Lecture 9: Decision Tree

Shuai Li

John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

https://shuaili8.github.io/Teaching/VE445/index.html

Motivation

Example 1

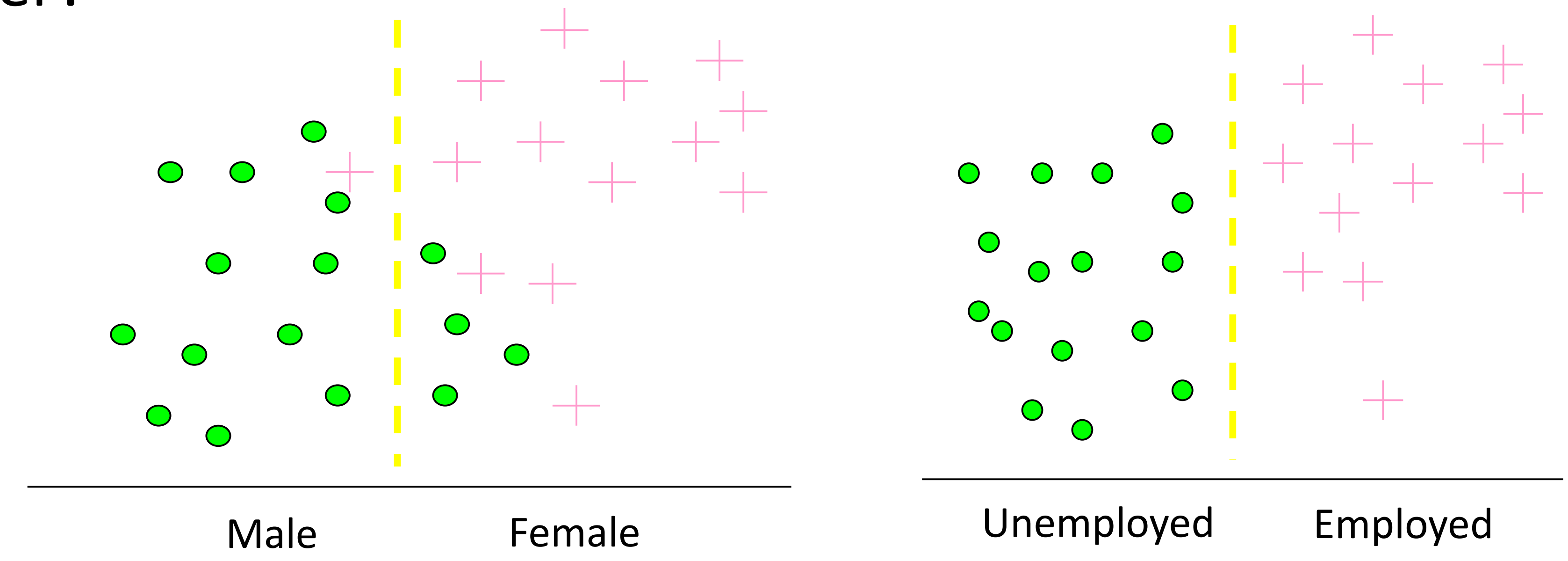
- Suppose you are a police officer and there was a robbery last night. There are several suspects and you want to find the criminal from them by asking some questions.

- You may ask: where are you last night?
- You are not likely to ask: what is your favorite food?

- Why there is a preference for the policeman? Because the first one can distinguish the guilty from the innocent. It is more informative.

Example 2

- Suppose we have a dataset of two classes of people. Which split is better?



- We prefer the right split because there is no outliers and it is more certain.

Entropy

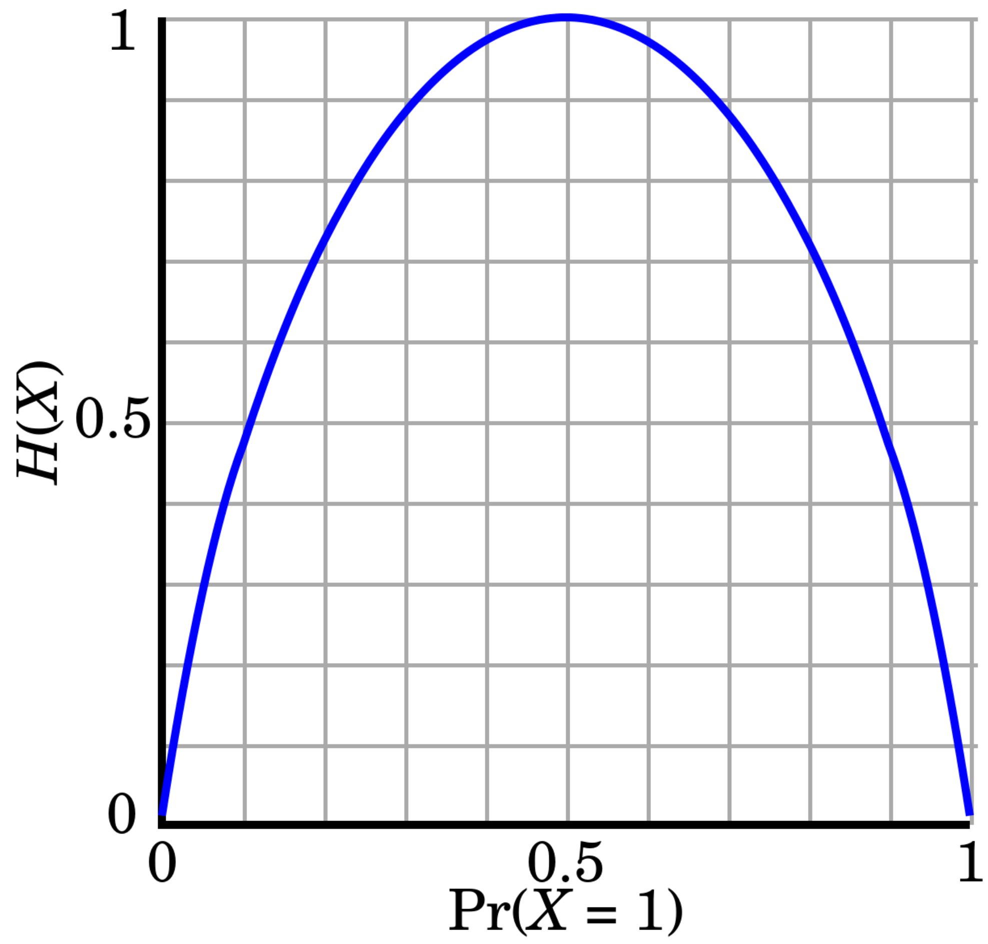
Entropy and uncertainty

- How to measure the level of informative (first example) and level of certainty (second example) in mathematics?

- The answer is Entropy
- Entropy = $\sum -p_i \log_2(p_i)$

Entropy

- Entropy $H(X) = \sum -p_i \log_2(p_i)$
  - p_i is the probability of class i, or that the proportion of class i in the set
  - Entropy is a measure of information, or uncertainty



Entropy distribution for a two-class example

Interpretation

- Most efficient code method (such as Hoffman code) assigns $-\log_2 P(x=i)$ bits to encode the message $x=i$

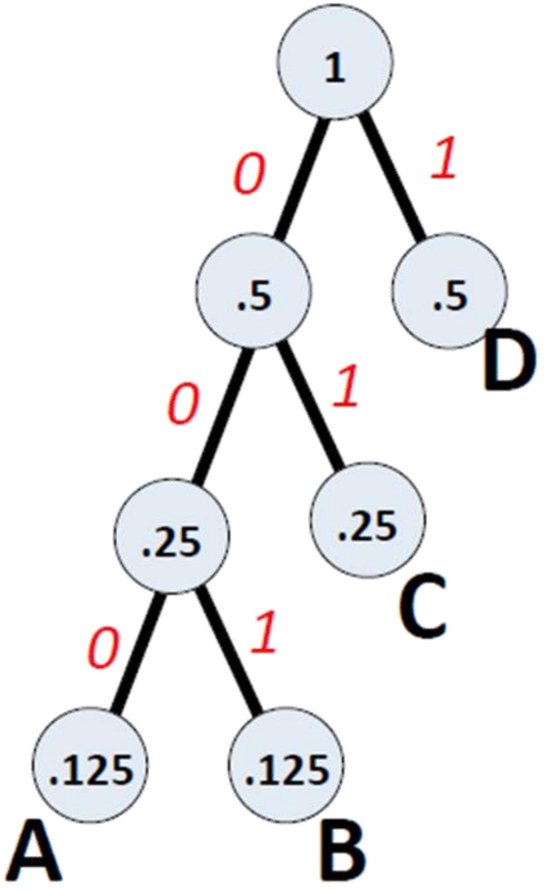
- So the expected number of bits to code a distribution p is $\sum -p_i \log_2(p_i)$

Example - Huffman code

- In 1952, MIT student David Huffman devised, in the course of doing a homework assignment, an elegant coding scheme which is optimal in the case where all symbols' probabilities are integral powers of $1/2$
- A Huffman code can be built in the following manner:
    - Rank all symbols in increasing order of probability of occurrence
    - Successively combine the two symbols of the lowest probability to form a new composite symbol; eventually we will build a binary tree where each node is the probability of all nodes beneath it
    - Trace a path to each leaf, noticing direction at each node

Example - Huffman code (cont.)

M	P
A	.125
B	.125
C	.25
D	.5



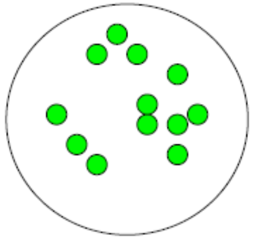
M	code	length	prob	
A	000	3	0.125	0.375
B	001	3	0.125	0.375
C	01	2	0.250	0.500
D	1	1	0.500	0.500
average message length				1.750

If we use this code to many messages (A,B,C or D) with this probability distribution, then, over time, the average bits/message should approach 1.75

Example – Two specific distributions

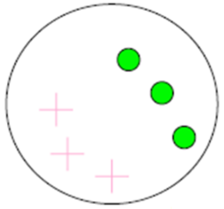
- What is the entropy of a group in which all examples belong to the same class?
  - entropy = - 1 $\log_2 1$ = 0

Minimum uncertainty



- What is the entropy of a group with 50% in either class?
  - entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1

Maximum uncertainty



Decision Tree

Information gain

- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned
- Information gain tells us how important a given attribute of the feature vectors is
  - Is used to decide the ordering of attributes in the nodes of a decision tree

- *Information Gain = Initial Entropy – Entropy with new information*

Example

- Given a dataset of 8 students about whether they like the famous movie *Gladiator*, calculate the entropy in this dataset

Like
Yes
No
Yes
No
No
Yes
No
Yes

Example (cont.)

- Given a dataset of 8 students about whether they like the famous movie *Gladiator*, calculate the entropy in this dataset

Like
Yes
No
Yes
No
No
Yes
No
Yes

$$E(Like) = -\frac{4}{8}\log\left(\frac{4}{8}\right) - -\frac{4}{8}\log\left(\frac{4}{8}\right)=1$$

Example (cont.)

- Suppose we now also know the gender of these 8 students, what is the new Entropy?

Gender	Like
Male	Yes
Female	No
Male	Yes
Female	No
Female	No
Male	Yes
Male	No
Female	Yes

Example (cont.)

- Suppose we now also know the gender of these 8 students, what is the new Entropy?
- The labels are divided into two small dataset based on the gender

Gender	Like
Male	Yes
Female	No
Male	Yes
Female	No
Female	No
Male	Yes
Male	No
Female	Yes

Like (male)
Yes
Yes
Yes
No

P(Yes | male) = 0.75

Like(female)
No
No
No
Yes

P(Yes | female) = 0.25

Example (cont.)

- Suppose we now also know the gender of these 8 students, what is the new Entropy?
  - P(Yes | male) = 0.75
  - P(Yes | female) = 0.25
  - $E(Like|male)$
    $= -\frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{3}{4}\log\left(\frac{3}{4}\right)$
    $= -0.25 * -2 - 0.75 * -0.41 = 0.81$
  - $E(Like|female)$
    $= -\frac{3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right)$
    $= -0.75 * -0.41 - 0.25 * -2 = 0.81$

Gender	Like
Male	Yes
Female	No
Male	Yes
Female	No
Female	No
Male	Yes
Male	No
Female	Yes

Example (cont.)

- Suppose we now also know the gender of these 8 students, what is the new Entropy?
  - $E(Like|\text{Gender})$
    $= E(Like|male) * P(male)$
    $\quad +E(Like|female) * P(female)$
    $= 0.5 * 0.81 + 0.5 * 0.81 = 0.81$
  - $\text{IG(Gender)} = \text{E(Like)} - \text{E(Like|Gender)}$
    $= 1 - 0.81 = 0.19$

Gender	Like
Male	Yes
Female	No
Male	Yes
Female	No
Female	No
Male	Yes
Male	No
Female	Yes

Example (cont.)

- Suppose we now also know the major of these 8 students, what about the new Entropy?

Major	Like
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Example (cont.)

- Suppose we now also know the major of these 8 students, what about the new Entropy?
- Three datasets are created based on major

Major	Like
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Like (math)
Yes
No
No
Yes

P(Yes|math) = 0.5

Like(history)
No
No

P(Yes|history) = 0

Like(cs)
Yes
Yes

P(Yes|cs) = 1

Example (cont.)

- Suppose we now also know the major of these 8 students, what about the new Entropy
  - $E(Like|Math) = -\frac{2}{4}\log\left(\frac{2}{4}\right) - \frac{2}{4}\log\left(\frac{2}{4}\right)$=1
  - $E(Like|CS) = -\frac{2}{2}\log\left(\frac{2}{2}\right) - \frac{0}{2}\log\left(\frac{0}{2}\right)$=0
  - $E(Like|history) = -\frac{2}{2}\log\left(\frac{2}{2}\right) - \frac{0}{2}\log\left(\frac{0}{2}\right)$=0
  - $E(Like|Major)$
    $= E(Like|math) * P(math)$
    $+E(Like|History) * P(History)$
    $+E(Like|cs) * P(cs)$
    $= 0.5 * 1 + 0.25 * 0 + 0.25 * 0 = 0.5$
  - $\mathrm{IG(Major) = E(Like) - E(Like|Major)}$
    $= 1 - 0.5 = 0.5$

Major	Like
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Example (cont.)

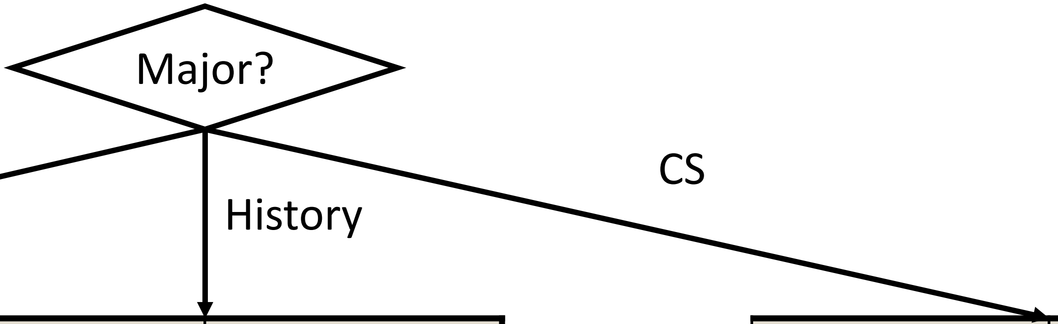
- Combine gender and major together
- As we have computed:
  - $\mathrm{IG(Gender)} = \mathrm{E(Like)} - \mathrm{E(Like|Gender)} = 1 - 0.81 = 0.19$
  - $\mathrm{IG(Major)} = \mathrm{E(Like)} - \mathrm{E(Like|Major)} = 1 - 0.5 = 0.5$

- Major is the better feature to predict the label “like”

Gender	Major	Like
Male	Math	Yes
Female	History	No
Male	CS	Yes
Female	Math	No
Female	Math	No
Male	CS	Yes
Male	History	No
Female	Math	Yes

Example (cont.)

- Major is used as the decision condition and it splits the dataset into three small one based on the answer



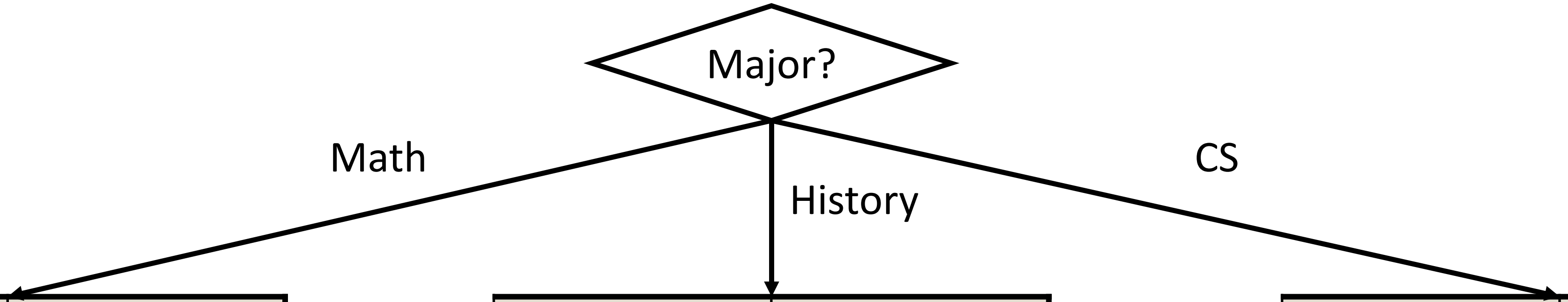
Gender	Like
Male	Yes
Female	No
Female	No
Female	Yes

Gender	Like
Female	No
Male	No

Gender	Like
Male	Yes
Female	Yes

Example (cont.)

- The history and CS subset contain only one label, so we only need to further expand the math subset

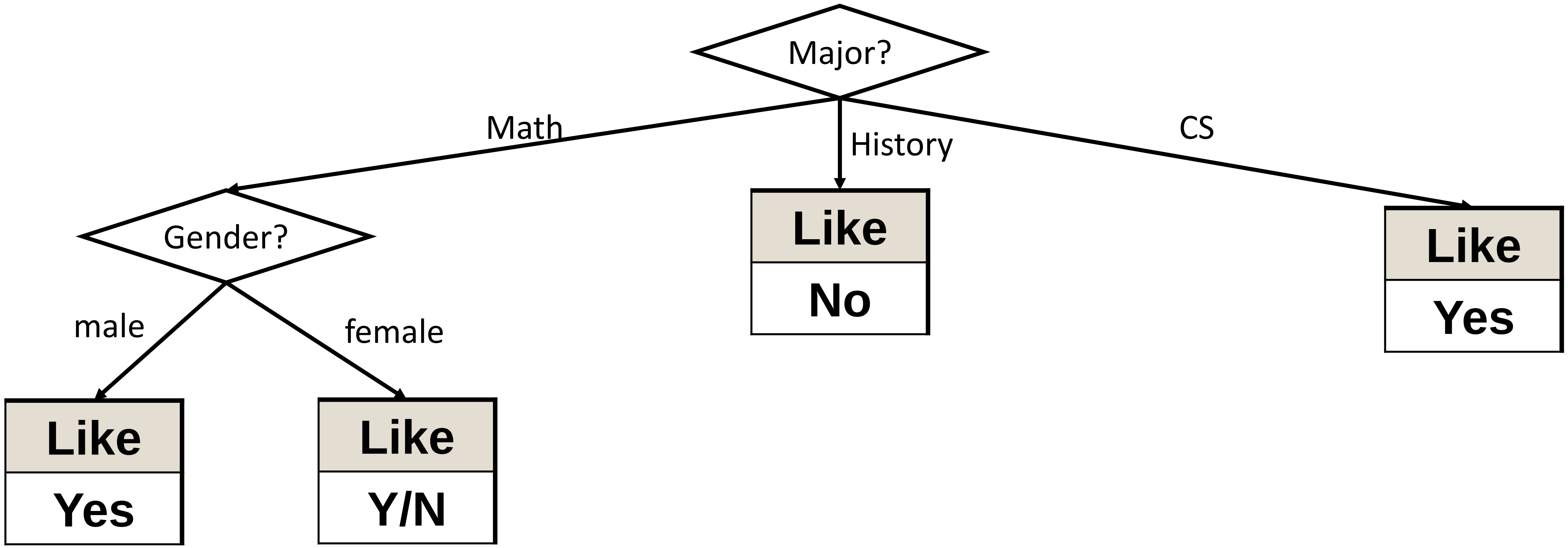


Gender	Like
Male	Yes
Female	No
Female	No
Female	Yes

Gender	Like
Female	No
Male	No

Gender	Like
Male	Yes
Female	Yes

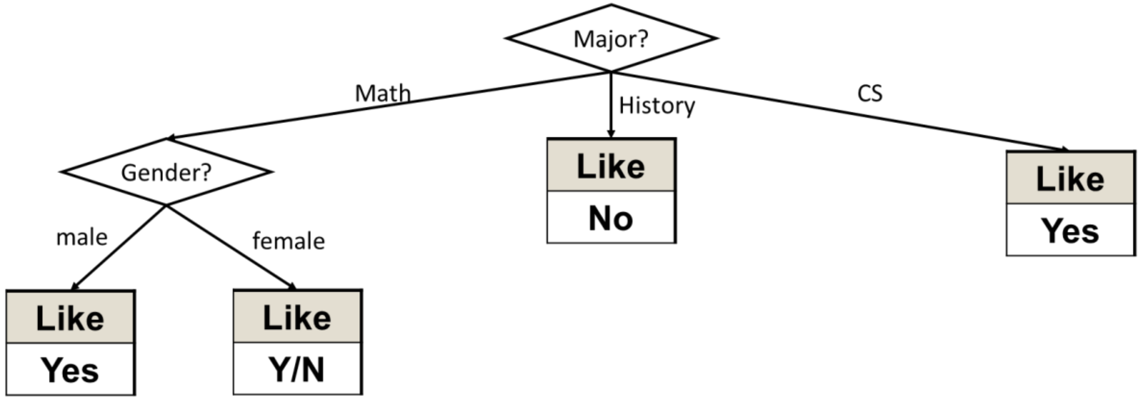
Example (cont.)



Example (cont.)

- In the stage of testing, suppose there come a female students from the CS department, how can we predict whether she like the movie Gladiator?
  - Based on the major of CS, we will directly predict she like the movie.

  - What about a male student and a female student from math department?



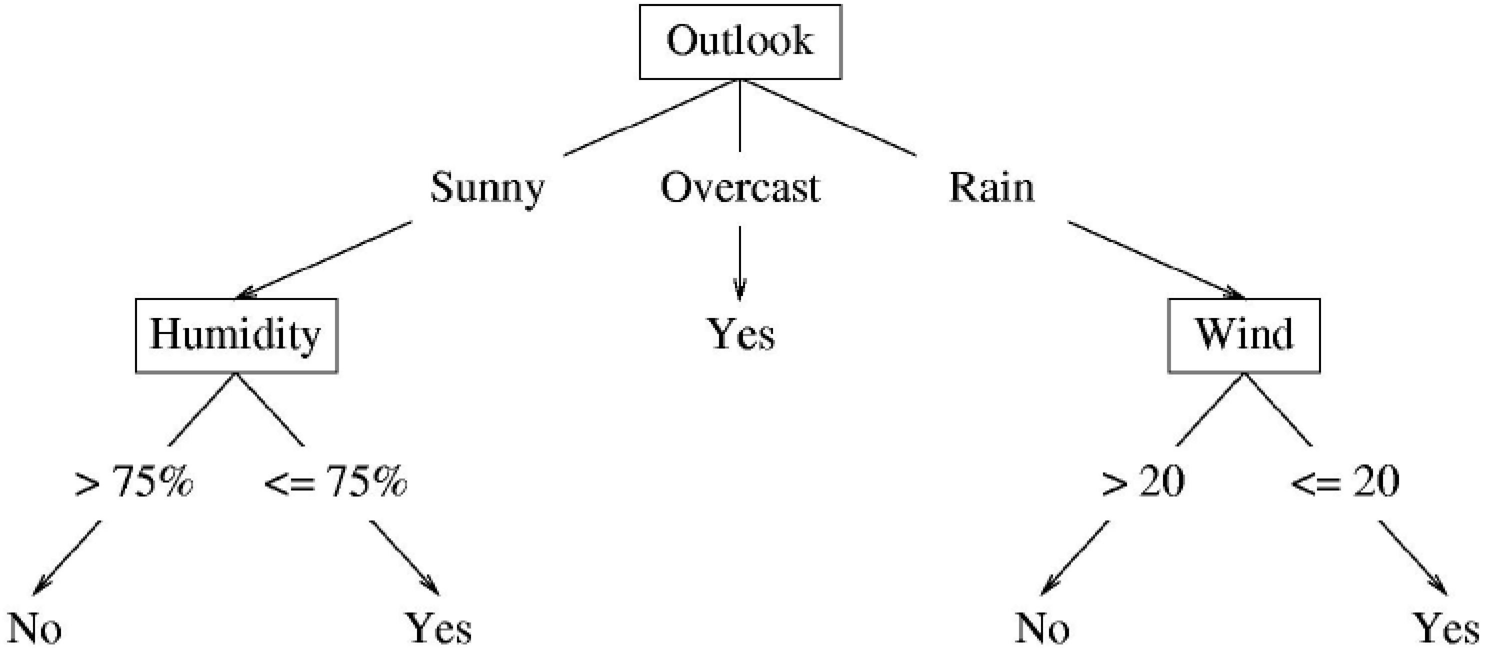
Summary

- During the training stage:
  - For a given dataset, the DT algorithm repeats the steps in the previous example, until the sub-dataset become non-dividable

- During the testing stage:
  - For a given sample, the DT algorithms go through the nodes in the tree based on the answer to each node

Continuous feature

- If features are continuous, internal nodes can test the value of a feature against a threshold



Standard Deviation

Continuous label (regression)

- Previously, we have learned how to build a tree for classification, in which the labels are categorical values
- The mathematical tool to build a classification tree is entropy in information theory, which can only be applied in categorical labels
- To build a decision tree for regression (in which the labels are continuous values), we need new mathematical tools
- The answer is standard deviation
- https://www.saedsayad.com/decision_tree_reg.htm

Standard deviation

- Standard deviation is used to calculate the homogeneity of a numerical sample. If the numerical sample is completely homogeneous its standard deviation is zero

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Standard deviation example

Hours Played
26
30
48
46
62
23
43
36
38
48
48
62
44
30

$$Count = n = 14$$

$$Average = \bar{x} = \frac{\sum x}{n} = 39.8$$

$$Standard\ Deviation = S = \sqrt{\frac{\sum(x - \bar{x})^2}{n}} = 9.32$$

$$Coeffeicient\ of\ Variation = CV = \frac{S}{\bar{x}} * 100\% = 23\%$$

Standard deviation example

Outlook	Hours Played
Rainy	26
Rainy	30
Overcast	48
Sunny	46
Sunny	62
Sunny	23
Overcast	43
Rainy	36
Rainy	38
Sunny	48
Rainy	48
Overcast	62
Overcast	44
Sunny	30

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

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

S(Hours, Outlook) = **P**(Sunny)***S**(Sunny) + **P**(Overcast)***S**(Overcast) + **P**(Rainy)***S**(Rainy)

= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87

= 7.66

Standard Deviation Reduction

- Standard Deviation Reduction (SDR) is the reduce from the original standard deviation of the label to the joined standard deviation between label and feature

$$SDR(T,X) = S(T) - S(T,X)$$

- In the example above, the original SD is 9.32, the joined standard deviation is 7.66

- The SDR is 9.32-7.66 = 1.66

Complete example dataset

Outlook	Temp.	Humidity	Windy	Hours Played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30

SDR for different feature

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR=0.17		

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

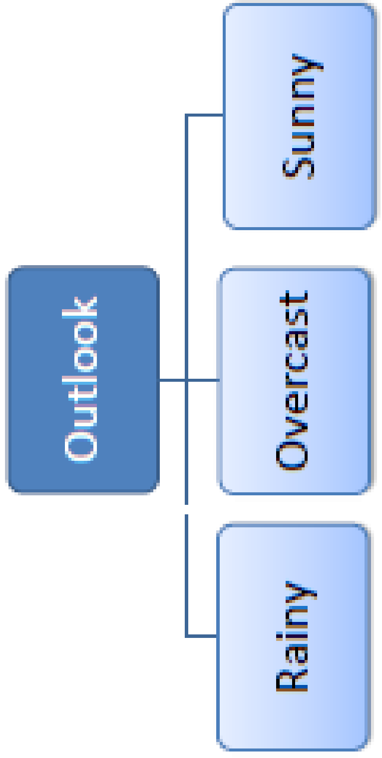
Largest SDR

- The attribute with the largest standard deviation reduction is chosen for the decision node.

★		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

Split the dataset

- The dataset is divided based on the values of the selected attribute. This process is run recursively on the non-leaf branches, until all data is processed



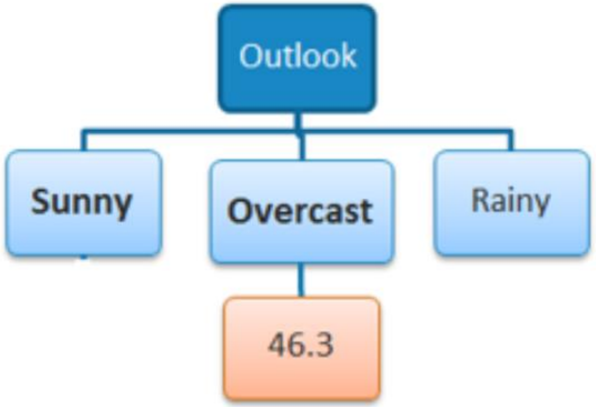
Outlook	Temp	Humidity	Windy	Hours Played
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Sunny	Mild	Normal	FALSE	46
Sunny	Mild	High	TRUE	30
Overcast	Hot	High	FALSE	46
Overcast	Cool	Normal	TRUE	43
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Rainy	Mild	Normal	TRUE	48

Stopping criteria

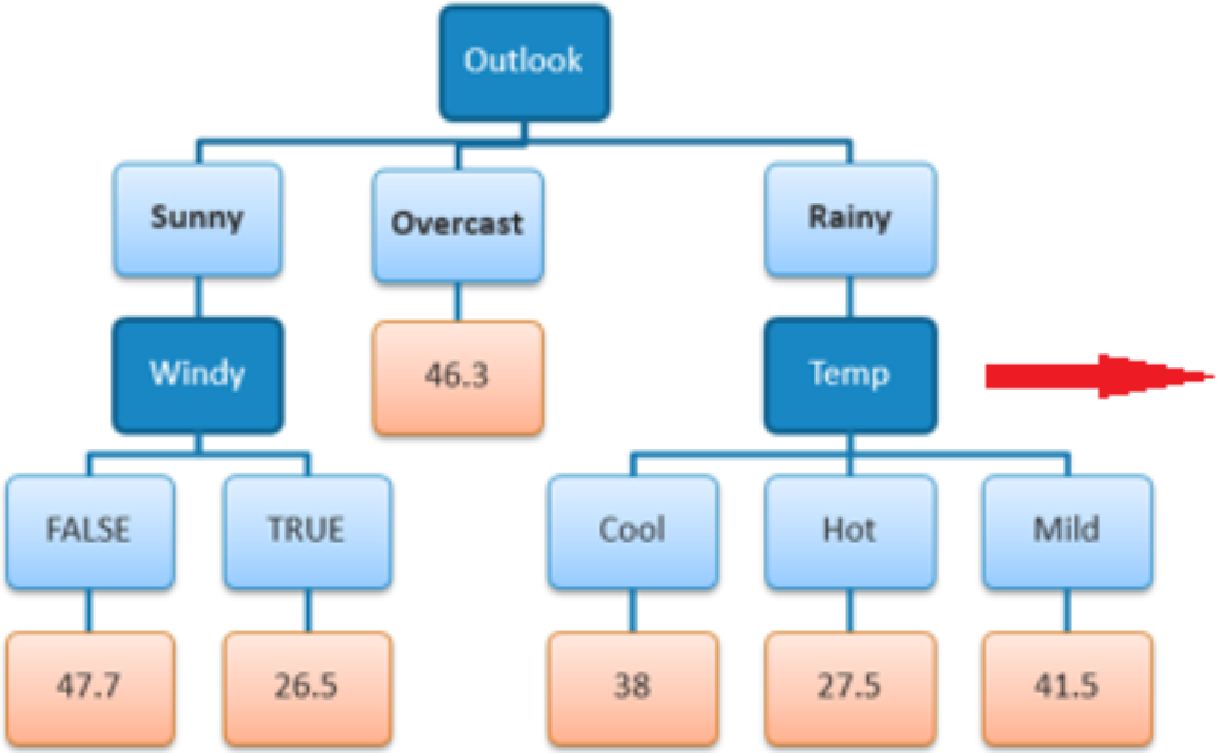
- The split of the dataset is stopped until the coefficient of variation is below defined threshold.
- For example, suppose the threshold is 10%, and the CV in overcast sub-dataset is 8%. Thus it does not need further split

Outlook - Overcast

		Hours Played (StDev)	Hours Played (AVG)	Hours Played (CV)	Count
Outlook	Overcast	3.49	46.3	8%	4
	Rainy	7.78	35.2	22%	5
	Sunny	10.87	39.2	28%	5



Final result



Temp	Hours Played
Cool	38
Hot	25
Hot	30
Mild	35
Mild	48