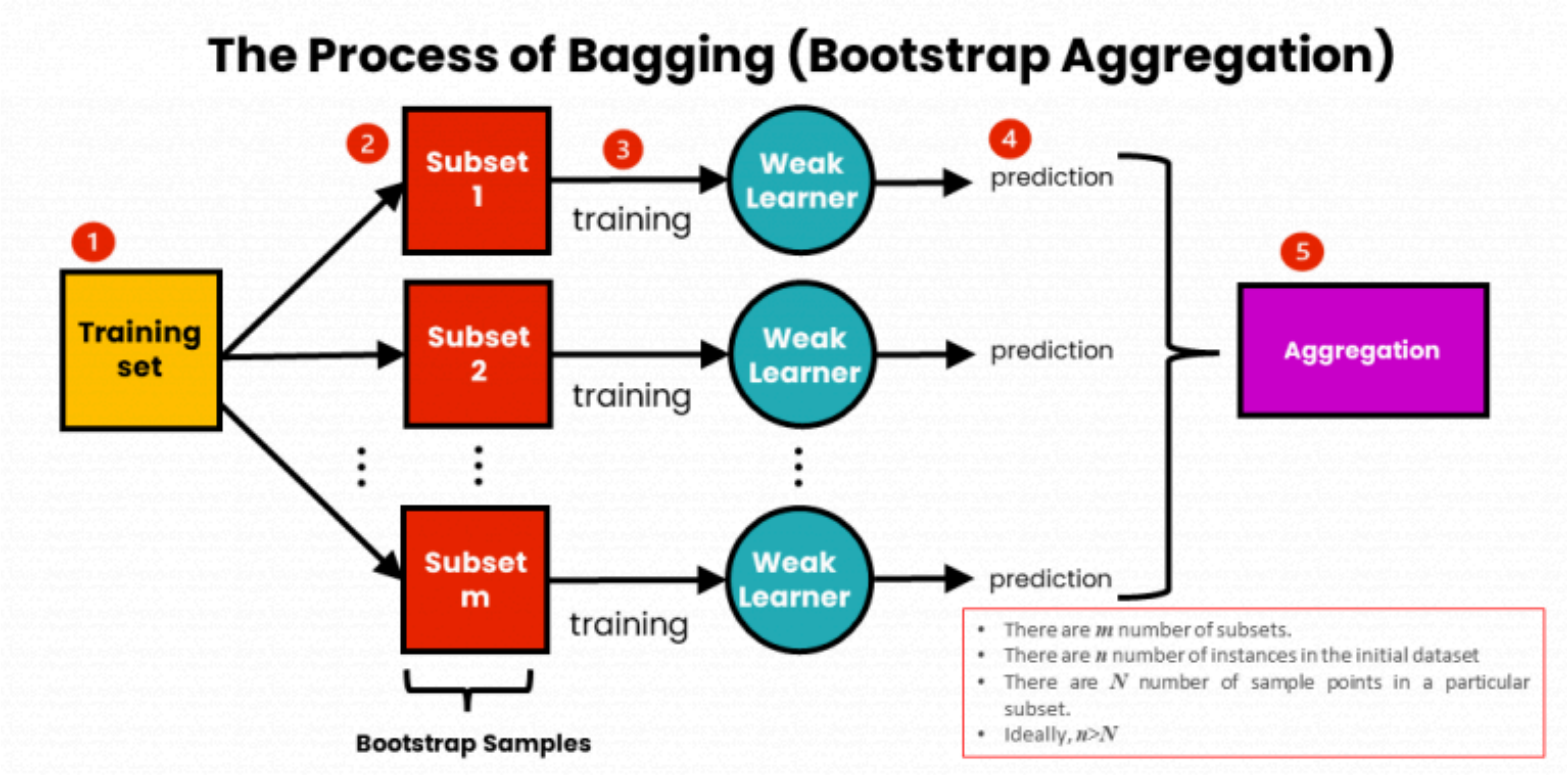
Max Voting

a prediction from each model counts as a single 'vote.' The most occurring 'vote' is chosen as the representative for the combined model.

Averaging

Using it generally for regression problems. It involves taking the average of the predictions. The resulting average is used as the overall prediction for the combined model.

Steps of Bagging



The steps of bagging are as follows:

1. We have an initial training dataset containing n-number of instances.
2. We create a m-number of subsets of data from the training set. We take a subset of N sample points from the initial dataset for each subset. Each subset is taken with replacement. This means that a specific data point can be sampled more than once.
3. For each subset of data, we train the corresponding weak learners independently. These models are homogeneous, meaning that they are of the same type.
4. Each model makes a prediction.
5. Aggregating the predictions into a single prediction. For this, using either max voting or averaging.

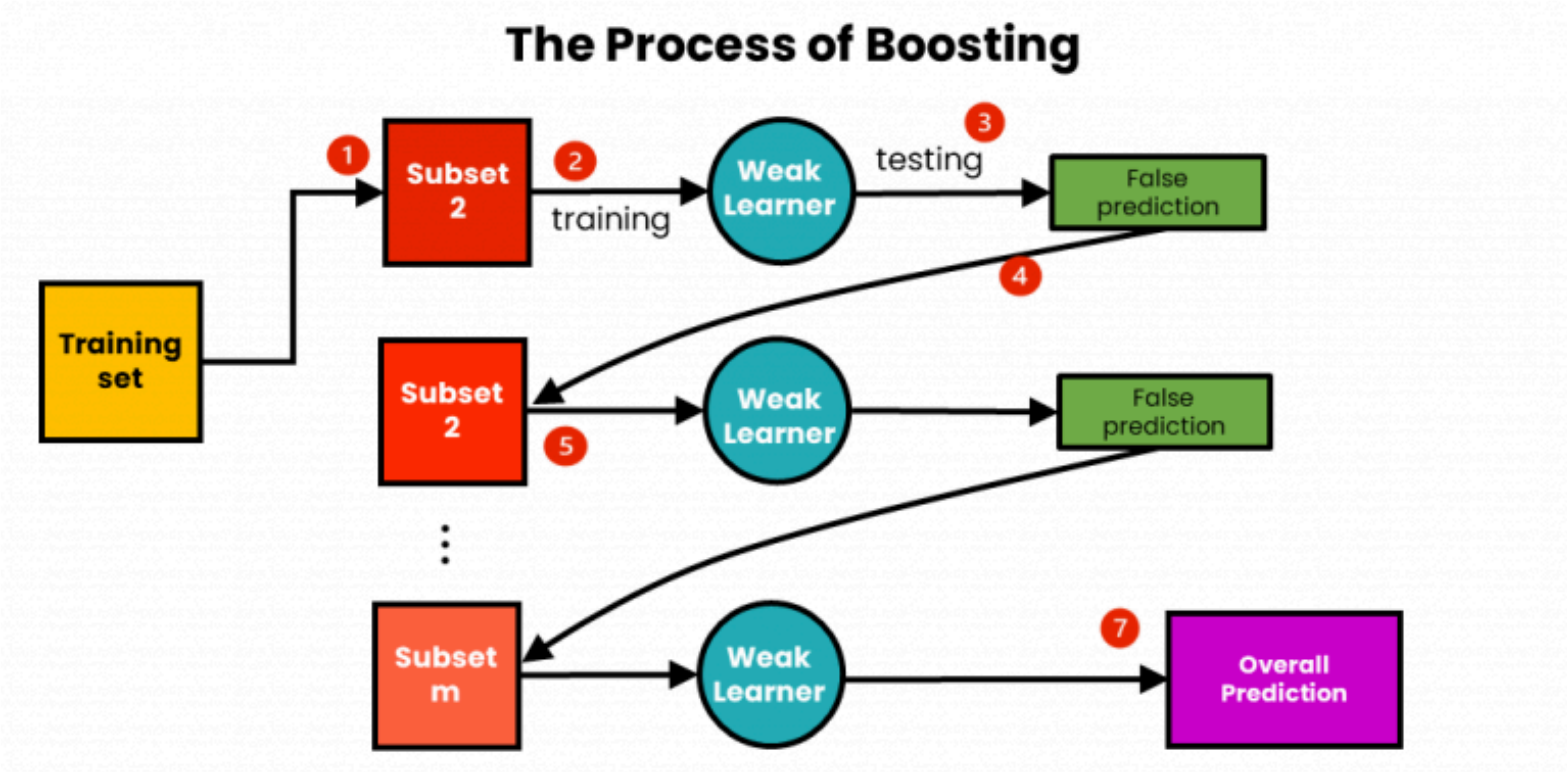
We use boosting to combine weak learners with high bias. Boosting aims to produce a model with a lower bias than the individual models. Like in bagging, the weak learners are homogeneous.

Boosting involves sequentially training weak learners. Here, each subsequent learner improves the errors of previous learners in the sequence. A sample of data is first taken from the initial dataset. Using this sample to train the first model, and the model makes its prediction. The samples can either be correctly or incorrectly predicted. The samples that are wrongly predicted are reused for training the next model. In this way, subsequent models can improve on the errors of previous models.

Unlike bagging, which aggregates prediction results at the end, boosting aggregates the results at each step.

Weighted averaging involves giving all models different weights depending on their predictive power. In other words, it gives more weight to the model with the highest predictive power. This is because the learner with the highest predictive power is considered the most important.

Steps of Boosting



Boosting works with the following steps:

- We sample m-number of subsets from an initial training dataset.
- Using the first subset, we train the first weak learner.

- Each data point with the wrong prediction is sent into the second subset of data, and this subset is updated.
- Using this updated subset, we train and test the second weak learner.
- We continue with the next subset until reaching the total number of subsets.
- We now have the total prediction. The overall prediction has already been aggregated at each step, so there is no need to calculate it.

Improving Model Accuracy with Stacking

We use stacking to improve the prediction accuracy of strong learners. Stacking aims to create a single robust model from multiple heterogeneous strong learners.

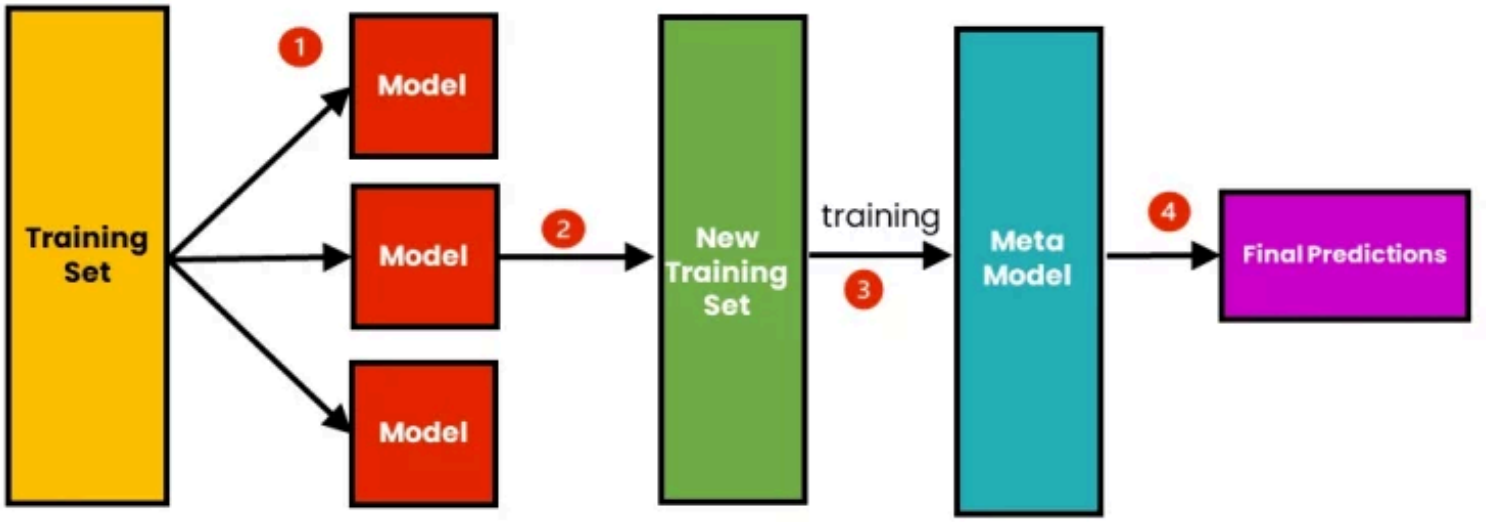
Stacking differs from bagging and boosting in machine learning in that:

- It combines strong learners
- It combines heterogeneous models
- It consists of creating a Metamodel.

Individual heterogeneous models are trained using an initial dataset. These models make predictions and form a single new dataset using those predictions. Using this new data set to train the metamodel, which makes the final prediction. Combining the prediction using weighted averaging.

Because stacking combines strong learners, it can combine bagged or boosted models.

Steps of Stacking



The steps of Stacking are as follows:

- We use initial training data to train m-number of algorithms.
- Using the output of each algorithm, we create a new training set.
- Using the new training set, we create a meta-model algorithm.
- Using the results of the meta-model, we make the final prediction. Combining the result using weighted averaging.

Implementation of Using Bagging, Boosting, and Stacking in Python

Sure! Let's explore how to implement ensemble models like bagging, boosting, and stacking in Python using popular libraries like scikit-learn, XGBoost, and others.

1. Bagging

We'll use the Random Forest classifier, which is a common implementation of bagging, using scikit-learn.


```
# Import necessary libraries
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Load dataset
iris = load_iris()
```

```
# Split dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Initialize and train the RandomForest classifier
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)
rf_clf.fit(X_train, y_train)
# Make predictions
y_pred = rf_clf.predict(X_test)
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Random Forest Classifier Accuracy: {accuracy:.2f}")
```

2. Boosting

We'll use XGBoost, a powerful boosting algorithm, to demonstrate boosting.

Copy Code

```
# Import necessary libraries
import xgboost as xgb
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Load dataset
iris = load_iris()
X = iris.data
y = iris.target
# Split dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Initialize and train the XGBoost classifier
xgb_clf = xgb.XGBClassifier(n_estimators=100, learning_rate=0.1, random_state=42)
xgb_clf.fit(X_train, y_train)
# Make predictions
y_pred = xgb_clf.predict(X_test)
# Evaluate the model
```

```
print(f"XGBoost Classifier Accuracy: {accuracy:.2f}")
```

3. Stacking

We'll use the Stacking Classifier from scikit-learn to implement stacking.

Copy Code

```
# Import necessary libraries
from sklearn.ensemble import StackingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Load dataset
iris = load_iris()
X = iris.data
y = iris.target
# Split dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Define base models
base_models = [
('rf', RandomForestClassifier(n_estimators=100, random_state=42)),
('svm', SVC(probability=True, random_state=42))
]
# Define meta-model
meta_model = LogisticRegression()
# Initialize and train the StackingClassifier
stacking_clf = StackingClassifier(estimators=base_models, final_estimator=meta_model)
stacking_clf.fit(X_train, y_train)
# Make predictions
y_pred = stacking_clf.predict(X_test)
```

```
print(f"Stacking Classifier Accuracy: {accuracy:.2f}")
```

When to use Bagging vs Boosting vs Stacking?

	Bagging	Boosting	Stacking
Purpose	Reduce Variance	Reduce Bias	Improve Accuracy
Base Learner Types	Homogeneous	Homogeneous	Heterogeneous
Base Learner Training	Parallel	Sequential	Meta Model
Aggregation	Max Voting, Averaging	Weighted Averaging	Weighted Averaging

You can use bagging to reduce your model's overfitting or variance, boosting to reduce underfitting or bias, or stacking to increase predictive accuracy.

Bagging and boosting both work with homogeneous weak learners. Stacking works using heterogeneous solid learners.

All three of these **methods** can work with either classification or regression problems.

One disadvantage of boosting is that it is prone to variance or overfitting. It is thus not advisable to use boosting to reduce variance. Boosting will do a worse job of reducing variance as compared to bagging.

On the other hand, the converse is true. It is not advisable to use bagging to reduce bias or underfitting. This is because bagging is more prone to bias and does not help reduce bias.

Stacked models have the advantage of better prediction accuracy than bagging or boosting. However, because they combine bagged or boosted models, they have the disadvantage of needing much more time and computational power. If you are looking for faster results, it's advisable not to use stacking. However, stacking is the way to go if you're looking for high accuracy.

Conclusion

In conclusion, understanding and effectively applying ensemble learning techniques like bagging, boosting, and stacking is crucial for enhancing the performance and robustness of machine learning models. By mastering these techniques, data scientists and machine learning practitioners can significantly improve the accuracy and reliability of their models, making ensemble learning an indispensable tool in data science. Whether tackling classification or regression problems, strategically bagging, boosting and stacking can lead to more robust and accurate predictive models, ultimately driving better decision-making and insights across various applications.

Hope you like the article! Bagging and boosting are essential ensemble techniques in machine learning. While bagging reduces variance by averaging multiple models, boosting sequentially improves predictions by focusing on errors. Bagging, boosting, and stacking enhance model performance through diverse approaches, highlighting the differences in bagging vs boosting methodologies. Understanding bagging boosting and stacking is crucial for effective model optimization in machine learning.

Elevate your machine learning skills with our 'Mastering Ensemble Learning Techniques' course! Dive deep into bagging, boosting, and stacking to enhance your model performance and accuracy—enroll today and become an expert in building robust predictive models!

If you want to know more about machine learning and AI concepts then enroll in our blackbelt plus program!

Key Takeaways

- Ensemble learning combines multiple machine learning models into a single model. The aim is to increase the performance of the model.
- Bagging aims to decrease variance, boosting aims to decrease bias, and stacking aims to improve prediction accuracy.
- Bagging and boosting combine homogenous weak learners. Stacking combines heterogeneous solid learners.
- Bagging trains models in parallel and boosting trains the models sequentially. Stacking creates a meta-model.

The media shown in this article is not owned by Analytics Vidhya and is used at the Author's discretion.

Mbali Kalirane

Algorithm | Beginner | Machine Learning

Free Courses

Generative AI - A Way of Life

Explore Generative AI for beginners: create text and images, use top AI tools, learn practical skills, and ethics.

Getting Started with Large Language Models

Master Large Language Models (LLMs) with this course, offering clear guidance in NLP and model training made simple.

Building LLM Applications using Prompt Engineering

This free course guides you on building LLM apps, mastering prompt engineering, and developing chatbots with enterprise data.

Improving Real World RAG Systems: Key Challenges & Practical Solutions

Explore practical solutions, advanced retrieval strategies, and agentic RAG systems to improve context, relevance, and accuracy in AI-driven applications.

Responses From Readers

What are your thoughts?...

Submit reply

Frequently Asked Questions

What is bagging and boosting?

A. Bagging and boosting are ensemble learning techniques. Bagging (Bootstrap Aggregating) reduces variance by averaging multiple

What is bagging strategy?

What is boosting in ML?

What is the concept of bagging?

Ultimate Guide To Boosting Algorithms

Stacking Algorithms in Machine Learning

Know About Ensemble Methods in Machine Learning

5 Easy questions on Ensemble Modeling everyone ...

Interview Questions on Bagging Algorithms in Ma...

Ensemble Stacking for Machine Learning and Deep...

Make This Article **Fun and Interactive** with Flash Cards and Quizzes!

Generative AI | Large Language Models | Building LLM Applications using Prompt Engineering | Building Your first RAG System using LlamaIndex | Stability.AI | MidJourney | Building Production Ready RAG systems using LlamaIndex | Building LLMs for Code | Deep Learning | Python | Microsoft Excel | Machine Learning | Decision Trees | Pandas for Data Analysis | Ensemble Learning | NLP | NLP using Deep Learning | Neural Networks | Loan Prediction Practice Problem | Time Series Forecasting | Tableau | Business Analytics

Popular Categories

Generative AI | Prompt Engineering | Generative AI Application | News | Technical Guides | AI Tools | Interview Preparation | Research Papers | Success Stories | Quiz | Use Cases | Listicles

Generative AI Tools and Techniques

GANs | VAEs | Transformers | StyleGAN | Pix2Pix | Autoencoders | GPT | BERT | Word2Vec | LSTM | Attention Mechanisms | Diffusion Models | LLMs | SLMs | StyleGAN | Encoder Decoder Models | Prompt Engineering | LangChain | LlamaIndex | RAG | Fine-tuning | LangChain AI Agent | Multimodal Models | RNNs | DCGAN | ProGAN | Text-to-Image Models | DDPM | Document Question Answering | Imagen | T5 (Text-to-Text Transfer Transformer) | Seq2seq Models | WaveNet | Attention Is All You Need (Transformer Architecture)

Popular GenAI Models

Llama 3.1 | Llama 3 | Llama 2 | GPT 4o Mini | GPT 4o | GPT 3 | Claude 3 Haiku | Claude 3.5 Sonnet | Phi 3.5 | Phi 3 | Mistral Large 2 | Mistral NeMo | Mistral-7b | Gemini 1.5 Pro | Gemini Flash 1.5 | Bedrock | Vertex AI | DALL.E | Midjourney | Stable Diffusion

Data Science Tools and Techniques

Python | R | SQL | Jupyter Notebooks | TensorFlow | Scikit-learn | PyTorch | Tableau | Apache Spark | Matplotlib | Seaborn | Pandas | Hadoop | Docker | Git | Keras | Apache Kafka | AWS | NLP | Random Forest | Computer Vision | Data Visualization | Data Exploration | Big Data | Common Machine Learning Algorithms | Machine Learning