Decision Tree (Classification Tree)

Harry Jiannan Wang, Ph.D.

Professor of Management Information Systems

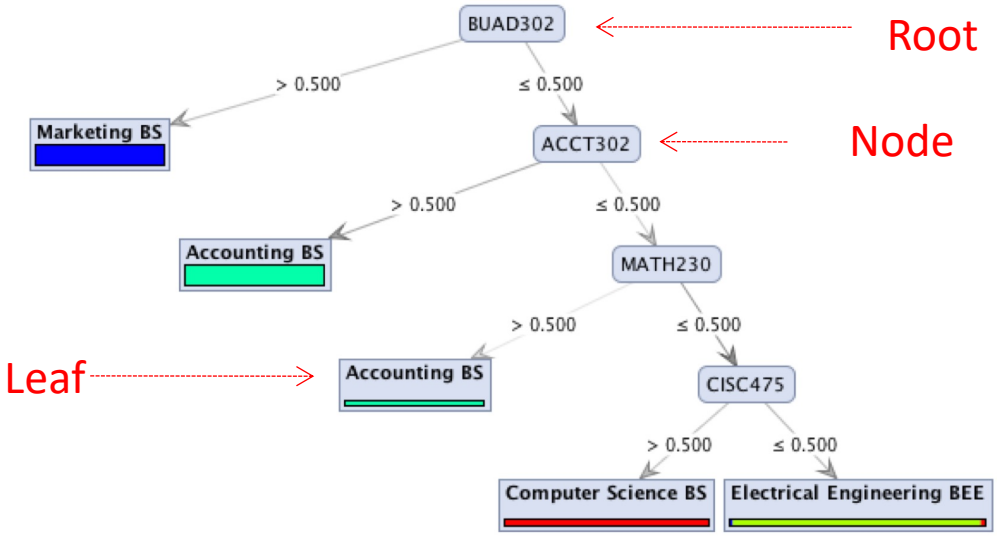
University of Delaware

Predictive Analytics

- **Predictive analytics** problems can be largely categorized into either **Class Prediction (Classification)** or **Numeric Prediction (Regression)** problems
- In classification, we try to use the information from the sample data to sort the data into *distinct classes*: (Classification Trees)
  - Predict whether a customer is a loyal or churn customer
  - Predict whether a loan is going to default or not
  - Predict whether a passenger of Titanic is going to survive or not
- In numeric prediction, we try to predict the *numeric value* of a target/label attribute (Regression Trees)
  - Predict the housing price
  - Predict how much a customer is going to spend in the next quarter
- Decision Trees can be used for both classification and regression predictions, which are often referred to as CART: **C**lassification **a**nd **R**egression **T**rees

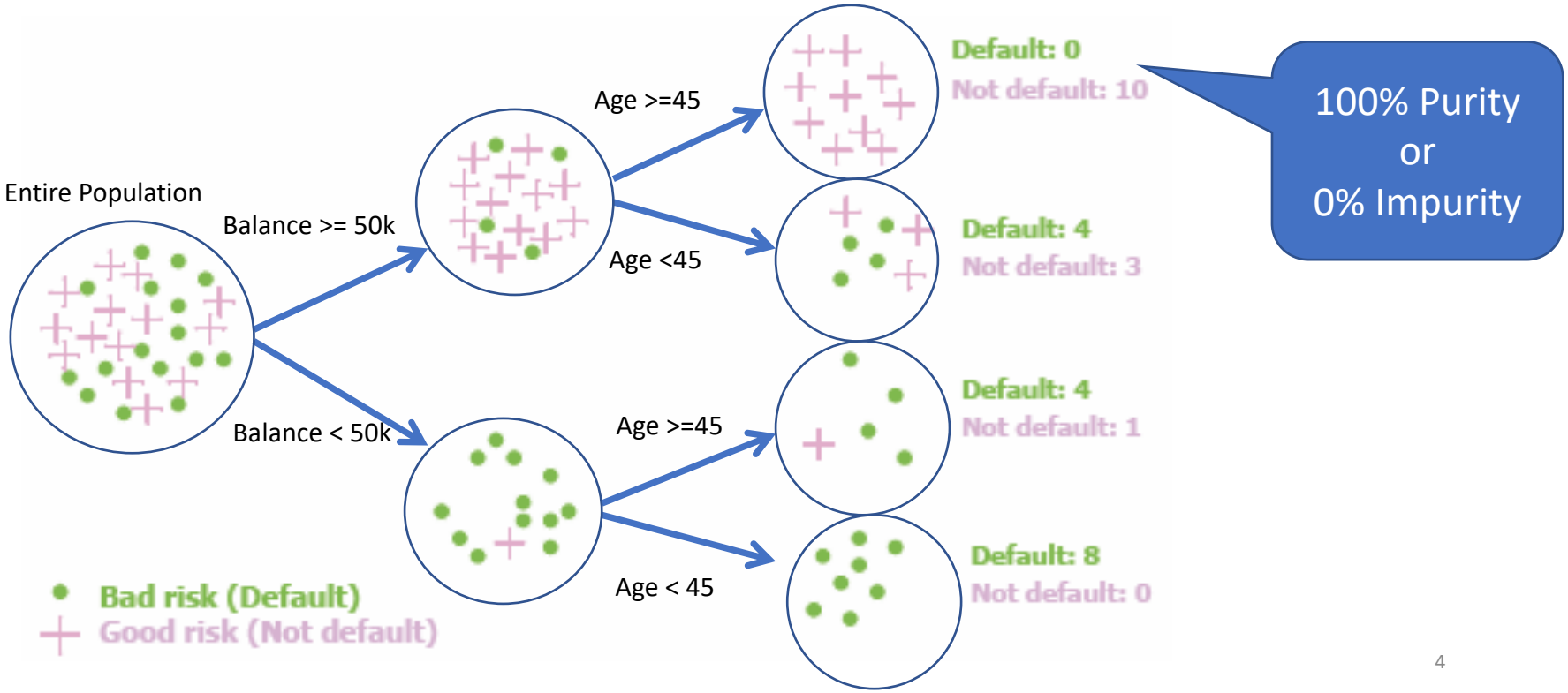
Classification Trees (Decision Trees)

- Decision Trees (DTs), or classification trees, are one of the most popular classification techniques
  - Easy to setup
  - Easy to interpret (especially for business users)
  - Computationally cheap
- Almost all data mining packages include DTs

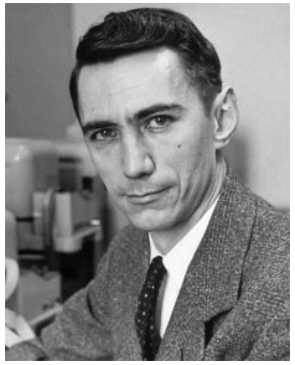


Homogeneity/Purity of Data

- The basic idea of a decision tree is to split data set based on the homogeneity (of the same kind) of data, i.e., reducing "Impurity"
- Impurity (uncertainty) is at maximum when all possible classes are equally represented, e.g., same number of "default" and "not default" in the following example



Entropy

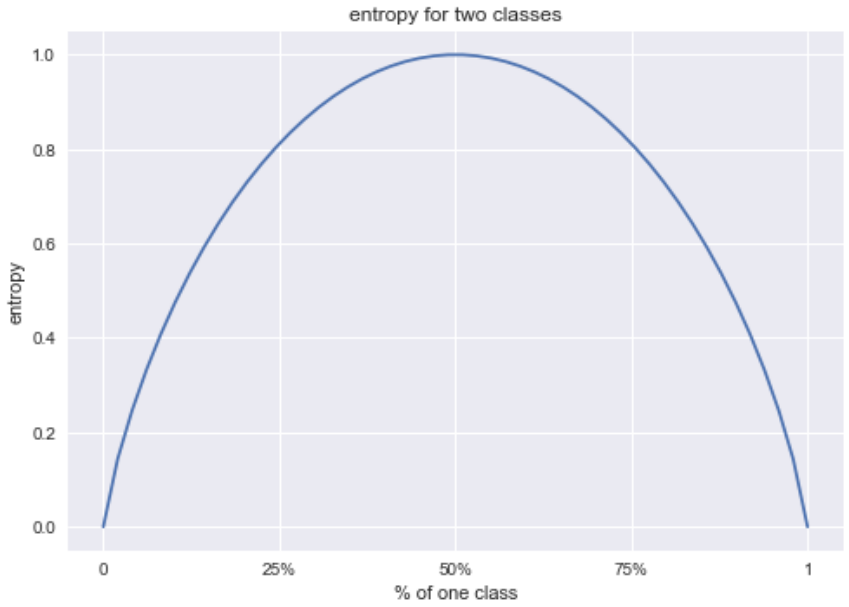


- Entropy is one of the most common measures for calculating impurity proposed by Claude Shannon known as "the father of information theory"

Entropy = $\sum_{i=1}^{n} -p(x_i)\log_2 p(x_i)$

$P(x_i)$ is the probability of class x_i in the data with total *n* classes

For example: churn dataset has two classes:
- "Default" (14 cases)
- "Not Default" (16 cases)



Entropy (entire dataset) =

$$-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$$

ID3 Decision Tree Algorithm

- Iterative Dichotomizer 3 (ID3)
  - ID3 was developed in 1986 by Ross Quinlan
  - Classic golf dataset: decide playing golf or not based on four attributes: Temperature, Humidity, Wind and Outlook
- Two key questions:
  - Where to split the data
  - When to stop splitting

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Sources:
https://en.wikipedia.org/wiki/ID3_algorithm
http://www.saedsayad.com/decision_tree.htm

Source: http://www.saedsayad.com/decision_tree.htm

Entropy Calculation using Frequency Table

- Calculate Entropy using the frequency table of the target variable:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

- Calculate Entropy given the frequency table of one feature variable and the target variable:

Target Feature

$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)
= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971
= 0.693

Information Gain/Entropy Reduction

- The information gain is based on the decrease in entropy after a dataset is split on an attribute (information gain = entropy reduction)
- Constructing a decision tree is all about finding attribute that returns the highest information gain at each split

Information Gain when splitting on Outlook attribute is 0.247

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)
= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971
= 0.693

- Choose attribute with the largest information gain as the decision node
- divide the dataset by its branches and repeat the same process on every branch

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Outlook

Sunny

Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

need for further splitting

Overcast

Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes

Already pure (Entropy is 0) No more splitting

Rainy

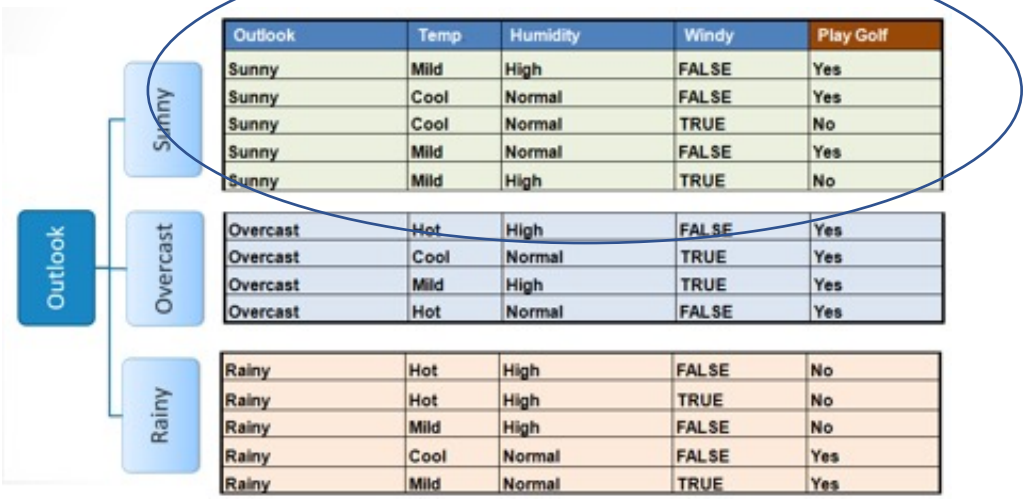
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

need for further splitting

- A branch with entropy of 0 is a leaf node.
- A branch with entropy more than 0 needs further splitting
- Repeat recursively on the non-leaf branches until all data is classified

Windy is chosen to be the next decision node here

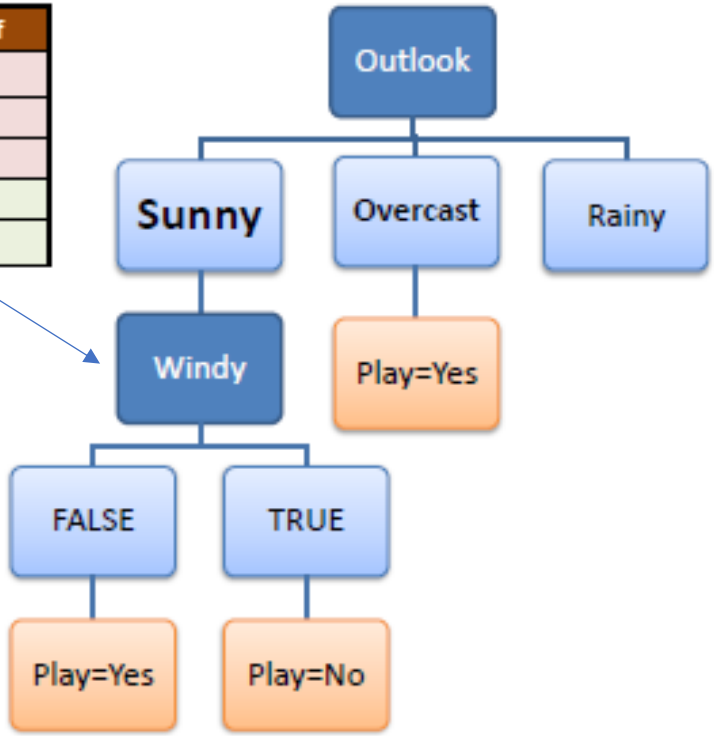
Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes

Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes



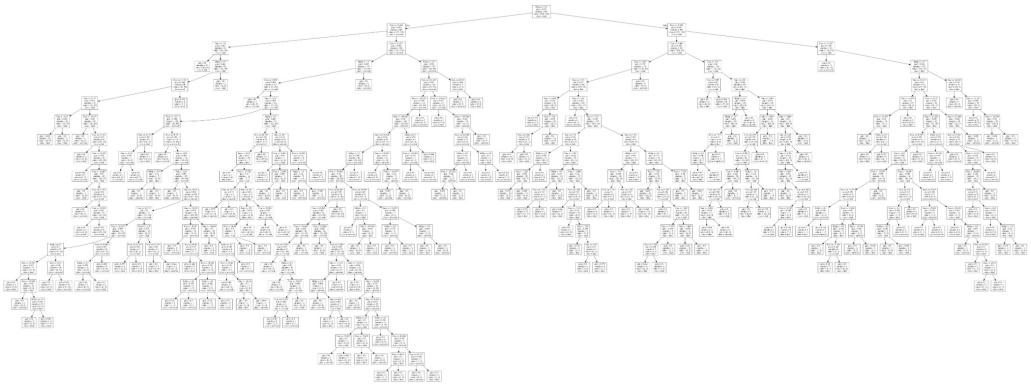
Split Points for Numeric Variables

- Method 1: use averages of available values
  - For Temperature: first split at Avg(64, 65) = 64.5, then Avg(65, 68) = 66.5, etc.
- Method 2: discretize values via “binning”
  - For temperature: >= 80 is “Hot”
    between 70 and 79 is “Mild”
    less than 70 is “Cool”

Temper...	Humidity
64	65
65	70
68	80
69	70
70	96
71	80
72	95
72	90
75	80
75	70
80	90
81	75
83	78
85	85

When to Stop?

- ID3 algorithm may lead to a very complex tree as shown below, which may work well for the training data but has poor prediction performance for the unseen data, which is called overfitting (we will discuss this more in the future)



- Pruning can be used to reduce overfitting:
  - Pre-pruning (stop growing the tree), such as setting max depth or minimal information gain
  - Post-pruning (fully grow the tree then remove unimportant nodes), such as Minimal Cost-Complexity Pruning https://scikit-learn.org/stable/modules/tree.html#minimal-cost-complexity-pruning

Questions?