Ensemble Learning Methods: Bagging, Boosting and Stacking

What is Ensemble Learning?

Ensemble learning is a machine learning technique combining 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, enhance performance, and increase the overall robustness of predictions, leading to improved results across various tasks in machine learning and data analysis.

How Did Ensemble Learning Come into Existence?

One of the first uses of ensemble methods was the bagging technique. This technique was developed to overcome instability in decision trees. In fact, 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 can't be relied on for making 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 the development of 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' importance. These applications include many critical systems. These include decision-

making systems, spam detection, autonomous vehicles, medical diagnosis, and many others. These systems are crucial because they have the ability to 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.

How Ensemble Learning Works?

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.
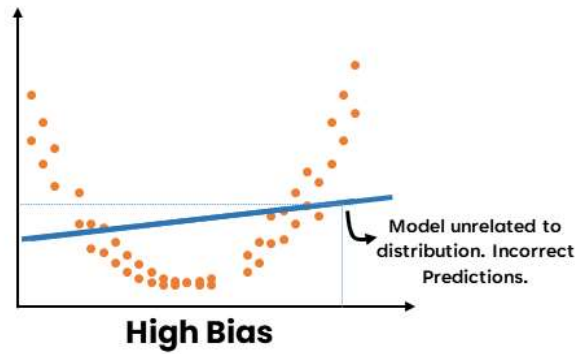
High Bias

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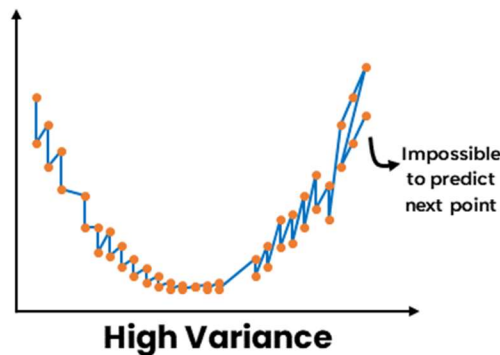

High Variance

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

As we know from the bias-variance trade-off, an underfit model has high bias and low variance, whereas an overfit model has high variance and low bias. In either case, there is no balance between bias and variance. For there to be a 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.

It aims to reduce the bias if we have a weak model with high bias and low variance. Ensemble learning will aim to reduce the variance if we have a weak model with high

variance and low bias. 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.
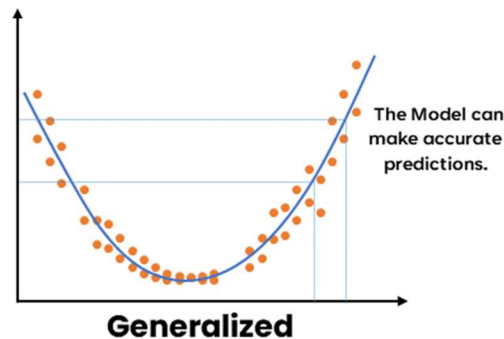
Monitoring Ensemble Learning Models

Ensemble learning improves a model's performance in mainly three ways:

- By reducing the variance of weak learners
- By reducing the bias of weak learners,
- By improving the overall accuracy of strong learners.

Bagging is used to reduce the variance of weak learners. Boosting is used to reduce the bias of weak learners. Stacking is used to improve the overall accuracy of strong learners.

Reducing Variance with Bagging

We use bagging for combining 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, subsets of data are taken from the initial dataset. These subsets of data are called bootstrapped datasets or, simply, bootstraps. Resampled 'with replacement' means an individual data point can be sampled multiple times. Each bootstrap dataset is used to train a weak learner.

Aggregating

The individual weak learners are trained independently from each other. Each learner makes independent predictions. The results of those predictions are aggregated at the end to get the overall prediction. The predictions are aggregated using either max voting or averaging.
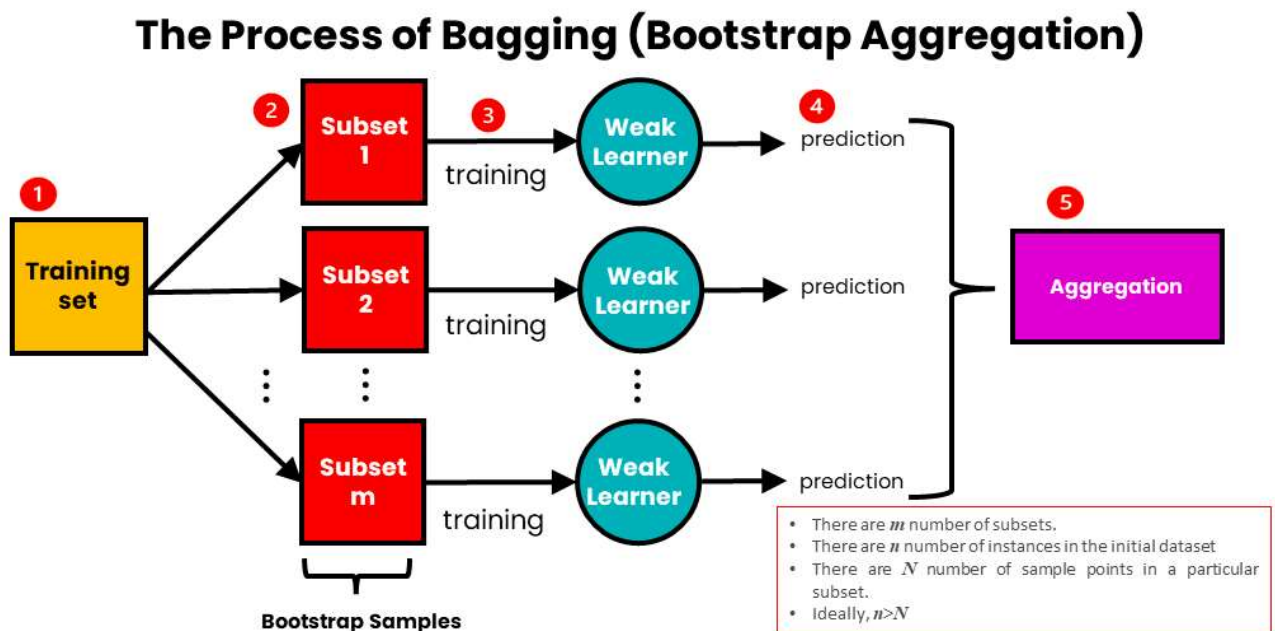
Max Voting

It is a commonly used for classification problems that consists of taking the mode of the predictions (the most occurring prediction). It is called voting because like in election voting, the premise is that 'the majority rules'. Each model makes a prediction. A prediction from each model counts as a single 'vote'. The most occurring 'vote' is chosen as the representative for the combined model.

Averaging

It is generally used 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 Process of Bagging (Bootstrap Aggregation)

- There are m number of subsets.
- There are n number of instances in the initial dataset
- There are N number of sample points in a particular subset.
- Ideally, $n > N$

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. The predictions are aggregated into a single prediction. For this, either max voting or averaging is used.

Reducing Bias by Boosting

We use boosting for combining weak learners with high bias. Boosting aims to produce a model with a lower bias than that of 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. This sample is used 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. They are aggregated using weighted averaging.

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

The Process of Boosting
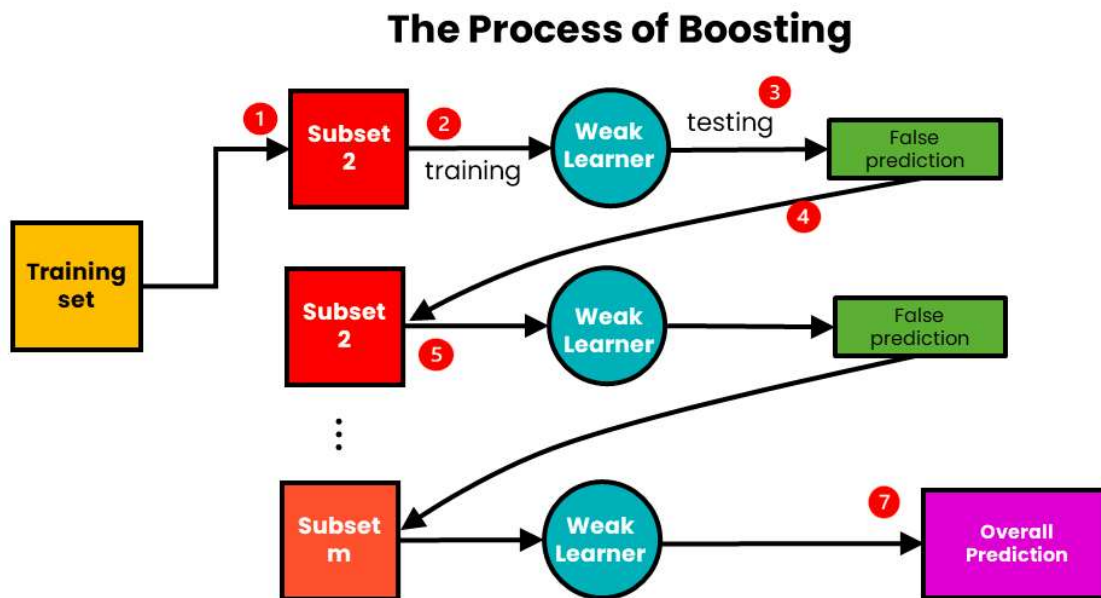
Boosting works with the following steps:

1. We sample m-number of subsets from an initial training dataset.
2. Using the first subset, we train the first weak learner.
3. We test the trained weak learner using the training data. As a result of the testing, some data points will be incorrectly predicted.
4. Each data point with the wrong prediction is sent into the second subset of data, and this subset is updated.
5. Using this updated subset, we train and test the second weak learner.

6. We continue with the following subset until the total number of subsets is reached.

7. 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 that:

- It combines strong learners
- It combines heterogeneous models
- It consists of creating a Metamodel. A metamodel is a model created using a new dataset.

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

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

Steps of Stacking

The Process of Stacking
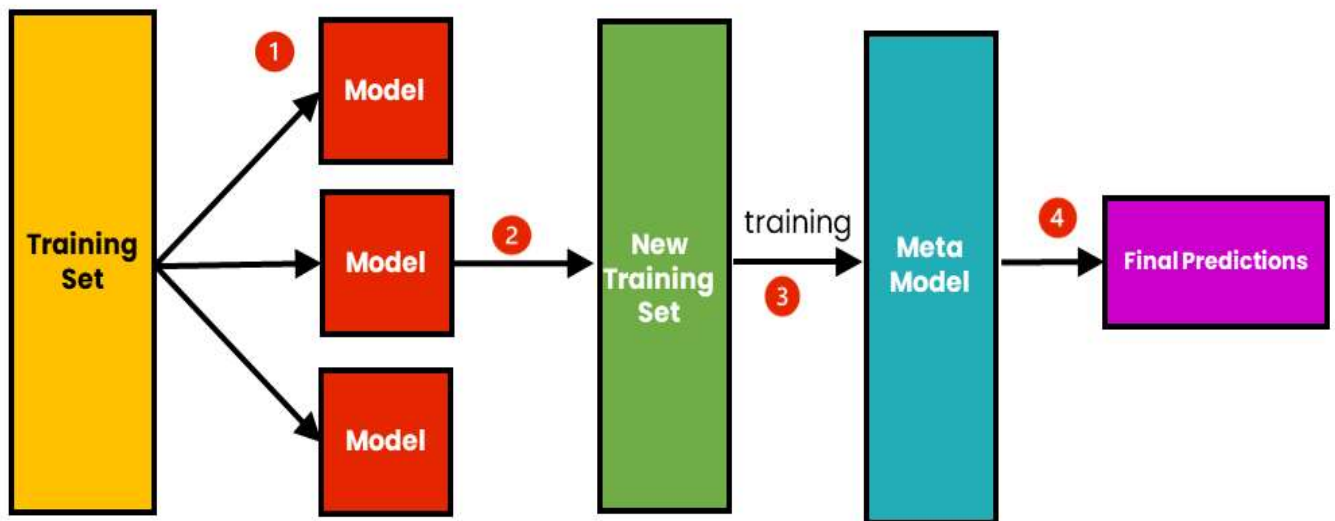
The steps of Stacking are as follows:

1. We use initial training data to train m-number of algorithms.
2. Using the output of each algorithm, we create a new training set.
3. Using the new training set, we create a meta-model algorithm.
4. Using the results of the meta-model, we make the final prediction. The results are combined using weighted averaging.

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

If you want to reduce the overfitting or variance of your model, you use bagging and if you are looking to reduce underfitting or bias, you use boosting. However, if you want to increase predictive accuracy, use stacking.

Bagging and boosting both works 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 for reducing variance. Boosting will do a worse job in 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. But 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

Bagging, boosting and stacking are important for ensuring the accuracy of models. They can help prevent undesirable consequences caused by inaccurate models. Below are some of the key takeaways from the article:

- 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.