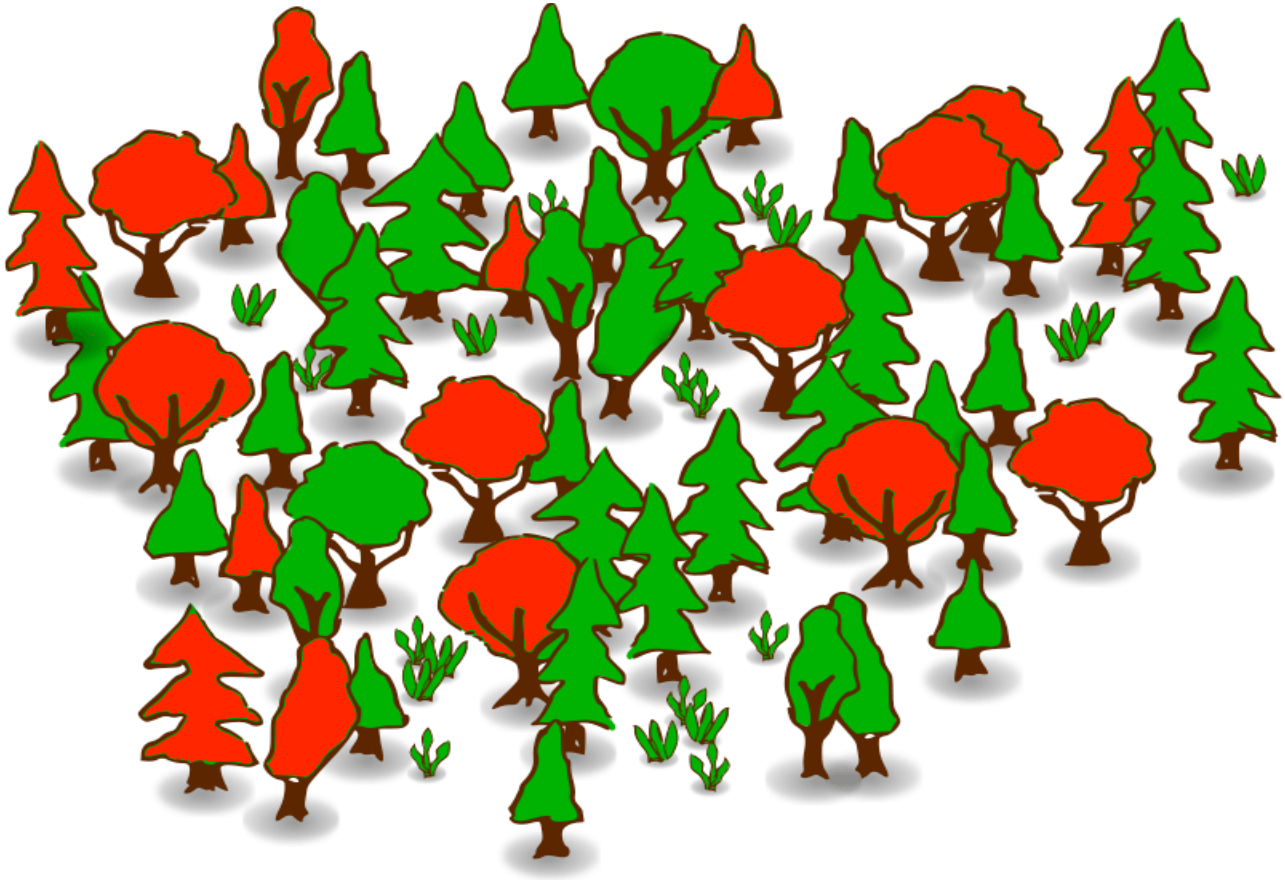
Ensemble Method: Random Forest



What is random forest?

Random forest is an ensemble machine learning algorithm. It is one of the most versatile machine learning algorithms capable of solving both classification and regression problems.

With its built-in ensembling capacity, the task of building a decent generalized model gets much easier.

Its ability to solve both regression and classification problems along with robustness to correlated features and variable importance plot gives enough head start to solve various problems.

What is random forest?

The fundamental idea behind a random forest is to combine many decision trees into a single model. Individually, predictions made by decision trees (or humans) may not be accurate, but combined together, the predictions will be closer to the mark on average.

It operates by building multiple decision trees, then combining their output to improve generalization ability of the model. The method of combining trees is known as an ensemble method.

Ensembling is nothing but a combination of weak learners (individual trees) to produce a strong learner.

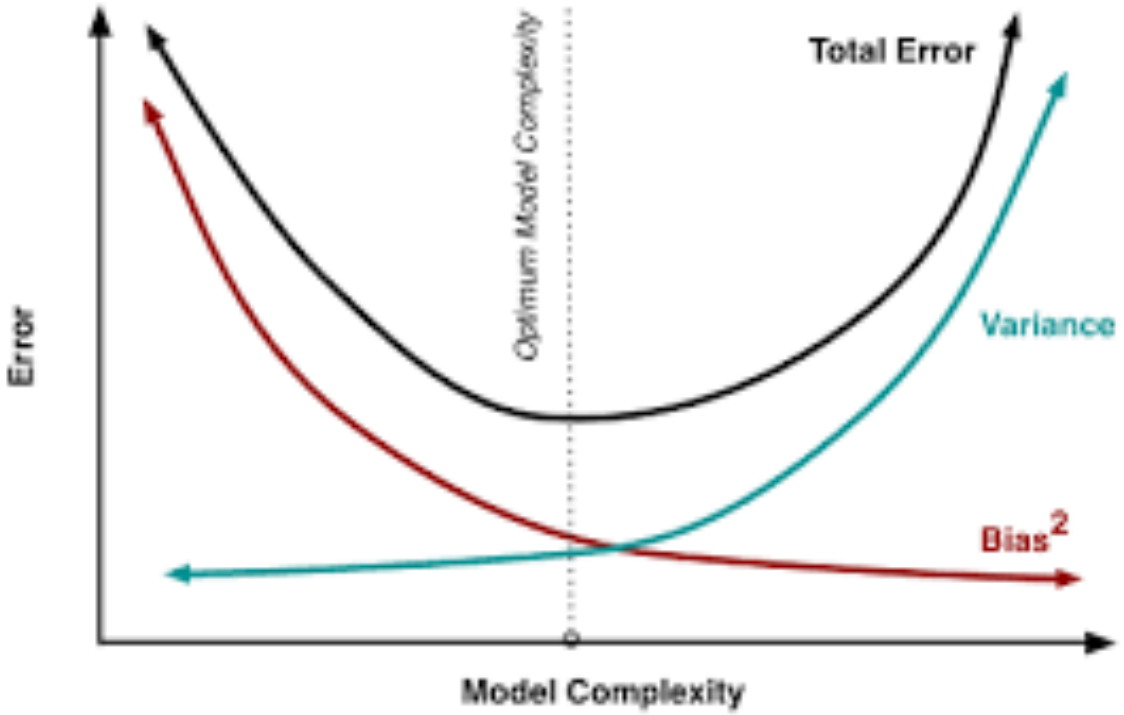
Problem with Decision Trees

A tree based model suffers from bias and variance problem and suffers from over fit problem.

If we make a small tree or simple model, it suffers from low variance and high bias, i.e., under fit

Decision tree are sensitive to the specific data on which they are trained. If the training data is changed the resulting tree can be quite different and in turn the predictions can be quite different

So there should be balance between these two types of errors. This is known as the trade-off balance of bias-variance errors.



Things to consider

Ensemble learning is one way to execute this trade off analysis.

The literary meaning of word “ensemble’ is group. Ensemble methods involve group of predictive models to achieve a better accuracy and model stability.

Ensemble methods impart supreme boost to tree base models

Some of the commonly used ensemble methods include: Bagging and Boosting

Bootstrap

Bootstrap is a statistical resample method

When you can’t afford to get more sample (medical data)

Used in statistic when you want to estimate a statistic of a random sample (a statistic mean, variance, mode etc.)

Drawing randomly with replacement set of available data.

Bootstrap

Ideally we would like to take more samples, but n is all what we have got

What to do?

What we do is sample our data set with replacement.

We repeat the above step for a large number of times, say P times. Once done we have P number of bootstrap random samples

We then take the statistic of each bootstrap random sample and average it.

Bootstrap

We are getting the range for that mean of the population.

Important: Bootstrapping does not give “better” estimates. It gives a better range for the estimate

Bootstrap

Original data (random sample)

{2,4, 5,6, 6 }, mean = 4.6

Bootstrap samples:

{2,5,5,6,6}, mean = 4.8

{4,5,6,6,6}, mean = 5.4

{2,2,4,5,5} , mean = 3.6

{2,2,2,4,6}, mean = 3.2

{4,6,6,6,6} = mean = 5.6

mean = 4.52

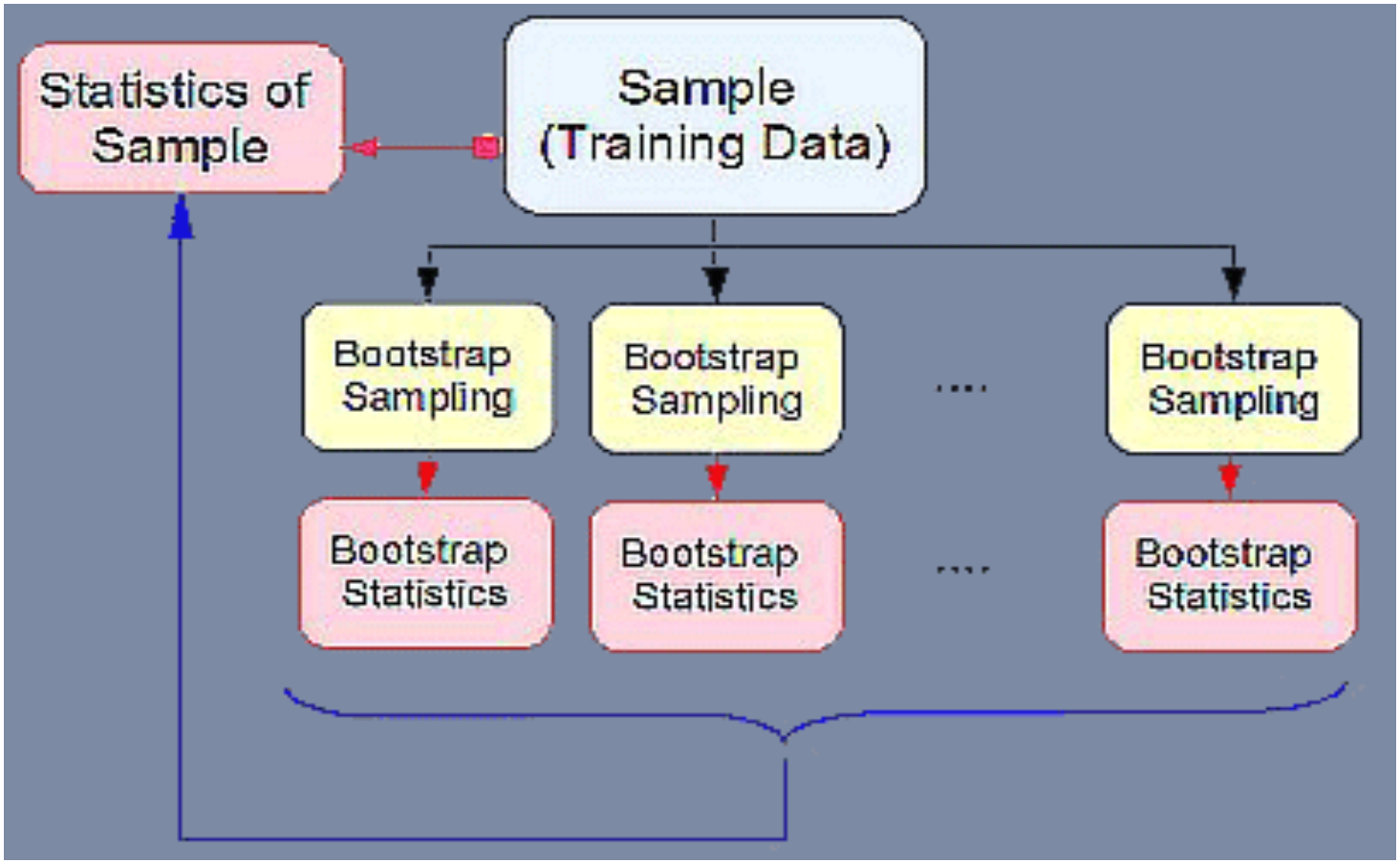
Bootstrap

- It is especially useful when the sample size that we are working with is small. These kind of techniques assume nothing about the distribution of our data.

- It comes in handy when there is doubt that the useful distribution assumptions and asymptotic results are valid and accurate.

- Bootstrapping is a nonparametric method which lets compute estimated standard errors, confidence intervals and hypothesis testing.

Bootstrap


Statistics of Sample
Sample (Training Data)
Bootstrap Sampling
Bootstrap Sampling
....
Bootstrap Sampling
Bootstrap Statistics
Bootstrap Statistics
....
Bootstrap Statistics

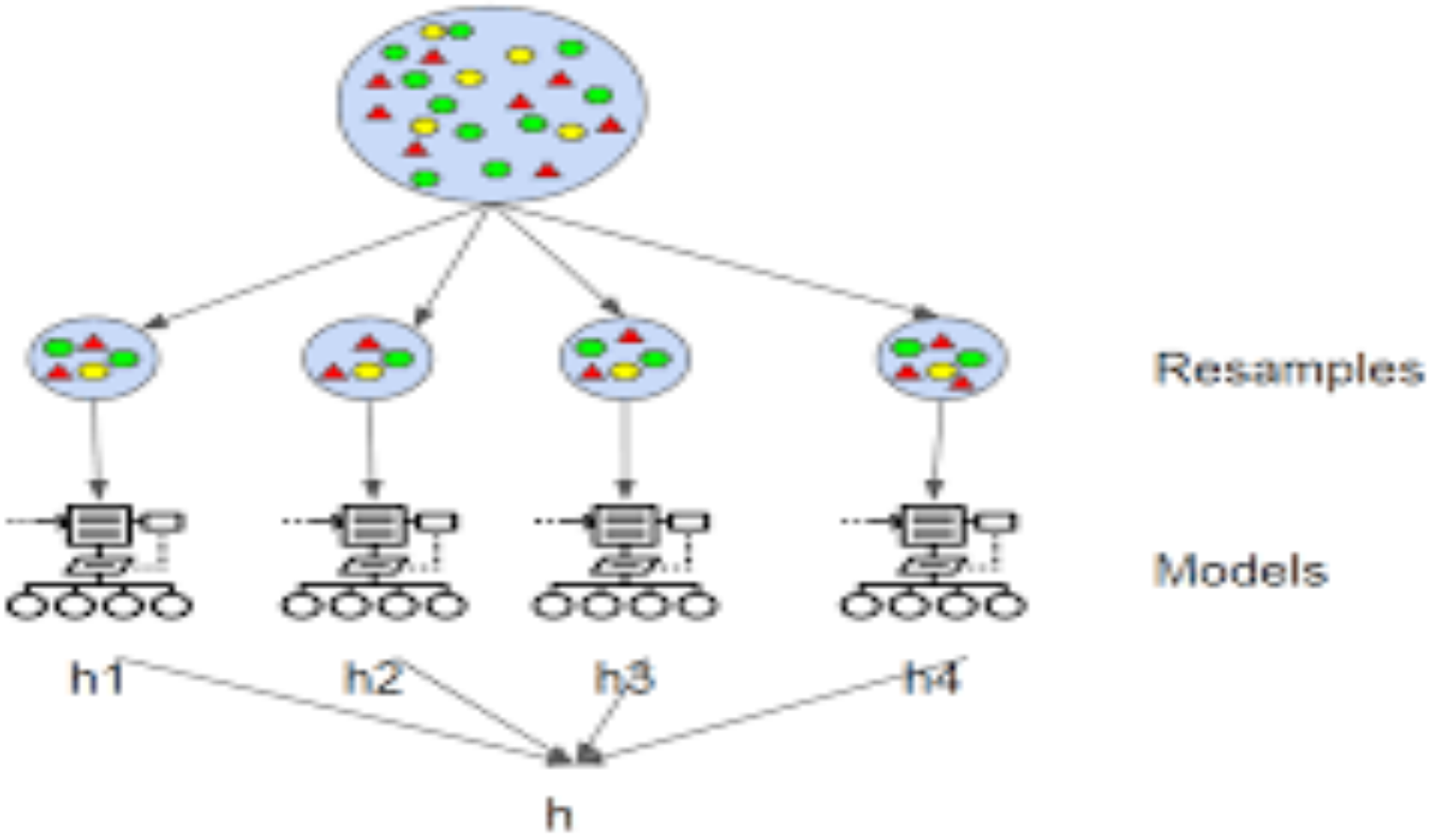
Bootstrap

- What are the assumption of Bootstrap?
- Bootstrap method has assumptions
  - Sample is a valid representative of the population
  - Bootstrap method will take sampling with replacement from the sample. Each sub sampling is independent and identical distribution (i.i.d). In other word, it assumes that the sub samples come from the same distribution of the population, but each sample is drawn independently from the other samples.

- Generally bootstrapping follows the same basic steps:
  1. Resample a given data set a specified number of times
  2. Calculate a specific statistics from each sample
  3. Find the standard deviation of the distribution of that statistic

Bagging (Bootstrap Aggregation)

Bagging is a technique used to reduce the variance by combining the result of multiple classifiers modeled on different sub-samples of the same data set.



Bagging

The following steps are followed:

1. **Create multiple Data sets:**
   - Sampling is done with replacement on the original data and new data sets are formed
   - The new data sets can have a fraction of the columns as well as rows, which are generally hyper-parameters in a bagging model

2. **Create multiple Classifiers:**
   - Classifiers are built on each data set
   - Generally the same classifier is built on each data set and predictions are made.

Bagging

3. **Combine Classifiers:**
    - The prediction of all the classifiers are combined using mean, median or mode value depending on the problem at hand

There are various implementations of bagging models, Random Forest is one of them.

What is random forest?

Let us say you want to go on a vacation to a certain place, and you are uncertain about the place. You ask 10 people who have visited that place. 7 of them recommended. Since the majority is in favor, you decide to go there. This is how we use ensemble techniques in out daily life.

Random Forest

- Random forest is a versatile machine learning method capable of performing both regression and classification tasks.
- It is one of the most popular and most powerful machine learning algorithms
- It is a type of ensemble machine learning algorithm called Bootstrap Aggregation or bagging
- It can also undertake dimensional reduction methods, treats missing values, outlier values and does a fairly good job.
- It is a type of ensemble learning method, where a group of weak models combine to form a powerful model.

Random Forest

- Random forest are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of overcoming over-fitting problem of individual decision tree

- In other words, random forests are an ensemble learning method for classification and regression that operate by constructing a lot of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.

- Random forest come at the expense of a some loss of interpretability, but generally greatly boosts the performance of the final model

How does it work?

- Multiple trees are grown during modeling in Random forest
- To classify a new object based on attributes, each tree gives a classification and votes for that class.
- The forest chooses the classification having the most votes and in case of regression, it takes the average of outputs by different trees

Random Forest

Working : Each tree is grown as follows:

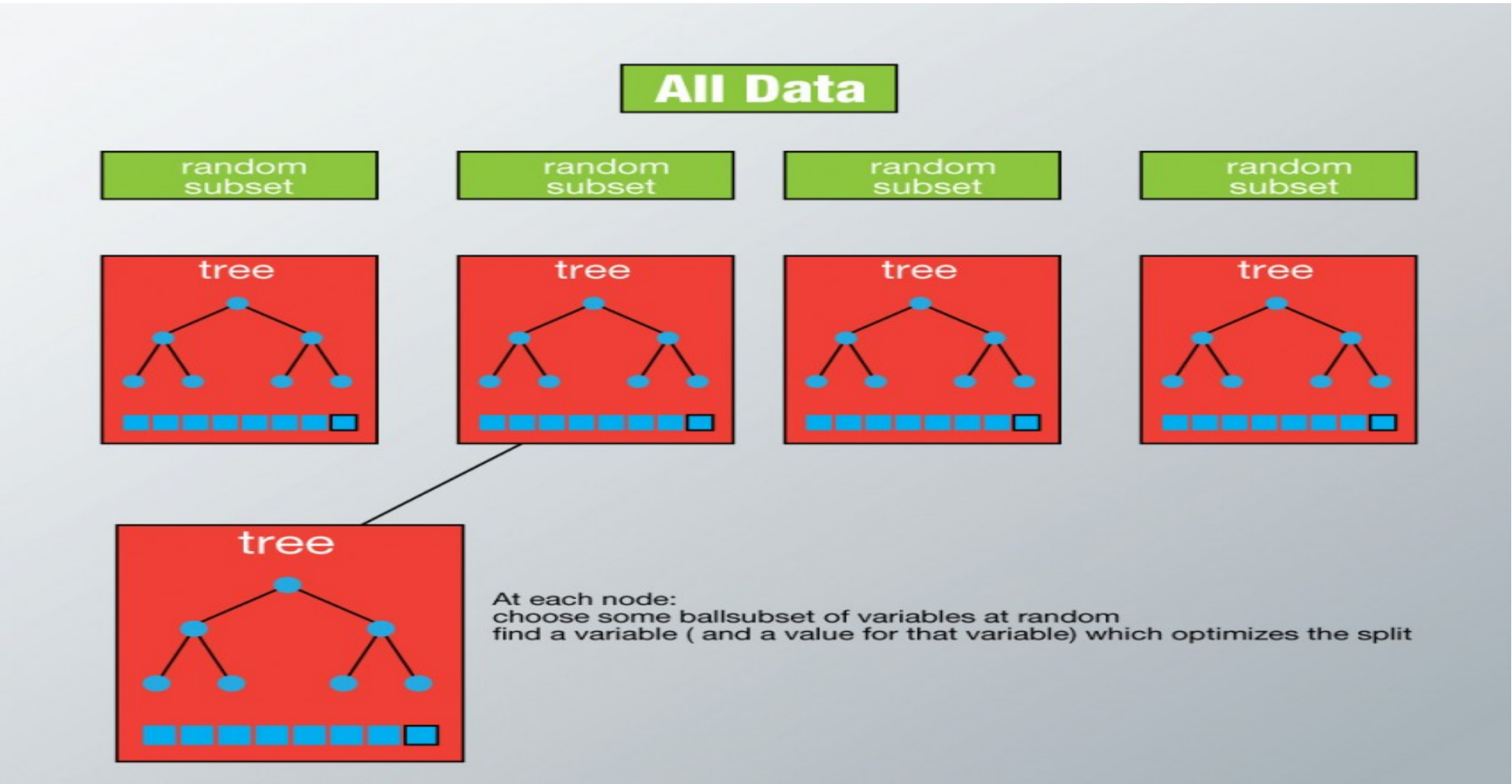
1. Assume number of cases in the training set is N. Then, sample of these N cases is taken at random but with replacement. This sample will be the training set for growing the tree.

2. If there are M input variables, a number m<M is specified such that at each node, m variables are selected at random out of the M. The best split of these m is used to split the node. The value of m is held constant while we grow the forest.

3. Each tree is grown to the largest extent possible

Random Forest

4. Predict new data by aggregating the predictions of the n trees (i.e., majority votes for classification, average for regression)



How does Random Forest work

Each tree is grown as follows:

1. **Random Record Selection**: Each tree is trained on roughly 2/3rd of the total training data. Cases are drawn at random with replacement from the original data.

2. **Random Variable Selection**: Random number of predictors (say, m) are selected at random out of all the predictor variables and the best split on these m is used to split the node.

- The number of features that can be searched at each split point (m) must be specified as a parameter to the algorithm.
  - For classification a good default is: m = sqrt(p)
  - For regression a good default is: m = p/3

How does Random Forest work

3. For each tree, using the leftover (1/3rd) data, calculate the misclassification rate, out of bag (OOB) error rate. Aggregate error from all trees to determine overall OOB error rate for the classification.

4. Each tree gives a classification on leftover data (OOB). The forest chooses the classification having the most votes over all the trees in the forest.

What is random in Random Forest?

Why the name ‘random forest?’ Well, much as people might rely on different sources to make a prediction, each decision tree in the forest considers a random subset of features when forming questions and only has access to a random set of the training data points. This increases diversity in the forest leading to more robust overall predictions and the name ‘random forest.’ When it comes time to make a prediction, the random forest takes an average of all the individual decision tree estimates. (This is the case for a regression task, such as our problem where we are predicting a continuous value of temperature. The other class of problems is known as classification, where the targets are a discrete class label such as cloudy or sunny. In that case, the random forest will take a majority vote for the predicted class).

Estimated Performance

- For each bootstrap sample taken from the training data, there will be samples left behind that were not included. These samples are called Out-of-Bag samples or OOB

- The performance of each model on its left out samples when averaged can provide an estimated accuracy of the bagged models. This estimated performance is often called the OOB estimate of performance

- These performance measures are reliable test error estimate and correlate well with cross validation estimates.

Variable Importance

- As trees are constructed, we can calculate how much the error drops for a variable at each split point
- In regression this may be drop in sum squared error and in classification this might be the Gini score
- These drops in error can be averaged across all the decision trees and output to provide an estimate of the importance of each input variable. The greater the drop when the variable was chosen, the greater the importance.

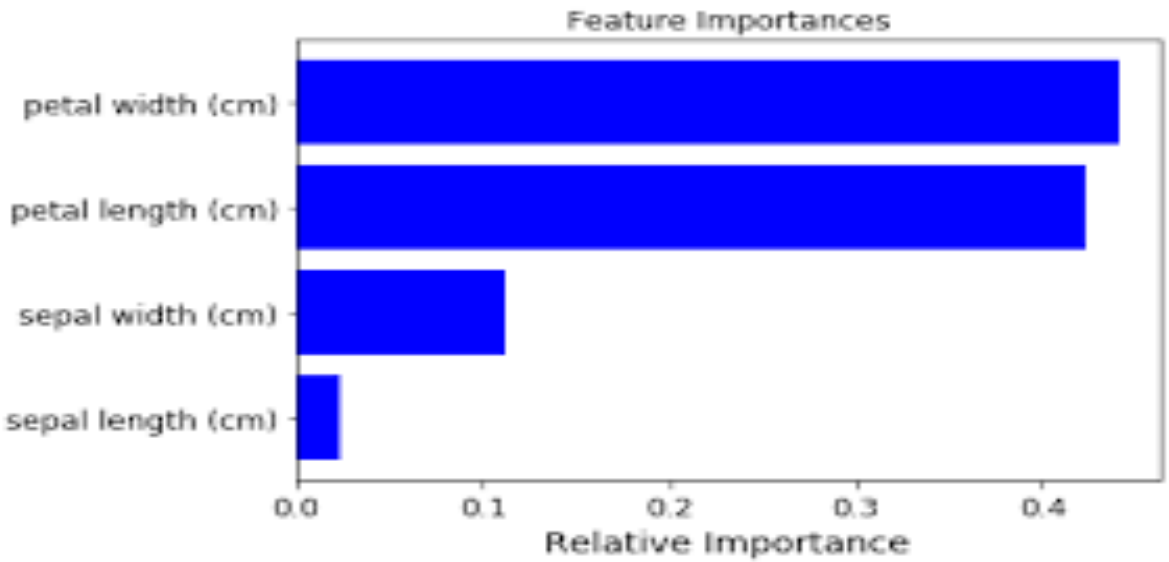
Variable Importance

- These outputs can help identify subsets of input variables that may be most or least relevant to the problem and suggest at the possible feature selection experiments one could perform where some features are removed from the dataset.

.

Advantages

- The model outputs importance of variable, which can be a very handy feature



Tuning Parameters

Random forests are considered as a black box models which takes in input and gives out predictions, without worrying too much about what is going on the back end. The black box models have few parameters to play with.

Each of these parameters have some effect on either performance of the model or the resource - speed. Tuning the parameter results in the optimized value of the parameters.

While model parameters are learned during training – such as the slope and intercept in a linear regression- hyperparameters must be set by the developer before training.

.

Parameters to tune

Parameters in random forest are either to increase the predictive power of the model or make it easier to train the model.

Scikit-Learn implements a set of default hyper-parameters for all models, but these are not guaranteed to be optimal for a problem. The best hyper-parameters are usually impossible to determine ahead of time, and tuning a model is where machine learning turns from a science into trial and error based engineering.

Tuning these parameters relies more on experimental results than theory, and thus best method to determine the optimal settings is to try many different combinations evaluate the performance of each model.

max_features:

These are the maximum number of features Random Forest is allowed to try on individual tree. Multiple options are available in scikit learn implementation.

1. Auto/None: Here no restrictions are put. It will simply take features which makes sense in every tree.
2. sqrt: square root of the total number of features in individual run. “log2” is another similar type
3. 0.2: This allows to take 20% of the variables in individual run. We can assign and value in a format “0.x” where x% of the features will be considered.

n_estimators :
It represents the number of trees in random forest. Usually higher number of trees give you better performance but makes your code slower. You should choose as high value as your processor can handle because this makes your predictions stronger and more stable. Therefore we do a parameter search to find the sweet spot.

min_sample_leaf :
It is minimum number of samples required to be at a leaf node. This is similar to min_samples_splits, however, this describes the minimum number of samples of samples at the leafs. A smaller leaf makes the model more prone to capturing noise in train data. You should try multiple leaf sizes to find the most optimum for your use case.

max_depth :

max_depth represents the depth of each tree. The deeper the tree, the more splits it has and it captures more information about the data. With larger depth, the model oevrfits. The tree perfectly predicts all of the training data, however, it fails to generalize the findings for new data.

min_samples_split:

It represents the minimum number of samples required to split an internal node. This can vary between considering at least one sample at each node to considering all of the the samples at each node. When we increase this parameter, each tree in the forest becomes more constrained as it has to consider more samples at each node.

There are a few attributes which have a direct impact on model training speed.

n_jobs :

This parameter tells the engine how many processors is it allowed to use. A value of “-1” means there is no restriction whereas a value of “1” means it can only use one processor.

random_state:

This parameter makes a solution easy to reproduce our results. A definite value of random_state will always produce same results if given with same parameters and training data.

oob_score:

This is a random forest cross validation method. This method tags every observation used in different trees. Then it finds out a maximum vote score for every observation based on only tress which did not use this particular observation to train itself.

.

Advantages

- Can solve both type of problems, i.e., classification and regression
- Can handle large data set with higher dimensionality. It can handle thousands of input variables
- Can identify most significant variables so it is considered as one of the dimensionality reduction methods

Advantages contd.

- Can estimate missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing errors in data sets where data in imbalanced
- Can be used for clustering, data views and outlier detection.
- Very little of no preprocessing is involved.

Disadvantages

- Random forest is not as good as for regression problem as it does not give precise continuous nature predictions. In case of regression, it does not predict beyond the range in the training data, and that they may over-fit data sets that are particularly noisy.

- Random forest feels like a black box approach. You may have very little control on what the model does. What you can do best is try different parameters and random seeds.

- It is biased towards the categorical variable having multiple levels (categories). It is because feature selection based on impurity reduction is biased towards preferring variables with more categories so variable selection (importance) is not accurate for this type of data.

Random Forest Machine Learning Algorithm

Applications of Random Forest Machine Learning Algorithms

- Random Forest algorithms are used by banks to predict if a loan applicant is a likely high risk.
- They are used in the automobile industry to predict the failure or breakdown of a mechanical part.
- These algorithms are used in the healthcare industry to predict if a patient is likely to develop a chronic disease or not.
- Recently, the algorithm has also made way into predicting patterns in speech recognition software and classifying images and texts.

Random Forest Machine Learning Algorithm

https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm