RANDOM FORESTS

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

https://web.csulb.edu/~tebert/teaching/lectures/551/random_forest.pdf

http://www.math.usu.edu/adele/RandomForests/Ovronnaz.pdf

https://www.princeton.edu/~aylinc/files/CS613-15-04.pdf

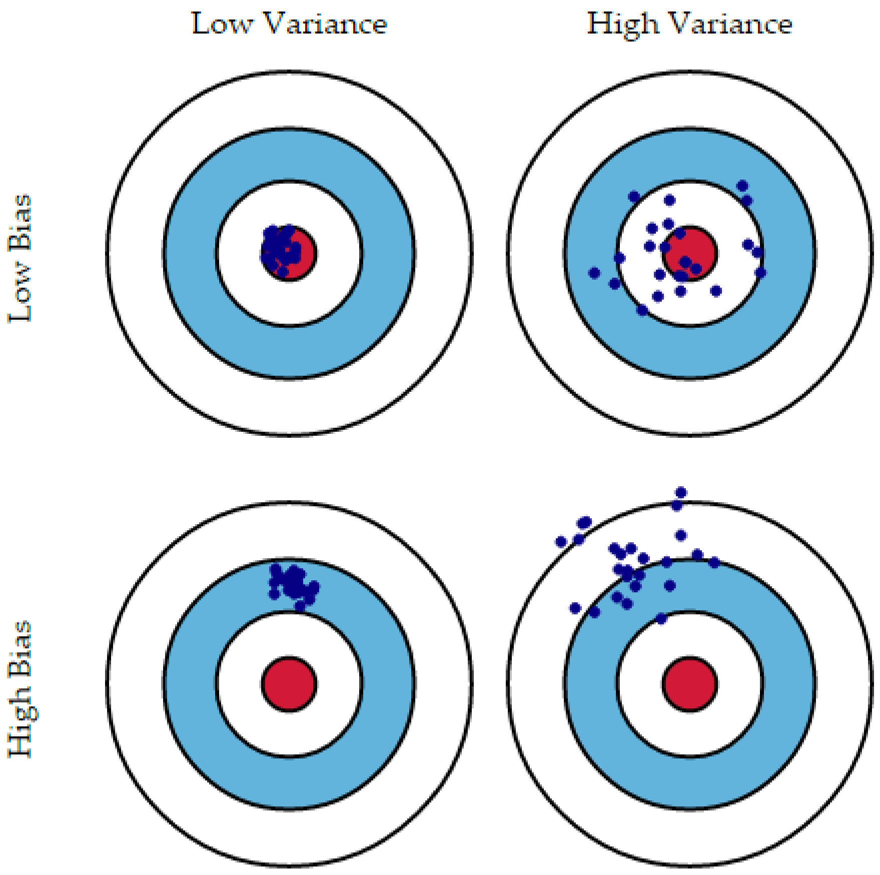
Ensemble methods

- A single decision tree does not perform well - But it is super fast
- What if we learn multiple trees?
- It combines multiple algorithms to obtain better predictive performance than the one from a single model.
- There is no predefined number of models to consider, and some business goals may require more models than others.
- We need to make sure they do not all just learn the same

Model Error and Reducing this Error with Ensembles

- The error emerging from any machine model can be broken down into three components mathematically:

Bias + Variance + Irreducible error



Bagging

- If we split the data in random different ways, decision trees give different results, high variance.
- **Bagging:** **B**ootstrap **agg**regat**ing** is a method that result in low variance.
- If we had multiple realizations of the data (or multiple samples), we could calculate the predictions multiple times and take the average of the fact that averaging multiple onerous estimations produce less uncertain results

Bagging

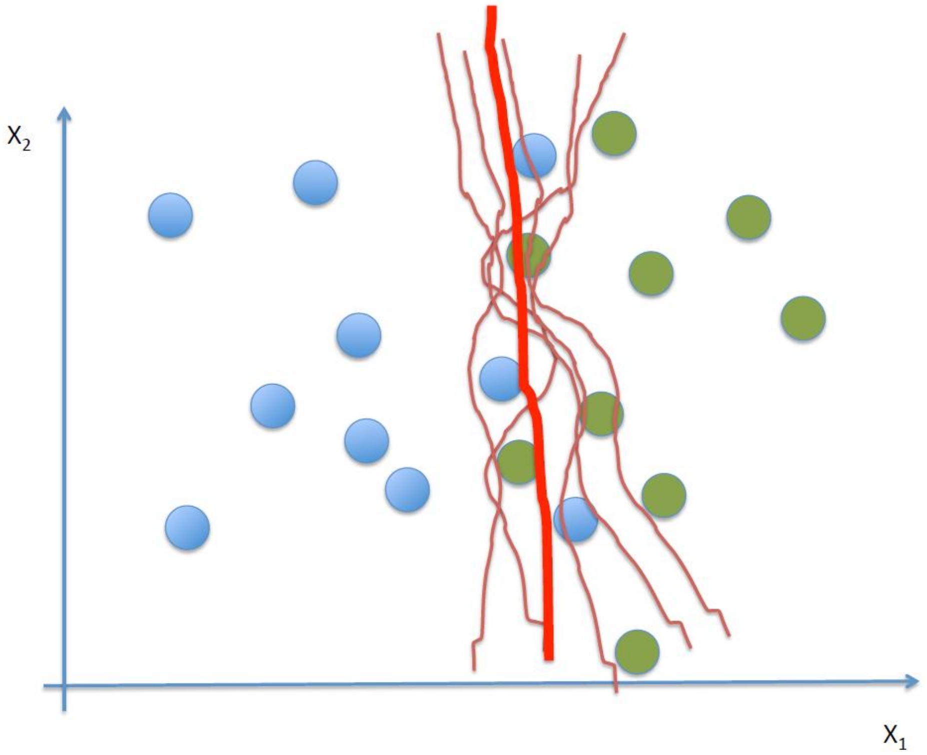
- Say for each sample *b*, we calculate $f^b(x)$, then:

$$\hat{f}_{ave}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_x^b$$

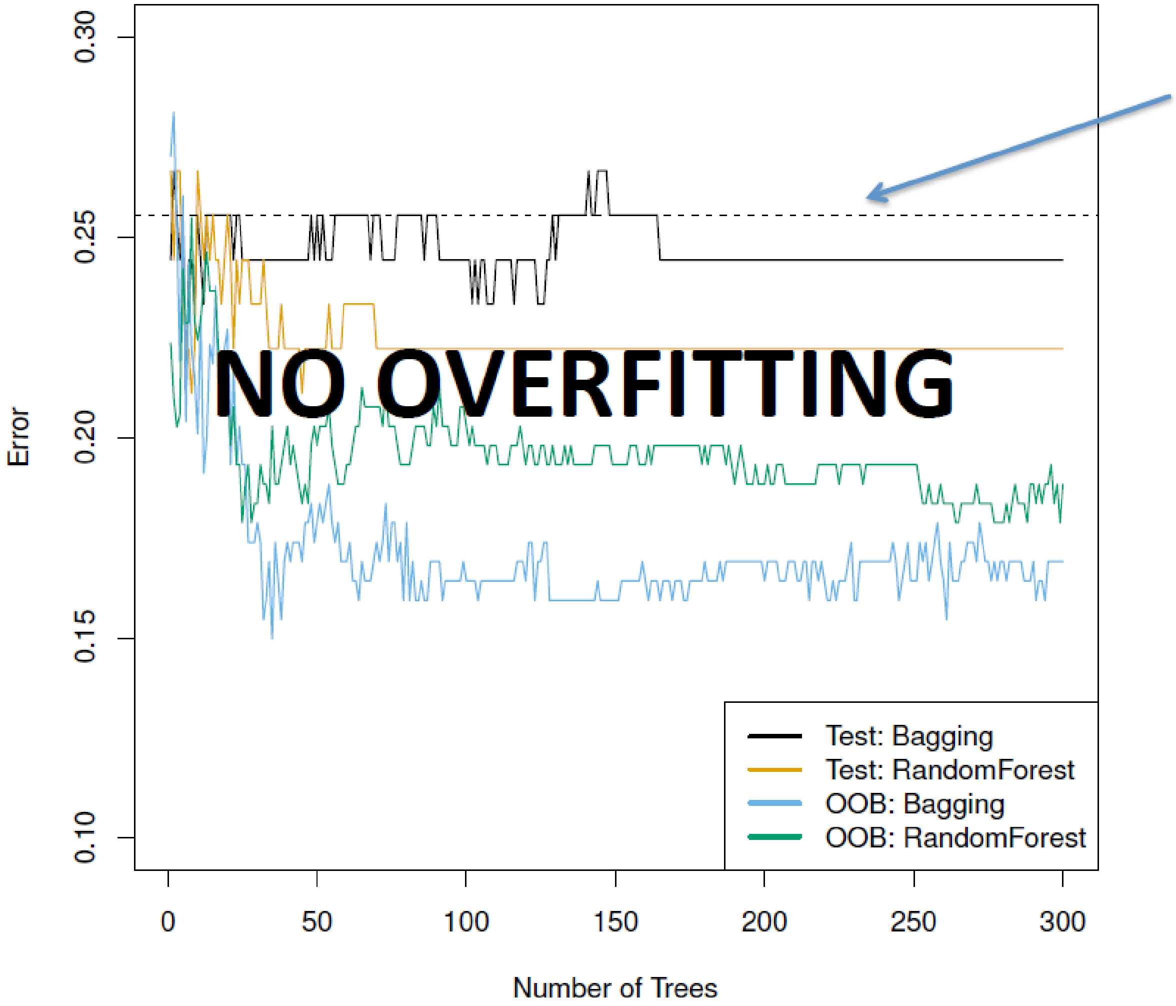
How? → Bootstrap

Bootstrap

- Construct B (hundreds) of trees (no pruning)
- Learn a classifier for each bootstrap sample and average them
- Very effective



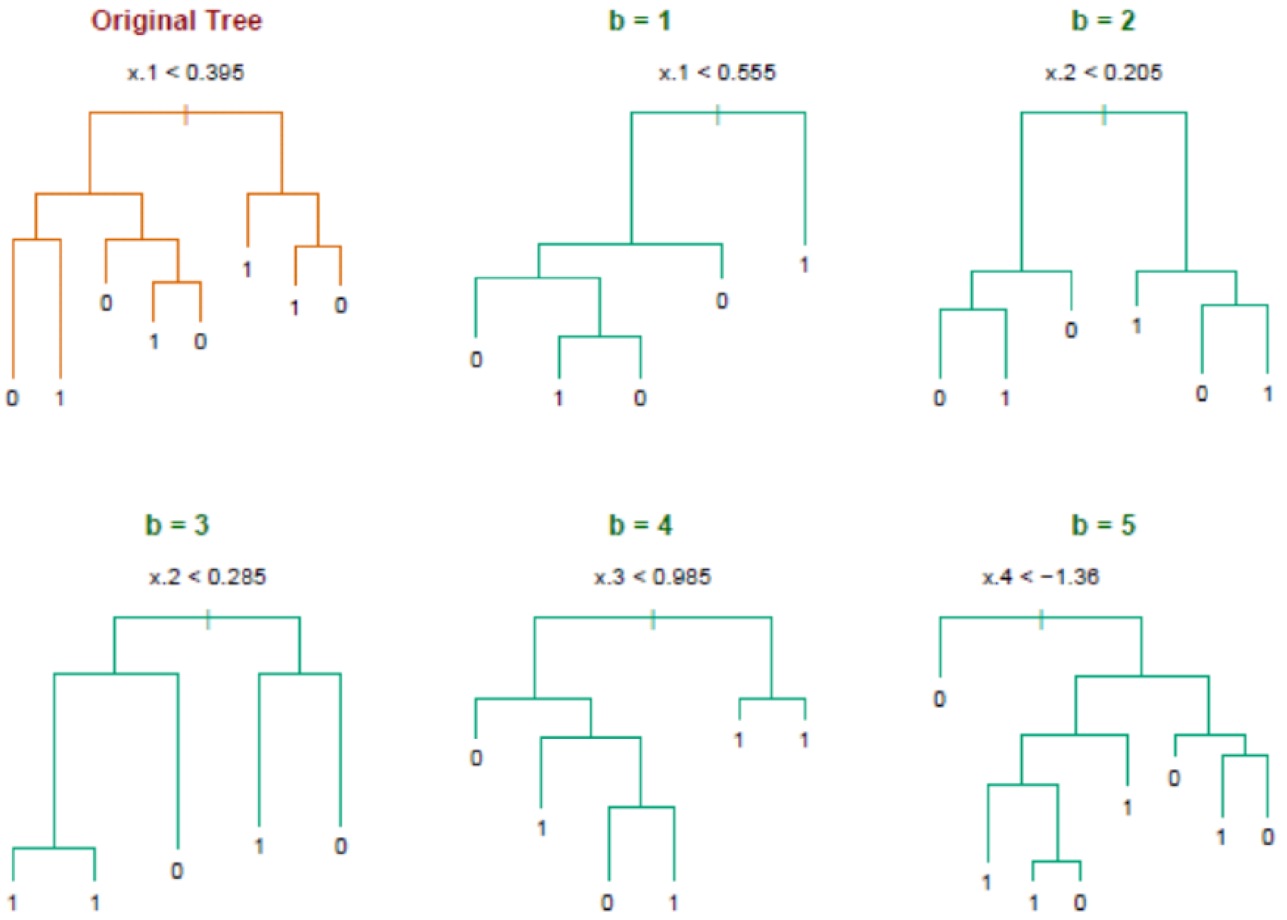
Bagging for classification: Majority vote



NO OVERFITTING

Test error

Bagging decision trees



Hastie et al.,"The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer (2009)

Out-of-Bag Error Estimation

- No cross-validation?
- Remember, in bootstrapping, we sample with replacement, and therefore, not all observations are used for each bootstrap sample. On average, 1/3 of them are not used!
- We call them out-of-bag samples (OOB)
- We can predict the response for the i-th observation using each of the trees in which that observation was OOB and do this for n observations
- Calculate overall OOB MSE or classification error

Bagging

- Reduces overfitting (variance)
- Normally uses one type of classifier
- Decision trees are popular
- Easy to parallelize

Variable Importance Measures

- Bagging results in improved accuracy over prediction using a single tree
- Unfortunately, difficult to interpret the resulting model. Bagging improves prediction accuracy at the expense of interpretability.
- Calculate the total amount that the RSS or Gini index is decreased due to splits over a given predictor, averaged over all B trees.

Bagging

- Each tree is identically distributed (i.d.)

$\rightarrow$ The expectation of the average of B such trees is the same as the expectation of any one of them

$\rightarrow$ The bias of bagged trees is the same as that of the individual trees

- i.d. and not i.i.d

Bagging

- An average of B i.i.d. random variables, each with variance σ^2, has variance: σ^2/B
- If i.d. (identical but not independent) and pair correlation ρ is present, then the variance is:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- As B increases the second term disappears but the first term remains

Why does bagging generate correlated trees?

- Suppose that there is one very strong predictor in the data set, along with a number of other moderately strong predictors.
- Then, all bagged trees will select the strong predictor at the top of the tree, and therefore, all trees will look similar.
- How do we avoid this?
- What if we consider only a subset of the predictors at each split?
- We will still get correlated trees unless …. we randomly select the subset!

Random Forest, Ensemble Model

- The random forest (Breiman, 2001) is an ensemble approach that can also be thought of as a form of the nearest neighbour predictor.
- Ensembles are a divide-and-conquer approach used to improve performance. The main principle behind ensemble methods is that a group of “weak learners” can form a “strong learner”.

Trees and Forests

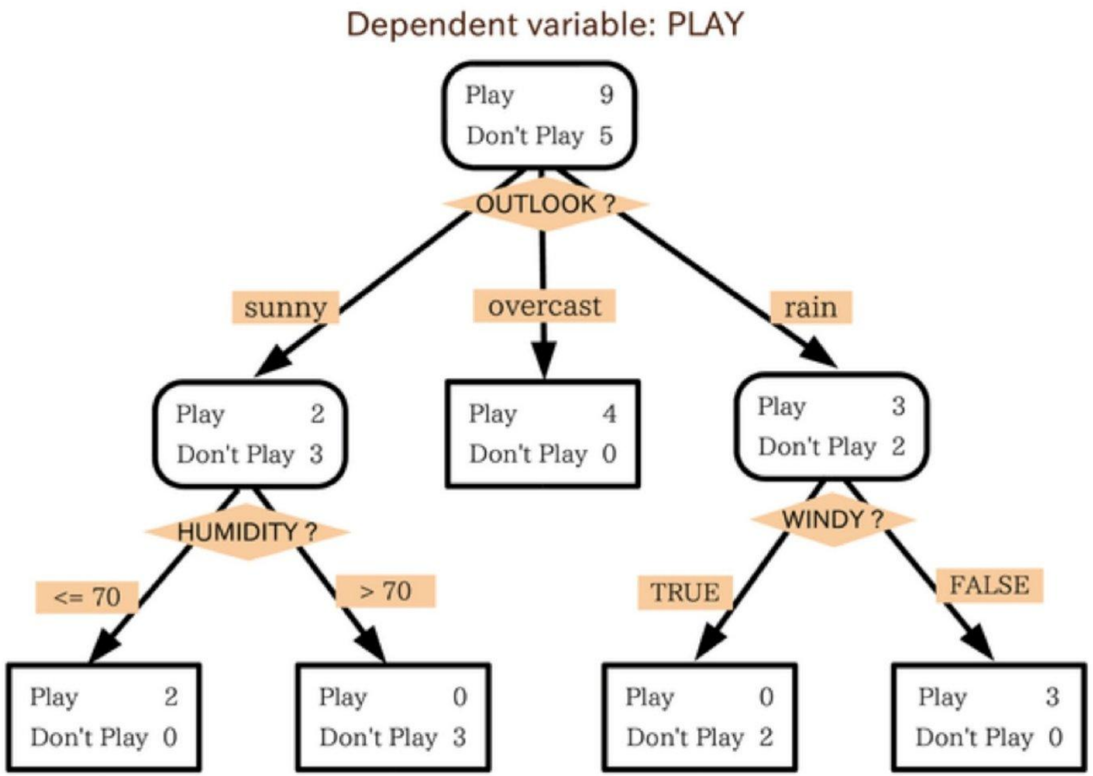
- The random forest starts with a standard machine learning technique called a “decision tree” which, in ensemble terms, corresponds to our weak learner.
- In a decision tree, an input is entered at the top and as it traverses down the tree the data gets bucketed into smaller and smaller sets.

Random Forest

- As in bagging, we build a number of decision trees on bootstrapped training samples each time a split in a tree is considered, a random sample of **m** predictors is chosen as split candidates from the full set of **p** predictors.
- **Note that if m = p, then this is bagging**.

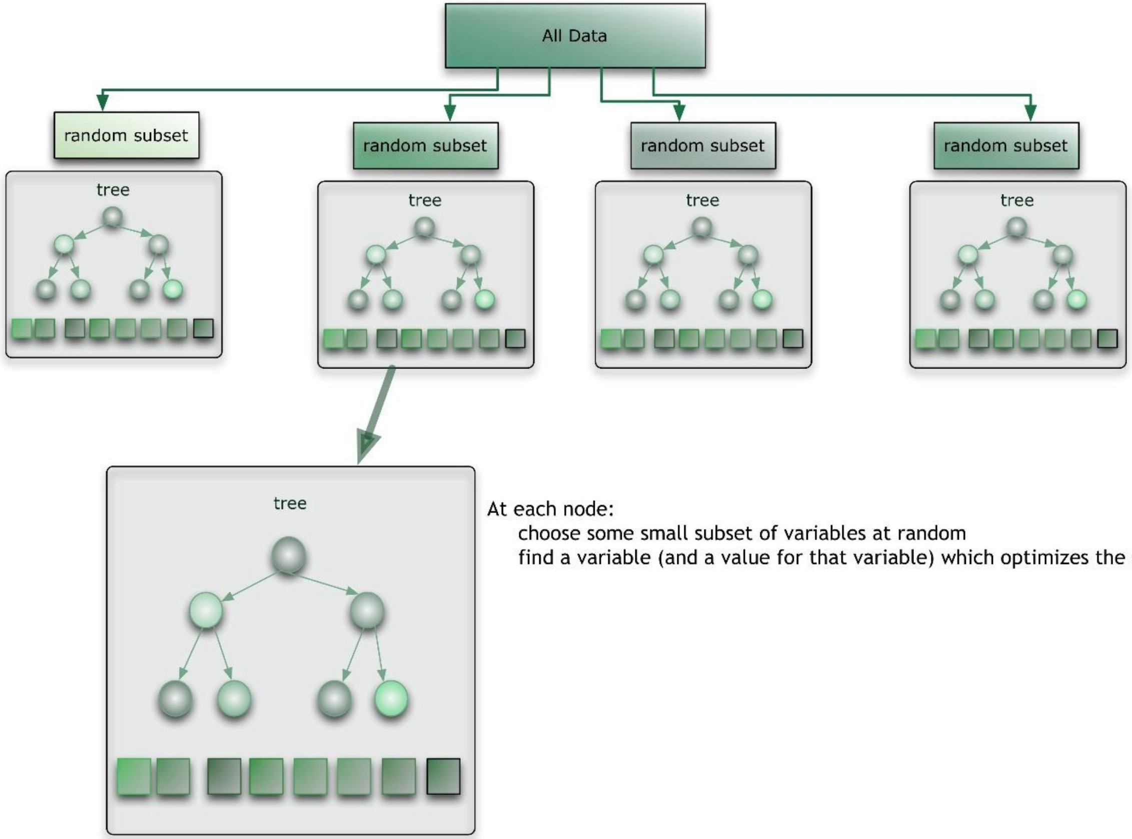
Trees and Forests

- In this example, the tree advises us, based on weather conditions, whether to play ball. For example, if the outlook is sunny and the humidity is less than or equal to 70, it's probably OK to play.



Trees and Forests

- The random forest takes this notion to the next level by combining trees with the notion of an ensemble. Thus, in ensemble terms, the trees are weak learners and the random forest is a strong learner.



At each node:
choose some small subset of variables at random
find a variable (and a value for that variable) which optimizes the split

Why Random Forest Works

- Mean Squared Error = Variance + Bias^2
- If trees are sufficiently deep, they have very small bias

- How could we improve the variance over that of a single tree?

Why Random Forest Works

$$
\begin{aligned}
Var\left(\frac{1}{B}\sum_{i=1}^{B} T_i(c)\right) &= \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B} Cov(T_i(x), T_j(x)) \\
&= \frac{1}{B^2}\sum_{i=1}^{B}\left(\sum_{j\neq i}^{B} Cov(T_i(x), T_j(x)) + Var(T_i(x))\right) \\
&= \frac{1}{B^2}\sum_{i=1}^{B}\left((B-1)\sigma^2 \cdot \rho + \sigma^2\right) \\
&= \frac{B(B-1)\rho\sigma^2 + B\sigma^2}{B^2} \\
&= \frac{(B-1)\rho\sigma^2}{B} + \frac{\sigma^2}{B} \\
&= \rho\sigma^2 - \frac{\rho\sigma^2}{B} + \frac{\sigma^2}{B} \\
&= \boxed{\rho\sigma^2} + \boxed{\sigma^2\frac{1-\rho}{B}}
\end{aligned}
$$

i=j

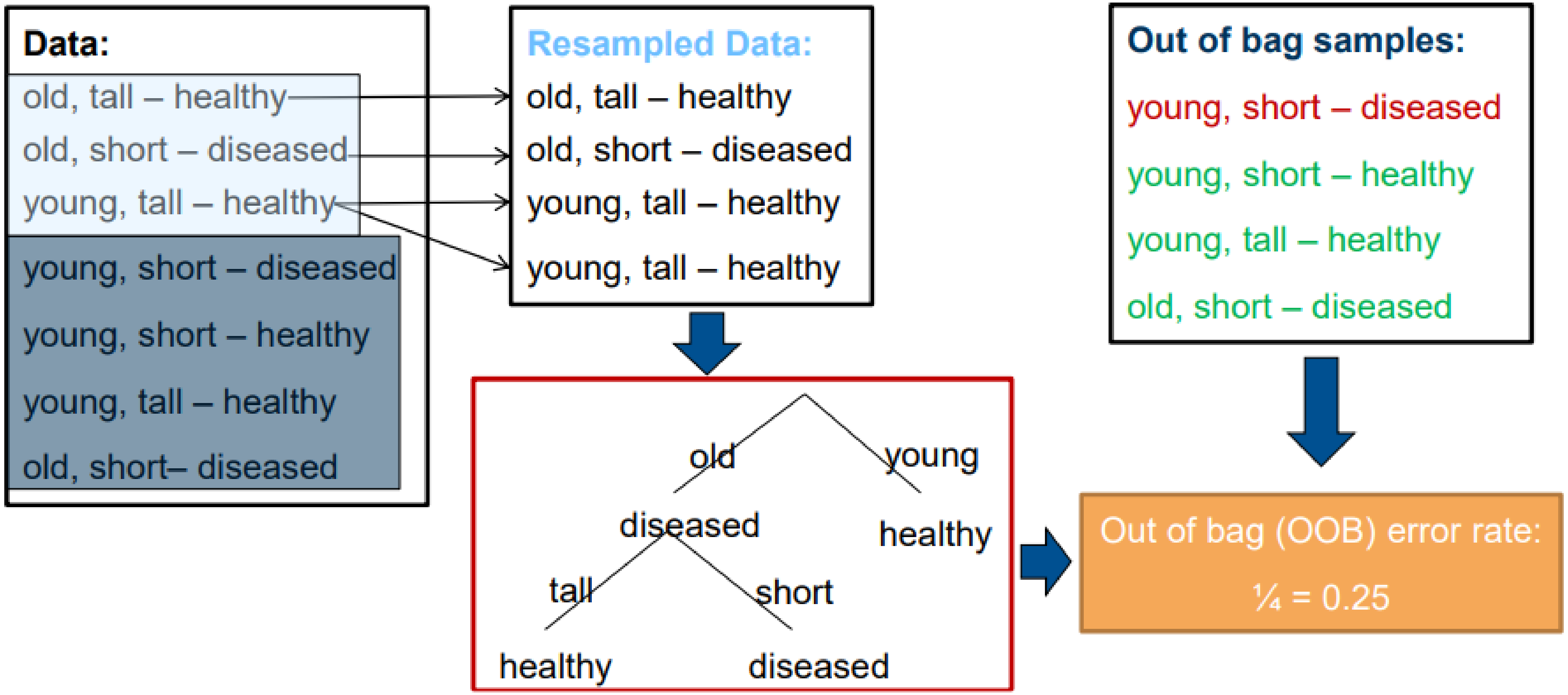
De-correlation gives better accuracy

Decreaes, if ρ decreases, i.e., if m decreases

Decreases, if number of trees B increases (irrespective of ρ)

Estimating generalization error:
Out-of bag (OOB) error

- Similar to leave-one-out cross-validation, but almost without any additional computational burden
- OOB error is a random number, since based on random resamples of the data



Data:

old, tall – healthy
old, short – diseased
young, tall – healthy
young, short – diseased
young, short – healthy
young, tall – healthy
old, short– diseased

Resampled Data:

old, tall – healthy
old, short – diseased
young, tall – healthy
young, tall – healthy

Out of bag samples:

young, short – diseased
young, short – healthy
young, tall – healthy
old, short – diseased

Out of bag (OOB) error rate:

¼ = 0.25

Random Forest Algorithm

- For b = 1 to B:

(a) Draw a bootstrap sample Z^* of size N from the training data.

(b) Grow a random-forest tree to the bootstrapped data by recursively repeating the following steps for each terminal node of the tree until the minimum node size n_{min} is reached.

i. Select m variables at random from the p variables.

ii. Pick the best variable/split-point among m.

iii. Split the node into two daughter nodes. Output the ensemble of trees.

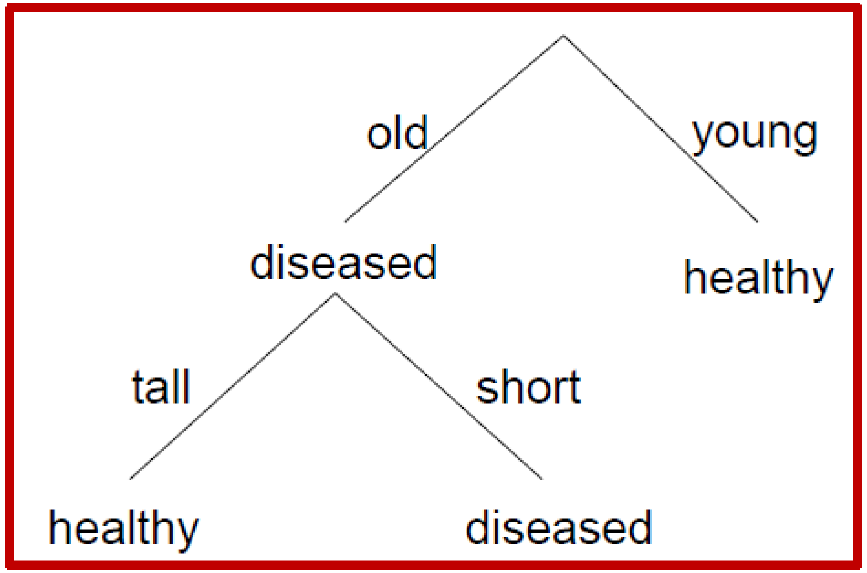
Random Forest Algorithm

- To make a prediction at a new point x we do:

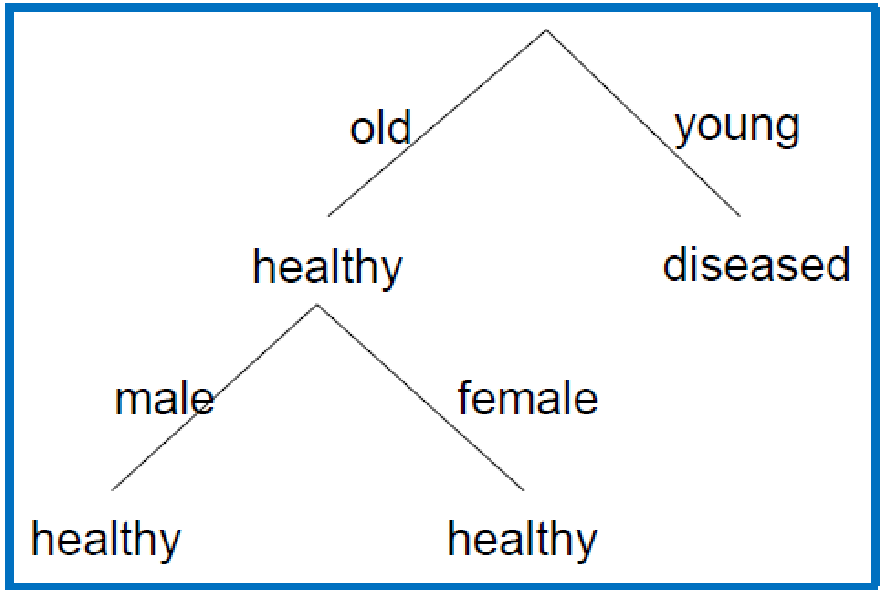
$\rightarrow$ For regression: average the results

$\rightarrow$ For classification: majority vote

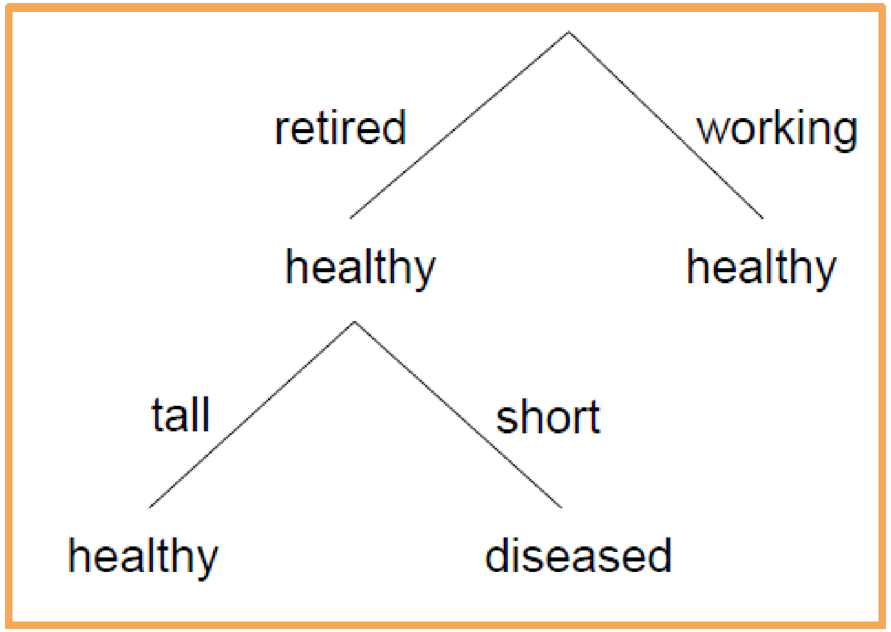
Tree 1



Tree 2



Tree 3



New sample:

old, retired, male, short

Tree predictions:

diseased, healthy, diseased

Majority rule:

diseased

Training the Algorithm

- For some number of trees *T*:
- Sample *N* cases at random with replacement to create a subset of the data. The subset should be about 66% of the total set.
- At each node:
  - For some number *m* (see below), *m* predictor variables are selected randomly from all the predictor variables.
  - The predictor variable that provides the best split, according to some objective function, is used to do a binary split on that node.
  - At the next node, choose another *m* variable randomly from all predictor variables and do the same.
- Depending upon the value of *m*, there are three slightly different systems:
- Random splitter selection: *m* =1
- Breiman’s bagger: *m* = total number of predictor variables
- Random forest: *m* << number of predictor variables. Breiman suggests three possible values for m: ½√*m*, √*m*, and 2√*m*

Running a Random Forest

- When a new input is entered into the system, it is run down all the trees. The result may either be an average or weighted average of many terminal nodes reached or, in the case of categorical variables, a voting majority.

Note that:

- With a large number of predictors, the eligible predictor set will be quite different from node to node.
- The greater the inter-tree correlation, the greater the random forest error rate, so one pressure on the model is to have the trees as uncorrelated as possible.
- As *m* goes down, both inter-tree correlation and the strength of individual trees go down. So some optimal value of *m* must be discovered.

Differences to standard tree

- Train each tree on Bootstrap **Resample** of data (Bootstrap resample of data set with N samples: Make new data set by drawing **with Replacement N samples**; i.e., some samples will probably occur multiple times in new data set)
- For each split, consider only m randomly selected variables
- Don’t prune
- Fit B trees in such a way and use average or majority voting to aggregate results

Random Forests Tuning

- The inventors make the following recommendations:

→ For classification, the default value for m is √p and the minimum node size is one.

→ For regression, the default value for m is p/3 and the minimum node size is five.

- In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters.
- Like with Bagging, we can use OOB and therefore RF can be fit in one sequence, with cross-validation being performed along the way. Once the OOB error stabilizes, the training can be terminated.

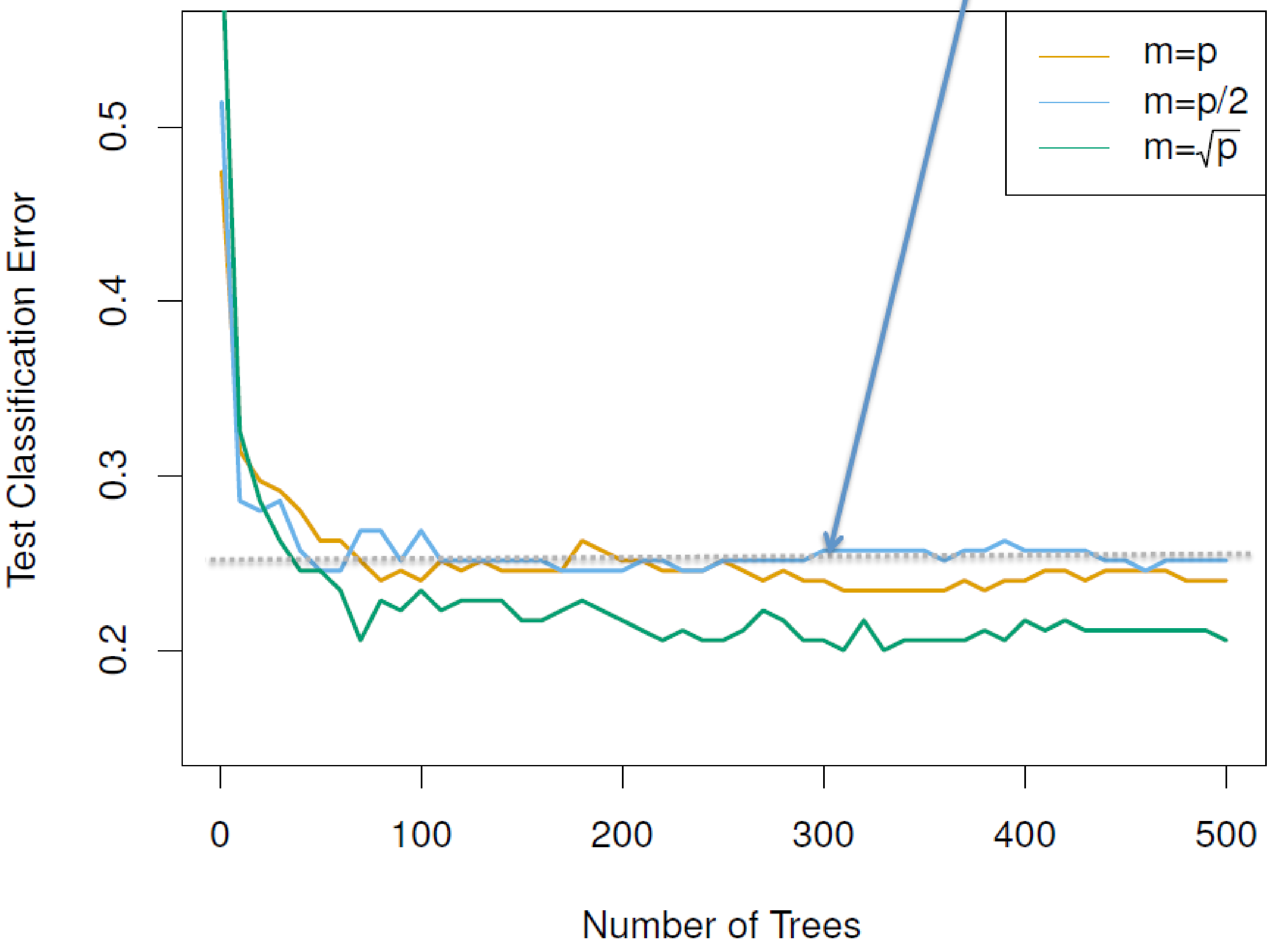
Advantages of Random Forest

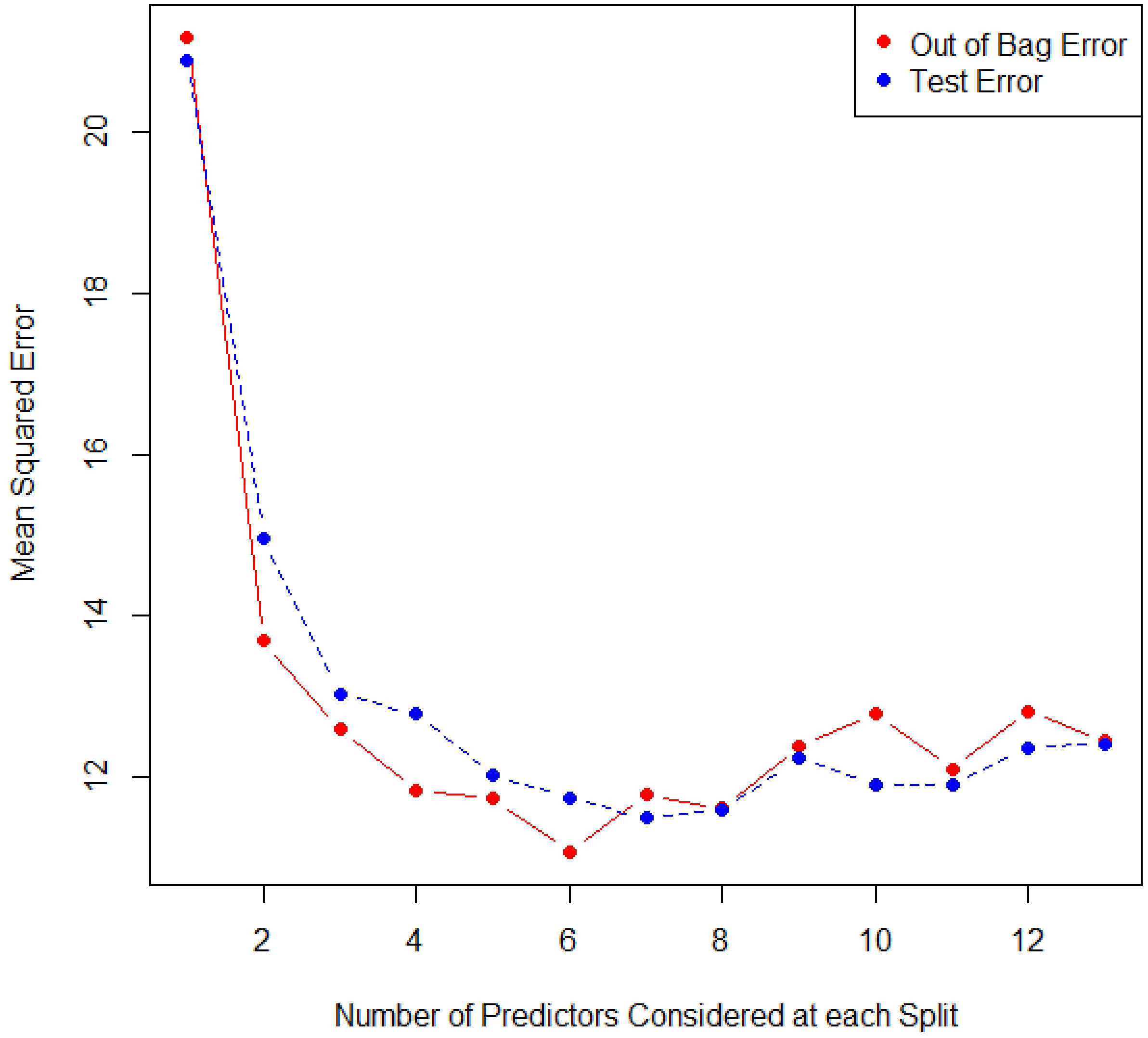
- No need for pruning trees
- Accuracy and variable importance generated automatically
- Overfitting is not a problem
- Not very sensitive to outliers in training data
- Easy to set parameters
- Good performance

Example

- 4,718 genes measured on tissue samples from 349 patients.
- Each gene has different expression
- Each of the patient samples has a qualitative label with 15 different levels: either normal or 1 of 14 different types of cancer.
- Use random forests to predict cancer type based on the 500 genes that have the largest variance in the training set.

Null choice (Normal)





Now what we observe is that the Red line is the Out of Bag Error Estimates and the Blue Line is the Error calculated on Test Set. Both curves are quite smooth and the error estimates are somewhat correlated too. The Error Tends to be minimized at around mtry=4.