Random Forests

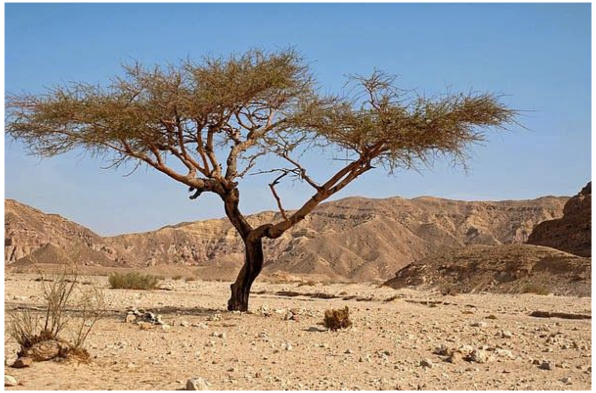
Data Science Immersive

FLATIRON SCHOOL

Objectives

- Decision Trees Review
- Apply Bootstrapping and Aggregating (Bagging) to Decision Trees
- Apply the Random Forests algorithm

Trees are pretty great :)



Quick Review of Classification Trees

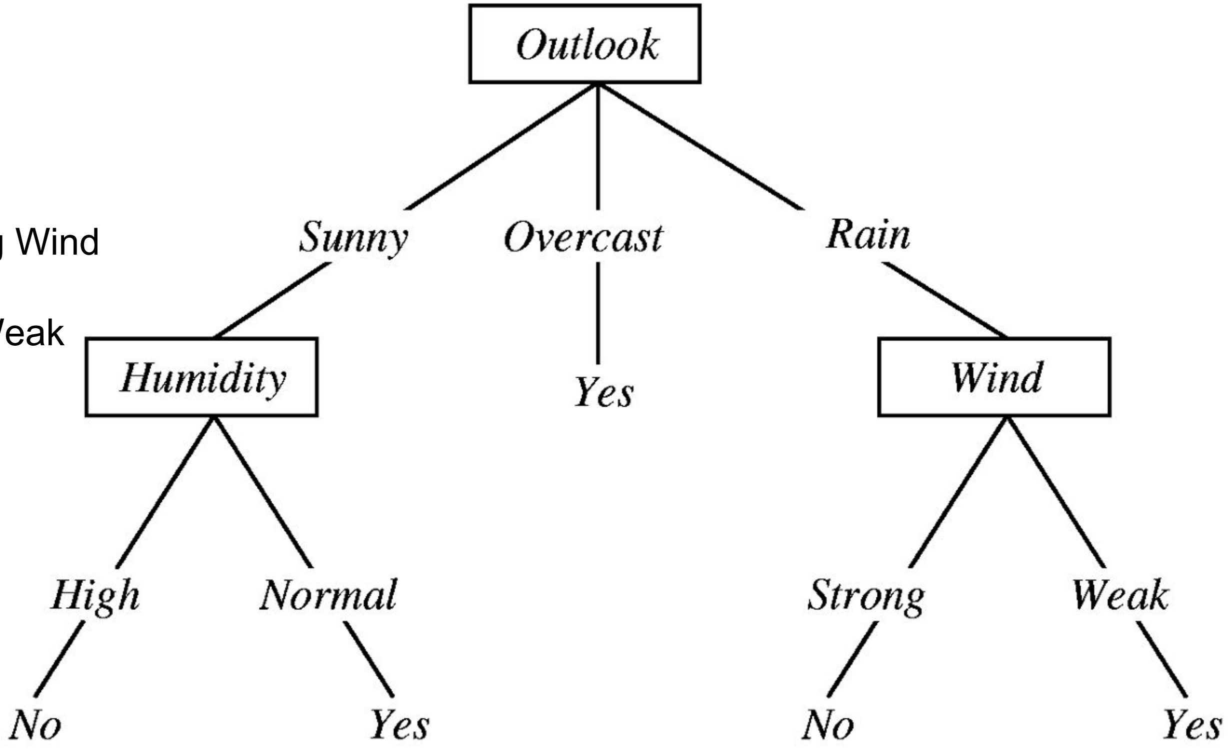
Should we go on a class field trip to Central Park?

Day 1: Rain, High Humidity, Strong Wind

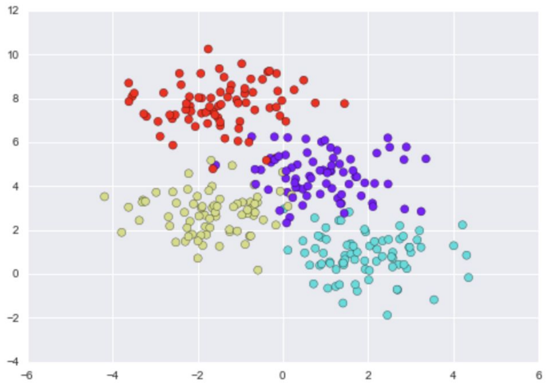
Day 2: Sunny, Normal Humidity, Weak Wind

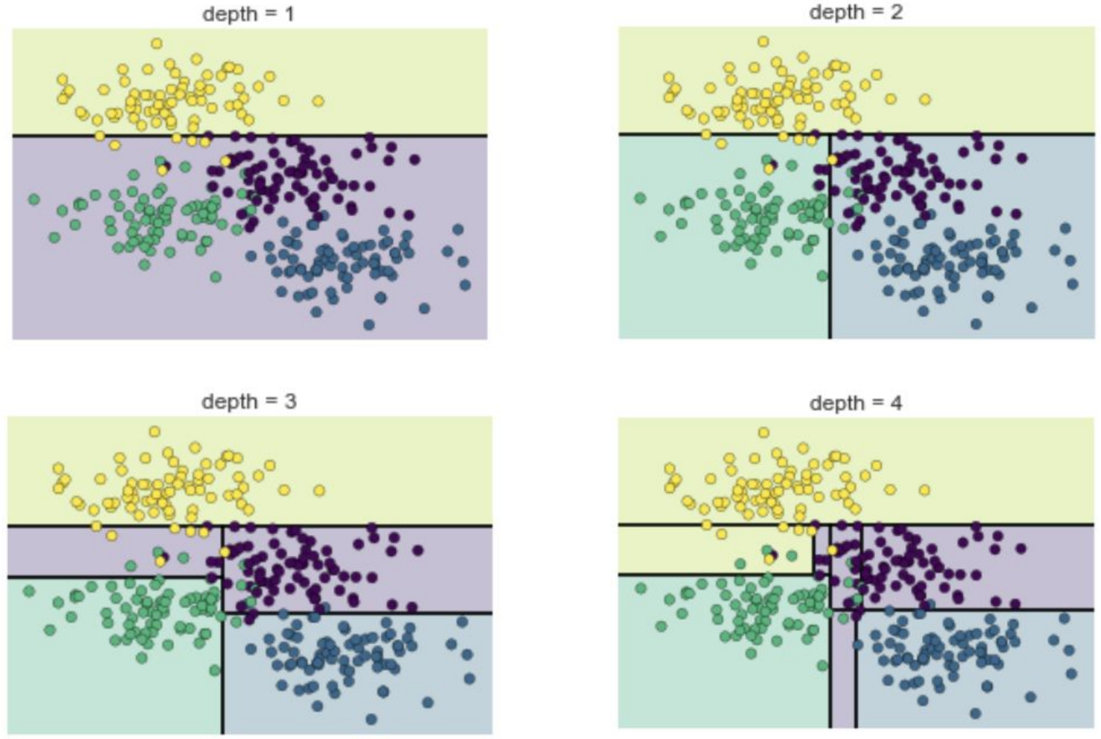
Non-parametric: We assume no underlying distribution or function

Greedy algorithm

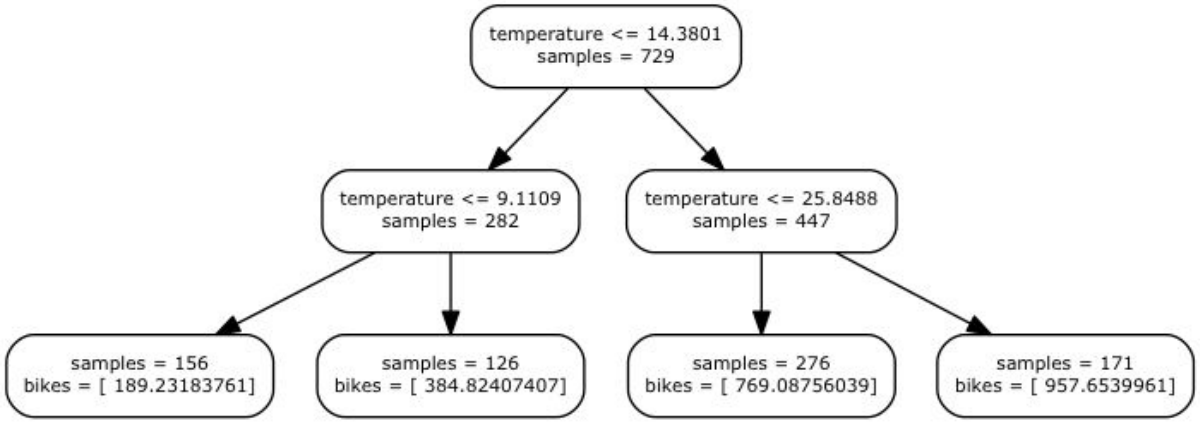


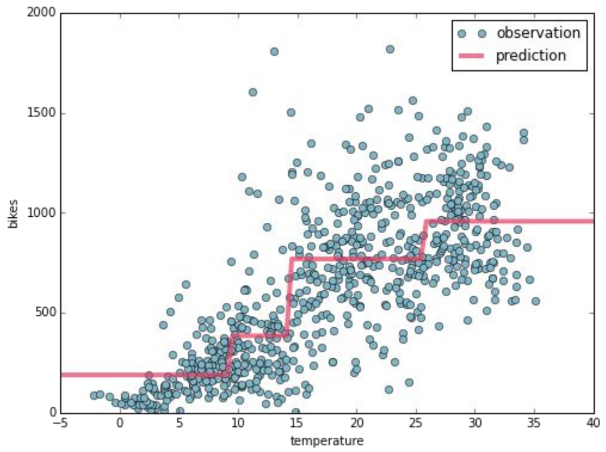
Visualizing Classification Decision Trees





Regression Decision Tree

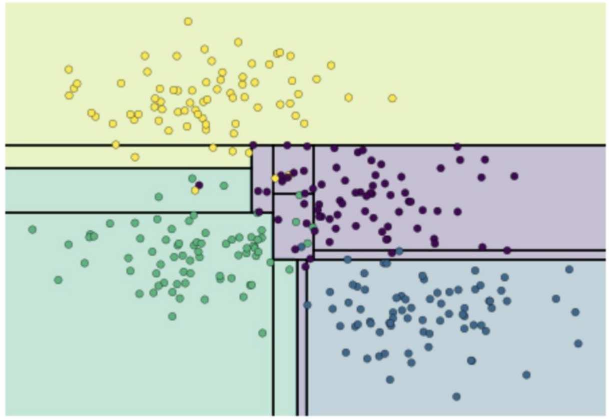


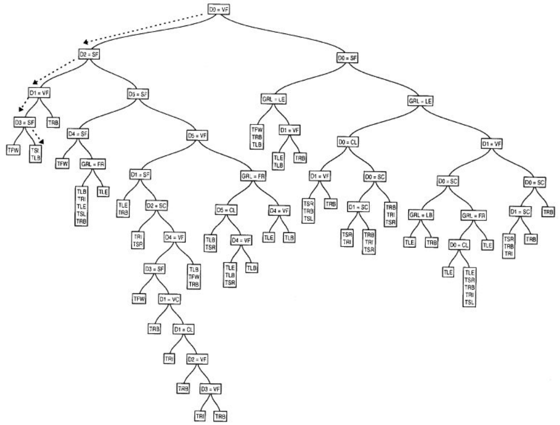


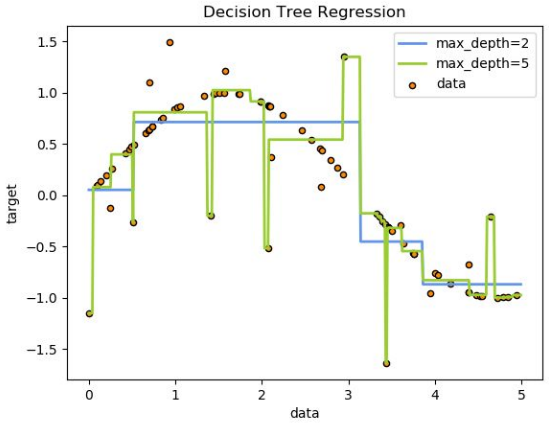
Impurity	Task	Formula	Description
Gini impurity	Classification	$\sum_{i=1}^{C} -f_i(1-f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels.
Entropy	Classification	$\sum_{i=1}^{C} -f_i \log(f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels.
Variance / Mean Square Error (MSE)	Regression	$\frac{1}{N}\sum_{i=1}^{N}(y_i-\mu)^2$	y_i is label for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$
Variance / Mean Absolute Error (MAE) (Scikit-learn only)	Regression	$\frac{1}{N}\sum_{i=1}^{N}\lvert y_i-\mu\rvert$	y_i is label for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$

Tree Depth and Overfitting

The deeper our trees go, the higher the chance that we overfit our data







Decision Tree Review

Can be used for both Regression and Classification problems (Classification and Regression Trees)

Advantages of Decision Trees:

- Work well with non-linear relationships
- Easy to interpret!
- Implicit feature selection
- They are fairly robust to outliers

Disadvantages of Decision Trees:

- Decision trees tend to overfit, especially if they are completely pure.
- Instability: Small changes in the input data can cause large changes to the structure of the tree.

Should we trust an expert or a crowd?

You are at an auction trying to determine whether or a specific painting is a fraud. You decide to ask around and get opinions from people at the auction site.

Option A:

An expert art inspector has an accuracy of 80%. There is only one on site.

Option B:

Other attendees, who are not quite as knowledgeable, accurately identify whether or not a piece of art is fake 60% of the time. There are at most 30 other attendees you can ask.

Assuming you can only have time for one option, which would you choose? Why?

Hint: Think about how you might be able to use a binomial distribution perhaps….

Wisdom of the Crowd!

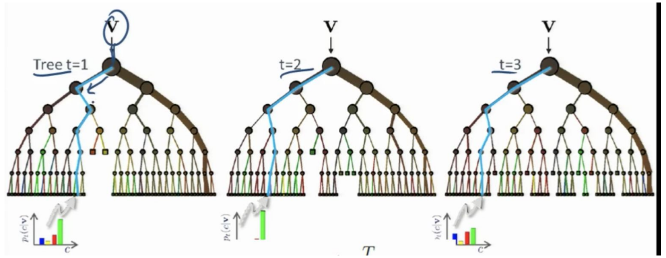
At a 1906 fair in Plymouth, England, statistician Francis Galton noticed how when 800 people guessed how much a “dressed” ox weighed. It turns out that the actual guess had only a 1% error from the median guess

Takeaway

We can learn a lot by combining many different learners. This is called an *ensemble method*.

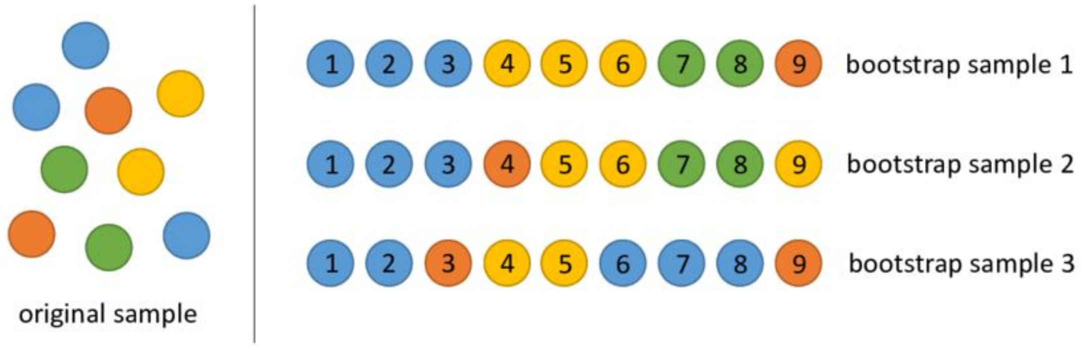
In essence, we can combine the output of models that are not accurate enough on their own. We can get wisdom from the crowd!

How can we grow a forest of trees?



Bootstrapping

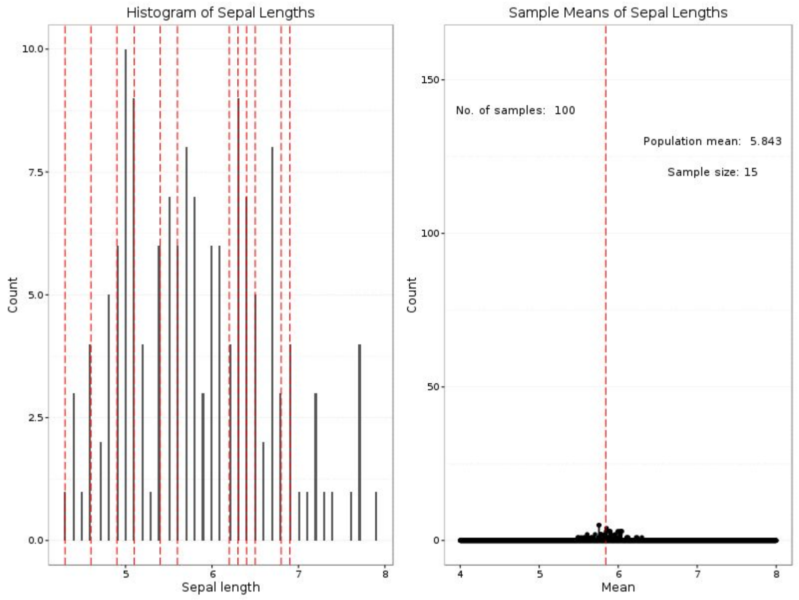
- Trees are prone to overfitting. (They have a high variance). This is especially true if the trees are built out to full “purity” in each of the leaves.
- To help prevent overfitting we take bootstrap samples **with replacement** from our training data that is the same size as our training data.



Bootstrapping

What principle allows us to use bootstrapping?

Central Limit Theorem



Aggregating

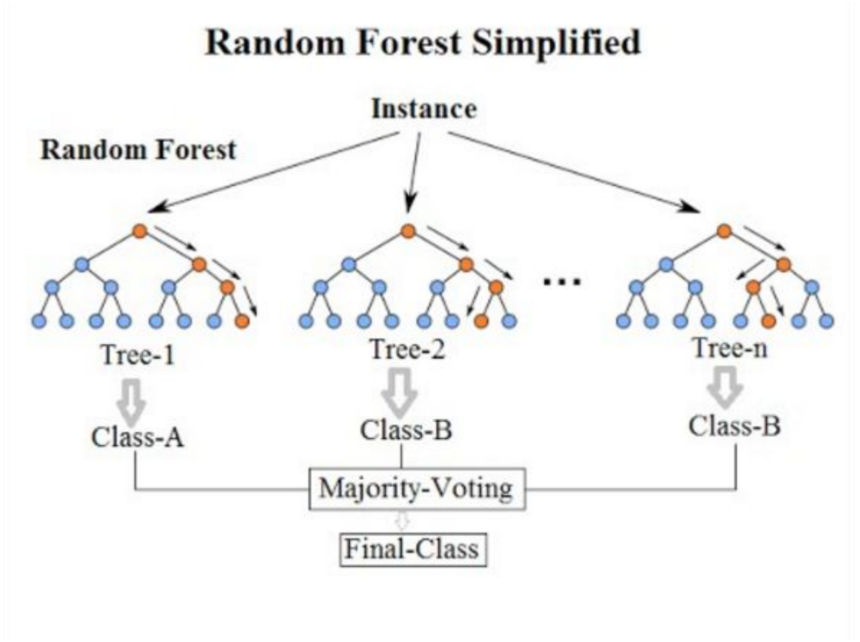
- Once we have taken bootstrapped samples, we fit a decision tree to each sample. This tree will have a **low bias and high variance**
- Repeat this process for however many trees you want in your model
- Now, we can feed data through all of the bootstrapped trees and take
    - Classification: whichever class is predicted most by the bootstrapped decision trees
    - Regression: take an average of the predicted values for each decision tree

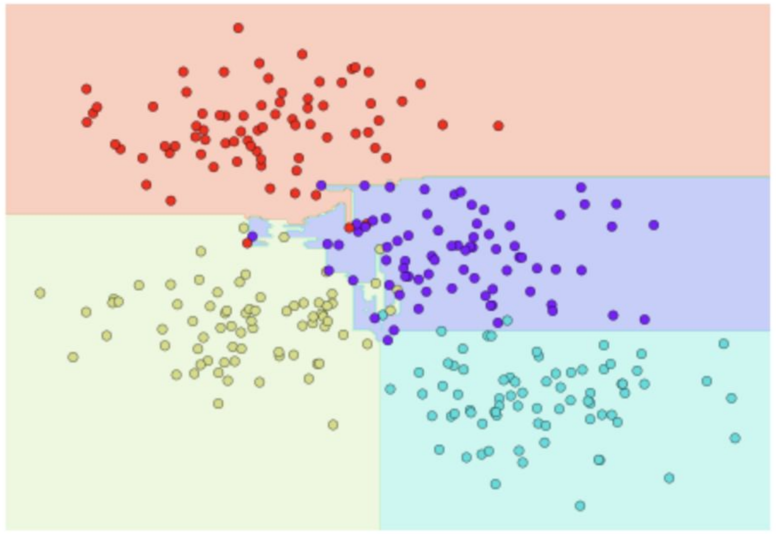
$$\tilde{y} = argmax\left(N_c(y_t^1), N_c(y_t^2), \ldots, N_c(y_t^n)\right)$$

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

Voting Method

Majority Class Labels (Majority/Hard Voting)



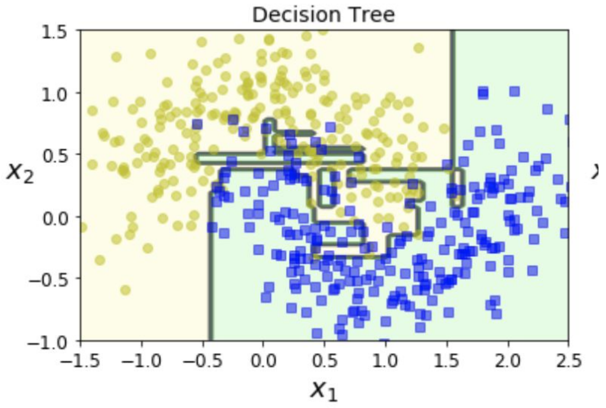


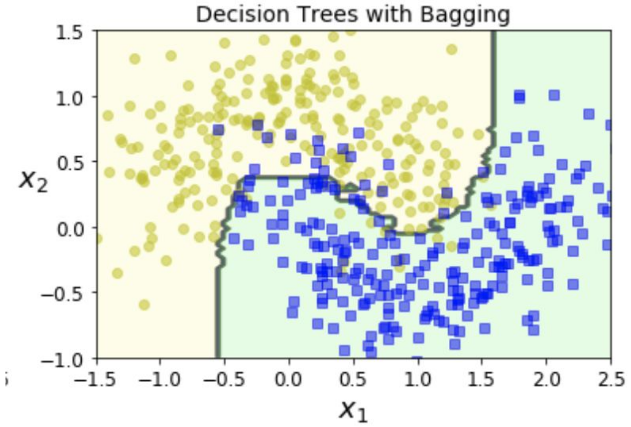
Bootstrapping + **Agg**regating

$\Downarrow$

Bagging

Decision Tree v. Decision Trees with Bagging



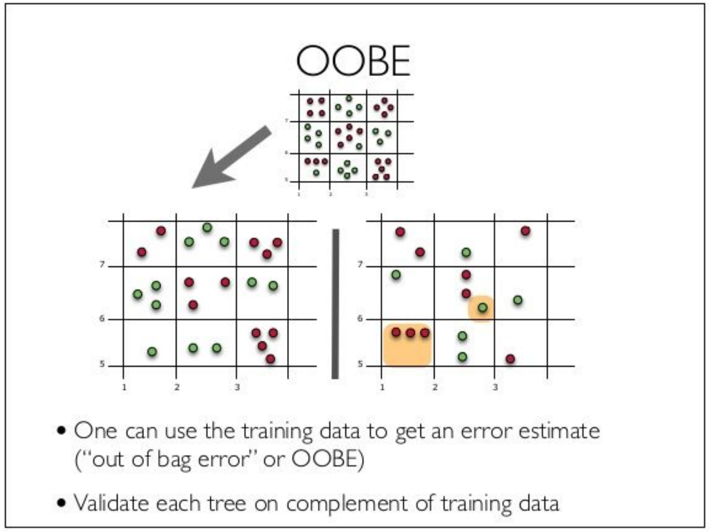


Out-of-Bag Error

In general only ⅔ of the observations will be trained on each model. (Why 2/3?)

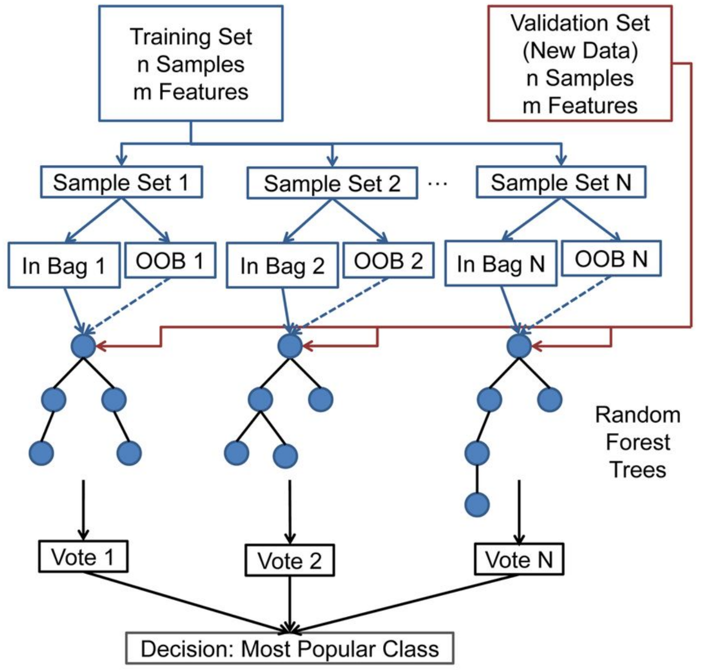
Performing cross validation on bagged classifiers/regressors is challenging because it can be computationally expensive.

Rather, we can look at every observation and make a prediction for each of the data points for which that data points was not used to create the tree.



- One can use the training data to get an error estimate ("out of bag error" or OOBE)
- Validate each tree on complement of training data

Out-of-Bag Error



It's essentially just a form of cross-validation!

The Issue with Bagging

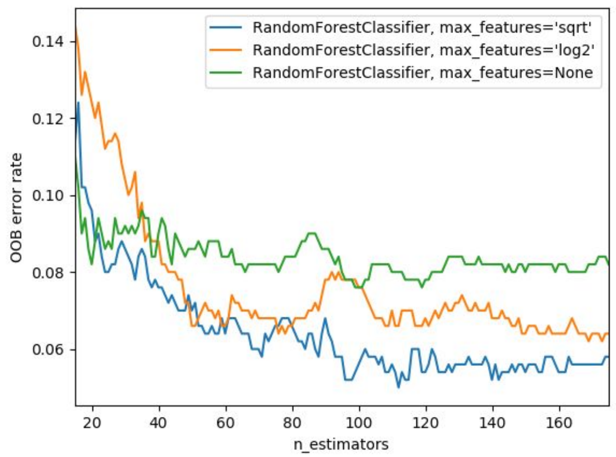
The issue with bagging is that each one of the trees might be correlated to each other. There might be a feature that is powerful in generating a separation between different categories, which results in trees that are correlated to one another despite being from bootstrapped samples.

We need to do something to ensure that the bootstrapped samples are not correlated with one another…..

Random Forests

Random Forests de-correlate each of the decision trees created in bagging by ensuring that at each split, only *m* features are considered for a given split. (typically *m* = sqrt(p))

This means that on average *(p-m)/p* splits will not even consider a given strong predictor. As we increase the number of trees, this will not lead to overfitting, meaning that we should make as many trees as possible until we have achieved an acceptable error rate.



Random Subspace Sampling Method

Important note: The *m* features are randomly chosen at each ***node*** not for the entire decision tree.

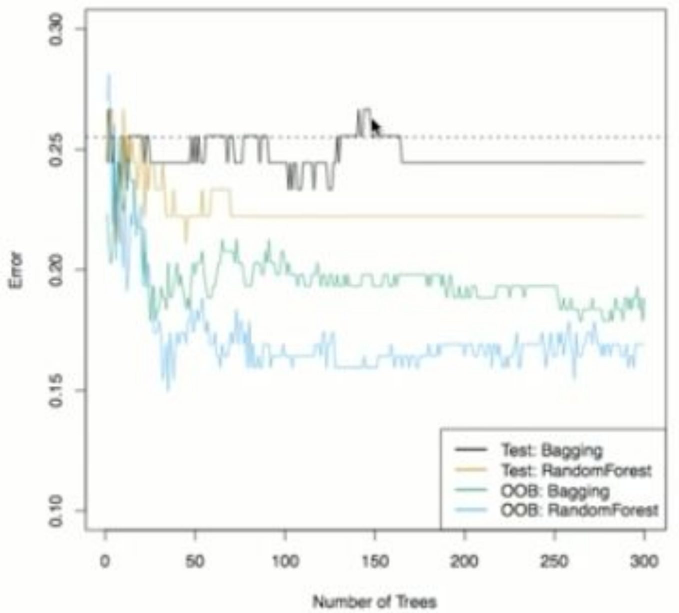
This is to ensure that our trees need to be decorrelated with one another

Our trees should have ***diverse*** opinions

Comparing Performances

In general:

Random Forest > Bagging > Decision Trees



Random Forest Advantages/Disadvantages

Advantages

- A very powerful model. Will nearly always outperform decision trees
- Able to detect non-linear relationships well
- Harder than other models to overfit

Disadvantages

- Not as interpretable as decision trees
- Many hyperparameters to tune (GridSearch is your friend!)

Random Forest Hyperparameters

n_estimators : the number of trees in the forest

criterion: “gini”,”entropy”

max_features: the number of random features to be considered when looking for the best split

max_depth: the maximum number of levels of a tree

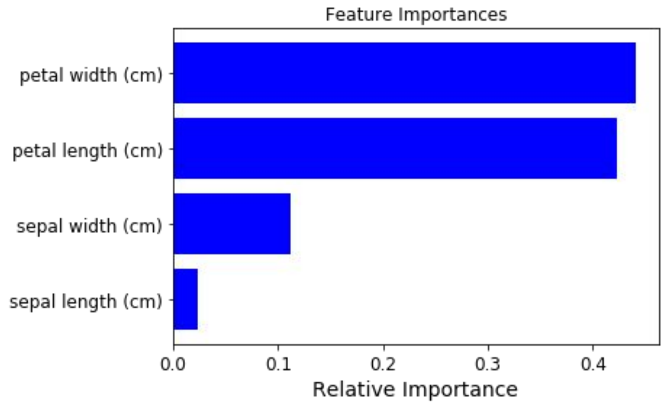
bootstrap: whether or not bootstrap samples are used to build trees

oob_score: whether or not to use out-of-bag samples to estimate the generalization accuracy

n_jobs: how many cores you want to use when training your trees

Feature Importances

We are able to obtain "feature importance" for each of the variables. This is calculated by averaging the total amount that the gini index is decreased/mean decrease impurity increased by splits over a given predictor, averaged over all B trees.



There are many other ways to determine the feature importances of Random Forest. Check them out here https://papers.nips.cc/paper/4928-understanding-variable-importances-in-forests-of-randomized-trees.pdf

On another note

In sklearn, you can create custom ensemble models by making use of the VotingClassifier/VotingRegressor. This is intended to be used with conceptually different models.

- Within the VotingClassifier, you can specify “Majority Class” vote or “Soft” vote.
    - Majority Class: Select the class that is predicted most
    - Soft: Take an average of the probabilities for each class, make decision based off of that

Question to ponder…….

How do Random Forests handle the bias-variance tradeoff? What would be another way of using ensembling methods to tackle the bias-variance tradeoff?

Additional Resources

https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

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