Section

Introduction to Random Forest in Machine Learning

December 11, 2020

Topics: Machine Learning

A random forest is a supervised machine learning algorithm that is constructed from decision tree algorithms. This algorithm is applied works. The article will present the algorithm’s features and how it is employed in real-life applications. It also points out the advantages and disadvantages of this algorithm.

What is a random forest?

A random forest is a machine learning technique that’s used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The ‘forest’ generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like scikit-learn).

Features of a Random Forest Algorithm

It’s more accurate than the decision tree algorithm.

It provides an effective way of handling missing data.

It can produce a reasonable prediction without hyper-parameter tuning.

It solves the issue of overfitting in decision trees.

In every random forest tree, a subset of features is selected randomly at the node’s splitting point.

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work.

A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further.

The nodes in the decision tree represent attributes that are used for predicting the outcome. Decision nodes provide a link to the leaves. The following diagram shows the three types of nodes in a decision tree.

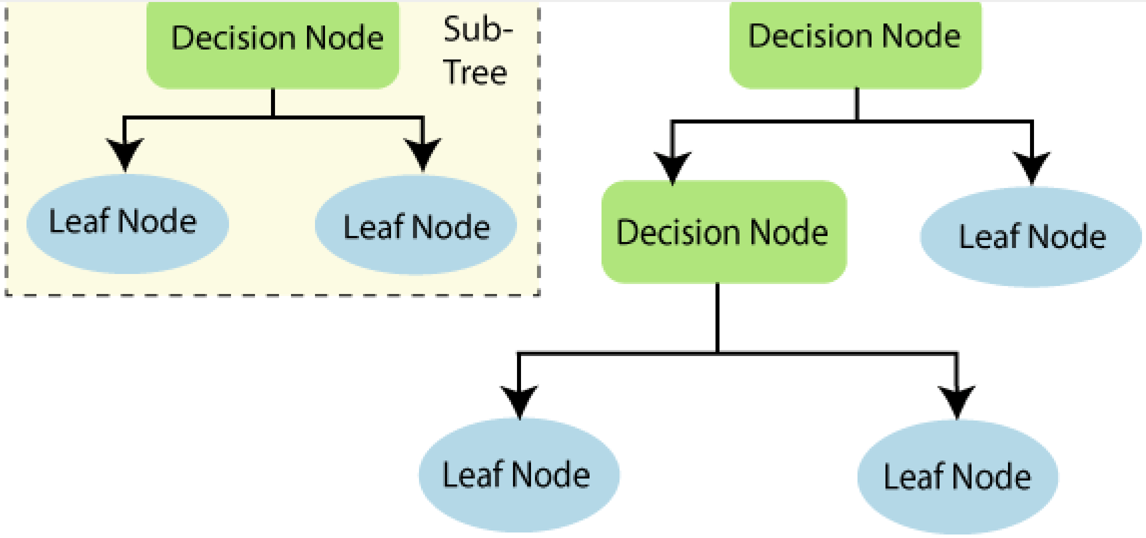


Image Source: Tutorials and Example

The information theory can provide more information on how decision trees work. Entropy and information gain are the building blocks of decision trees. An overview of these fundamental concepts will improve our understanding of how decision trees are built.

The information gain concept involves using independent variables (features) to gain information about a target variable (class). The entropy of the target variable (Y) and the conditional entropy of Y (given X) are used to estimate the information gain. In this case, the conditional entropy is subtracted from the entropy of Y.

Information gain is used in the training of decision trees. It helps in reducing uncertainty in these trees. A high information gain means that a high degree of uncertainty (information entropy) has been removed. Entropy and information gain are important in splitting branches, which is an important activity in the construction of decision trees.

Let’s take a simple example of how a decision tree works. Suppose we want to predict if a customer will purchase a mobile phone or not. The features of the phone form the basis of his decision. This analysis can be presented in a decision tree diagram.

decision tree will appear as follows.

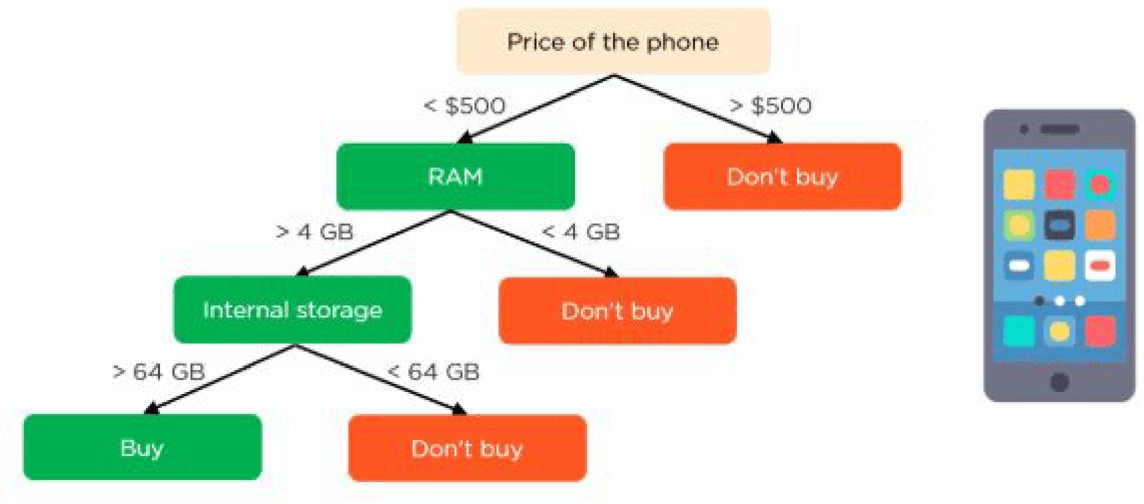


Image Source: Simplilearn

Applying decision trees in random forest

Bagging involves using different samples of data (training data) rather than just one sample. A training dataset comprises observations and features that are used for making predictions. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output.

Our first example can still be used to explain how random forests work. Instead of having a single decision tree, the random forest will have many decision trees. Let’s assume we have only four decision trees. In this case, the training data comprising the phone’s observations and features will be divided into four root nodes.

The root nodes could represent four features that could influence the customer’s choice (price, internal storage, camera, and RAM). The random forest will split the nodes by selecting features randomly. The final prediction will be selected based on the outcome of the four trees.

Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.

A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system. The diagram below shows a simple random forest classifier.

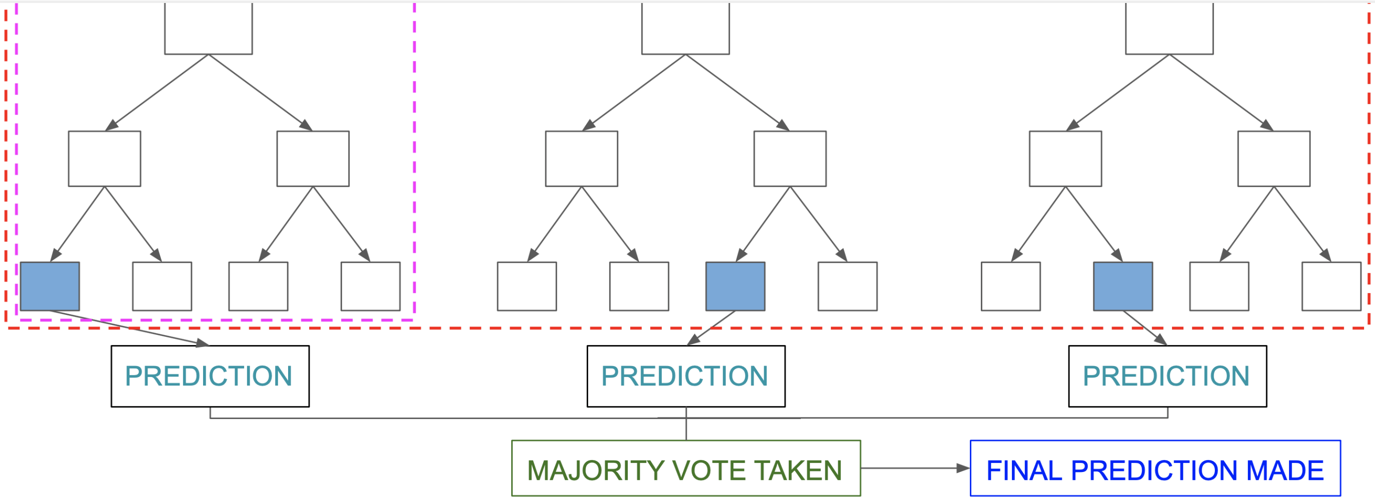
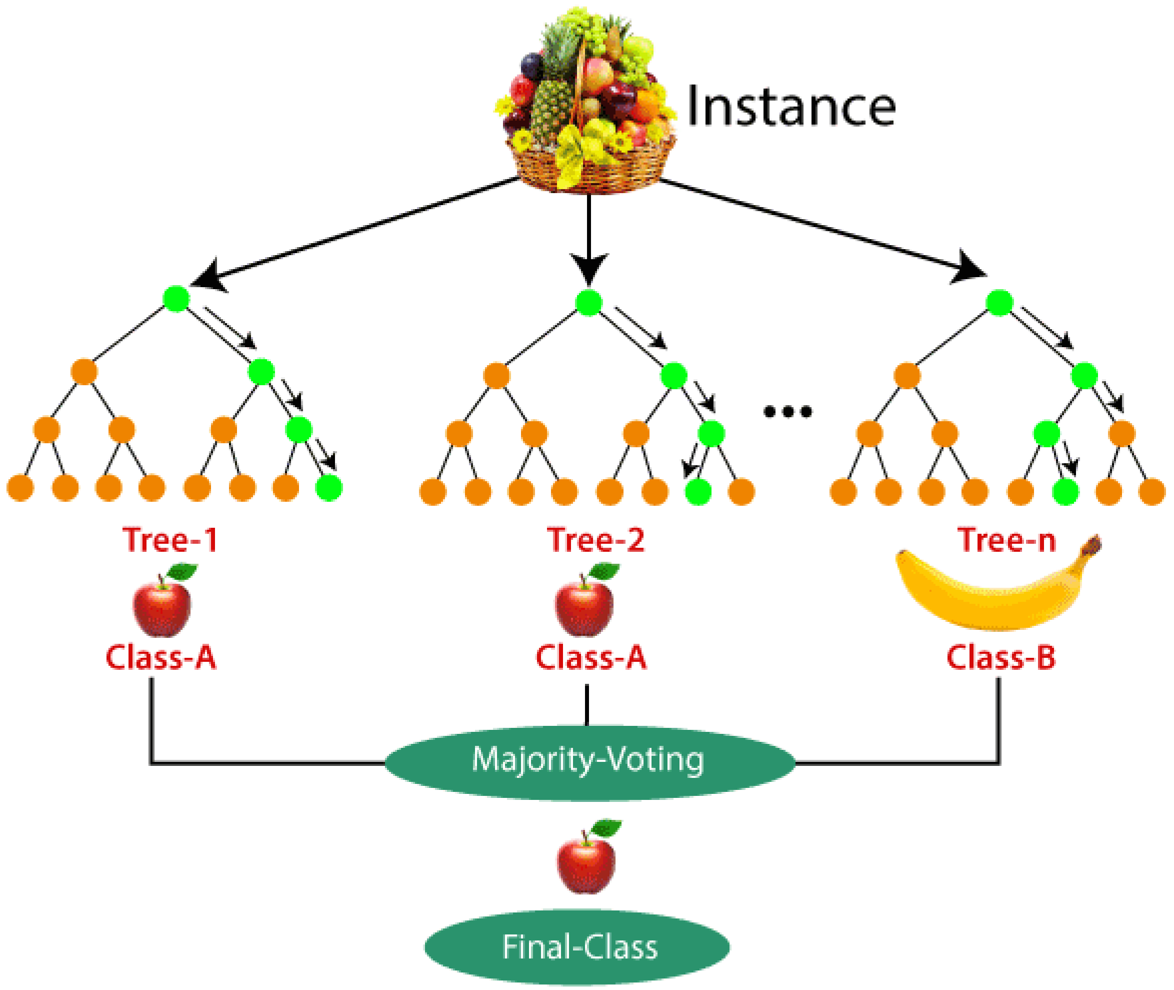


Image Source: Medium

Let's take an example of a training dataset consisting of various fruits such as bananas, apples, pineapples, and mangoes. The random forest classifier divides this dataset into subsets. These subsets are given to every decision tree in the random forest system. Each decision tree produces its specific output. For example, the prediction for trees 1 and 2 is *apple*.


Instance
Tree-1
Class-A
Tree-2
Class-A
Tree-n
Class-B
Majority-Voting
Final-Class

Regression is the other task performed by a random forest algorithm. A random forest regression follows the concept of simple regression. Values of dependent (features) and independent variables are passed in the random forest model.

We can run random forest regressions in various programs such as SAS, R, and python. In a random forest regression, each tree produces a specific prediction. The mean prediction of the individual trees is the output of the regression. This is contrary to random forest classification, whose output is determined by the mode of the decision trees' class.

Although random forest regression and linear regression follow the same concept, they differ in terms of functions. The function of linear regression is y=bx + c, where y is the dependent variable, x is the independent variable, b is the estimation parameter, and c is a constant. The function of a complex random forest regression is like a blackbox.

Banking

Random forest is used in banking to predict the creditworthiness of a loan applicant. This helps the lending institution make a good decision on whether to give the customer the loan or not. Banks also use the random forest algorithm to detect fraudsters.

Health care

Health professionals use random forest systems to diagnose patients. Patients are diagnosed by assessing their previous medical history. Past medical records are reviewed to establish the right dosage for the patients.

Stock market

Financial analysts use it to identify potential markets for stocks. It also enables them to identify the behavior of stocks.

When to avoid using random forests

Random forest algorithms are not ideal in the following situations:

Extrapolation

Random forest regression is not ideal in the extrapolation of data. Unlike linear regression, which uses existing observations to estimate values beyond the observation range. This explains why most applications of random forest relate to classification.

Sparse data

Random forest does not produce good results when the data is very sparse. In this case, the subset of features and the bootstrapped sample will produce an invariant space. This will lead to unproductive splits, which will affect the outcome.

A random forest produces good predictions that can be understood easily.

It can handle large datasets efficiently.

The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.

Disadvantages of random forest

When using a random forest, more resources are required for computation.

It consumes more time compared to a decision tree algorithm.

Conclusion

The rain forest algorithm is a machine learning algorithm that is easy to use and flexible. It uses ensemble learning, which enables organizations to solve regression and classification problems.

Resources

Dev Skrol

Free Code Camp

Towards Data Science

Peer Review Contributions by: Lalithnarayan C

About the author

Onesmus Mbaabu

Technology of China (UESTC), Sichuan Province, China. His interests include economics, data science, emerging technologies, and information systems. His hobbies are playing basketball and listening to music.

This article was contributed by a student member of Section's Engineering Education Program. Please report any errors or innaccuracies to enged@section.io.

Want to learn more about the EngEd Program?

+ QUICK LINKS // More Section offerings

Join our Slack community

Add to Slack

Company	Resources	Support
About	Blog	Docs
Careers	Case Studies	Community Slack

Section supports many open source projects including:

varnish cache logo cloud native computing foundation logo the linux foundation logo lf edge logo

More than one instance of Sumo is attempting to start on this page. Please check that you are only loading Sumo once per page.