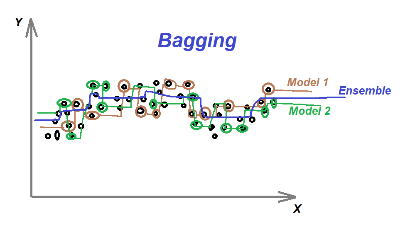
Random Forests

Why ensembles:

* Single decision trees often have high variability, which means a different sample would produce a very different decision tree.
* Therefore, we need to introduce the idea of combining multiple models (many decision trees) into a single “ensemble” model (a forest).

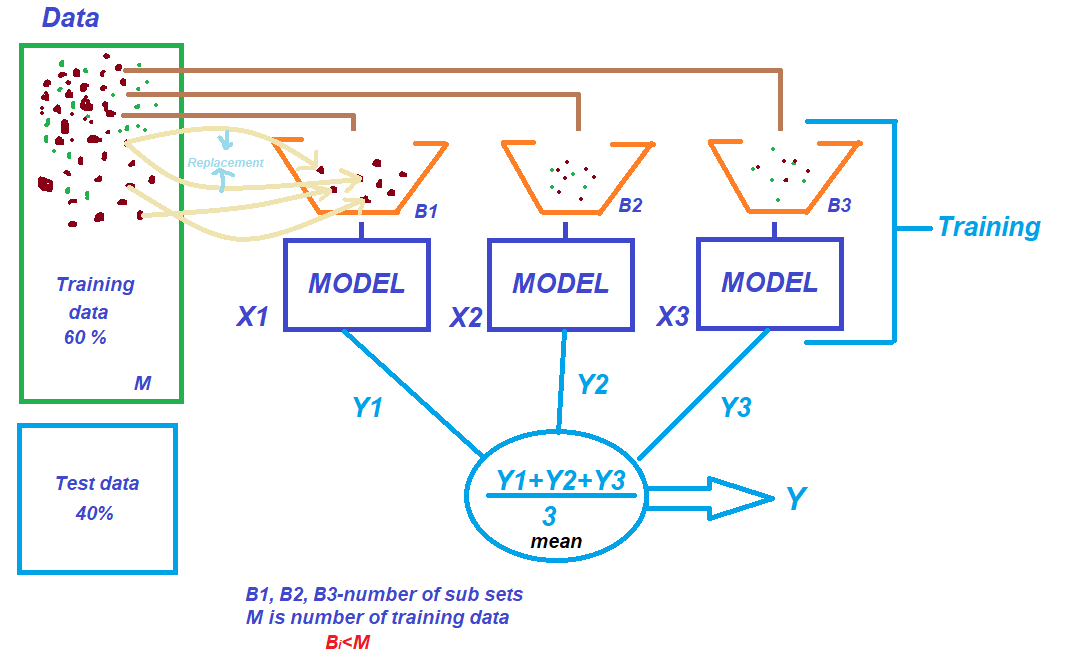
Some ensemble models:

* Bagging
* Random Forests
* Boosting

Bagging (bootstrap aggregating)

* Bagging is one of the ensemble methods aims to reduce variance, and thus avoid overfitting.
* Majority vote for classification and mean for regression.

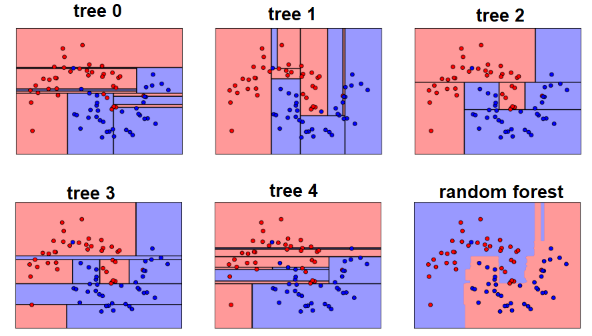
The algorithm(regression): (wud classification be different)

1. Choose randomly sub sets (bags) from training data with replacement.

* In general, we choose 60% of the data as our training set and 40% data as our test set (out-of-bag observations)
* The data size in each bag is the same as the full training dataset.

1. Build separate decision tree using each bag to get prediction MSE.
2. Average all prediction MSE to get a single prediction.

Random Forests:

* Combining multiple decision trees into a single ensemble model to grow a forest of trees. Therefore, we can get a less variable and more accurate result.
* Decorrelate the tree by considering only a subset of the predictor for each split so that some weak predictors will have more chance to be considered.
* Majority vote for classification and mean for regression.

The algorithm

1. Randomized bootstrapped copies

* Random, small set of predictors is chosen for consideration

1. Randomized predictor split

* Best split is found within the subset of predictors

1. Average all predictions

Variable Importance:

* Reflect predictive power of variables
* calculated based on OOB
* Two ways to measure for classification
* Accuracy-based importance: how much accuracy decreases when the variable is permuted

Step 1: Measure prediction accuracy on OOB sample on a single tree

Step 2: Keeping all other variables the same, randomly shuffle (permute) the values of each variable used in the tree

Step 3: Measure the decrease OOB prediction accuracy on shuffled data

Step 4: Calculate the mean decreases of all variables across all trees

* Gini-based importance: decrease of Gini impurity when a variable is chosen to split a node
* Use Gini impurity to decide which variable to split as each node
* Calculate the mean Gini decrease across all trees (when that variable is chosen to split a node)
* Variable importance for Regression
* Percentage increase in mean square error : calculated by shuffling the values of the out-of-bag samples
* Increase in node purity : based on the reduction in sum of squared errors whenever a variable is chosen to split

Code:

**randomForest() :** Can be used to perform both random forest and bagging.

**ntree:** Number of trees to grow. This number should not be set too small since we want every input row to get predicted at least few times.

**mtry:** How many predictors should be considered for each split of the tree.

* By default, randomForest() uses p/3 variables when building regression trees and square root of p when building classification trees.

**importance** : should importance of predictors be assessed?

**replace:** should sampling of observations be with replacement?

**cutoff**(classification only): the “winning” class for an observation is the one with the maximum ratio of proportion of votes to cutoff. (majority vote win)

**samplesize:** sizes of sample to draw. In classification, the elements of sampsize indicate the numbers to be drawn from each class.

