Ensemble Models

Random Forests and Other Ensembles

Erin Hoffman
September 2020

// FLATIRON SCHOOL

Agenda

1. **Motivation**
2. **Random Forests (Bagging+)**
   a. Layers of randomization
   b. Synchronic Aggregation
3. **Other Ensembles**
   a. Stacking
   b. Boosting
4. **SciKit-Learn Implementation**

1. Motivation

We have a lot of models already...why do we want to add ensembles?

Motivation

An Old Mantra

Data Collection:

- One data point is good, but more data points are better!

Bootstrapping:

- One sample is good, but more samples are better!

Modeling:

- One model is good, but more models are better!

Motivation

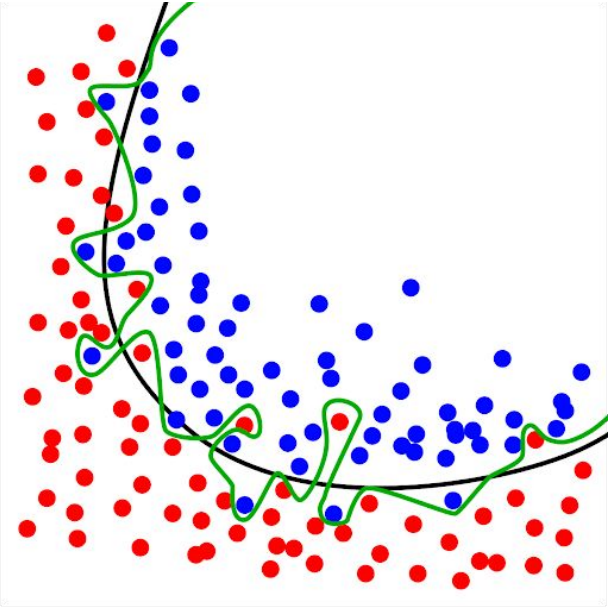
Model *Composed of* Other Models

- Building models beyond the first is good for comparison's sake
- But we can also combine models together to form new models!

2. Random Forests

More trees, more randomness = reducing variance without introducing too much bias by “pruning”

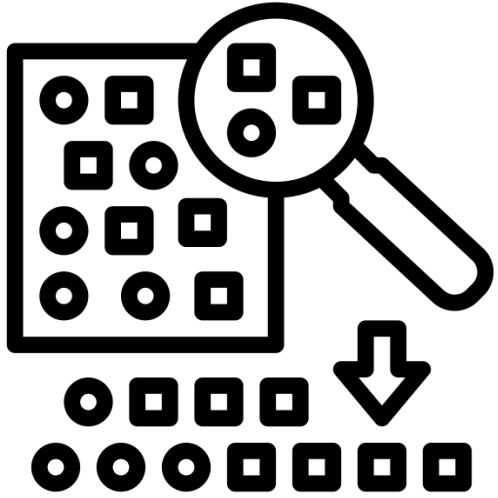
Bagging



Strategy

- Many models naturally **overfit**
- **Randomization** creates new models
  - New models overfit in different ways
- **Aggregation** smooths over different ways of overfitting

Bagging



Created by Becris
from Noun Project

Synchronic Aggregation

- Bagging = **B**ootstrapping + **Agg**regating
- Algorithm to repeat many times:
  - Create a sample from your data
  - Train a model (e.g. a decision tree) on that sample
- A **bagging model** is an average over those many models

Matchup

Decision Tree vs. Bagging

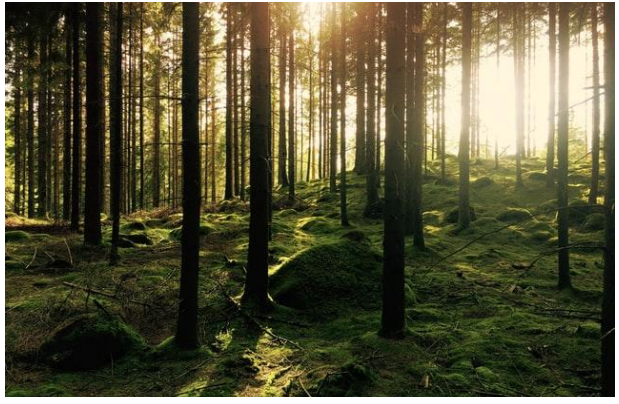
Decision Tree

- Less computationally complex
  - Faster to fit
  - Smaller model size

Bagging

- Better variance than un-pruned tree (less overfitting)
- Better vias than pruned tree (less underfitting)

Random Forests



Bagging+

- We already have Level 1 of randomization:
  - Train each model on a random sample of data rows/records
- Let's add Level 2 of randomization:
  - Choose a **random set of features at each decision point**
- A **random forest** has both of these levels of randomization

Job Interview Question

What is “random” about a random forest?

1. Each **tree** (model) uses a random sample of **records** (rows of the dataset)
2. Each **decision point** uses a random sample of **features** (columns of the dataset)

Matchup

Bagging vs. Random Forest

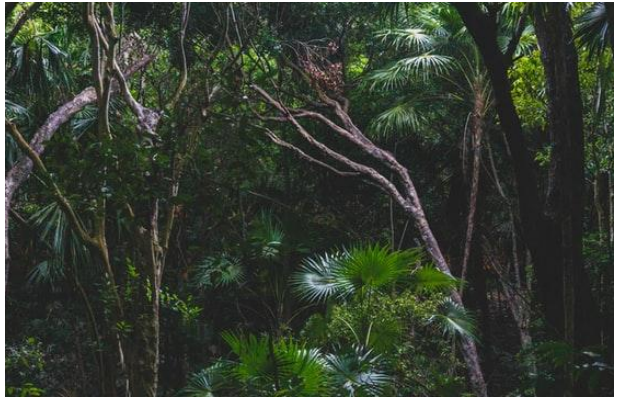
Bagging

- Sometimes can get the same results with fewer trees (depends on the data)

Random Forest

- Most of the time, better variance (less overfitting) due to increased randomness, which reduces correlation between trees

Extra Trees



Extremely Randomized Trees

- We already have Level 1 and Level 2 of randomization
- Add Level 3 of randomization:
  - **Randomly choose a decision point**, rather than finding the one with the most information gain
- An **extra trees** model can have all three levels
  - Although the SciKit-Learn implementation skips Level 1 by default

Matchup

Random Forest vs. Extra Trees

Random Forest

- Usually (but not always) gets better performance

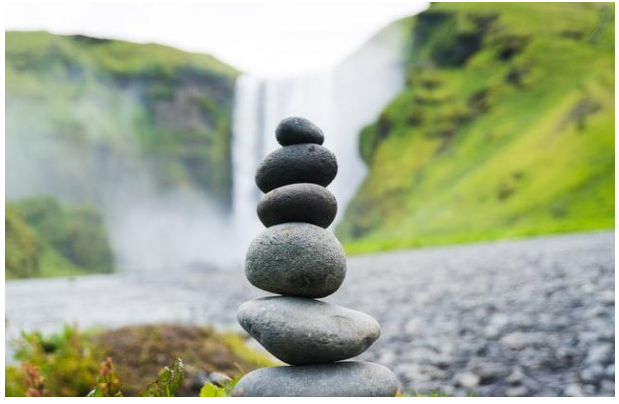
Extra Trees

- Randomly choosing a decision point is much faster than checking all of the options

3. Other Ensembles

Random forest models are probably the most popular, but let's go over some other kinds of ensembles

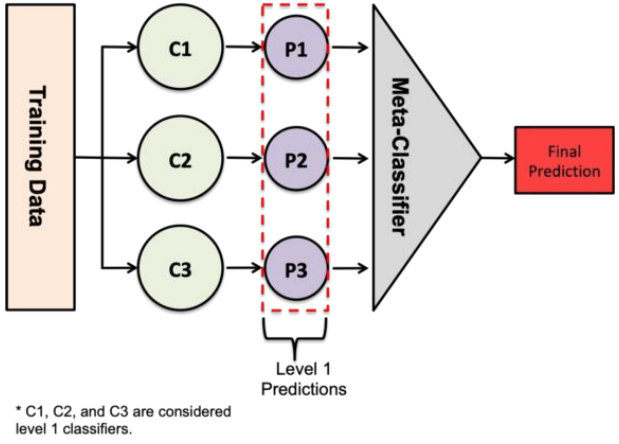
Stacking



Different Models, Same Data

- Similar to a bagging approach, stacking is a form of **averaging**
- Unlike a bagging approach, stacking typically uses the **same training data** for every model
- The innovation of stacking is the use of **different kinds of models**

Stacking



Meta-Classifier/Meta-Regressor

- First, we ask several different models to make predictions about the target
- Rather than taking a simple average or vote to determine the outcome, feed these results into a **final model that makes the prediction based on the other models' predictions**
- If it seems like we are approaching a neural network...you are correct!

Boosting

Strategy

- Prevent overfitting from the start
- Train an *underfit* (bad) model
- Improve the bad model by making quantitative use of the **residuals** of the bad model

4. SciKit-Learn Implementation

We are not going to code this from scratch, we're going to use the tools available to us!

SciKit-Learn Implementation

Regression	Classification
BaggingRegressor	BaggingClassifier
RandomForestRegressor	RandomForestClassifier
ExtraTreesRegressor	ExtraTreesClassifer
GradientBoostingRegressor	GradientBoostingClassifier
StackingRegressor	StackingClassifier