Decision Tree

Decision Trees (DTs) are a **supervised learning** technique that predict values of responses by learning decision rules derived from features. They can be used in both a regression and a classification context.

A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute.

Each branch represents the outcome of the test, and each leaf node represents a class label.

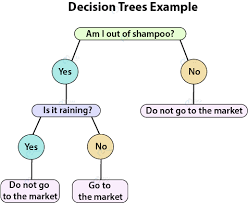
A decision tree is a decision support tool that uses a tree – like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

It is one way to display an algorithm that only contains conditional control statements.

DTs are a supervised learning technique that predict values of responses by learning decision rules derived from features.

It provides an effective method of Decision Making because they: Clearly lay out the problem so that all options can be challenged.

If the goal is exploratory analysis, we should prefer a single Decision Tree, as to understand the data relationship in a tree hierarchy structure. DT gives simple model or accuracy desired.



Naïve Bayes

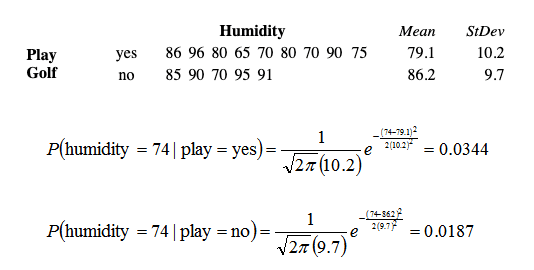
Naive Bayes classification is a form of **supervised learning**. It is considered to be supervised since naive Bayes classifiers are trained using labelled data, i.e., ... This contrasts with unsupervised learning, where there is no pre-labelled data available.

It is **a classification technique based on Bayes' Theorem with an assumption of independence among predictors**. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Missing Data: Naive bayes can handle missing data.

Use Log Probabilities: To calculate joint probabilities, you need to multiply probabilities together. Naive Bayes can help with this large probability.

Segment The Data: Identifying and separating out segments that are easily handled by a simple probabilistic approach like Naive Bayes can give you increase performance and focus on the elements of the problem that are most difficult to model.



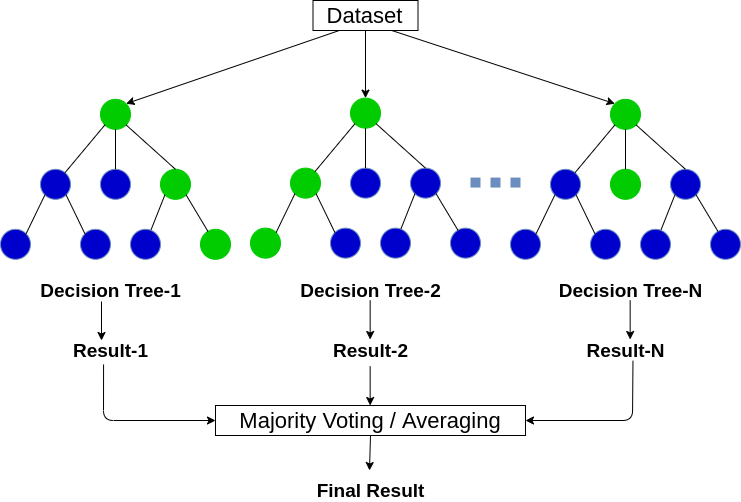
Random Forest

A random forest is a **supervised machine** learning algorithm that is constructed from decision tree algorithms. This algorithm is applied in various industries such as banking and e-commerce to predict behaviour and outcomes.

The random forest algorithm **establishes the outcome based on the predictions of the decision trees**. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

Random Forest well on most data sets. Random Forest is highly robust to over fitting problem, which means it should have less variance. And as per variance trade-off it can have high bias.

It is an extremely flexible and can handle unnormalized data easily. Higher accuracy on datasets with multiple features.



random forest is a group of decision trees. However, there are some differences between the two. A decision tree tends to create rules, which it uses to make decisions. A random forest will randomly choose features and make observations, build a forest of decision trees, and then average out the results.

