PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Machine Learning Techniques

DATASCI 420

Lesson 05-01 Decision Trees, Part 1

Common Types of Modeling Tasks

- Classification: categorization into types
- Scoring: predicting or estimating a value
- Ranking: Ordering items by affinities
- Clustering: grouping similar items
- Relations/correlations: determine potential causes of effects
- Characterization: report generation

Refresher on Supervised vs. Unsupervised Learning

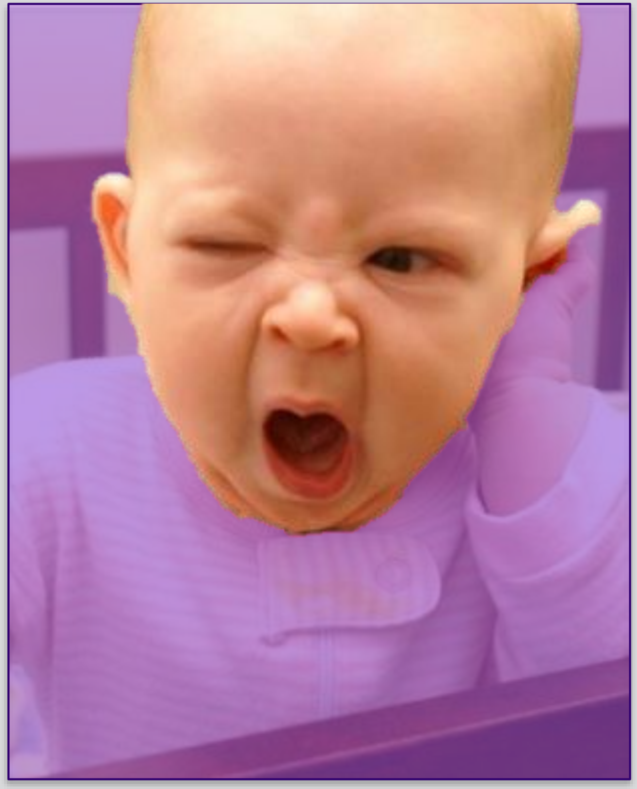
Supervised Learning

- **Definition**: Uses ‘labeled’ training data
- Each set of input features (could be single value or a vector) is accompanied by the “correct” classification or “signal”.

Supervised Learning Examples

- Fraud Detection: when we have examples in the data where fraud = True/False is known
- Patient Readmission: when we know which patients were readmitted to the hospital
- Recommendation Systems: when we know which items were presented to customers resulted in added to the cart or purchased

Supervised Learning for Face Recognition



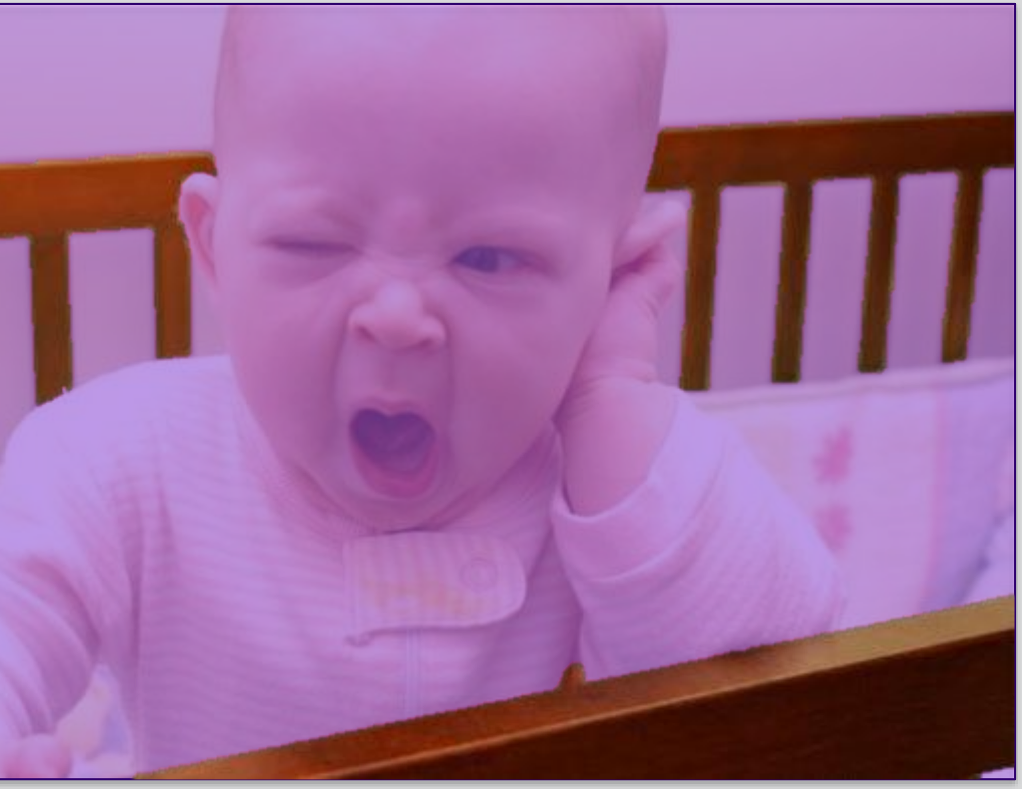
For example: a set of pixels in an image that represent a face. The non-face pixels are "False" and the face pixels are "True"

All of the characteristics of each pixel in this image are represented as input vector; the *Face* or *Not-Face* value is the "signal"

Unsupervised Learning

> Definition: unlabeled data is used to create a model *inferred* by the input vector; no correct classification is present.

Unsupervised Learning in Image Recognition



Left to it's own devices, an image recognition system might first look for sharp edges.

Decision Trees: Classification or Categorization

- What is a decision tree?
- When to use a decision tree (and when not to)
- How to create and tune them

Decision Trees: Definition and Popularity

- Decision trees are a commonly used type of supervised learning for *classification*
- Decision Trees enable automation of grouping “like” things using mathematical relationships between those classes and the input variables
- Decision trees are easy to interpret, understand, and explain
- They are “white box” in that each decision point (branch in the tree) clearly indicates not only the decision, but the path leading to each branch

Types of Decision Trees

- CART – Most commonly used
  - **Classification Trees**: where the target variable is categorical and the tree is used to identify the "class" of a target variable
  - **Regression Trees**: where target variable is continuous and the terminal nodes of the tree contain the predicted output variable values

NOTE: Both of these types can take categorical and continuous input variables

What about these?

Covered in upcoming lectures; in brief

- ID3 – The earliest decision tree algorithm (circa 1975)
- C4.5 and 5 – a faster “boosted” successor to ID3 that uses an ensemble (model on top of a model) to improve accuracy
- Gradient Boosted Trees – sequentially adding predictors to an ensemble each one correcting the predecessor
- Random Forest – trains on a random subset of features and then averages out their predictions

Which is best?

It’s best to try several techniques and see which best suits the domain and problem space. Don’t limit yourself to one.

Vocabulary

Root Node: Top of the tree—represents the entire dataset

Splitting (arc/edge): Process of dividing a node into two or more sub-nodes

Decision Node: A test on a single attribute that results in a split

Leaf/Terminal Node: Nodes that are not split

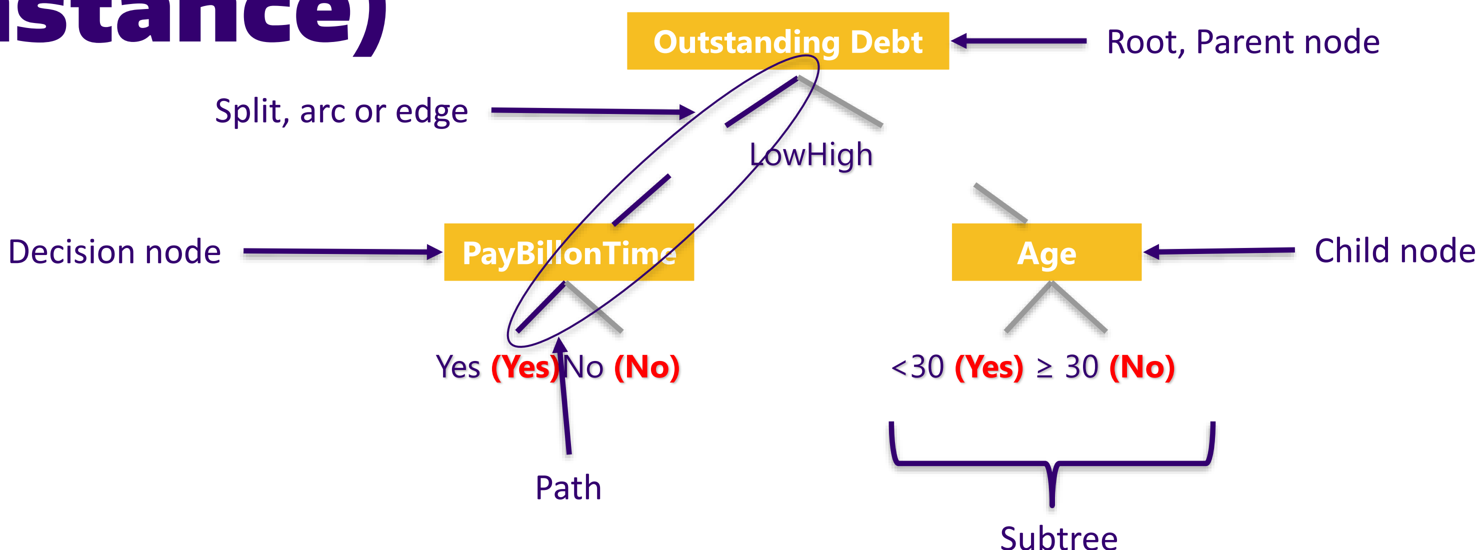
Pruning: When we remove sub-nodes of a decision node, this process is called pruning (ant. Splitting)

Branch/Sub-Tree: A section of entire tree is called branch or sub-tree

Parent/Child Nodes: A node, which is divided into sub-nodes is called a parent node; sub-nodes are the children of parent nodes

Path: a disjunction test that traverses the tree from root to a decision node

Decision trees classify instances by sorting them from the root of the tree to some leaf nodes (which provides the classification of the instance)



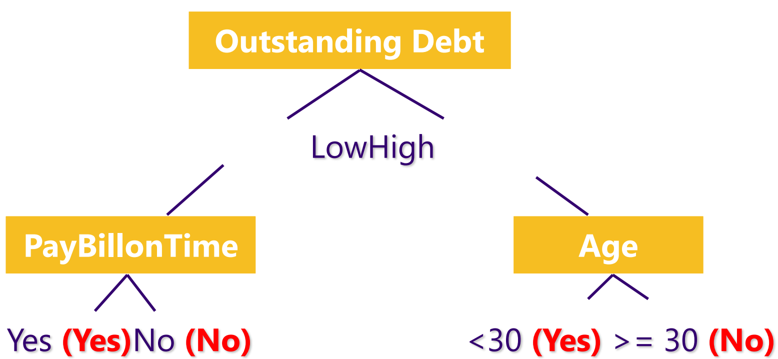
How do we split?

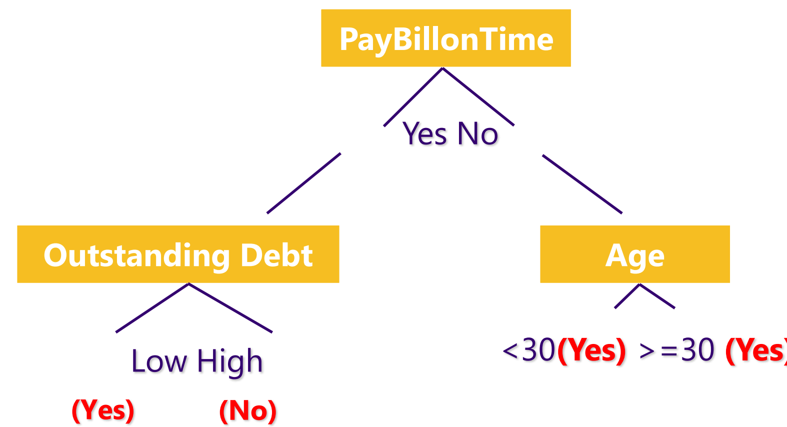
Which attribute should be used first?

Outstanding Debt?

Pay bill on time?

Age?





Determining the Split Attribute

- Each **attribute** (feature) is assigned a score
- The attribute which maximizes the score during each iteration is chosen as the **Split** attribute
- Different methods to compute the score. E.g.,
  - Gini Impurity (node homogeneity)
  - Chi-Square (statistical significance)
  - Information Gain (entropy)

Entropy

Binary Classification - Given a collection S, with (p) and failure (q) example of a target

$$Entropy(S) = -p\log_2 p \; - q\log_2 q$$

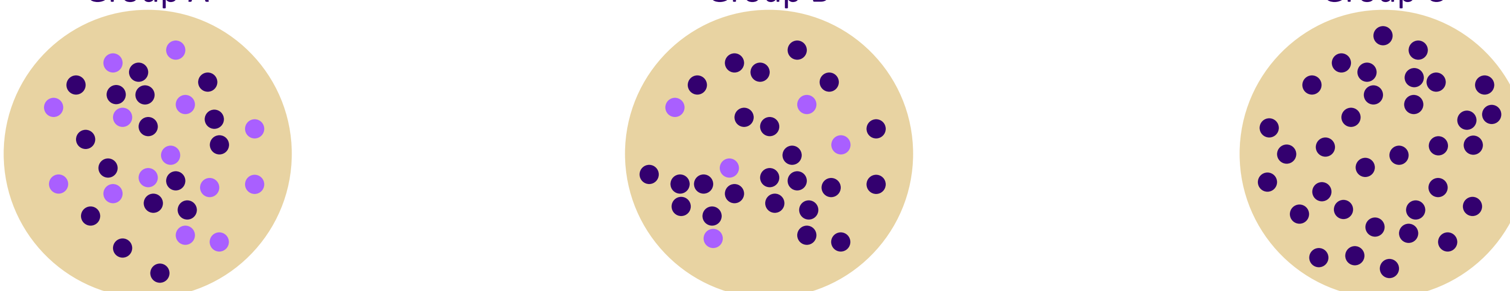
The formula: "–" (minus) percentage of success * (times) the $\log_2$ of the percentage of success subtracted from the percentage of failure * the $\log_2$ of the percentage of failure.

Log allows us to calculate based on two "states" at a time

Understanding Entropy

Consider the splits based on the three nodes below:

Group A | Group B | Group C



Entropy = 1
$-15/30*\log(15/30) - 15/30*\log(15/30)$

Entropy = 0.211632
$-25/30*\log(25/30) - 5/30*\log(5/30)$

Entropy = 0

Entropy is the degree of disorganization. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% – 50%), it has entropy of one.

Let’s build a theoretical decision tree

Example Training Data: Play Tennis

Day	Outlook	Temperature	Humidity	Wind	Play Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Start With the Entropy of Play vs. Not Play

- *S* is a collection of 14 examples
- 9 are positive examples (**playing**), 5 are negative examples (**not playing**)
  Notation: [9+, 5-]
- Entropy(*S*)
  $= -(9/14)\log_2 (9/14) - (5/14)\log_2(5/14)$
  $= 0.940$

Information Gain

Information Gain – is the expected reduction in entropy if attribute was chosen as the splitting attribute

= Entropy_before - Entropy_after

= 0.940 – Entropy_after_split

Four Attributes to Consider

Outlook, Temperature, Humidity and Wind

TEMPERATURE	Play = Yes	Play = No	
Hot	2/4	2/4	4/14
Mild	4/6	2/6	6/14
Cool	3/4	1/4	4/14

OUTLOOK	Play = Yes	Play = No	
Sunny	2/5	3/5	5/14
Overcast	4/4	0	4/14
Rain	3/5	2/5	5/14

HUMIDITY	Play = Yes	Play = No	
High	3/7	4/7	7/14
Normal	6/7	1/7	7/14

WIND	Play = Yes	Play = No	
Strong	3/9	3/5	6/14
Weak	6/9	2/5	8/14

Choosing the Best Split Variable

- Humidity – high

  $= \left(\frac{3}{7}\log_2\frac{3}{7}\right) - \left(\frac{4}{7}\log_2\frac{4}{7}\right)$

  $= .985$

- Humidity – normal

  $= \left(\frac{6}{7}\log_2\frac{6}{7}\right) - \left(\frac{1}{7}\log_2\frac{1}{7}\right)$

  $= .592$

- Gain = S, Humidity [+7, -7]

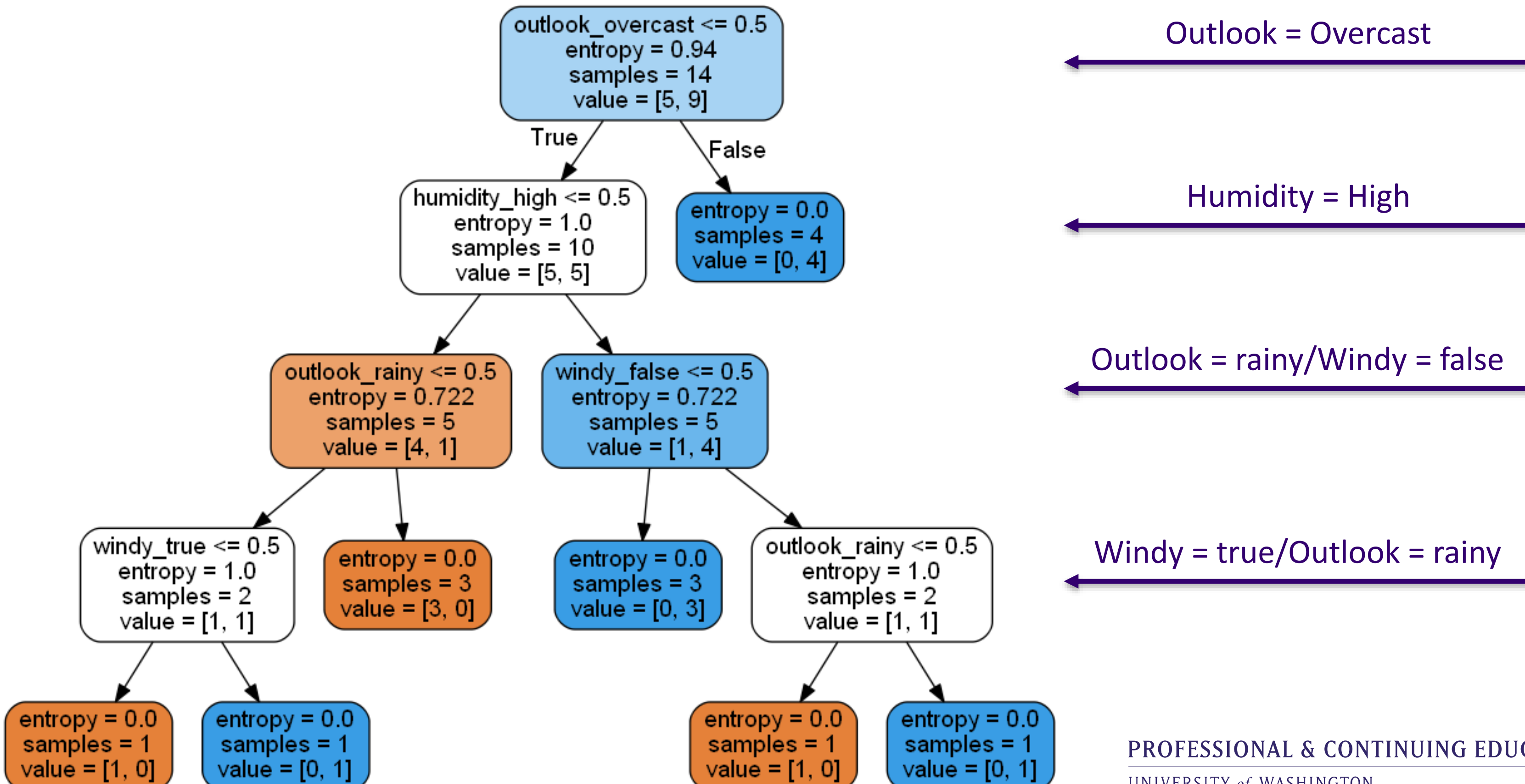
  $= .940 - \frac{7}{14}(.985) - \frac{7}{14}(.592)$

  $= .151$

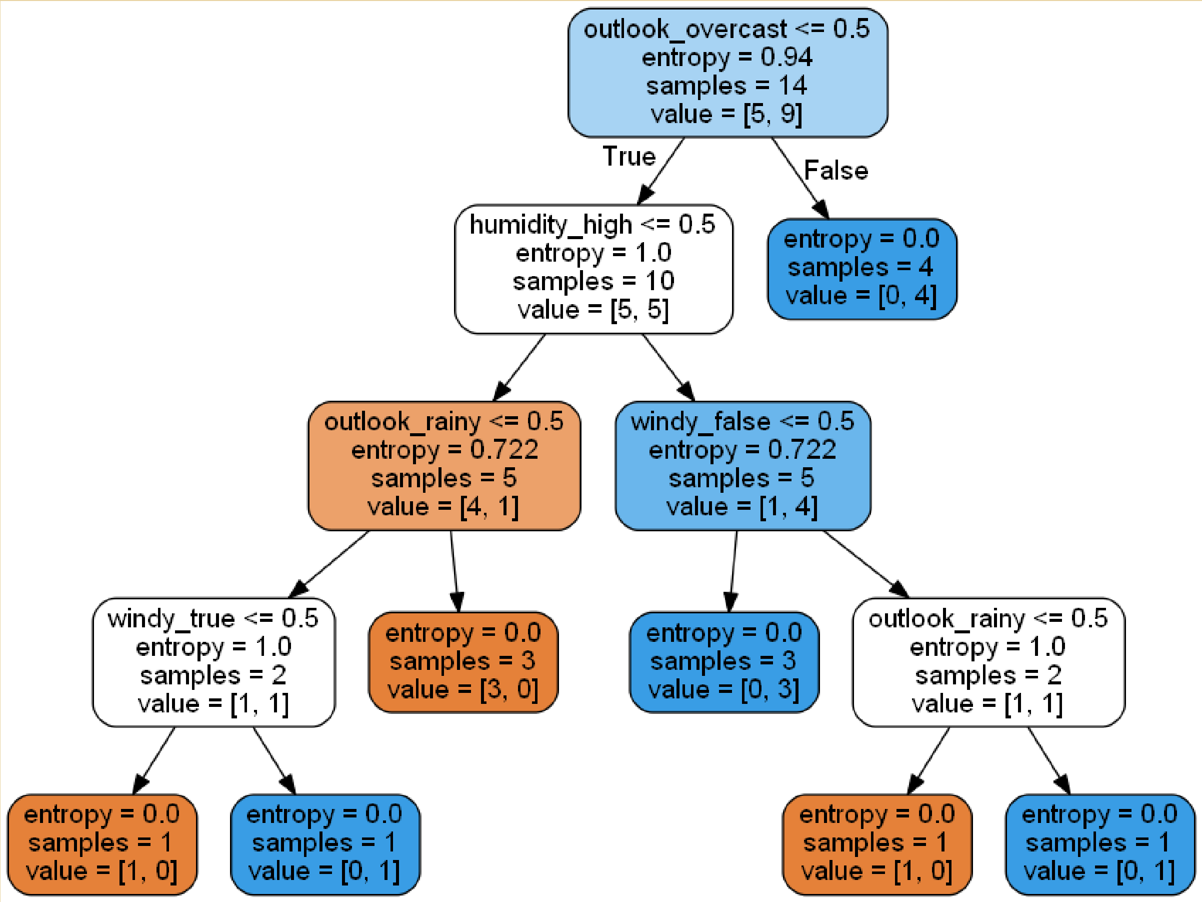
HUMIDITY	Play = Yes	Play = No	*S*, Humid
High	3/7	4/7	7/14
Normal	6/7	1/7	7/14

Rinse and repeat for every variable and state

Next Split – based on the previous split…



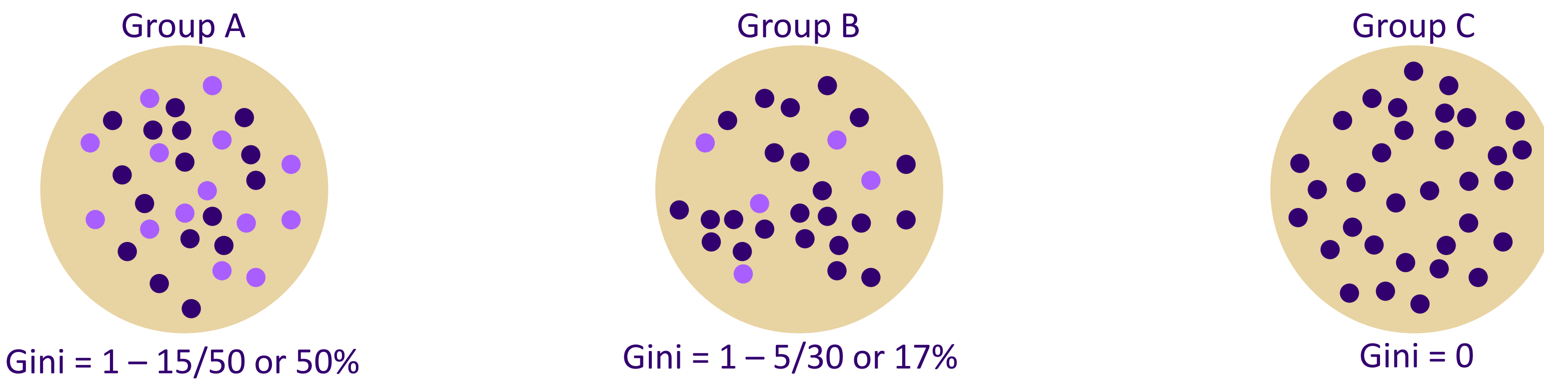
So of the input variables – which had the lowest predictive value?



TEMP	Play = Yes	Play = No	
Hot	2/4	2/4	4/14
Mild	4/6	2/6	6/14
Cool	3/4	1/4	4/14

Understanding Gini Impurity (Default)

Consider the splits based on the three nodes below:



The Gini Coefficient is the degree of node impurity.

$$G_i = 1 - \sum_{k=1}^{n} {P_{i,k}}^2$$

Gini vs. Entropy

- Usually create similar trees
- When the differ:
  - Gini tends to be faster (greedier) and isolates the most frequently occurring class in one side of the tree
  - Entropy tends to be more *balanced*
- It is worth testing BOTH

Let’s take a look at this example in a Jupyter Notebook