Entropy

Entropy

- Entropy is the mathematical concept of randomness or disorder.
- An increase in entropy represents a loss of order.
- Thus a decrease in entropy (loss of disorder) is information gain.
    - This allows us to quantify how much “information gain” we get by making decisions.

Entropy Defined

For a **discrete** random variable X with outcomes x_1, …., x_n

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_b P(x_i)$$

- H is the entropy.
- P(x) is the probability of the outcome.
- b is the base and is typically 2, e, or 10.
  - b = 2 gives rise to the phrase “bits of entropy”

Entropy Example: Fair Coin Flip

- x_1 = heads, $P(x_1) = 0.50$
- x_2 = tails, $P(x_2) = 0.50$

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_b P(x_i)$$

[redacted]

[redacted] $-\frac{1}{2} \cdot (-1) - \frac{1}{2} \cdot (-1)$

[redacted] 1

Note that the choice of b = 2 is unrelated to the fact we have two classes.

Entropy Example: Weighted Coin Flip

- x_1 = heads, $P(x_1) = 0.60$
- x_2 = tails, $P(x_2) = 0.40$

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_b P(x_i)$$

[redacted]

[redacted] $-(0.60) \cdot (-0.737) - (0.40) \cdot (-1.322)$

[redacted] 0.971

If we compare, we see the entropy has decreased, thus information has been gained.

DataScience@SMU

Gini

Gini Impurity

- Gini Impurity was the original measure of information gain in decision trees.
  - Do not confuse with Gini coefficient, which is a term in economics.
- Gini Impurity is a measurement of probability of incorrect classification for a new sample.
  - In other words, if you have a set of labels, the Gini Impurity is the probability of a wrong assigned label.

$$G = \sum_{i=1}^{C} P(i) \cdot (1 - P(i))$$

Gini Impurity Example: Fair Coin Toss

- 50 heads and 50 tails
- Assign the labels to all 100 examples
- P(i) = 0.50

[redacted]

- G = 0.50

Gini Impurity Example: Weighted Coin Toss

- 40 heads and 60 tails
- Assign the labels to all 100 examples

[redacted]

- G = 0.48
- Once again, G is lowered, so information is gained

DataScience@SMU

Partition Trees

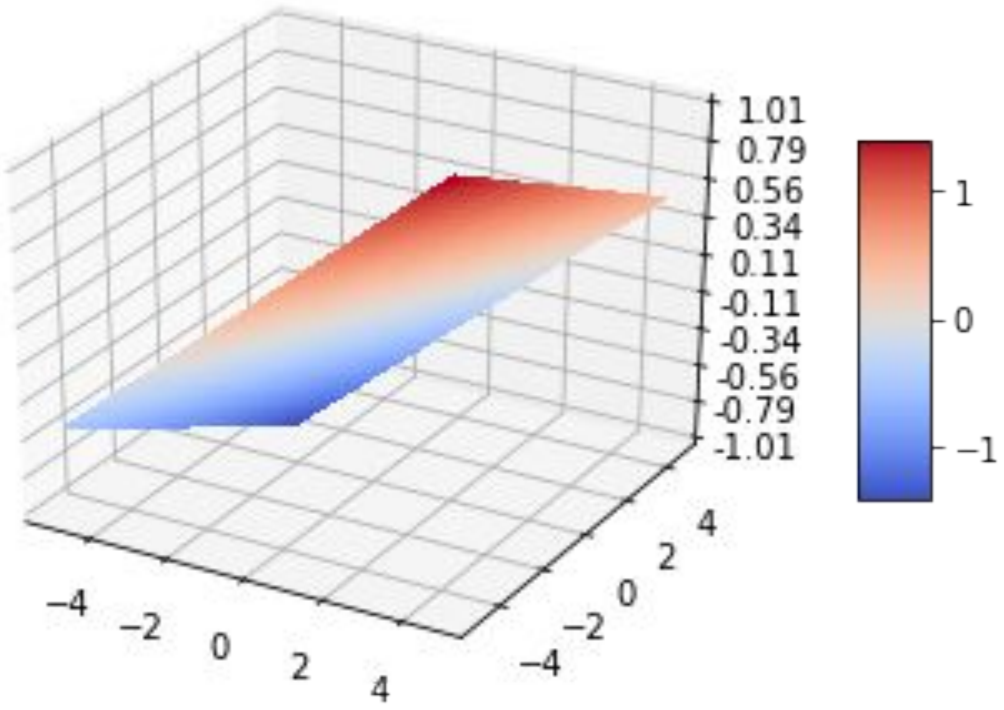
Partition Trees

- Partition trees form the basis of more modern and efficient algorithms like XGBoost and Random forest.
- It’s still important to understand how they work.
- The first method was called CART for Classification and Regression Trees.

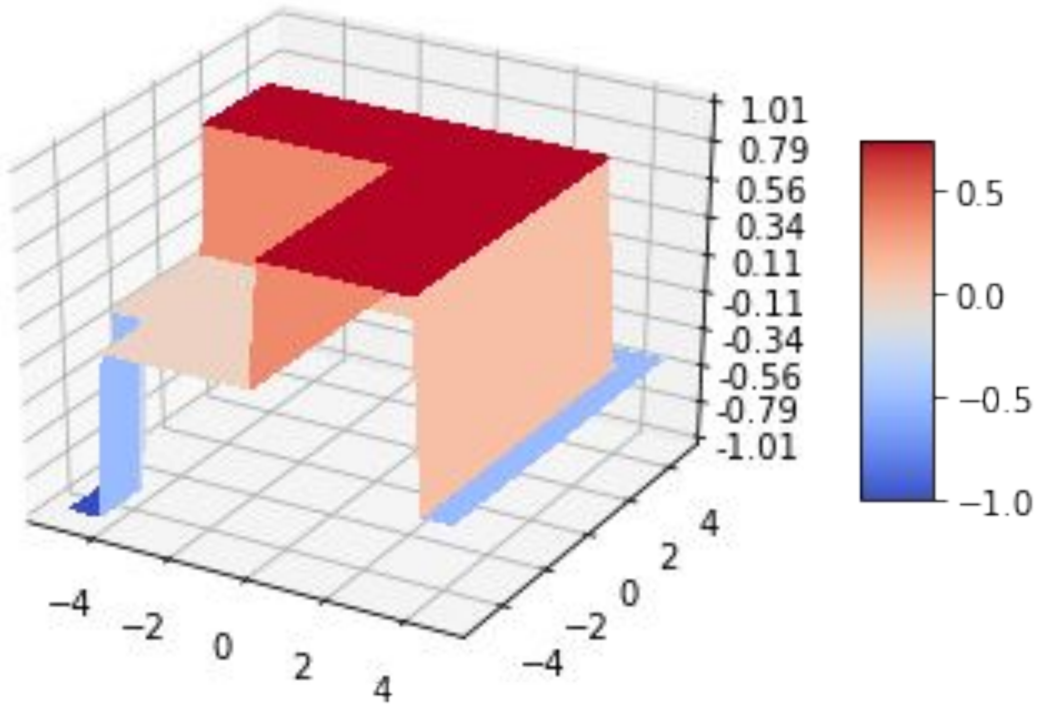
Partition Trees (cont.)

- What does a partition tree do?
  - A partition tree fits a nonlinear surface onto data.
  - In plain English: It is our first nonlinear classifier.
- Regression produces a predictive plane (in n dimension, it produces an n-dimensional plane).
- Trees produce multiple planes or surfaces.
- Let’s look at a picture to help.

Linear vs. Nonlinear



Linear model has a single continuous plane—all predictions are on this single plane.



Partition trees can have multiple surfaces or values. These surfaces can be as complex as your algorithm allows.

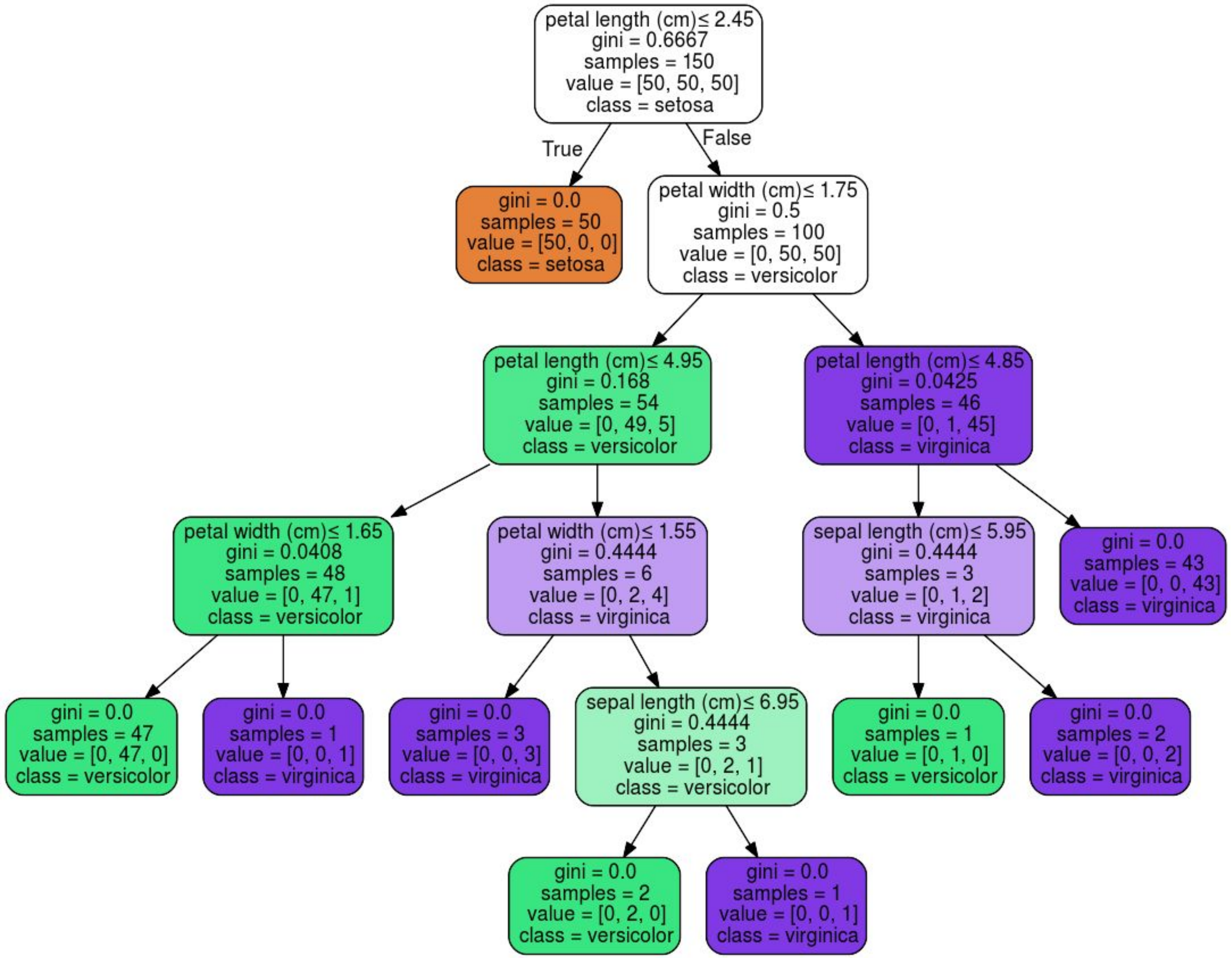
Partition Trees as Decision Trees

Partition trees “partition” the data into bins. The partitions are often called decisions.

- Example: Is the value of variable $x < 5$?
  - Yes: Go to bin A
  - No: Go to bin B

Example Decision Tree

The color of the box indicates the class (Iris dataset used). Using the left exit arrow indicates the test was true; using the right arrow indicates the test was false.



How Do We Find the Decision/Partition Boundaries?

- Sort through our data and make decisions based on the input values.
- The decision with the maximum information gain (Gini or entropy) becomes the rule.
- This process is repeated for each division until stopping criteria are met:
  - Complexity: the minimum amount of information (negative entropy, Gini) gain
  - Max depth: the number of decisions in a chain
  - Samples per X: The algorithm must have n number of samples to make a split, leaf or node

DataScience@SMU

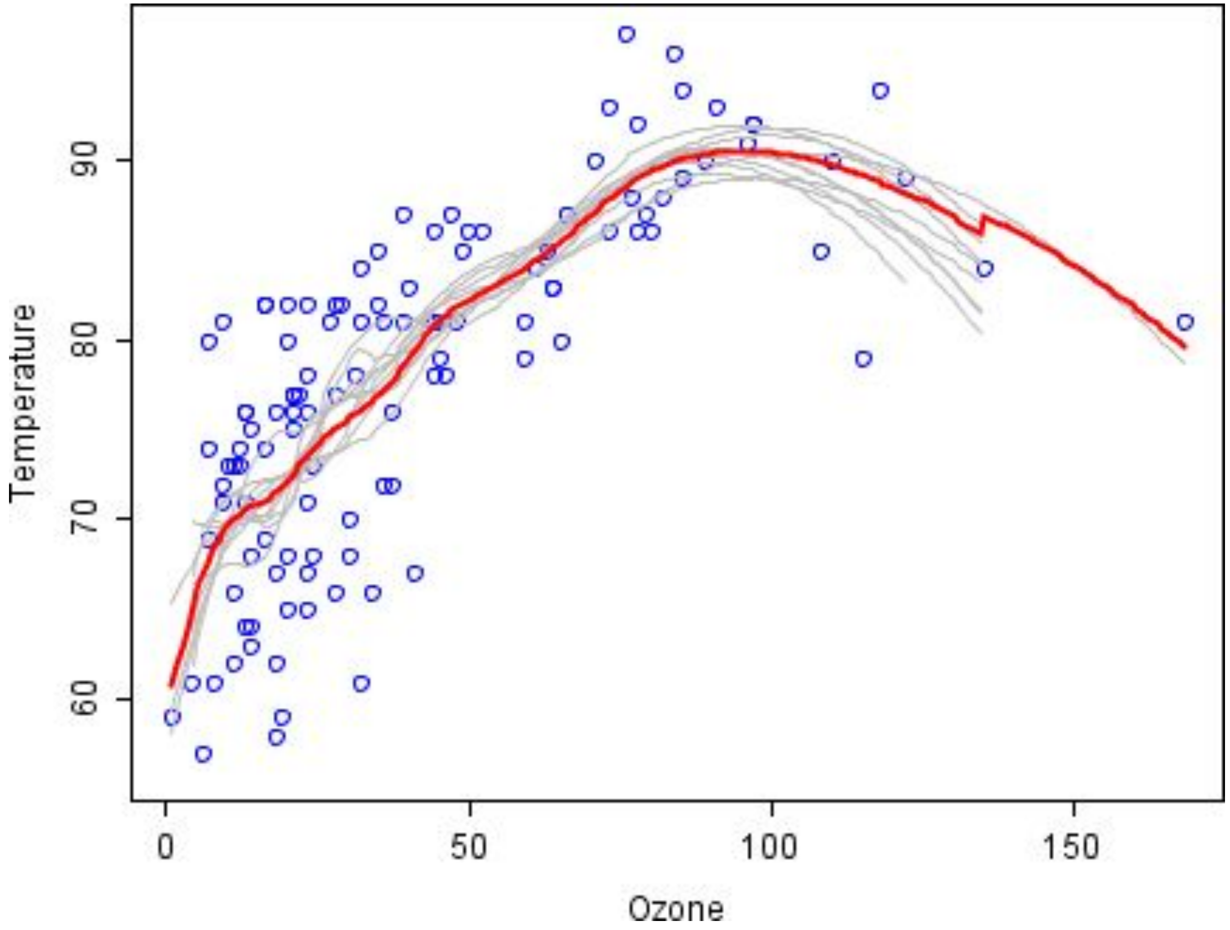
Bagging

Bootstrap Aggregation: Bagging

- Bagging is using sampling with replacement to build a small model.
- Using multiple models and averaging the final output produces a better general fit.

Example

- The individual gits in grey are quite noisy and display overfitting. By taking the average, a much more generalized fit is achieved.
- For a full mathematical treatment of bagging, please refer to: [Bagging Predictors]



DataScience@SMU

Random Forest

Random Forest

- Random forest is simply CART trees with bagging.
  - Pick a subset of data
  - Pick a subset of features to do splits by
  - Build multiple (default is about 100) classifiers and average the outcome
- The key idea is that each tree is uncorrelated by picking random subsets of the data.
- The bagging then combines all the trees' different "looks" of the data for a generalized fit.
- Random forest is your basic starting point.

DataScience@SMU