

Introduction Ensemble and Random Forest

Brian Wright

Ensemble Methods

- Ensemble methods are more or less aggregated predictions of many different algorithms in order to increase predictive accuracy.
- Often in solving a machine learning problem building a ensemble model comes at the end of the process after trying several different approaches you can combine them into one all knowing predictor!
- We will focus on Random Forrest, an ensemble of decision tress but also discuss bagging and boosting

Ensemble Methods

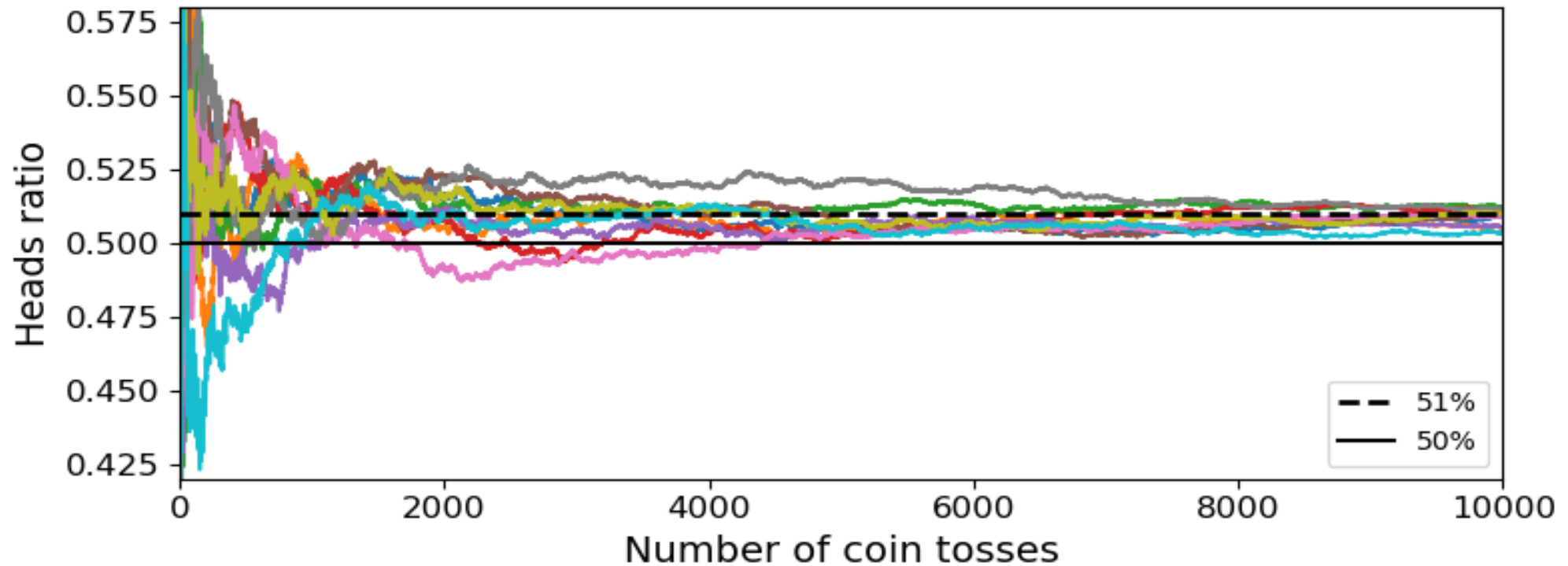
- Example: suppose we have developed several different classifiers a KNN model, Logistic Regression, an SVM and a Decision Tree.
- We could use the majority vote process to build the final classification of our data points, based on below what would be the outcome for this single data point?

Model	KNN	Logistic Regression	Support Vector Machine	Decision Tree
Prediction	1	0	1	1

- This is called **hard voting**, another approach is **soft voting (discuss later)**
- This process often works better than using single **weak learners** – or algorithms that predict only slightly better than random guessing

- Why do the models work better together?
 - ❖ Essentially scale increases the probability of finding a majority vote.
 - ❖ As an example if we have a bias coin flip that gives us 51% chance of heads and 49% chance of tails the more we toss the coin the higher the probability of getting a majority vote for heads.
 - 1,000 = 75% chance
 - 10,000 = 97% chance
- Works the same way for model building, the reliability of the results simply increase with scale.
 - ❖ Works best if the models are perfectly independent, meaning the error terms don't correlate which is hard when using one approach on a single dataset, that's why combining approaches can sometimes result in better outcomes

Ensemble Methods: Probabilities Converge (Code Available)



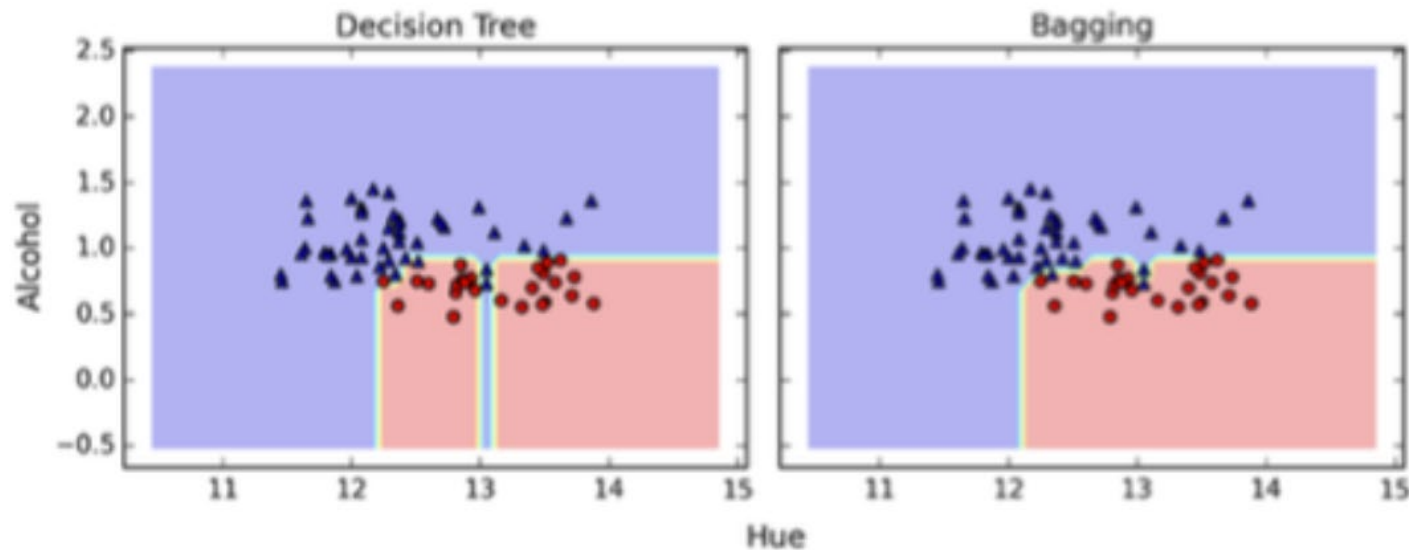
Ensemble Methods: Random Forest – power in numbers

- Ensemble methods – Essentially instead of building a single tree we are going to build a whole bunch
- We can limit the growth of the trees but don't have to use any of the hyper-parameters
- Set the number of trees grown and track the error classification rate of our algorithm
- Can be used for again for both Regression or Classification
- Random Forest uses a combination of **bagging and boosting**

Ensemble Methods:

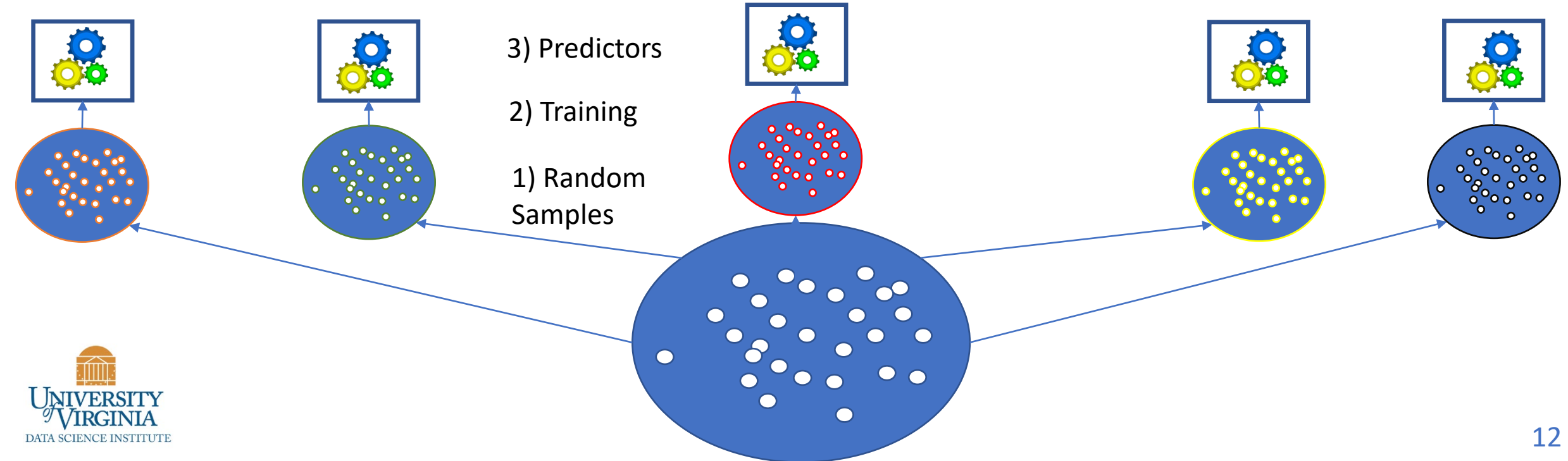
Tree base Boosting, Bagging and Random Forest

- Bagging – Goal is to reduce the complexity of models that have a tendency to **overfit** through “bootstrap aggregation”. Uses majority voting process to select features, which works to **reduce variance** (less overfitting) by creating simpler decision boundaries.
 - ❖ Grow a whole bunch of trees and use distribution of results to make predictions.




Ensemble Methods: Bagging

- As discussed one approach is to use several methods and combine them to produce a more powerful outcome
- You can also use the same technique but re-sample the data numerous times to produce different results, one such method is **Bagging**
- **Bagging** – is sampling with replacement, meaning that the entire dataset is available for every sub-sample. (without replacement is often called **pasting**)




Sampling with replacement

Sampling without replacement



Obs	X1	X2	Y1	Y2
1	2.5	3.6	4.8	3.7
2	2.8	4.7	-2.8	7.1
3	5.8	9.7	9.1	13

Obs	X1	X2	Y1	Y2
1	2.5	3.6	4.8	3.7
2	2.8	4.7	-2.8	7.1
1	2.5	3.6	4.8	3.7



Obs	X1	X2	Y1	Y2
2	2.8	4.7	-2.8	7.1
1	2.5	3.6	4.8	3.7
3	5.8	9.7	9.1	13

Ensemble Methods

- If we keep building decision trees on the same dataset, we would essentially get the same decision trees every time...

Obs	X1	X2	Y1	Y2
1	2.5	3.6	4.8	3.7
2	2.8	4.7	-2.8	7.1
3	5.8	9.7	9.1	13



Obs	X1	X2	Y1	Y2
2	2.8	4.7	-2.8	7.1
1	2.5	3.6	4.8	3.7
3	5.8	9.7	9.1	13



Obs	X1	X2	Y1	Y2
3	5.8	9.7	9.1	13
1	2.5	3.6	4.8	3.7
2	2.8	4.7	-2.8	7.1



Ensemble Methods

- Use a subset of attributes generated by bagging to build original data sets to make decision trees...want the trees to uncorrelated...that's the goal

Obs	X1	X2	Y1	Y2
1	2.5	3.6	4.8	3.7
1	2.5	3.6	4.8	3.7
3	5.8	9.7	9.1	13



Obs	X1	X2	Y1	Y2
2	2.8	4.7	-2.8	7.1
1	2.5	3.6	4.8	3.7
2	2.8	4.7	-2.8	7.1



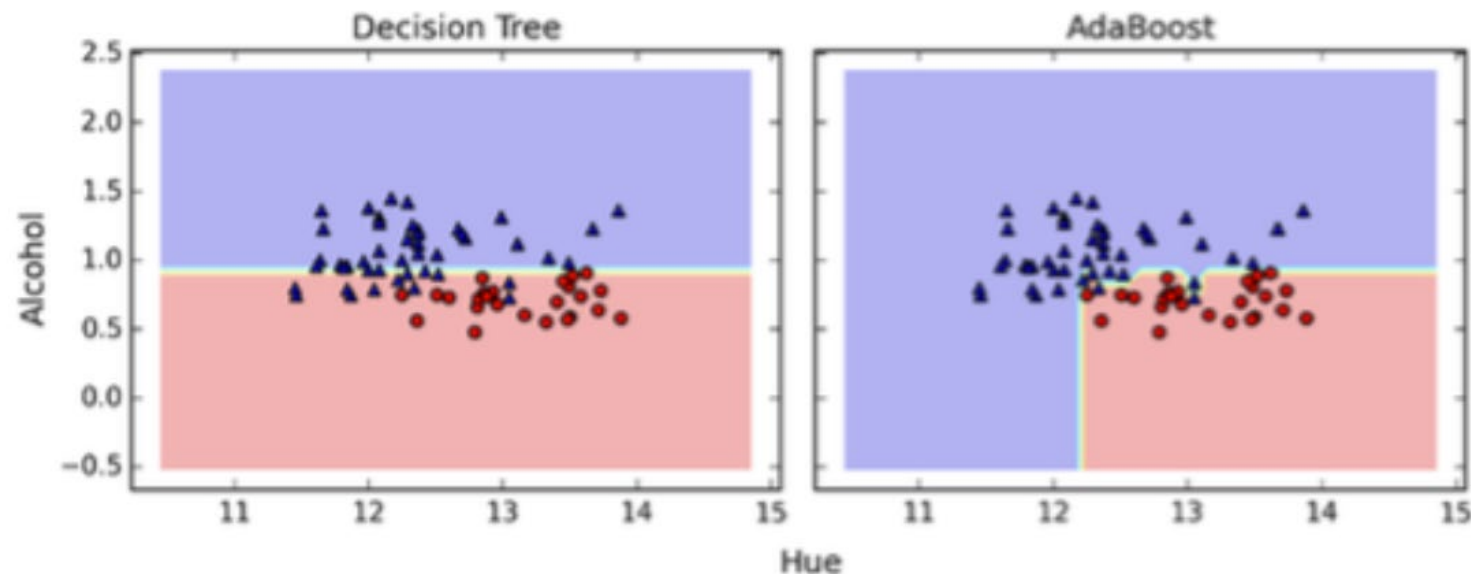
Obs	X1	X2	Y1	Y2
3	5.8	9.7	9.1	13
3	5.8	9.7	9.1	13
2	2.8	4.7	-2.8	7.1



Ensemble Methods

Tree base Boosting, Bagging and Random Forest

- Boosting – Combines a series of “weak learners” together to make a more powerful predictor. Think of combining a series of single or two level split trees that learn from each others' weaknesses. This **reduces bias** and helps the model work against **underfitting**.
 - ❖ An example is Adaboost which will uses simple trees to fit a model and then dependent on the error or mis-classification increases the weight on the error terms and refits the model. It will do this over and over again, thus “Adaptive Boosting”



Tree base Boosting, Bagging and Random Forest

- Random Forest – Is technically a “bagging” model but it uses random bootstrapped samples to model fit that includes reductions in the feature space.
 - ❖ Draws random samples from training data – Mostly Bagging – Reduces Variance
 - ❖ Draws random subsets of features – Mostly Boosting – Reduces Bias
- It is this combination of process which makes Random Forest such a powerful and widely used algorithm –
 - ❖ The main tuning parameter by Mtry = number of variables to use for each tree

Ensemble Methods: Random Forest: Variable Importance

- One of the really nice features associated with Random Forrest is it's ability to select variables that are most "important" to training the model.
- This is done by determining which variables are closest to the root of the tree and then moving outward or other methods associated with error reduction of the variables
- Random Forrest does this at scale and the output is generated as a part of the training process

Ensemble Methods: Random Forest: Variable Importance

- The Random Forest Package in R is basically a wrapper for the original fortran original program written by Leo Breiman and Adel Cutler from Berkeley
- Their original website is still quite useful:
https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm