Published in Geek Culture

Aman Gupta Follow

Apr 26, 2021 · 5 min read · Listen

XGBoost versus Random Forest

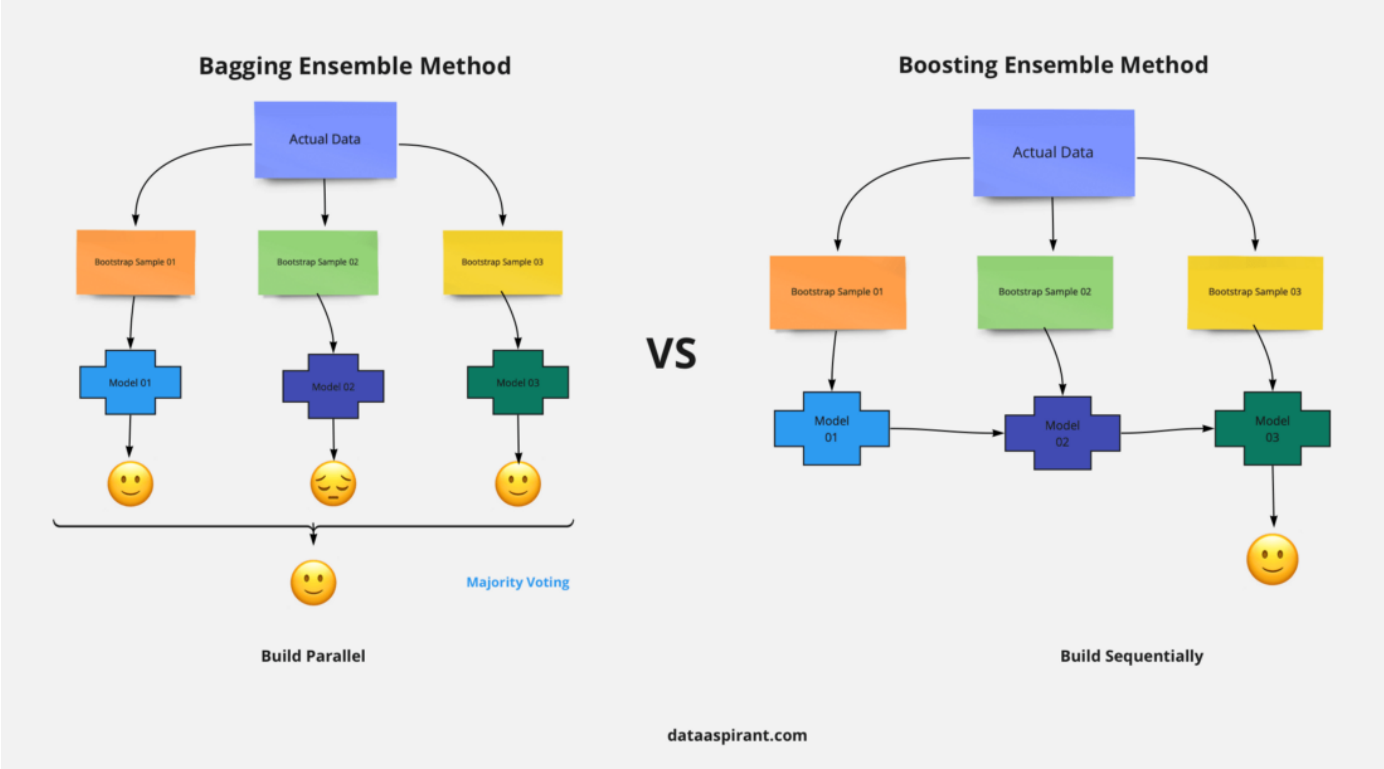


Image Credits: https://dataaspirant.com/

I was recently working on a Market Mix Model, wherein I had to predict sales from impressions. While working on an aspect of it I was confronted with the problem of choosing between a Random Forest and a XG Boost. This led to the inception of this article.

is to minimize the loss function of the model by adding weak learners using gradient descent. Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. As gradient boosting is based on minimizing a loss function, different types of loss functions can be used resulting in a flexible technique that can be applied to regression, multi-class classification, etc. Gradient boosting does not modify the sample distribution as weak learners train on the remaining residual errors of a strong learner (i.e., pseudo-residuals). By training on the residuals of the model, it gives more importance to misclassified observations. Intuitively, new weak learners are added to concentrate on the areas where the existing learners are performing poorly. The contribution of each weak learner to the final prediction is based on a gradient optimization process to minimize the overall error of the strong learner.

Random Forest is a bagging technique that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.
A Random Forest has two random elements —
1. Random subset of features.
2. Bootstrap Samples of data.

Comparing the Contenders




Image Credits: https://www.everypixel.com/

Boosting happens to be iterative learning which means the model will predict something initially and self analyses its mistakes as a predictive toiler and give more weightage to the data points in which it made a wrong prediction in the next iteration. After the second iteration, it again self analyses its wrong predictions and gives more weightage to the data points which are predicted as wrong in the next iteration. This process continues as a cycle. Hence technically, if a prediction has been done, there is an at most surety that it did not happen as a random chance but with a thorough understanding and patterns in the data. Such a model that prevents the occurrences of predictions with a random chance is trustable most of the time.

mode of this collection as the prediction of this forest depending upon the nature of data (either continues or categorical). At a high level, this seems to be fine but there are high chances that most of the trees could have made predictions with some random chances since each of the trees had their own circumstances like class imbalance, sample duplication, overfitting, inappropriate node splitting, etc.

Let us now score these two algorithms based on the below arguments.

XGBoost (1) & Random Forest (0):

XGBoost straight away prunes the tree with a score called "Similarity score" before entering into the actual modeling purposes. It considers the "Gain" of a node as the difference between the similarity score of the node and the similarity score of the children. If the gain from a node is found to be minimal then it just stops constructing the tree to a greater depth which can overcome the challenge of overfitting to a great extend. Meanwhile, the Random forest might probably overfit the data if the majority of the trees in the forest are provided with similar samples. If the trees are completely grown ones then the model will collapse once the test data is introduced. Therefore, major consideration is given to distributing all the elementary units of the sample with approximately equal participation to all trees.

XGBoost (2) & Random Forest (0):

XGBoost is a good option for unbalanced datasets but we cannot trust random forest in these types of cases. In applications like forgery or fraud detection, the classes will be almost certainly imbalanced where the number of authentic transactions will be huge when compared with unauthentic transactions. In XGBoost, when the model fails to predict the anomaly for the first time, it gives more preferences and weightage to it in the upcoming iterations thereby increasing its ability to predict the class with low participation; but we cannot assure that random forest will treat the class imbalance with a proper process.

XGBoost (3) & Random Forest (0):

One of the most important differences between XG Boost and Random forest is that




affect almost all trees in the forest which can alter the prediction. Also, this is not a good approach when we expect test data with so many variations in real-time with a pre-defined mindset of hyperparameters for the whole forest but XG boost hyperparameters are applied to only one tree at the beginning which is expected to adjust itself in an efficient manner when iterations progress. Also, the XGBoost needs only a very low number of initial hyperparameters (shrinkage parameter, depth of the tree, number of trees) when compared with the Random forest.

XGBoost (4) & Random Forest (0):

When the model is encountered with a categorical variable with a different number of classes then there lies a possibility that Random forest may give more preferences to the class with more participation.

XGBoost (5) & Random Forest (0):

XGBoost may more preferable in situations like Poisson regression, rank regression, etc. This is because trees are derived by optimizing an objective function.

XGBoost (5) & Random Forest (1):

Random forests are **easier to tune** than Boosting algorithms.

XGBoost (5) & Random Forest (2):

Random forests **easily adapt to distributed computing** than Boosting algorithms.

XGBoost (5) & Random Forest (3):

Random forests will not overfit almost certainly if **the data is neatly pre-processed and cleaned** unless similar samples are repeatedly given to the majority of trees.

Ting...Ting...Ting...!!
The winner of this argument is **XGBoost!**

Disclaimer: These are my personal views. These views are independent of the fact that the choice of an algorithm hugely depends on the data at hand as well.




Sign up for Geek Culture Hits

By Geek Culture

Subscribe to receive top 10 most read stories of Geek Culture — delivered straight into your inbox, once a week. Take a look.

Emails will be sent to jimjywang@gmail.com. Not you?

Get this newsletter




← (https://www.educba.com/random-forest-vs-decision-tree/)

→ (https://www.educba.com/dynamodb-vs-cassandra/)

Difference Between Random Forest vs XGBoost

The following article provides an outline for Random Forest vs XGBoost. A machine learning technique where regression and classification problems are solved with the help of different classifiers combinations so that decisions are based on the outcomes of the decision trees is called

(https://www.educba.com/data-science/)

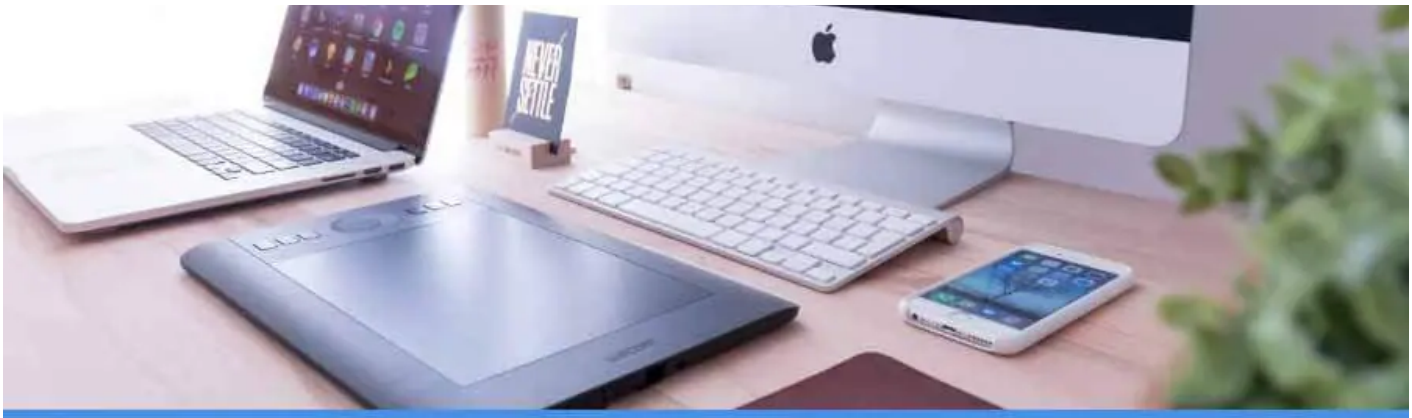
perfectly well along with cross-validation of facts and figures.

Head to Head Comparison Between Random Forest vs XGBoost (Infographics)

Below are the top 5 differences between Random Forest vs XGBoost:

EDUCBA

(https://www.educba.com/data-science/)



Random Forest	XGBoost
In Random Forest, the decision trees are built independently so that if there are five trees in an algorithm, all the trees are built at a time but with different features and data present in the algorithm. This makes developers to look into the trees and model it in parallel.	XGBoost builds one tree at a time so that each data pertaining the decision tree is taken into account and the data is filled if there are any missing data. This helps developers to work with gradient algorithm along with the decision tree algorithm for better results.

Random Forest	XGBoost
Once all the decision trees are built, the results are calculated by taking the average of all the decision tree values. This makes the developers to wait for building all the decision trees to the end and the cumulative results are taken into account.	While developers are building the decision trees, the results are calculated and added up for the next tree and hence the gradient of the results are considered. This helps developers to get an idea of the results even if the decision trees take

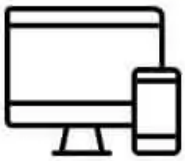
EDUCBA

(https://www.educba.com/data-science/)

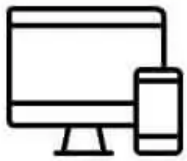
The calculation takes time and it is not accurate when compared to XGBoost. So, developers do not completely depend on Random Forest if there are other algorithms available. But, this is easy to do calculations even for beginners.

Since gradient of the data is considered for each tree, the calculation is faster and the precision is accurate than Random Forest. This makes developers to depend on XGBoost than Random Forest. XGBoost is complex than any other decision tree algorithms.

Random Forest	XGBoost



If the field of study is bioinformatics or multiclass object detection, Random Forest is the best choice as it is easy to tune and works well even if there are lots of missing data and more noise. Overfitting will not happen easily.



With accurate results, XGBoost is hard to work with if there are lots of noise. Also, it is hard to tune as well. If the data is real time so the data is unbalanced, we can use XGBoost where it performs exceptionally well.

Random Forest	XGBoost

Random Forest has many trees with leaves of equal weight so that high accuracy and precision can be obtained easily with the available data. This makes the developers to add more features to the data and look how it performs for all the data given to the algorithm.

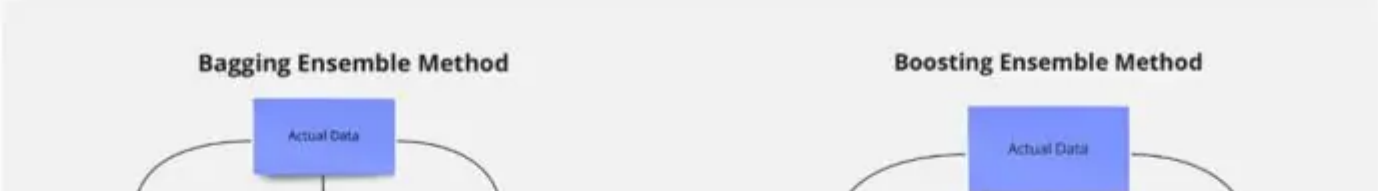
XGBoost does not account for the number of leaves present in the algorithm. If the model predictability is not good, the algorithm performs better with more number of leaves in the decision tree. This improves the bias and the results completely depends on

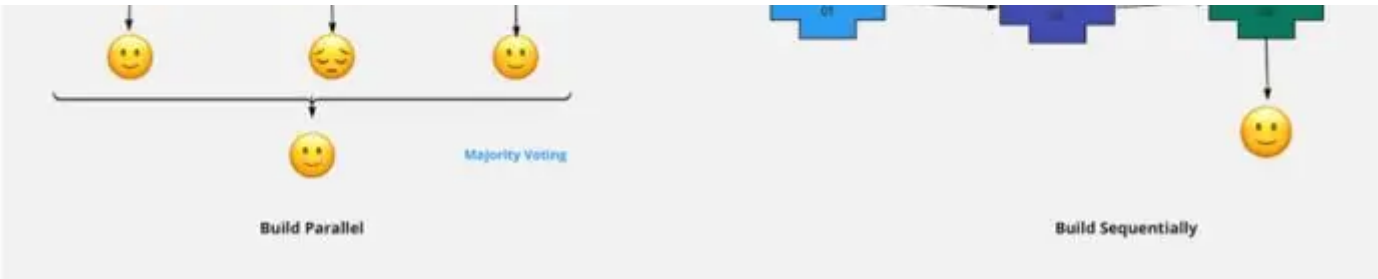
(https://www.educba.com/data-science/)

XGBoost

Let us discuss some of the major key differences between Random Forest vs XGBoost:

- Random Forest and XGBoost are decision tree algorithms where the training data is taken in a different manner. XGBoost trains specifically the gradient boost data and gradient boost decision trees. The training methods used by both algorithms is different. We can use XGBoost to train the Random Forest algorithm if it has high gradient data or we can use Random Forest algorithm to train XGBoost for its specific decision trees. Also, we can take samples of data if the training data is huge and if the data is very less, we can use the entire training data to know the gradient of the same.





- XGBoost helps in numerical optimization where the loss function of the data is minimized with the help of weak learners so that iteration happens in the local function in a differentiable manner. Sample is not modified here but different levels of importance are given to each feature in the data. Random Forest is mostly a bagging technique where various subsets are considered and an average of each subset is calculated. Either random subset of features or bootstrap samples of data is taken for each experiment in the data.
- Random subsamples of data are selected for Random Forest where the growing happens in parallel and overfitting is reduced with the combination of several underfitting features in the algorithm. Only a random subset of features is selected always that are included in the decision tree so that the result is not dependent on any subset of data. Overfitting is reduced with the help of regularization parameters in XGBoost that helps to select features based on weak and strong features in the decision tree. Algorithm is the combination of sequential growth by combining all the previous iterations in the decision trees. Optimal values of each leaf are calculated and hence the overall gradient of the tree is given as the output.
- Several hyperparameters are involved while calculating the result using XGBoost. Some include regularization rate, subsample, minimum weights, maximum depths, and learning rates. Though XGBoost is noted for better performance and high speed, these hyperparameters always stop developers from looking into this algorithm. Hyperparameters are not needed in Random Forest and developers can easily understand and visualize

Let's discuss the top comparison between Random Forest vs XGBoost.

Random Forest	XGBoost
In Random Forest, the decision trees are built	XGBoost builds one tree at a time so that

the algorithm. This makes developers look into the trees and model them in parallel.	developers to work with gradient algorithms along with the decision tree algorithm for better results.
Once all the decision trees are built, the results are calculated by taking the average of all the decision tree values. This makes the developers to wait for building all the decision trees to the end and the cumulative results are taken into account.	While developers are building the decision trees, the results are calculated and added up for the next tree and hence the gradient of the results is considered. This helps developers to get an idea of the results even if the decision trees take time.
The calculation takes time and it is not accurate when compared to XGBoost. So, developers do not completely depend on Random Forest if there are other algorithms available. But, this is easy to do calculations even for beginners.	Since gradient of the data is considered for each tree, the calculation is faster and the precision is accurate than Random Forest. This makes developers to depend on XGBoost than Random Forest. XGBoost is complex than any other decision tree algorithms.

If the field of study is bioinformatics or multiclass object detection, Random Forest is the best choice as it is easy to tune and works	With accurate results, XGBoost is hard to work with if there are lots of noise. Also, it is hard to tune as well. If the data is real-time so

equal weight so that high accuracy and precision can be obtained easily with the available data. This makes the developers add more features to the data and look at how it performs for all the data given to the algorithm.	leaves present in the algorithm. If the model predictability is not good, the algorithm performs better with more leaves in the decision tree. This improves the bias and the results completely depends on the data present in the algorithm.

Conclusion

It is important to have knowledge of both algorithms to decide which one to use for our data. If the dataset has no many differentiations and we are new to decision tree algorithms, it is better to use Random Forest as it provides a visualized form of the data as well. If we want to explore more about decision trees and gradients, XGBoost is good option.

Recommended Articles

This is a guide to Random Forest vs XGBoost. Here we discuss key differences with infographics and comparison table respectively. You may also have a look at the following articles to learn more –

1. Regression vs Classification (https://www.educba.com/regression-vs-classification/)
2. Supervised Learning vs Deep Learning (https://www.educba.com/supervised-learning-vs-deep-learning/)
3. Machine Learning vs Predictive Analytics (https://www.educba.com/machine-learning-vs-

(https://www.educba.com/data-science/)

Contact Us (https://www.educba.com/contact-us/?source=footer)

Verifiable Certificate (https://www.educba.com/data-science/verifiable-certificate/?source=footer)

Reviews (https://www.educba.com/data-science/reviews/?source=footer)

Terms and Conditions (https://www.educba.com/terms-and-conditions/?source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

Android (https://play.google.com/store/apps/details?id=com.educba.www)

Resources

Free Courses (https://www.educba.com/data-science/free-courses/?source=footer)

Database Management (https://www.educba.com/data-science/data-science-tutorials/database-management-tutorial/?source=footer)

Machine Learning (https://www.educba.com/data-science/data-science-tutorials/machine-learning-tutorial/?source=footer)

All Tutorials (https://www.educba.com/data-science/data-science-tutorials/?source=footer)

Certification Courses

All Courses (https://www.educba.com/data-science/courses/?source=footer)

(https://www.educba.com/data-science/)

Hadoop Certification Training (https://www.educba.com/data-science/courses/hadoop-certification-training/?source=footer)

Cloud Computing Training Course (https://www.educba.com/data-science/courses/cloud-computing-training-course/?source=footer)

R Programming Course (https://www.educba.com/data-science/courses/r-programming-course/?source=footer)

AWS Training Course (https://www.educba.com/data-science/courses/aws-training-course/?source=footer)

SAS Training Course (https://www.educba.com/data-science/courses/sas-training-course/?source=footer)