Bagging & Random Forests

Morning:

- Background
- Ensemble Motivations
- Bagging
- Random Forests

Afternoon:

- Tuning Params
- CVinRFakaOOB
- Feature Importance
- Proximity
- Interpretation

Background

14 Fun Facts About Trees

Trees are more than just part of our natural landscape. They provide shelter and food for wildlife, absorb carbon dioxide and produce breathable air, and add to the beauty of the world.

Below we've compiled some of our favorite tree facts, many of which you might be learning for the first time. Enjoy!

1. Trees are the longest living organisms on Earth, and never die of old age.

2. Trees drink about 2,000 liters of water each year.

3. Strategically planting trees and shrubs can save you up to 25 percent on your energy bills. Not only do they provide shade in the summer, but serve as a windbreak in the winter, too. (Get more information on smart landscaping.)

4. "Moon trees" were grown from seeds taken to the moon during the Apollo 14 mission in early 1971. NASA and USFS wanted to see if being in space or the moon's orbit caused the seeds to grow differently.

Cool picture of a tree

Background

Forest

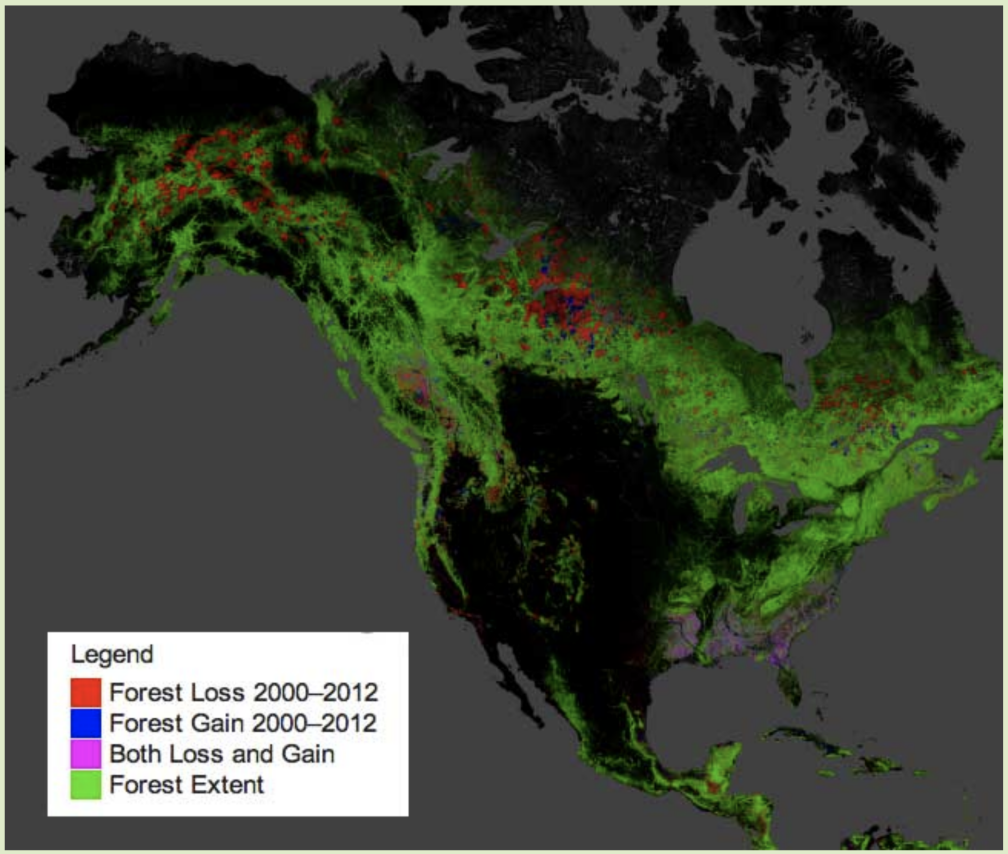
From Wikipedia, the free encyclopedia

This article is about a community of trees. For other uses, see Forest (disambiguation).
For a broader coverage related to this topic, see Plant community.

A **forest** is a large area dominated by trees.[1] Hundreds of more precise definitions of forest are used throughout the world, incorporating factors such as tree density, tree height, land use, legal standing and ecological function.[2][3][4] According to the widely used[5][6] Food and Agriculture Organization definition, forests covered four billion hectares (15 million square miles) or approximately 30 percent of the world's land area in 2006.[4]

Forests are the dominant terrestrial ecosystem of Earth, and are distributed across the globe.[7] Forests account for 75% of the gross primary productivity of the Earth's biosphere, and contain 80% of the Earth's plant biomass.[7]

Background

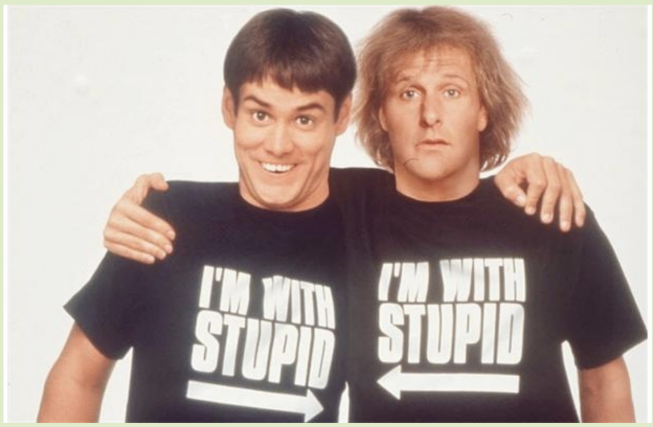

Legend
Forest Loss 2000–2012
Forest Gain 2000–2012
Both Loss and Gain
Forest Extent

Motivation: Pose a Question

Question:

Which would you choose to help you predict the price of a stock?

VS.



80% VS. 51%

Motivation: Pose a Question

Question:

Which would you choose to help you predict the price of a stock?

VS.

80% VS. ? %

Motivation: Pose a Question

Cumulative binomial probability refers to the probability that the value of a binomial random variable falls within a specified range.

Definition. The **probability mass function of a binomial random variable *X*** is:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

We denote the **binomial distribution** as $b(n, p)$. That is, we say: $X \sim b(n, p)$ where the tilde (~) is read "as distributed as," and n and p are called **parameters** of the distribution.

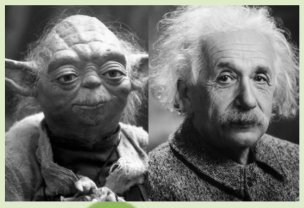
The **cumulative** mass/density function (or sometimes called distribution function) is

$$F(x) = Pr\{X \leq x\} = \sum_{k=1}^{x} \binom{n}{x} p^x (1-p)^{n-x}$$

Motivation: Pose a Question

Taking 2000 ‘dummies’ and pooling their opinions, you can beat a ‘genius’.

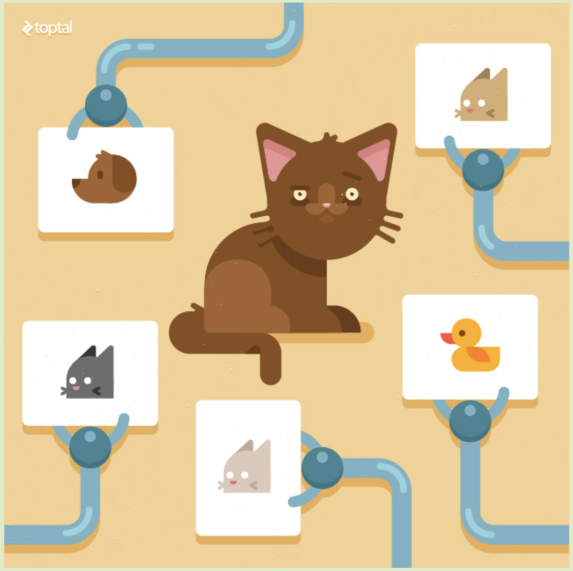
$$80\% \; < \; \binom{2000}{1001}(.51)^{1001}(.49)^{999} + \binom{2000}{1002}(.51)^{1002}(.49)^{998} + \ldots$$



How many ‘dummies’ do we need to get at least 99% accuracy?

Ensemble

By pooling the predictions of multiple weak learners, we can create a strong classifier. The overall prediction is much better than any individual could do on its own.



Ensemble

So, an ensemble model is really just a collection of prediction models, where each model gets a voice in the final prediction.

Weak Learner

A weak learner is defined as a predictor (classifier) whose accuracy is just barely better than chance.

Which of these are weak learners?

20% 50% 90% 81% 51% 34% 49%

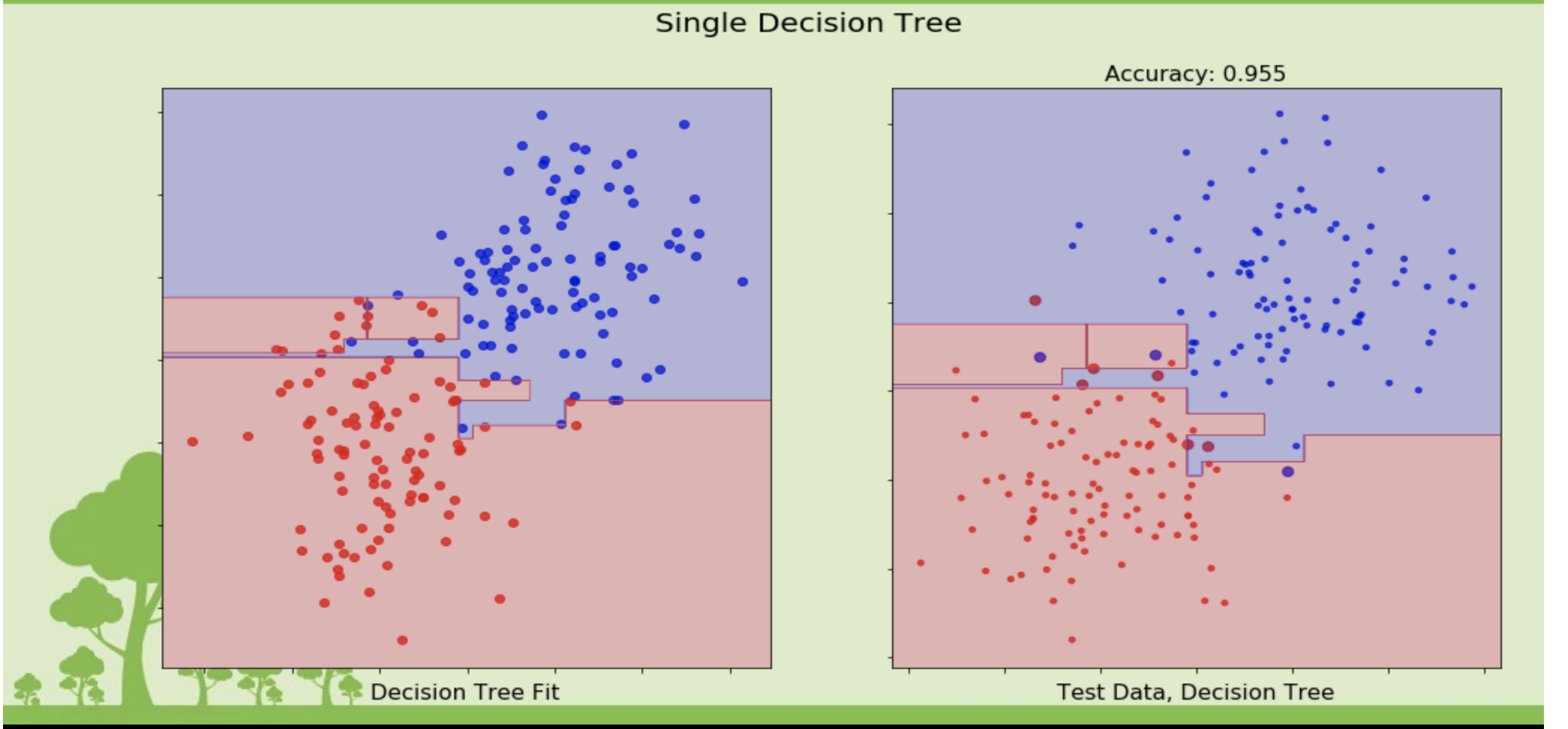
Wisdom of The Crowd

What makes a crowd wise?

1) Diversity of Opinion - range of opinions
2) Independence - each prediction not influenced by other opinions
3) Decentralization - specializations / local knowledge / expertise
4) Aggregation - mechanism to aggregate all predictions

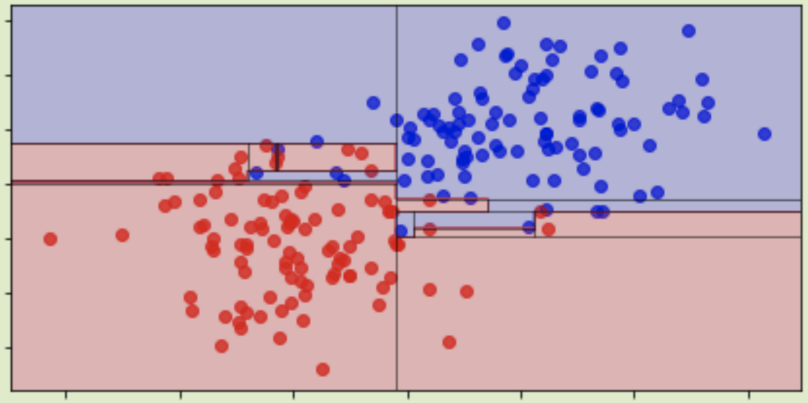
Individual Predictor

Lets use decision trees as our individual learners.

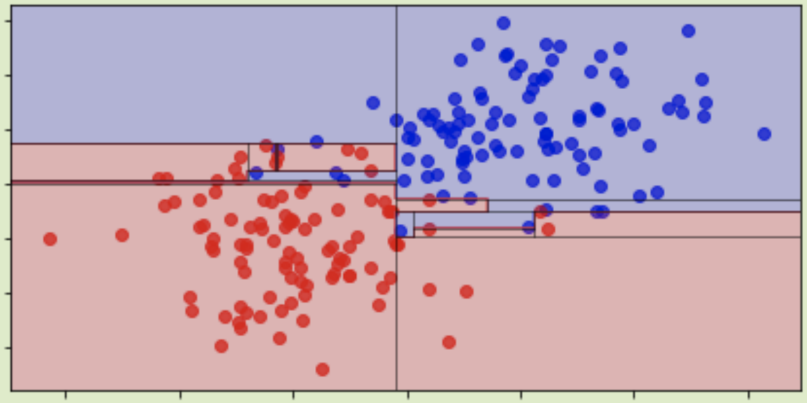


Ensemble Predictor

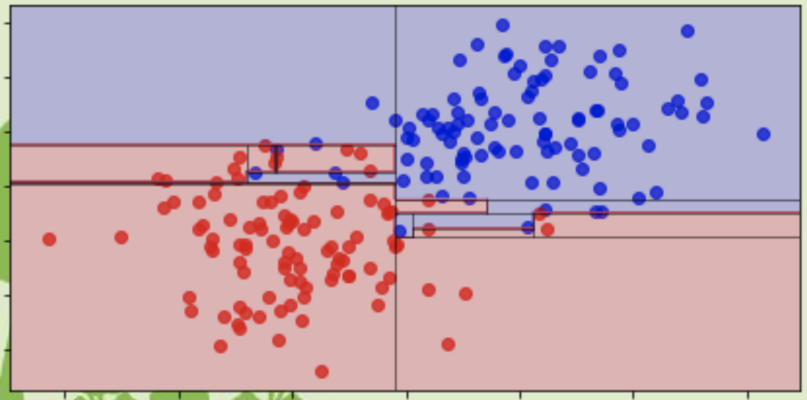
Train several models and take a majority vote.



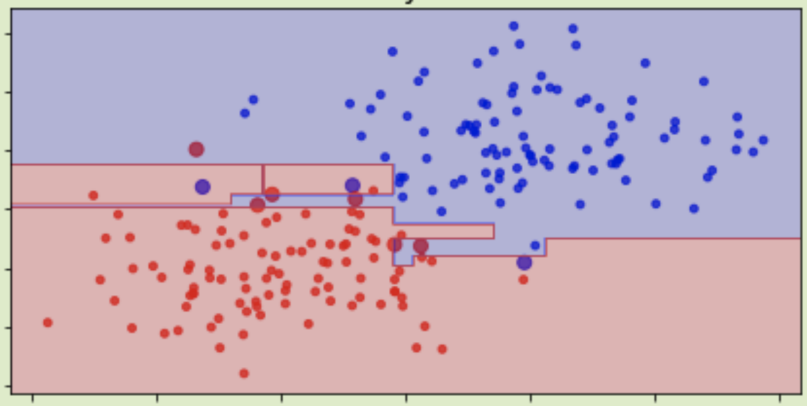
Same Tree



Accuracy: 0.955



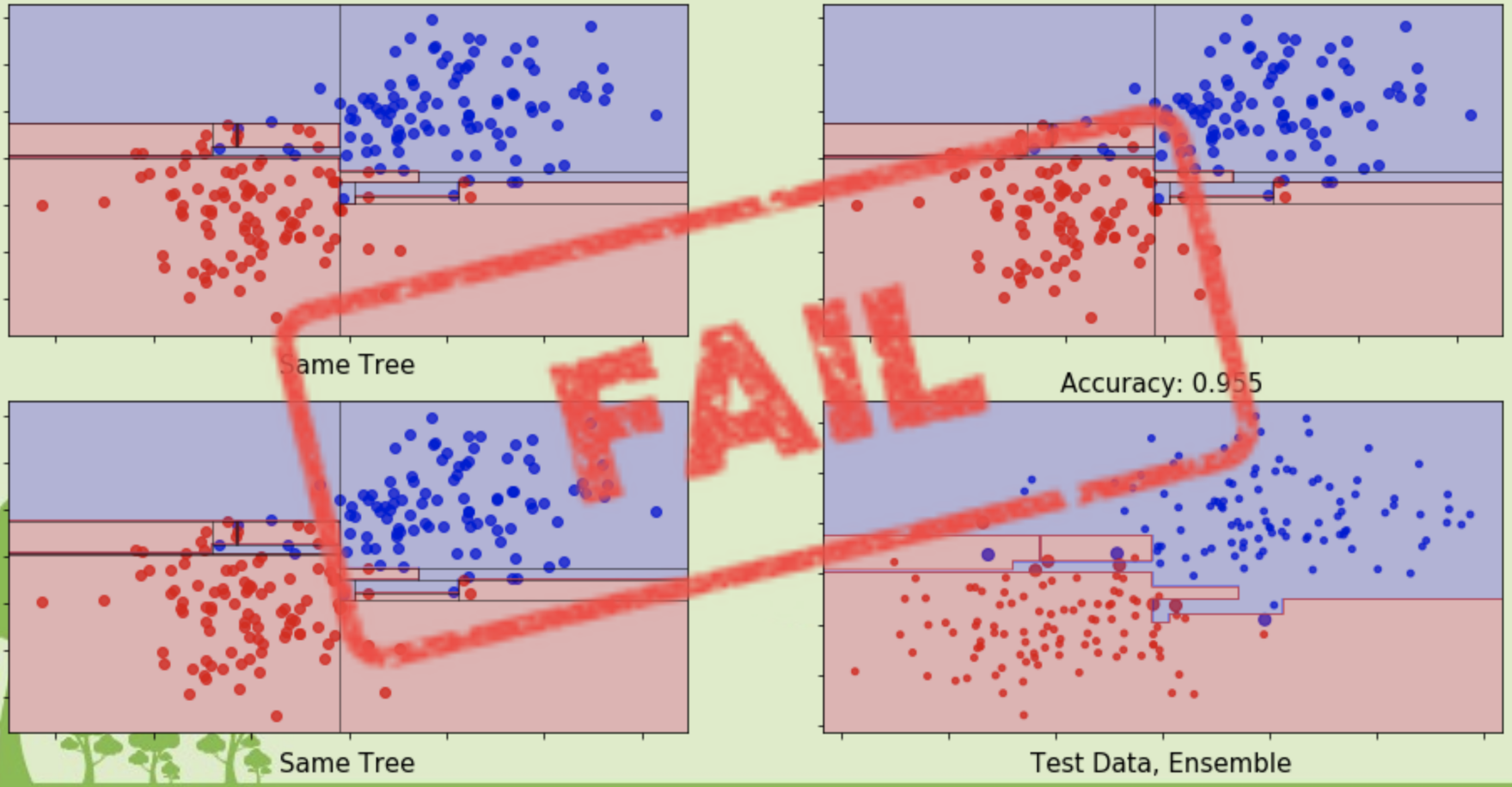
Same Tree



Test Data, Ensemble

Ensemble Predictor

Train several models and take a majority vote.



Ensemble Predictor

How to make train ‘different’ models?

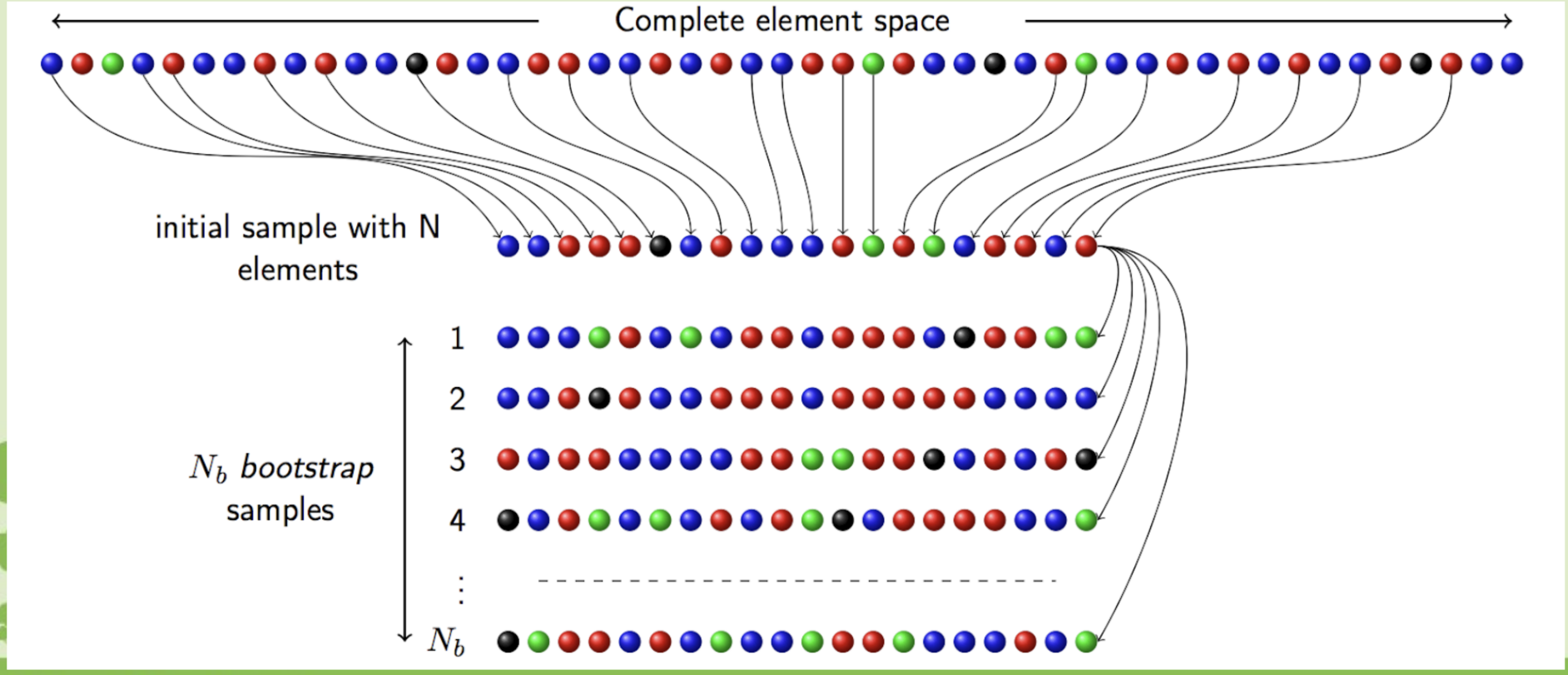
We just need to go back to our population and get new samples. For each one of these samples we can build a new predictor (decision tree). Now we will have different decision trees and we will be able to get these great WOTC results.

In Texas, these are what we call shoes!

These here are called laces!
But, y’all call these bootstraps!

Bootstrapping
Come again?

Bootstrapping is just the process of generating more samples (same size) by sampling with replacement from our original sample.



BootstrapAggregating - Bagging
ShoelaceAggregating - Shagging

Now lets try our ensemble method again. This time, each individual predictor (tree) will be built using a new bootstrapped sample.

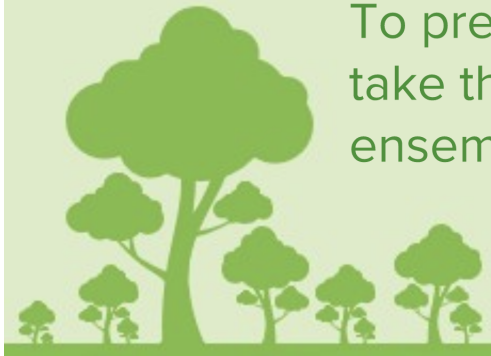
Bagged Decision Trees

Train many decision trees separate from each other each by this process:

1. Take a bootstrap sample of your data
2. Train a single decision tree on the bootstrap sample
3. Do step 1 & 2 for each tree in your ensemble

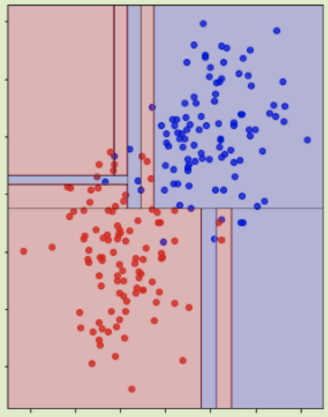
(Now the model is built)

To predict: we make a prediction with each of the above trees and then take the average (regression) or majority vote (classification) as our ensemble prediction.

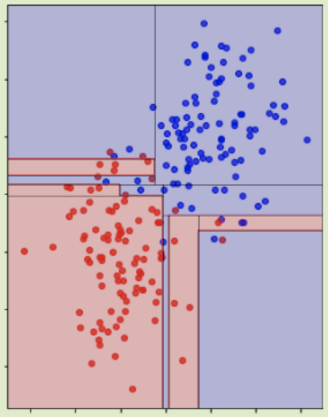


BootstrapAggregating - Bagging

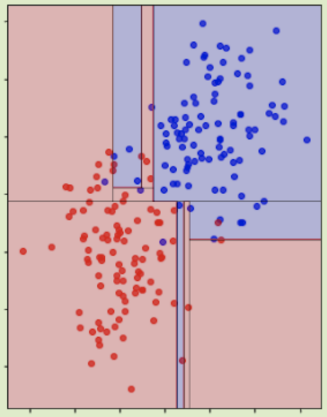
6 Tree Ensemble



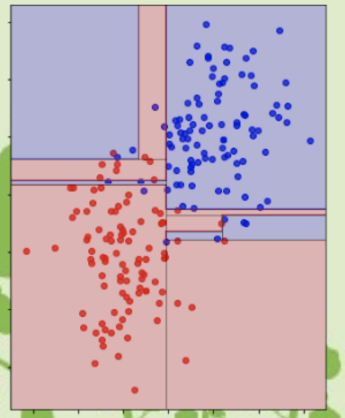
Bagged Tree



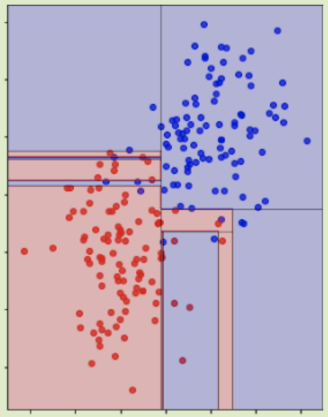
Bagged Tree



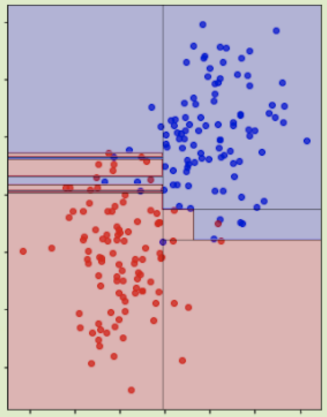
Bagged Tree



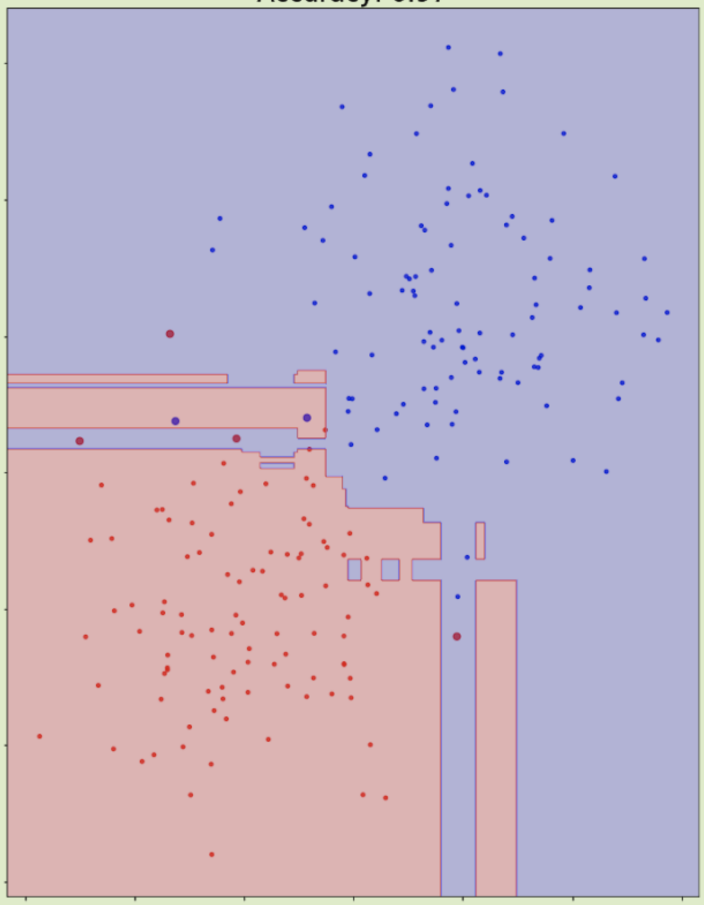
Bagged Tree



Bagged Tree



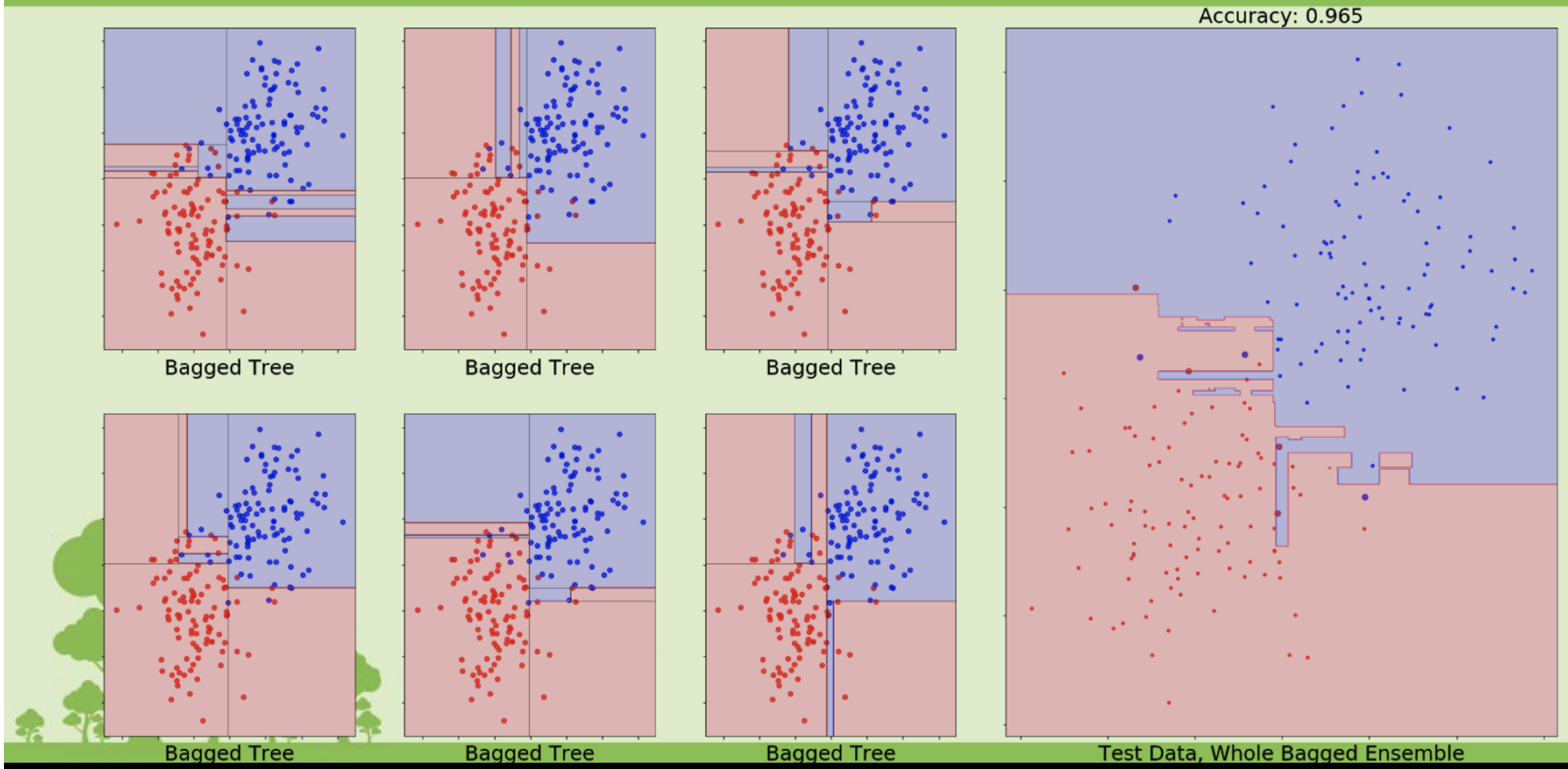
Bagged Tree



Test Data, Whole Bagged Ensemble

BootstrapAggregating - Bagging

1000 Tree Ensemble



BootstrapAggregating - Bagging

Benefits

1. Bootstrapped trees provide unbiased, high variance predictors
2. Averaged estimators are lower variance than single predictors
3. Averaged predictors are still unbiased
4. Bootstrapping explores the dataset, allowing for trees to have different opinions / areas of expertise

Next Quest: Decorate the Trees!
(Decorrelate)

Why are trees correlated in the first place?

- Bootstrapped samples tend to be about the same (sorry, you're SOL on this)
- The trees tend to have the same splits (the most influential features tend to be chosen first) (Let attack this problem!)
- The apple doesn't fall far from the tree.

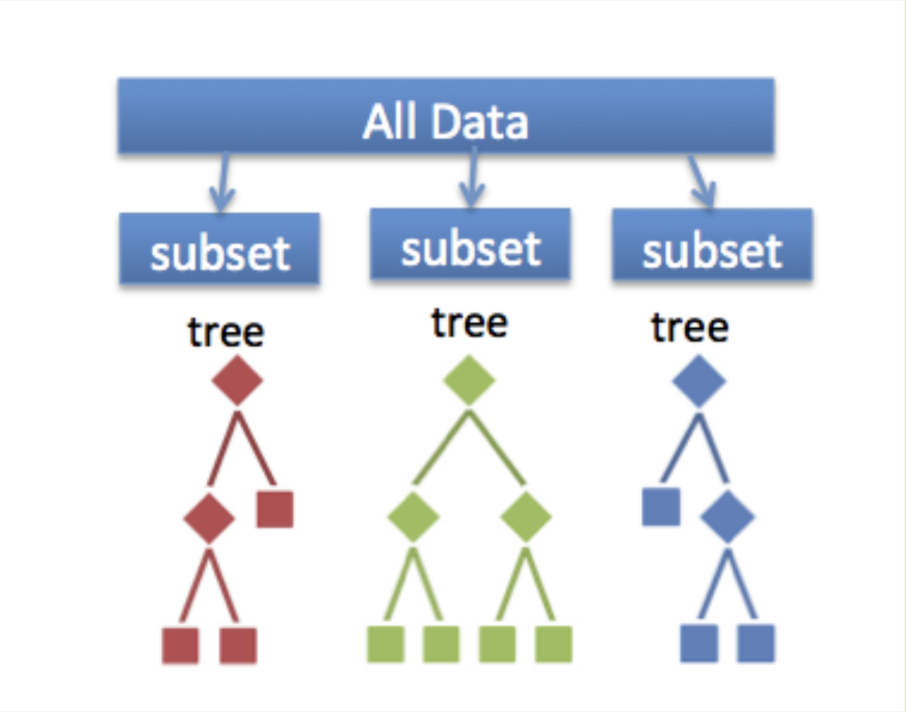
Next Quest: Decorate the Trees!

(Decorrelate)

- Lets take away this free reign that the trees have been given, to choose whatever the best possible feature is in the set of all features.
- We can do this by restricting the number of features the trees can split on.
- We will randomly select k features at each split for the tree to consider.

Random Forest

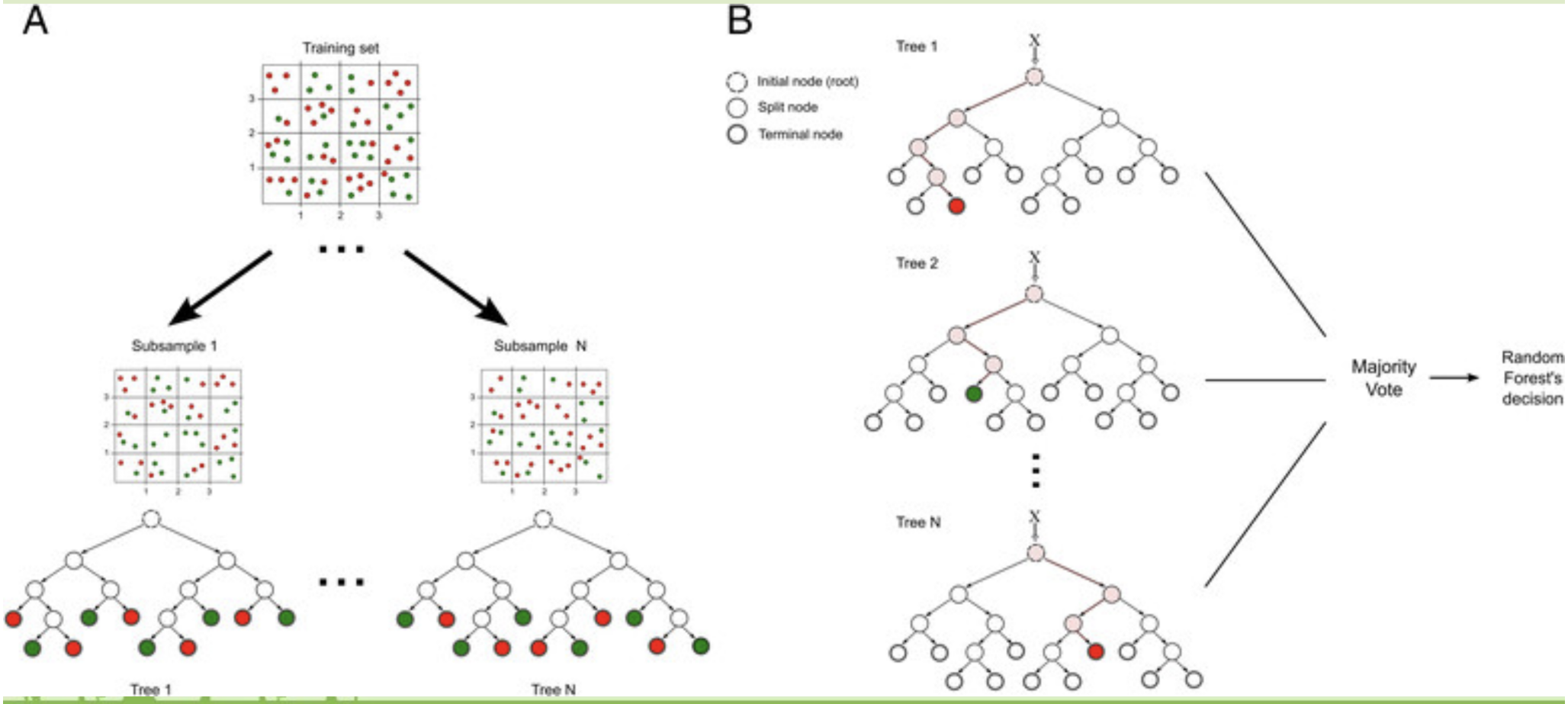
Gnarly Looking Trees



Lets visually look at these decorrelated trees!

Random Forest

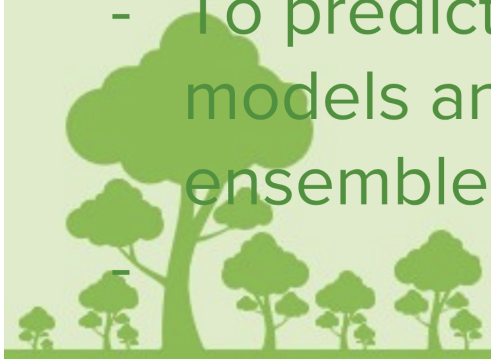
Process



Random Forest

Algorithm

- Train many decision trees separate from each other each by this process:
- 1. Take a bootstrap sample of your training data
- 2. Train a single decision tree to the bootstrap sample by the regular process except that: at each node only consider a subset of features to split on.
- To predict we make a prediction with each of the above models and then take the average of that as our ensemble prediction
-



Random Forest

Afternoon:

- Tuning Params
- CVinRFakaOOB
- Feature Importance
- Proximity
- Interpretation

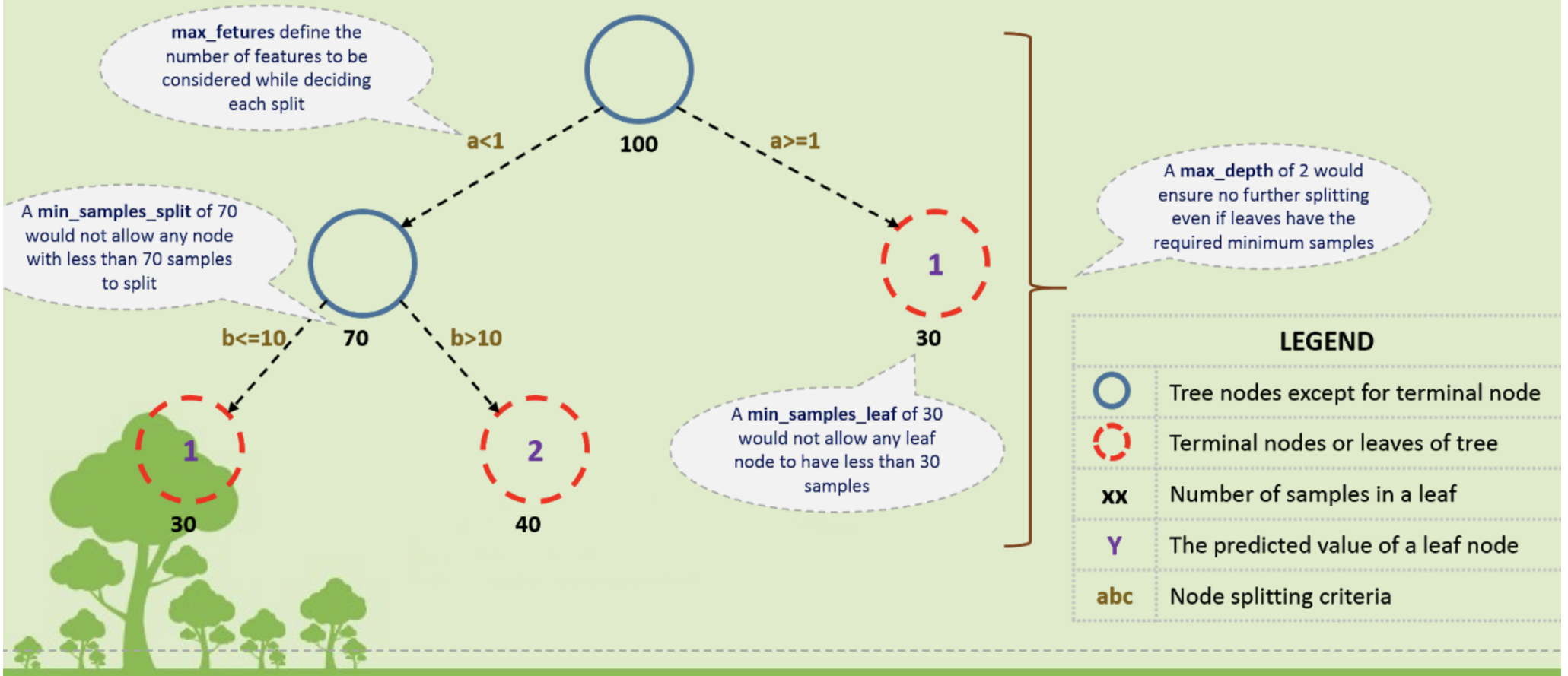
Random Forest

Tuning

- number of features considered at each split
- number of trees (i.e., number of bootstrapped samples)
- sample size of each bootstrapped sample (maybe)
- tree characteristics
  - min samples split
  - min samples leaf
  - max depth

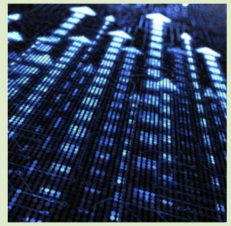
Random Forest

Tuning



Random Forest
Cost / Speed-Up

- Individual trees are expensive to grow
  - Do the trees depend on one another in any way?
  - Nope, so we can grow them in parallel
    - Specify the number of CPUs to use



n_jobs : integer, optional (default=1)

The number of jobs to run in parallel for both fit and predict. If -1, then the number of jobs is set to the number of cores.

Random Forest
Creative Validation

- Typically we use CV to test a model
- How did we construct a single tree
- How about we use the samples not included in our current tree construction to test that tree?
- When the number of bootstrapped samples is large, this approximates the LOO test error estimation.
- So we don’t need to split our data.
    - CVinRFakaOOB

Random Forest

OOB - Out Of Bag Error

OOB is the mean prediction error on each training sample x_i, using only the trees that did not have x_i in their bootstrap sample.

- Build all the trees
- Keep track of the data points left out of each tree.
- Run these data points down the tree after it was built
- Keep track of the predictions on these points.
- Average the predictions for each point
- Average the accuracy for all points

Random Forest

Interpretation: Feature Importance

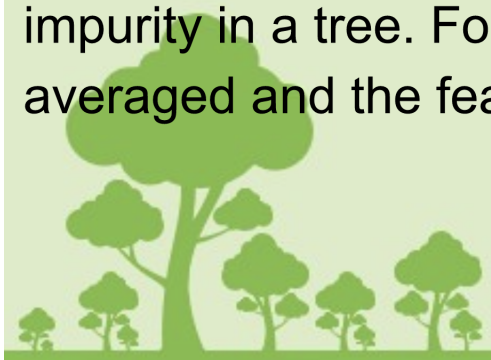
- What can we get out of this forest we have grown?
  - In bagged trees we were always choosing the most influential feature first.
  - Now we are restricting the features but we can still determine which features are being used more than others.

Random Forest

Interpretation: Feature Importance

Mean decrease impurity

Random forest consists of a number of decision trees. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the (locally) optimal condition is chosen is called impurity. For classification, it is typically either Gini impurity or information gain/entropy and for regression trees it is variance. Thus when training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are rankeD according to this measure.



Random Forest

Interpretation: Feature Importance

Mean decrease impurity

1. you initialize an array feature_importances of all zeros with size n_features.
2. you traverse the tree: for each internal node that splits on feature i you compute the error reduction of that node multiplied by the number of samples that were routed to the node and add this quantity to feature_importances[i].

Random Forest

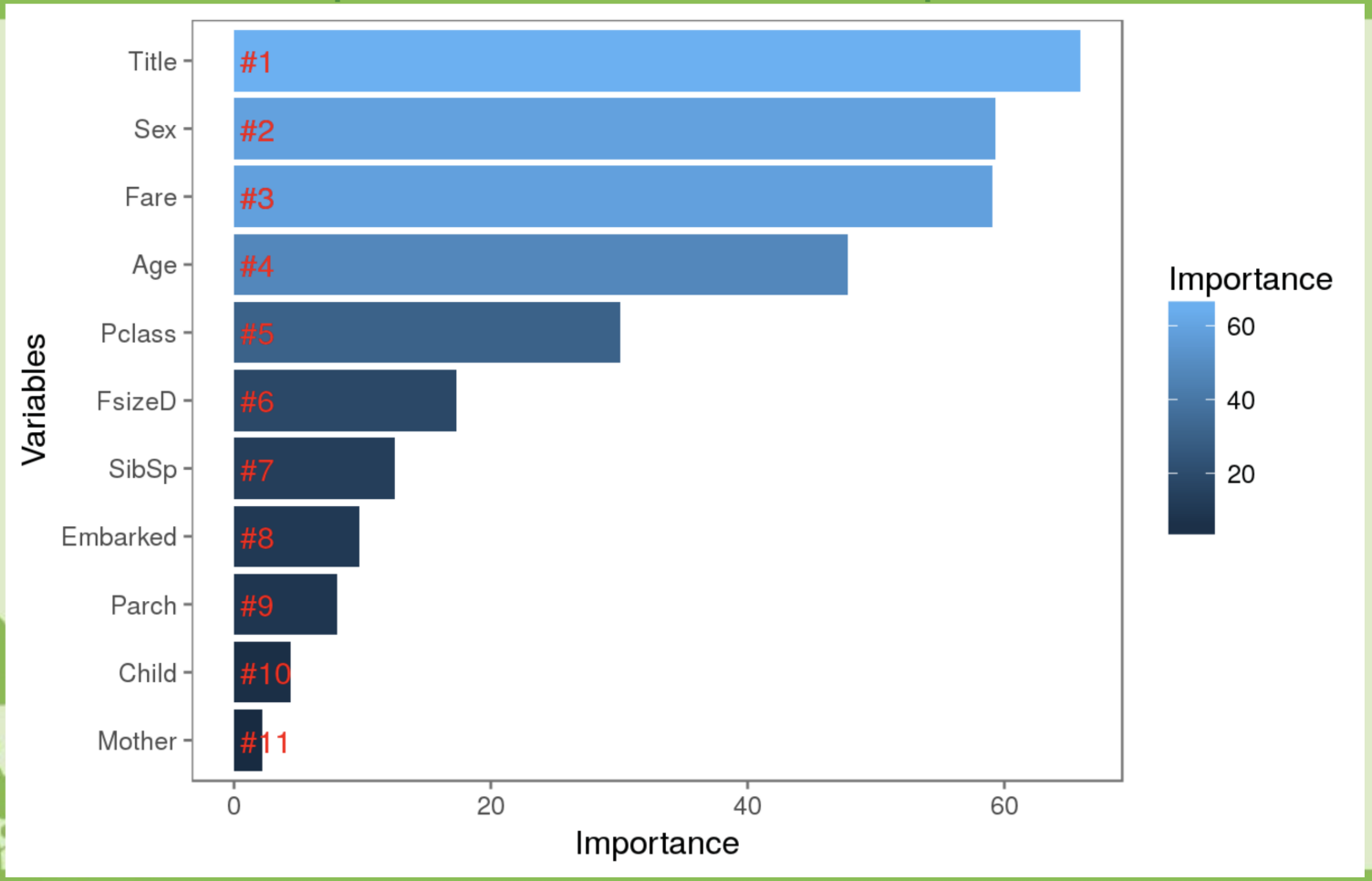
Interpretation: Feature Importance

Mean decrease accuracy

Another popular feature selection method is to directly measure the impact of each feature on accuracy of the model. The general idea is to permute the values of each feature and measure how much the permutation decreases the accuracy of the model. Clearly, for unimportant variables, the permutation should have little to no effect on model accuracy, while permuting important variables should significantly decrease it.

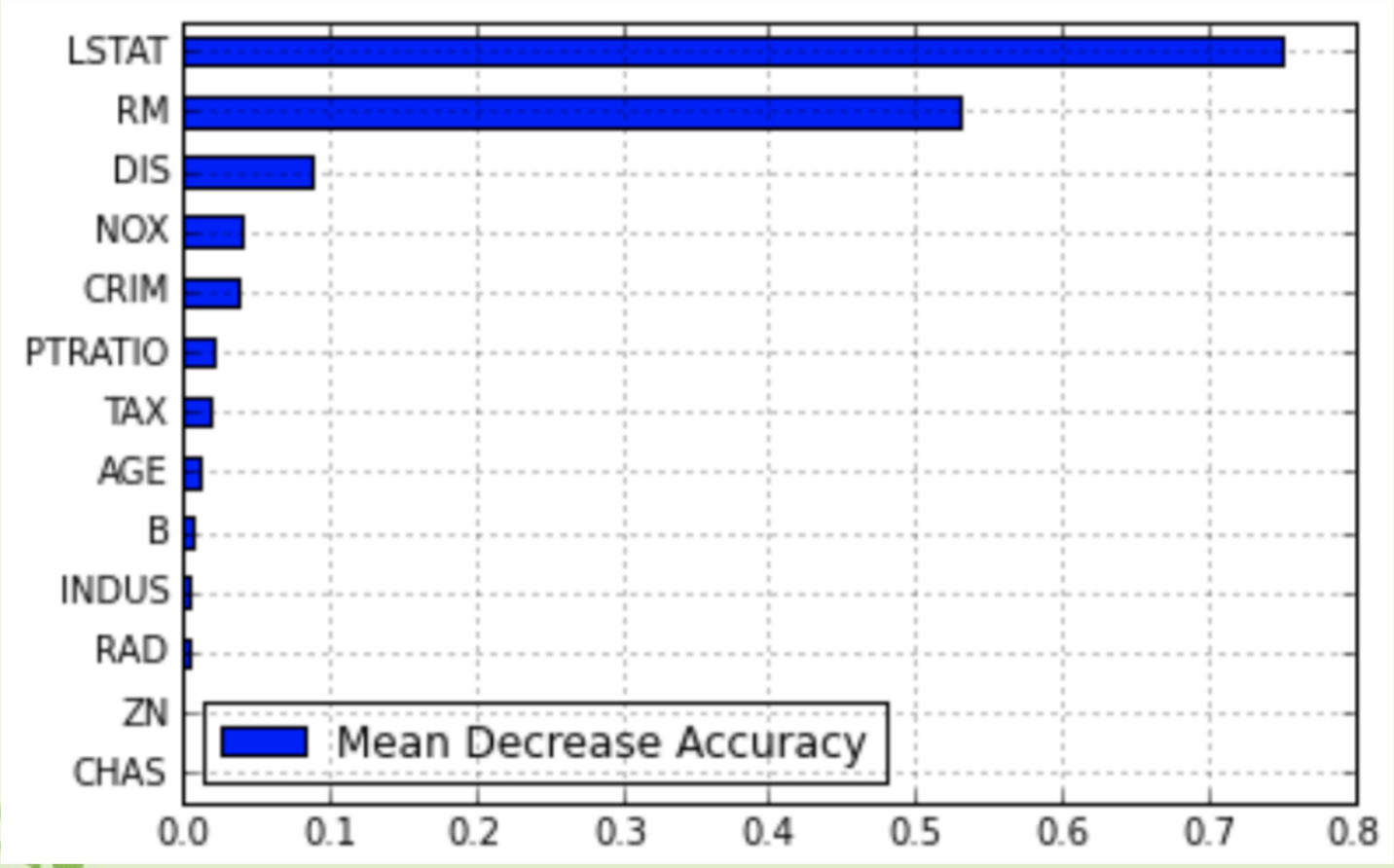
Random Forest

Interpretation: Feature Importance



Random Forest

Interpretation: Feature Importance



Random Forest

Interpretation: Proximity

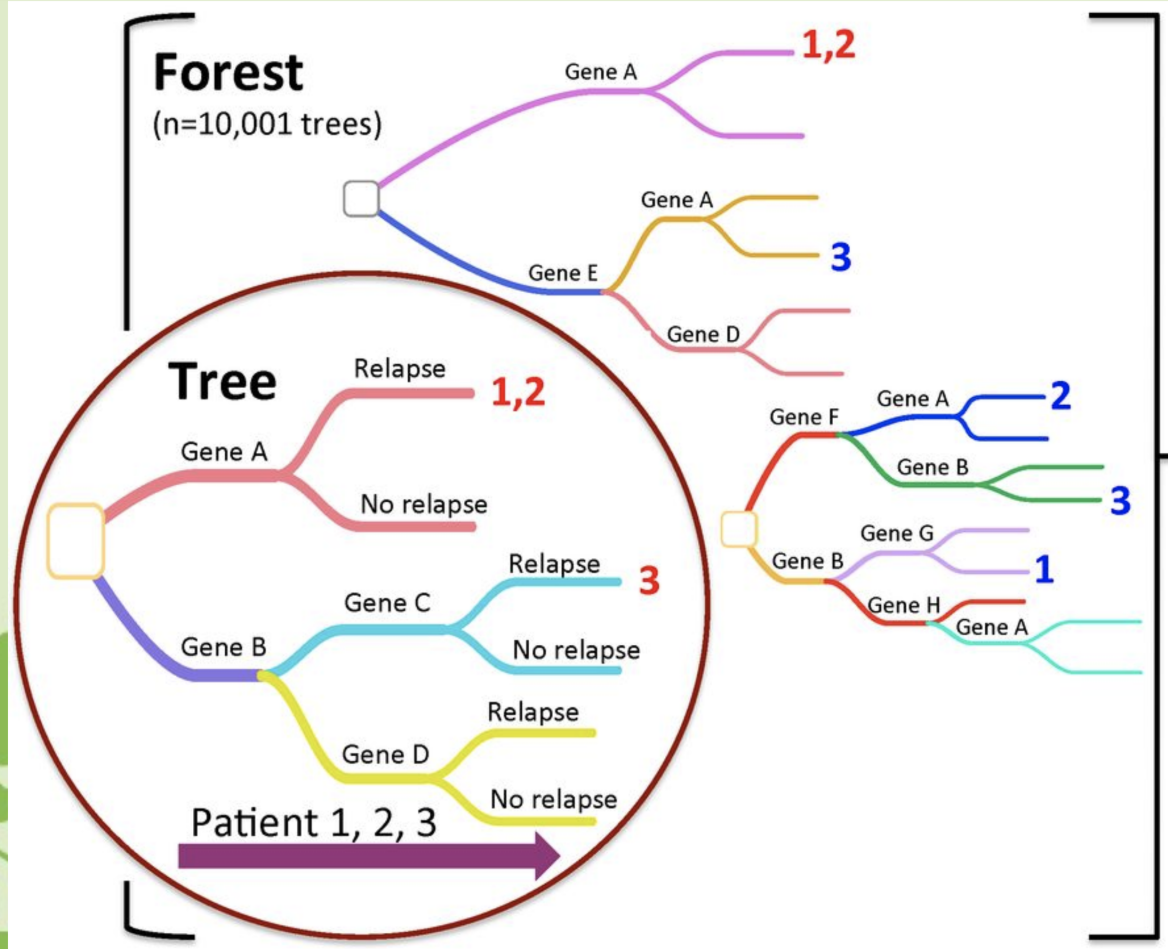
Proximity is the closeness or nearness between pairs of points

After a tree is grown, put all of the data, both training and oob, down the tree. If cases i and j are in the same terminal node increase their proximity by one. At the end, normalize the proximities by dividing by the number of trees.

Proximities can be used in replacing missing data, locating outliers, and producing illuminating low-dimensional views of the data.

Random Forest

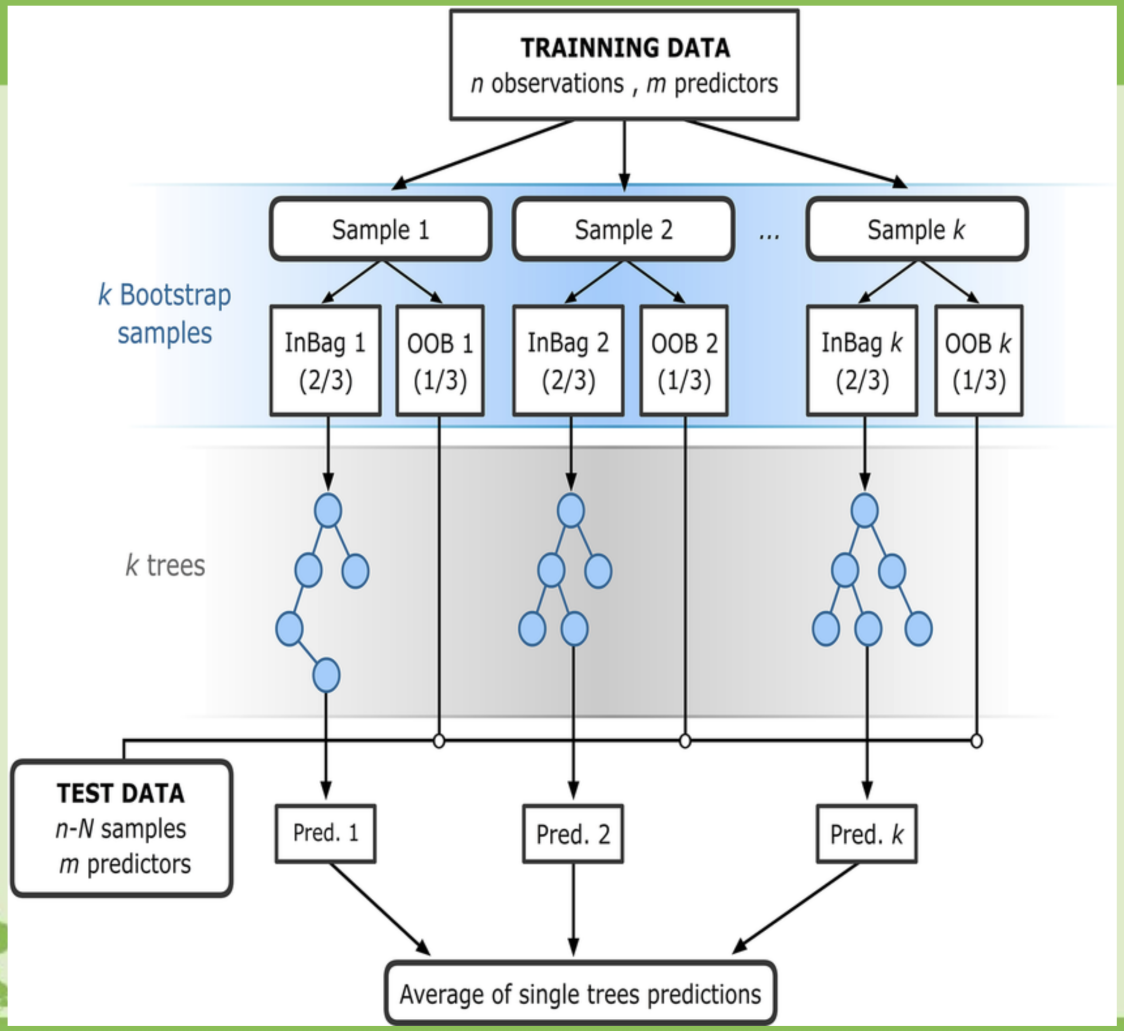
Interpretation: Feature Importance



- The more often a gene is chosen as a splitter variable, the higher its "Variable Importance" – This can be used to prioritize which genes to select for an assay with limited gene measurements

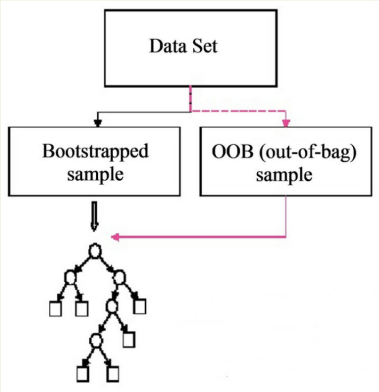
Gene	Var. Imp.
Gene A	0.67
Gene B	0.20
Gene D	0.13
…	…

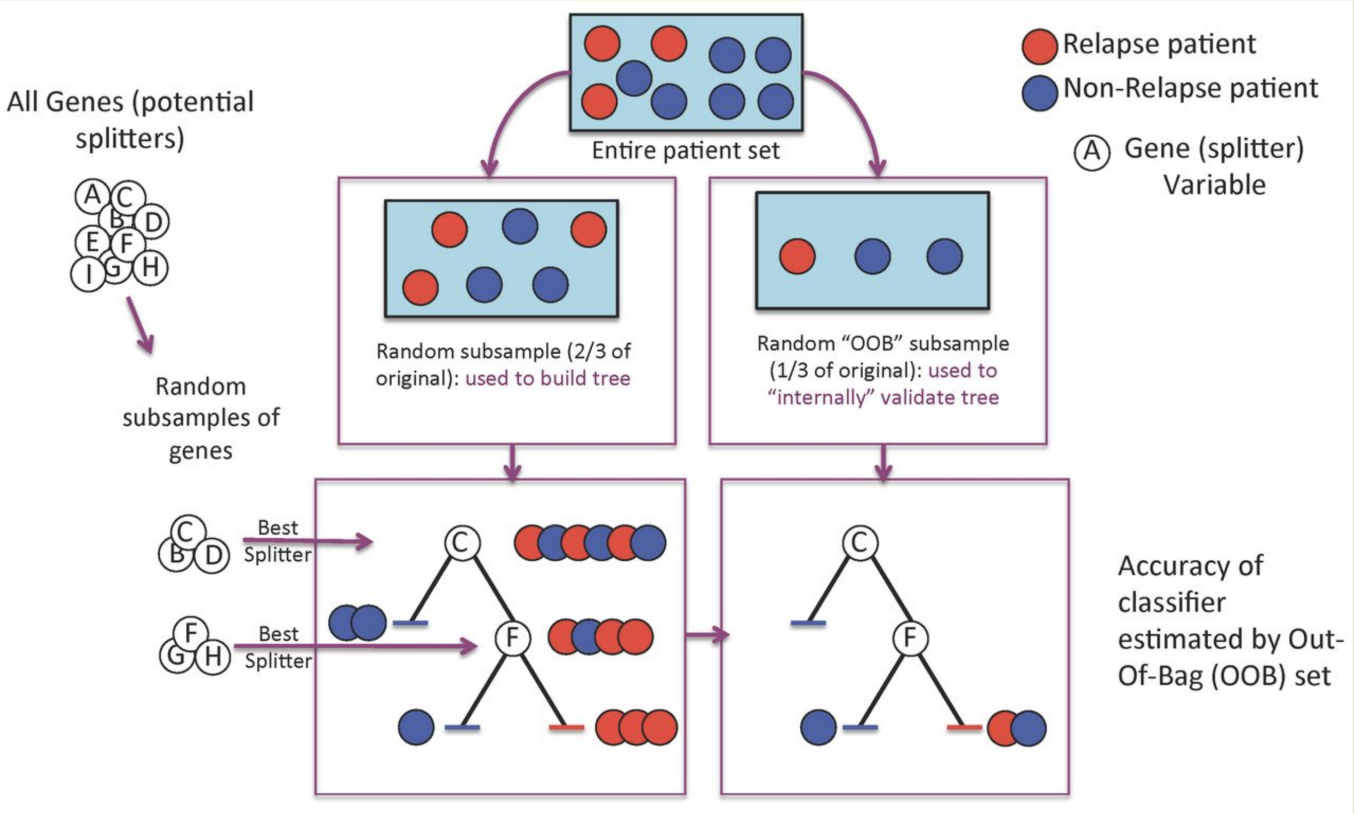
Random Forest



Random Forest

Interpretation: Feature Importance





Notes

Only produces good results for high variance, low bias classifiers. (DTs are great!)