Random Forest

This article discusses the random forest algorithm and its operation. The essay will discuss the algorithm's characteristics and how it is applied in real-world situations. Additionally, it discusses the algorithm's merits and downsides.

What is a random forest?

A random forest is a type of machine learning technique that is used to address problems involving regression and classification. It makes use of ensemble learning, which is a technique for solving complicated problems by combining several classifiers.

A random forest method is composed of a large number of decision trees. The random forest algorithm generates a 'forest' that is trained via bagging or bootstrap aggregation. Bagging is a meta-algorithm used in ensembles to increase the accuracy of machine learning systems.

The (random forest) method generates the outcome based on the decision trees' predictions. It forecasts by averaging or summing the output of several trees. Increasing the number of trees improves the outcome's precision.

A random forest method overcomes the drawbacks of a decision tree algorithm. It decreases overfitting and boosts precision in datasets. It generates forecasts without requiring numerous package configurations (like scikit-learn).

The Random Forest Algorithm's Characteristics

- It outperforms the decision tree algorithm in terms of accuracy.
- It enables the appropriate management of missing data.
- It is capable of producing a reasonable prediction in the absence of hyperparameter adjustment.
- It resolves the issue of decision tree overfitting.
- At the node's splitting point in each random forest tree, a subset of features is randomly selected.

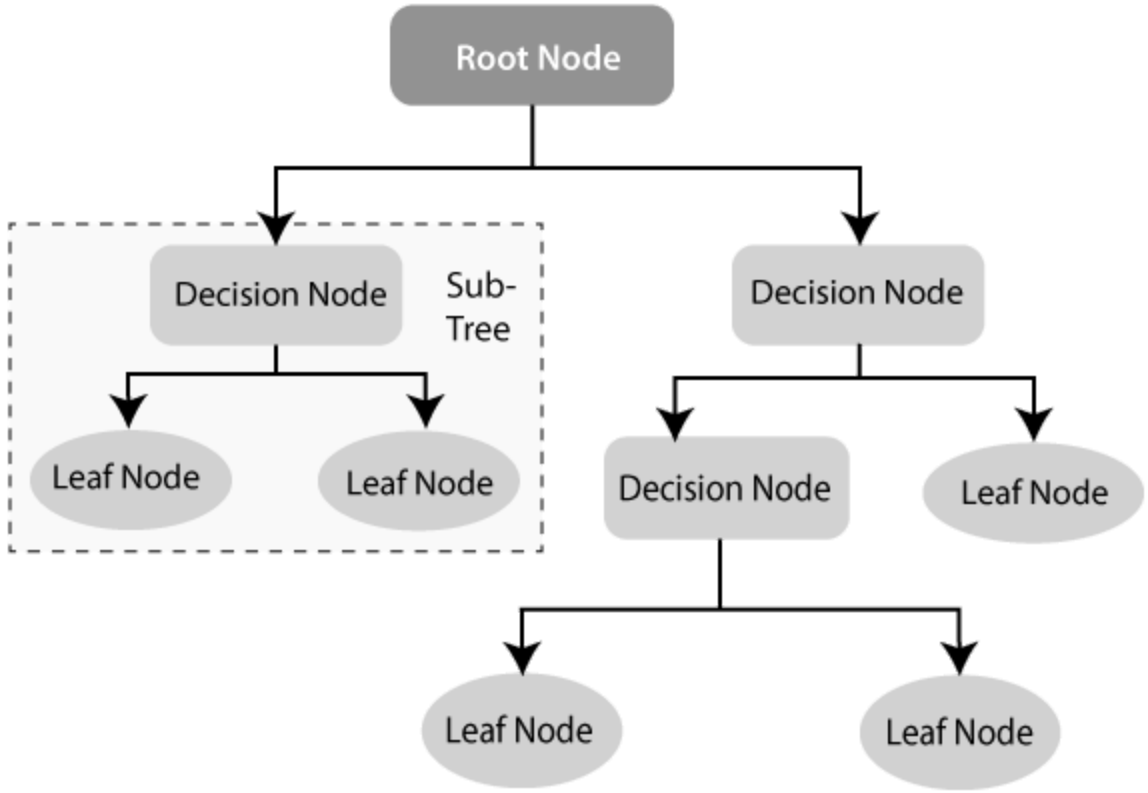
How does it work?

Understanding decision trees

A random forest algorithm's building components are decision trees. A decision tree is a type of decision support technique that takes the form of a tree. A review of decision trees will assist us in comprehending the operation of random forest algorithms.

Three components comprise a decision tree: decision nodes, leaf nodes, and a root node. A decision tree method divides a training dataset into branches that subdivide further into other branches. This procedure is repeated until a leaf node is reached. Further segregation of the leaf node is not possible.

The decision tree's nodes indicate the attributes that are utilised to forecast the outcome. The decision nodes connect the leaves. The following diagram illustrates the three different types of nodes that exist in a decision tree.



Nodes of the Decision Tree

Information theory can shed more light on the operation of decision trees. Decision trees are constructed using entropy and information gain. A review of these essential concepts will help us comprehend how decision trees are constructed.

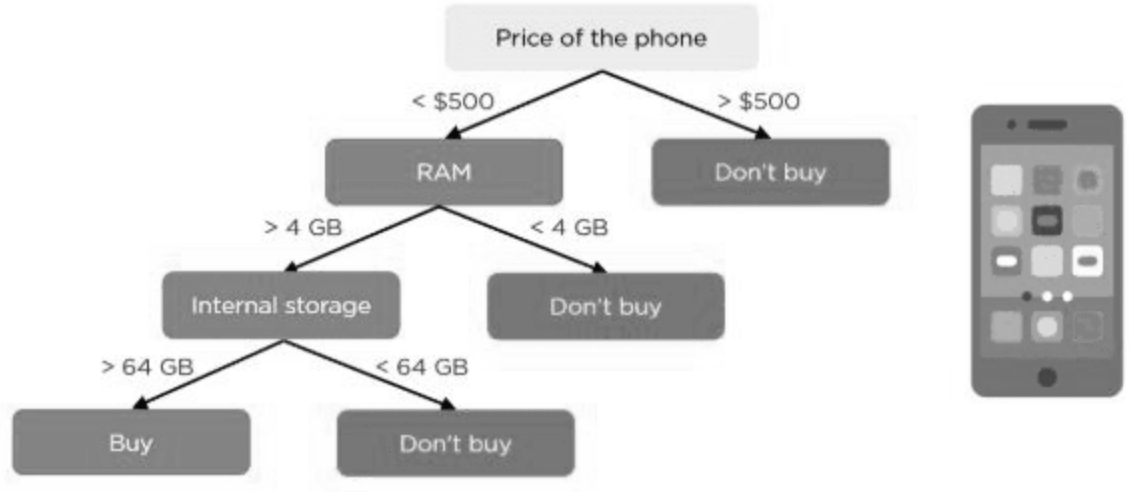
Entropy is a measure of uncertainty. Information gain is a measure of how much uncertainty is removed from a target variable when a set of independent variables is used.

The idea of information gain entails the use of independent variables (features) to elicit data about a target variable (class). The information gain is estimated using the entropy of the target variable (Y) and the conditional entropy of Y (given X). The conditional entropy is deducted from the entropy of Y in this scenario.

Information gain is a technique used to train decision trees. It contributes to the reduction of uncertainty in these trees. A significant information gain indicates that a great degree of uncertainty has been removed (information entropy). Splitting branches, which is a critical action in the creation of decision trees, requires entropy and information gain.

Consider a straightforward demonstration of how a decision tree works. Assume we want to forecast whether a customer will purchase a phone or not. His judgement is based on the phone's specifications. A decision tree diagram can be used to illustrate this analysis.

The decision's root node and decision nodes depict the phone's above-mentioned attributes. The leaf node denotes the final outcome, which might be either purchasing or not purchasing. The primary criteria for selection are the price, internal storage, and Random Access Memory (RAM). The following diagram depicts the decision tree.



A Decision Tree in Action

Application of decision trees in random forest

The primary distinction between the decision tree and random forest algorithms is that the latter randomly establishes root nodes and segregates nodes. The random forest generates the required forecast using the bagging approach.

Bagging is a technique that utilises multiple samples of data (training data) rather than a single sample. A training dataset is a collection of observations and attributes used to make predictions. Decision trees generate distinct outputs based on the training data input into the random forest algorithm. These outputs will be ranked, and the one with the highest score will be chosen as the final result.

Our initial illustration can still be used to demonstrate how random forests work. Rather than a single decision tree, the random forest will consist of several decision trees. Assume we only have four decision trees. In this example, the training data will be separated into four root nodes based on the phone's observations and features.

The root nodes may symbolise four distinct characteristics that may impact the customer's choice (price, internal storage, camera, and RAM). The random forest will split the nodes randomly based on their features. The ultimate forecast will be chosen based on the four trees' output.

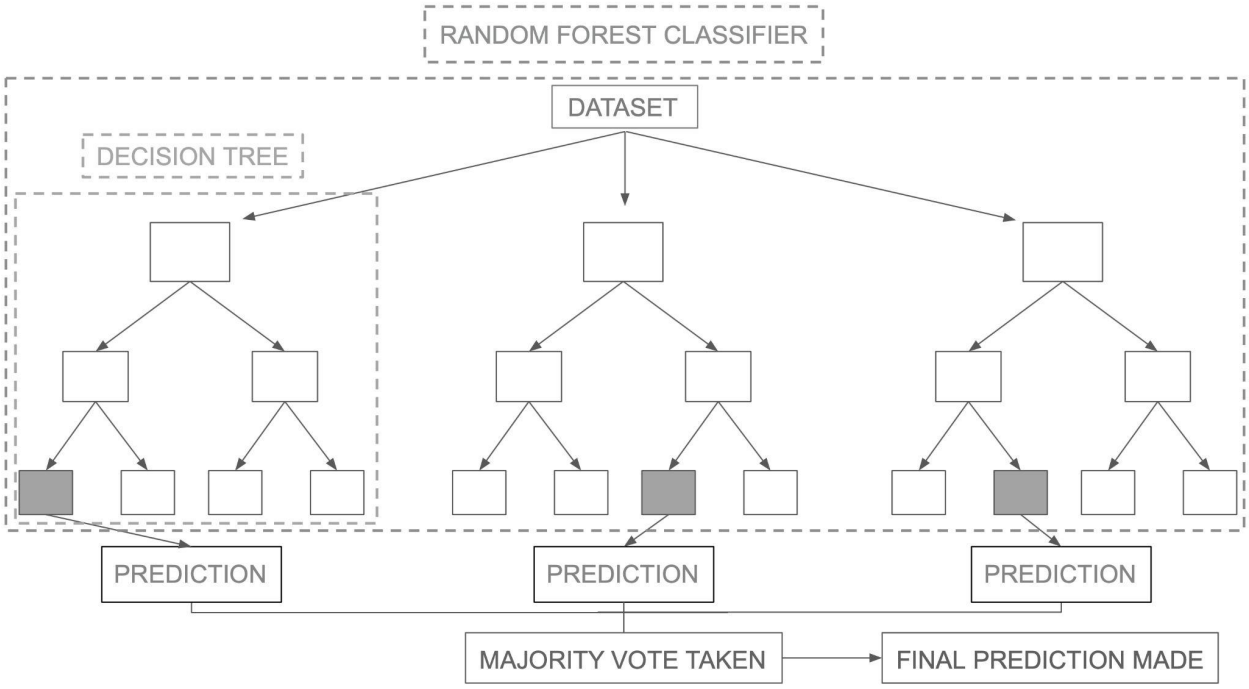
Most decision trees will choose the final outcome. If three trees forecast purchasing and one tree forecasts not purchasing, the final prediction will be purchasing. In this instance, the consumer is expected to purchase the phone.

Random forest classification

Classification in random forests is accomplished through the use of an ensemble methodology. The training data is fed into several decision trees for training. This dataset contains observations and features that will be randomly chosen during node splitting.

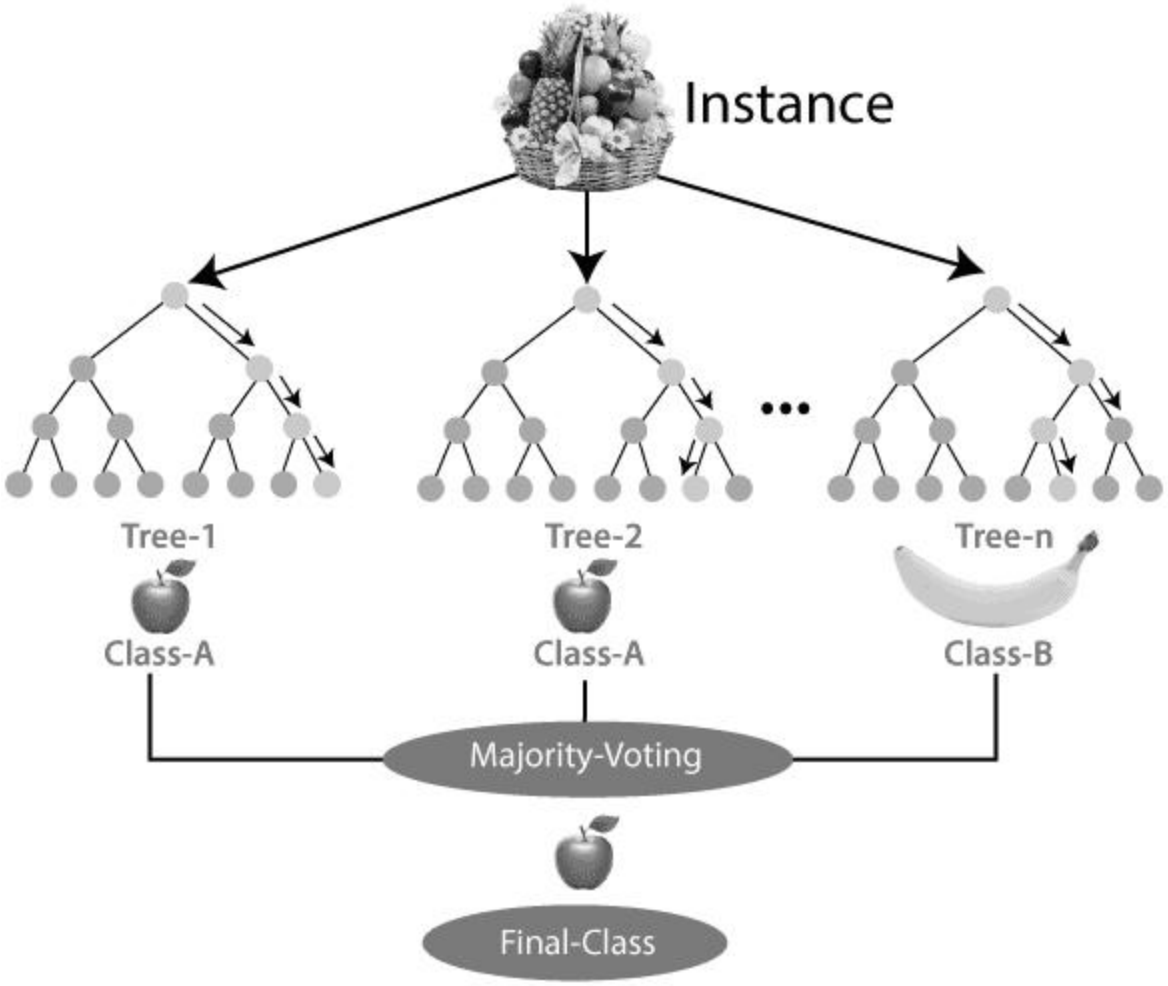
A rainforest ecosystem is composed of numerous decision trees. Each decision tree is composed of three types of nodes: decision nodes, leaf nodes, and a root node. Each tree's leaf node represents the decision tree's final result. The final product is chosen using a majority-voting procedure. In this situation, the output selected by the majority of decision trees

becomes the rain forest system's final output. The graphic below illustrates a straightforward random forest classifier.



Consider a training dataset composed of a variety of fruits, such as bananas, apples, pineapples, and mangoes. This dataset is subdivided using the random forest classifier. Each decision tree in the random forest system is given a subset of these subsets. Each decision tree generates a distinct outcome. For instance, apples are predicted for trees 1 and 2.

Another decision tree (n) anticipated the outcome would be banana. The random forest classifier aggregates majority votes in order to provide the final prediction. The majority of decision trees predicted apples. This causes the classifier to make the final prediction of apples.



A Random Forest Classifier in Action

Regression in random forests

Regression is the other task that a random forest algorithm performs. A random forest regression is based on the simple regression principle. The random forest model passes the values of dependent (features) and independent variables.
We can perform random forest regressions in a variety of different languages, including SAS, R, and Python. Each tree in a random forest regression produces a unique forecast. The regression result is the mean forecast of the individual trees. This is in contrast to random forest classification, which produces output based on the decision trees' class mode.

While both random forest and linear regression are based on the same notion, their purposes are distinct. Linear regression has the formula y=bx + c where y is the dependent variable, x is the independent variable, b is the estimation parameter, and c is a constant. A sophisticated random forest regression's function is similar to that of a blackbox.

Random forest applications

Several random forest applications include the following:

Banking

In banking, random forests are used to forecast a loan applicant's creditworthiness. This enables the lending organisation to make an informed choice about whether or not to extend the loan to the consumer. Additionally, banks employ the random forest algorithm to identify fraudsters.

Medical care

Health workers diagnose patients using random forest systems. Patients are diagnosed by an examination of their prior medical history. Previous medical data are evaluated to determine the appropriate dosage for each patient.

The stock exchange

It is used by financial analysts to determine possible markets for stocks. Additionally, it enables them to ascertain the stock's behaviour.

E-commerce

Rain forest algorithms enable e-commerce businesses to forecast a customer's choice based on previous consumption behaviour.

When to avoid using random forests

In the following cases, random forest methods are not optimal:

Extrapolation

Random forest regression is not the optimal method for data extrapolation. In contrast to linear regression, which makes use of current observations to estimate values outside the observation range, logistic regression makes use of unobserved values. This explains why the majority of random forest applications are related to classification.

Sparse data

When the data is extremely scarce, random forest does not generate good results. The combination of the subset of features and the bootstrapped sample results in an invariant space in this scenario. This will result in ineffective divides, which will have an impact on the outcome.

Advantages of random forest

- It is capable of doing regression as well as classification tasks.
- A random forest generates accurate predictions that are clearly understandable.
- It is capable of effectively handling huge datasets.
- The random forest algorithm is more accurate than the decision tree method at predicting outcomes.

Disadvantages of random forest

- When a random forest is used, additional computational resources are required.
- It is slower than a decision tree algorithm.

The rain forest algorithm is a simple and flexible machine learning technique. It makes use of ensemble learning to assist companies in resolving regression and classification issues.

This is a good technique for developers because it addresses the issue of dataset overfitting. It is a very useful tool for creating accurate predictions necessary for organisational strategic decision-making.