Introduction to Tree Methods

PIERIAN DATA
www.pieriandata.com

Reading Assignment

Chapter 8 of

Introduction to Statistical Learning

By Gareth James, et al.

Tree Methods

Let’s start off with a thought experiment to give some motivation behind using a decision tree method.

Tree Methods

Imagine that I play Tennis every Saturday and I always invite a friend to come with me.

Sometimes my friend shows up, sometimes not.

For him it depends on a variety of factors, such as: weather, temperature, humidity, wind etc..

I start keeping track of these features and whether or not he showed up to play with me.

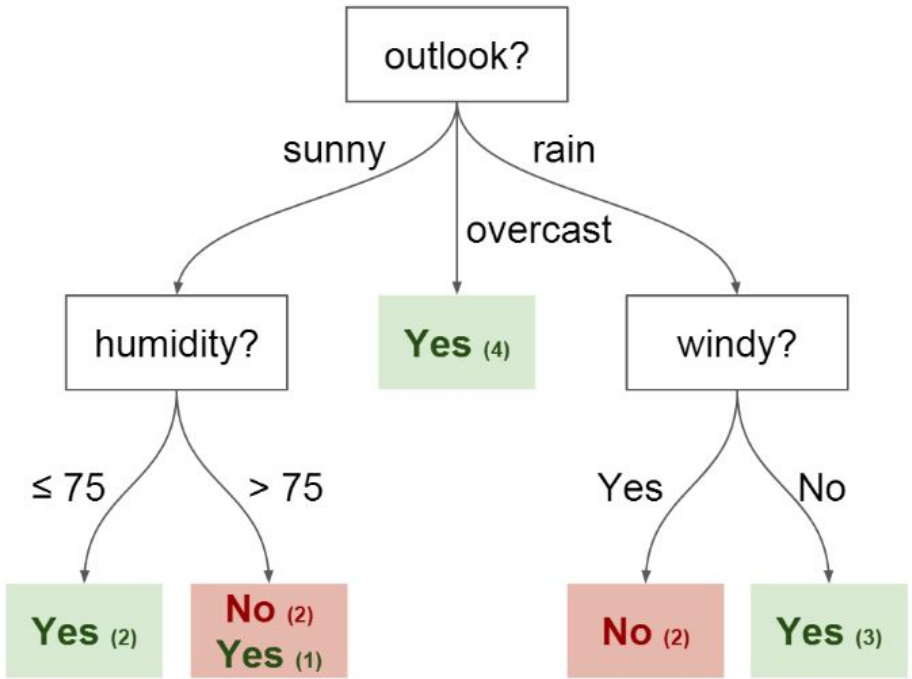
Tree Methods

Temperature	Outlook	Humidity	Windy	Played?
Mild	Sunny	80	No	Yes
Hot	Sunny	75	Yes	**No**
Hot	Overcast	77	No	Yes
Cool	Rain	70	No	Yes
Cool	Overcast	72	Yes	Yes
Mild	Sunny	77	No	**No**
Cool	Sunny	70	No	Yes
Mild	Rain	69	No	Yes
Mild	Sunny	65	Yes	Yes
Mild	Overcast	77	Yes	Yes
Hot	Overcast	74	No	Yes
Mild	Rain	77	Yes	**No**
Cool	Rain	73	Yes	**No**
Mild	Rain	78	No	Yes

Tree Methods

I want to use this data to predict whether or not he will show up to play.

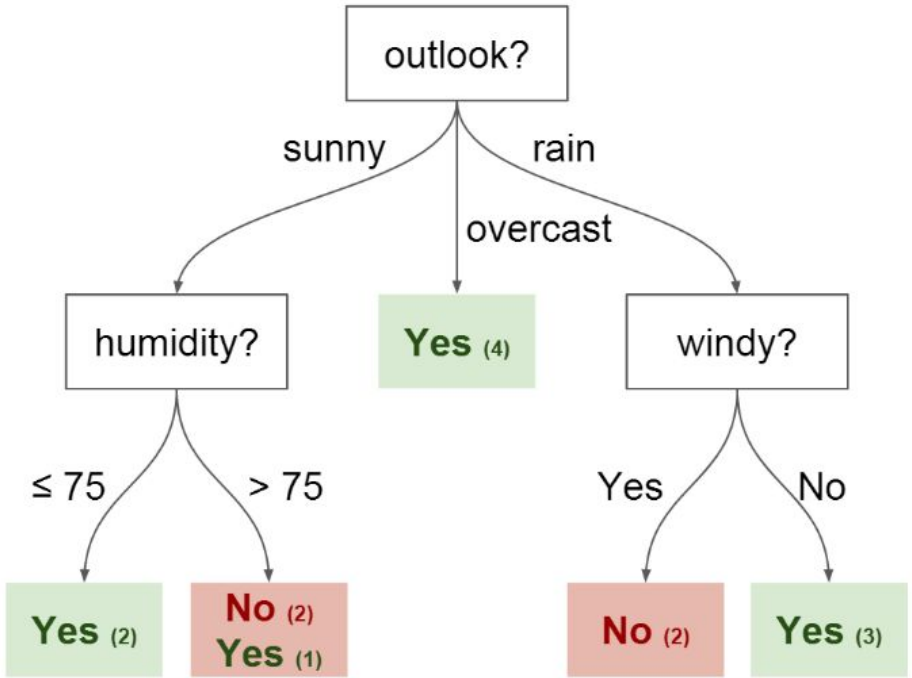
An intuitive way to do this is through a Decision Tree



Tree Methods

In this tree we have:

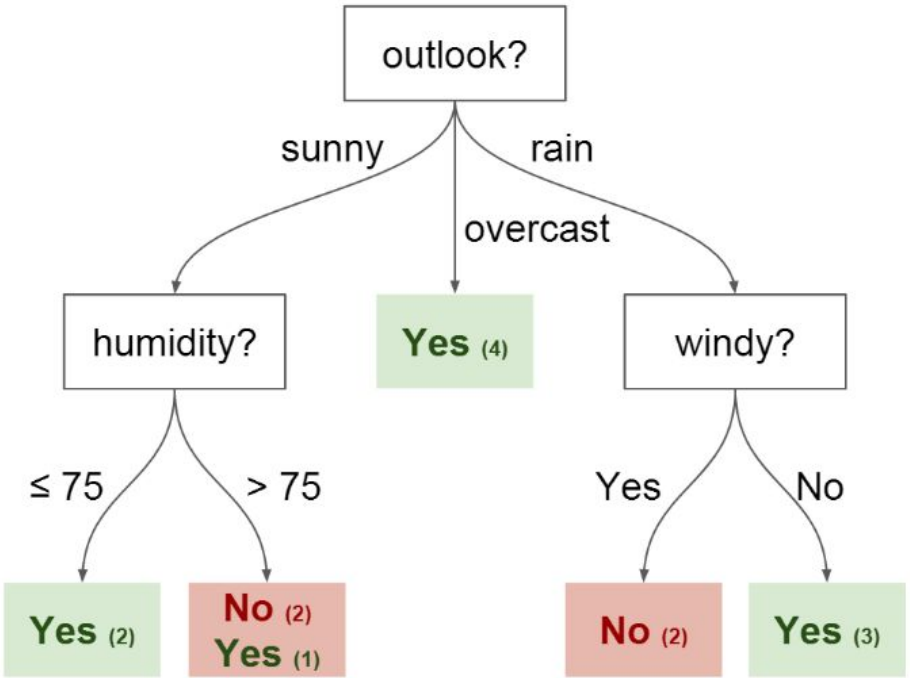
- Nodes
  - Split for the value of a certain attribute
- Edges
  - Outcome of a split to next node



Tree Methods

In this tree we have:

- Root
  - The node that performs the first split
- Leaves
  - Terminal nodes that predict the outcome



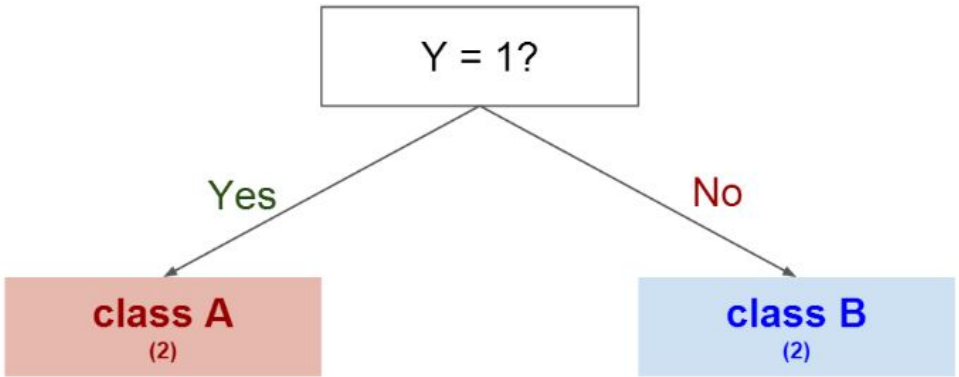
Intuition Behind Splits

Imaginary Data with 3 features (X,Y, and Z) with two possible classes.

X	Y	Z	Class
1	1	1	A
1	1	0	A
0	0	1	B
1	0	0	B

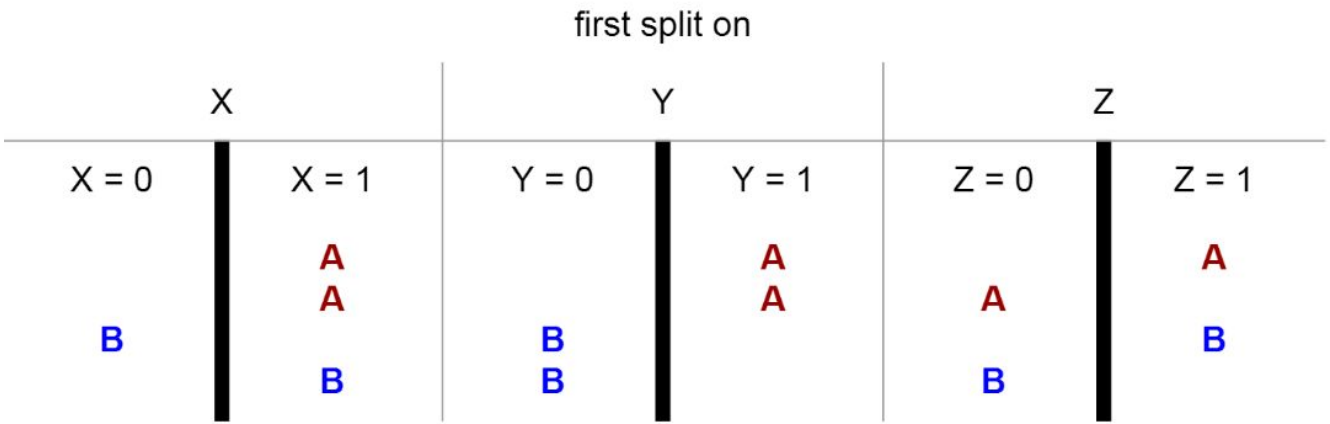
Intuition Behind Splits

Splitting on Y gives us a clear separation between classes



Intuition Behind Splits

We could have also tried splitting on other features first:



Intuition Behind Splits

Entropy and Information Gain are the Mathematical Methods of choosing the best split. Refer to reading assignment.

$$Entropy:$$

$$H(S) = -\sum_{i} p_i(S)\, log_2\, p_i(S)$$

$$Information\ Gain:$$

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{S} H(S_v)$$

Random Forests

To improve performance, we can use many trees with a random sample of features chosen as the split.

- A new random sample of features is chosen for **every single tree at every single split**.
- For **classification**, m is typically chosen to be the square root of p.

Random Forests

What's the point?

- Suppose there is **one very strong feature** in the data set. When using “bagged” trees, most of the trees will use that feature as the top split, resulting in an ensemble of similar trees that are **highly correlated**.

Random Forests

What's the point?

- Averaging highly correlated quantities does not significantly reduce variance.
- By randomly leaving out candidate features from each split, **Random Forests "decorrelates" the trees**, such that the averaging process can reduce the variance of the resulting model.

Example with R

Let's go to RStudio and begin to explore an example, then you'll have a project to test your understanding!