

# 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>



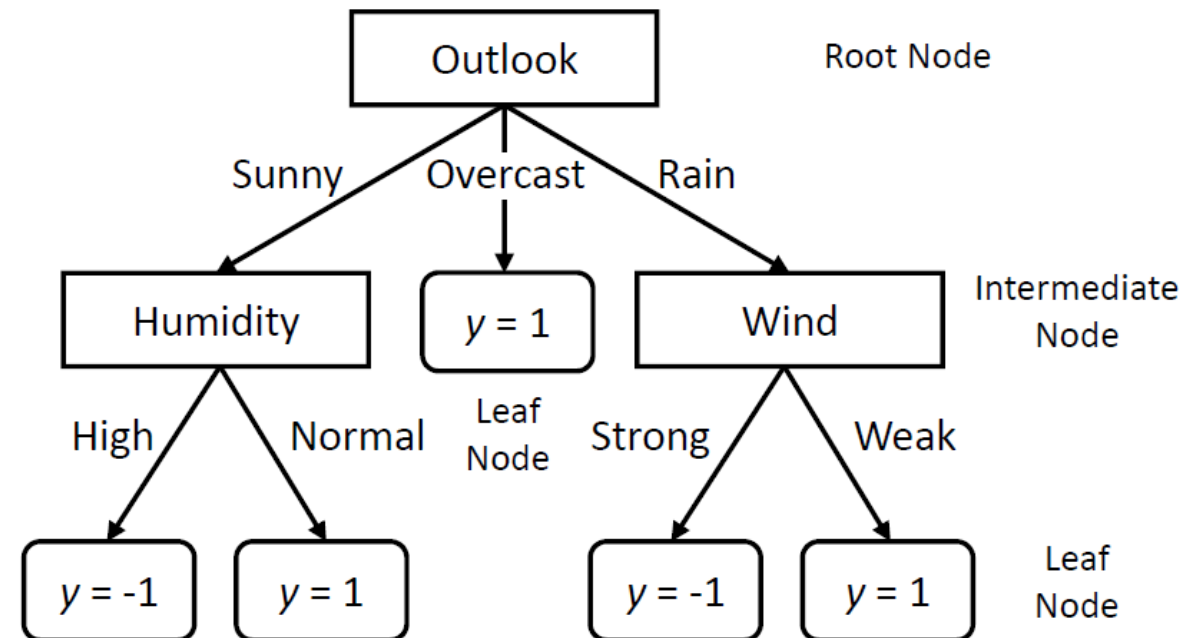
# Outline

- Tree models
- Information theory
  - Entropy, cross entropy, information gain
- Decision tree
- Continuous labels
  - Standard deviation

# Tree models

- Tree models
  - Intermediate node for splitting data
  - Leaf node for label prediction
- Discrete/categorical data example

Predictors				Response
Outlook	Temperature	Humidity	Wind	Class Play=Yes Play=No
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

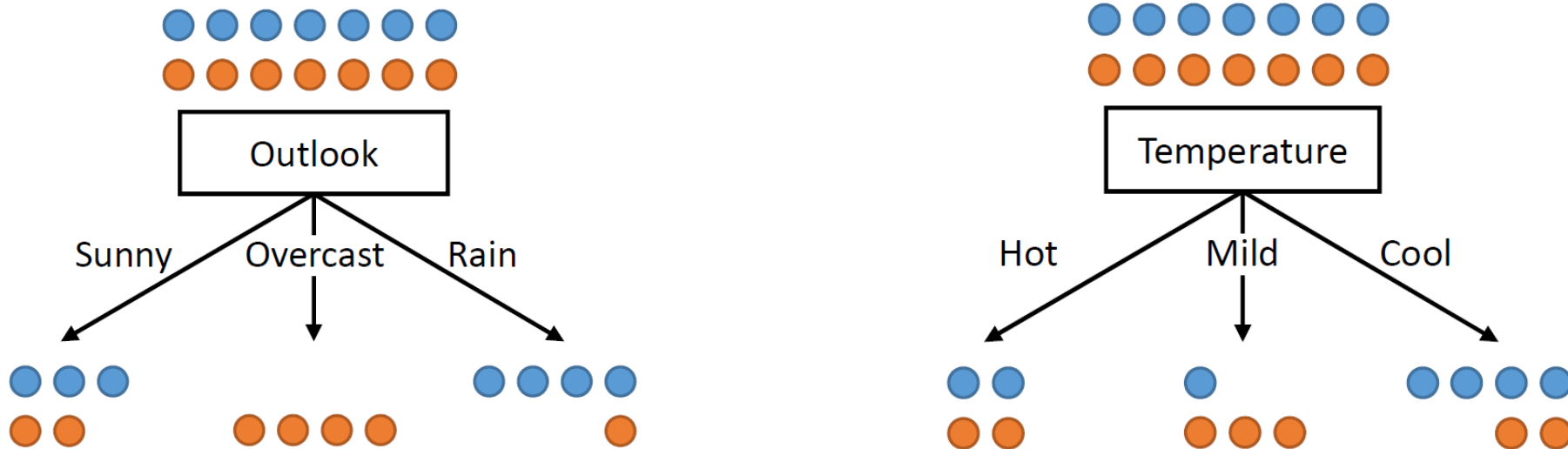


# Tree models

- Tree models
  - Intermediate node for splitting data
  - Leaf node for label prediction
- Discrete/categorical data example
- Key questions for decision trees
  - How to select node splitting conditions?
  - How to make prediction?
  - How to decide the tree structure?

# Node splitting

- Which node splitting condition to choose?



- Choose the features with higher classification capacity
  - Quantitatively, with higher **information gain**

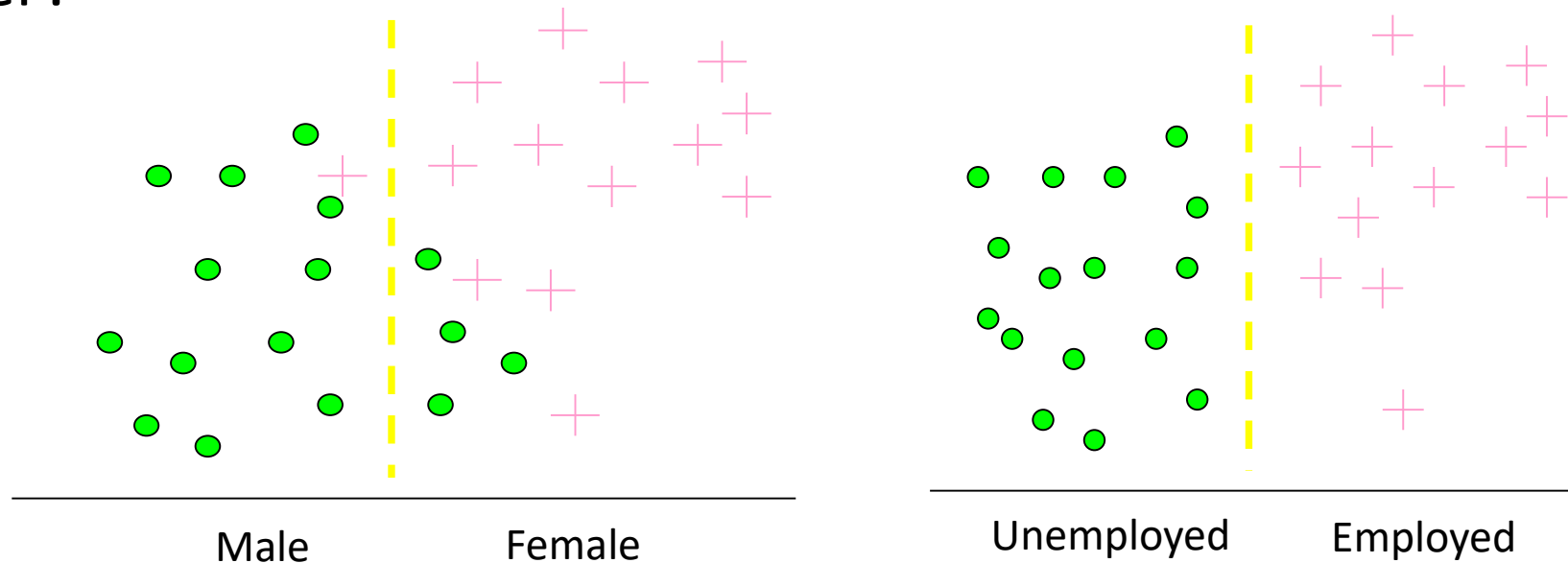
# Information Theory

# Motivating 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**.

# Motivating 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

- How to measure the level of **informative** (first example) and level of **certainty** (second example) in mathematics?
- **Entropy** (more specifically, Shannon entropy) is the expected value (average) of the information contained in each message
- Suppose  $X$  is a random variable with  $n$  discrete values

$$P(X = x_i) = p_i$$

then the entropy is

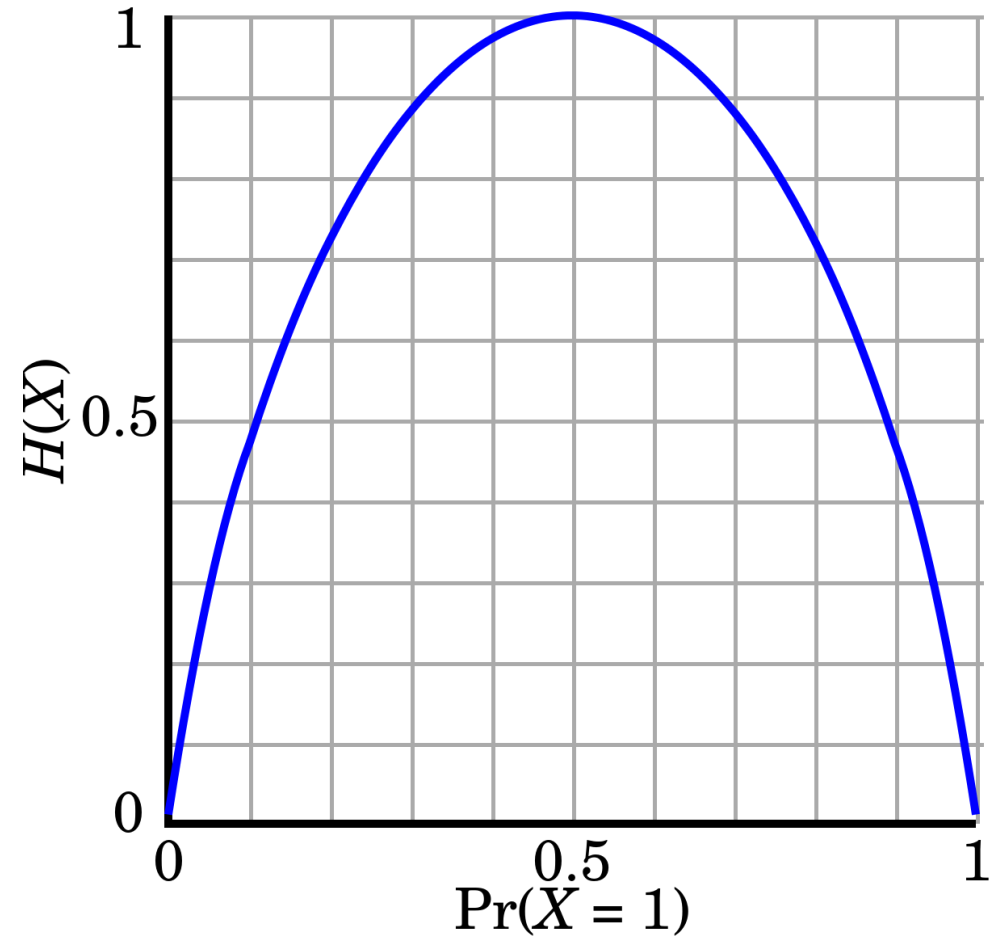
$$H(X) = - \sum_{i=1}^n p_i \log_2(p_i)$$

- Easy to verify  $H(X) = - \sum_{i=1}^n p_i \log_2(p_i) \leq - \sum_{i=1}^n \frac{1}{n} \log_2 \frac{1}{n} = \log n$

# Illustration

- Entropy for binary distribution

$$H(X) = -p_1 \log_2(p_1) - p_0 \log_2(p_0)$$

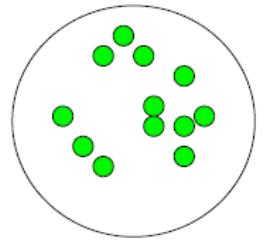


# Entropy examples

- What is the entropy of a group in which all examples belong to the same class?

- Entropy =  $-1 \log_2 1 = 0$

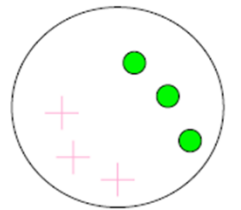
Minimum uncertainty



- 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



# Cross entropy

- Cross entropy is used to measure the **difference/distance** between two random variable distributions

$$H(X, Y) = - \sum_{i=1}^n P(X = i) \log P(Y = i)$$

- Continuous version

$$H(p, q) = - \int p(x) \log q(x) dx$$

# Recall on logistic regression

- Binary classification

$$p_{\theta}(y = 1|x) = \sigma(\theta^{\top} x) = \frac{1}{1 + e^{-\theta^{\top} x}}$$

$$p_{\theta}(y = 0|x) = \frac{e^{-\theta^{\top} x}}{1 + e^{-\theta^{\top} x}}$$

- Cross entropy loss function

is also convex in  $\theta$

$$\mathcal{L}(y, x, p_{\theta}) = -y \log \sigma(\theta^{\top} x) - (1 - y) \log(1 - \sigma(\theta^{\top} x))$$

- Gradient

$$\begin{aligned} \frac{\partial \mathcal{L}(y, x, p_{\theta})}{\partial \theta} &= -y \frac{1}{\sigma(\theta^{\top} x)} \sigma(z)(1 - \sigma(z))x - (1 - y) \frac{-1}{1 - \sigma(\theta^{\top} x)} \sigma(z)(1 - \sigma(z))x \\ &= (\sigma(\theta^{\top} x) - y)x \end{aligned}$$

$$\theta \leftarrow \theta + \eta(y - \sigma(\theta^{\top} x))x$$

$$\boxed{\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1 - \sigma(z))}$$

$$\theta_{\text{new}} \leftarrow \theta_{\text{old}} - \eta \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

# Conditional entropy

- Entropy  $H(X, Y) = -\sum_{i=1}^n P(X = i) \log P(Y = i)$
- Specific conditional entropy of  $X$  given  $Y = y$

$$H(X|Y = y) = -\sum_{i=1}^n P(X = i|Y = y) \log P(Y = i|Y = y)$$

- Conditional entropy of  $X$  given  $Y$

$$H(X|Y) = \sum_y P(Y = y) H(X|Y = y)$$

- Information gain of  $X$  given  $Y$

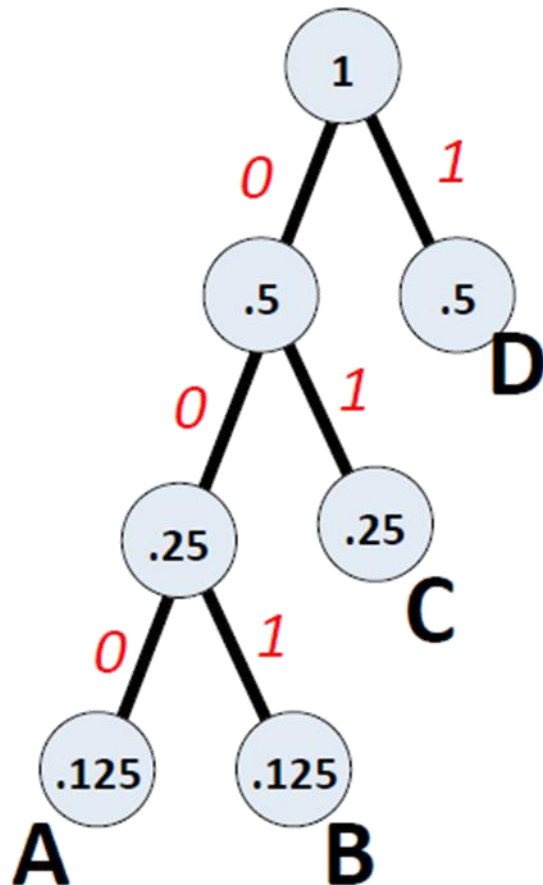
$$I(X, Y) = H(X) - H(X|Y)$$

# 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



# Interpretation from coding perspective

- Usually entropy denotes the best coding (in theory), with the minimal average message length
- When the probability distribution is composed of  $\frac{1}{2^i}$ , then the Huffman code is the entropy

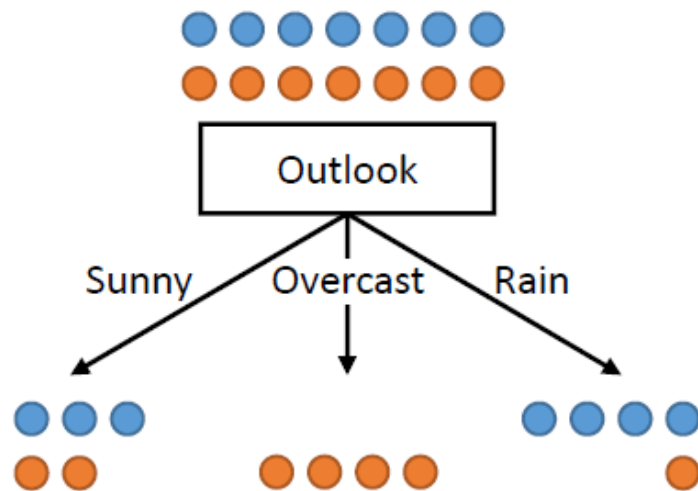
# Decision Tree

# Node Splitting

- Information gain

$$H(X|Y = v) = - \sum_{i=1}^n P(X = i|Y = v) \log P(X = i|Y = v)$$

$$H(X|Y) = \sum_{v \in \text{values}(Y)} P(Y = v) H(X|Y = v)$$



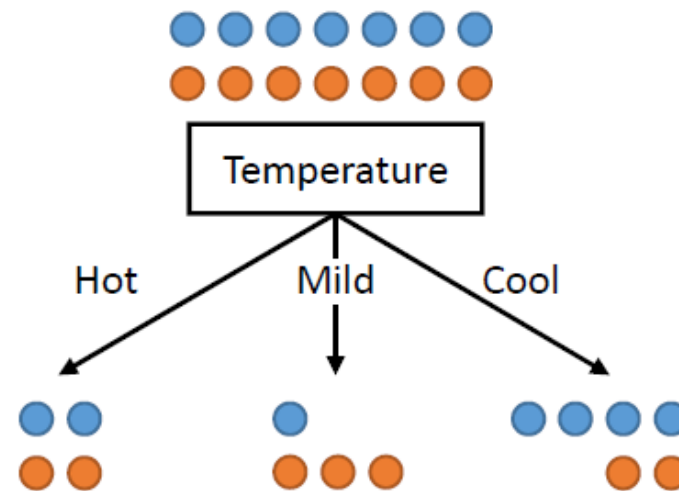
$$H(X|Y = S) = -\frac{3}{5} \log \frac{3}{5} - \frac{2}{5} \log \frac{2}{5} = 0.9710$$

$$H(X|Y = O) = -\frac{4}{4} \log \frac{4}{4} = 0$$

$$H(X|Y = R) = -\frac{4}{5} \log \frac{4}{5} - \frac{1}{5} \log \frac{1}{5} = 0.7219$$

$$H(X|Y) = \frac{5}{14} \times 0.9710 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.7219 = 0.6046$$

$$I(X, Y) = H(X) - H(X|Y) = 1 - 0.6046 = 0.3954$$



$$H(X|Y = H) = -\frac{2}{4} \log \frac{2}{4} - \frac{2}{4} \log \frac{2}{4} = 1$$

$$H(X|Y = M) = -\frac{1}{4} \log \frac{1}{4} - \frac{3}{4} \log \frac{3}{4} = 0.8113$$

$$H(X|Y = C) = -\frac{4}{6} \log \frac{4}{6} - \frac{2}{6} \log \frac{2}{6} = 0.9183$$

$$H(X|Y) = \frac{4}{14} \times 1 + \frac{4}{14} \times 0.8113 + \frac{5}{14} \times 0.9183 = 0.9111$$

$$I(X, Y) = H(X) - H(X|Y) = 1 - 0.9111 = 0.0889$$

# Node splitting

- 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** of  $X$  given  $Y$ 
$$I(X, Y) = H(X) - H(X|Y)$$

# 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

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

## Example (cont.)

- Suppose we now also know the gender of these 8 students, what is the conditional entropy on gender?

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 conditional entropy on gender?
- The labels are divided into two small dataset based on the gender

Like (male)
Yes
Yes
Yes
No

$$P(\text{Yes} \mid \text{male}) = 0.75$$

Like(female)
No
No
No
Yes

$$P(\text{Yes} \mid \text{female}) = 0.25$$

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 conditional entropy on gender?

- $P(\text{Yes}|\text{male}) = 0.75$
- $P(\text{Yes}|\text{female}) = 0.25$
- $H(\text{Like}|\text{male})$   
$$= -\frac{1}{4} \log\left(\frac{1}{4}\right) - \frac{3}{4} \log\left(\frac{3}{4}\right)$$
$$= -0.25 * -2 - 0.75 * -0.41 = 0.81$$
- $H(\text{Like}|\text{female})$   
$$= -\frac{3}{4} \log\left(\frac{3}{4}\right) - \frac{1}{4} \log\left(\frac{1}{4}\right)$$
$$= -0.75 * -0.41 - 0.25 * -2 = 0.81$$

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 conditional entropy on gender?

- $E(\text{Like}|\text{Gender})$   
 $= E(\text{Like}|\text{male}) * P(\text{male})$   
 $+ E(\text{Like}|\text{female}) * P(\text{female})$   
 $= 0.5 * 0.81 + 0.5 * 0.81 = 0.81$
- $I(\text{Like}, \text{Gender}) = E(\text{Like}) - E(\text{Like}|\text{Gender})$   
 $= 1 - 0.81 = 0.19$

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 conditional entropy on major?

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 conditional entropy on major?
- Three datasets are created based on major

Like (math)
Yes
No
No
Yes

$$P(\text{Yes}|\text{math}) = 0.5$$

Like(history)
No
No

$$P(\text{Yes}|\text{history}) = 0$$

Like(cs)
Yes
Yes

$$P(\text{Yes}|\text{cs}) = 1$$

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

- $H(\text{Like}|\text{Math}) = -\frac{2}{4}\log\left(\frac{2}{4}\right) - \frac{2}{4}\log\left(\frac{2}{4}\right)=1$
- $H(\text{Like}|\text{CS}) = -\frac{2}{2}\log\left(\frac{2}{2}\right) - \frac{0}{2}\log\left(\frac{0}{2}\right)=0$
- $H(\text{Like}|\text{history}) = -\frac{2}{2}\log\left(\frac{2}{2}\right) - \frac{0}{2}\log\left(\frac{0}{2}\right)=0$
- $H(\text{Like}|\text{Major})$   
     $= H(\text{Like}|\text{math}) \times P(\text{math})$   
     $+ H(\text{Like}|\text{History}) \times P(\text{History})$   
     $+ H(\text{Like}|\text{cs}) \times P(\text{cs})$   
     $= 0.5 * 1 + 0.25 * 0 + 0.25 * 0 = 0.5$
- $I(\text{Like}, \text{Major}) = E(\text{Like}) - E(\text{Like}|\text{Major})$   
     $= 1 - 0.5 = 0.5$

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

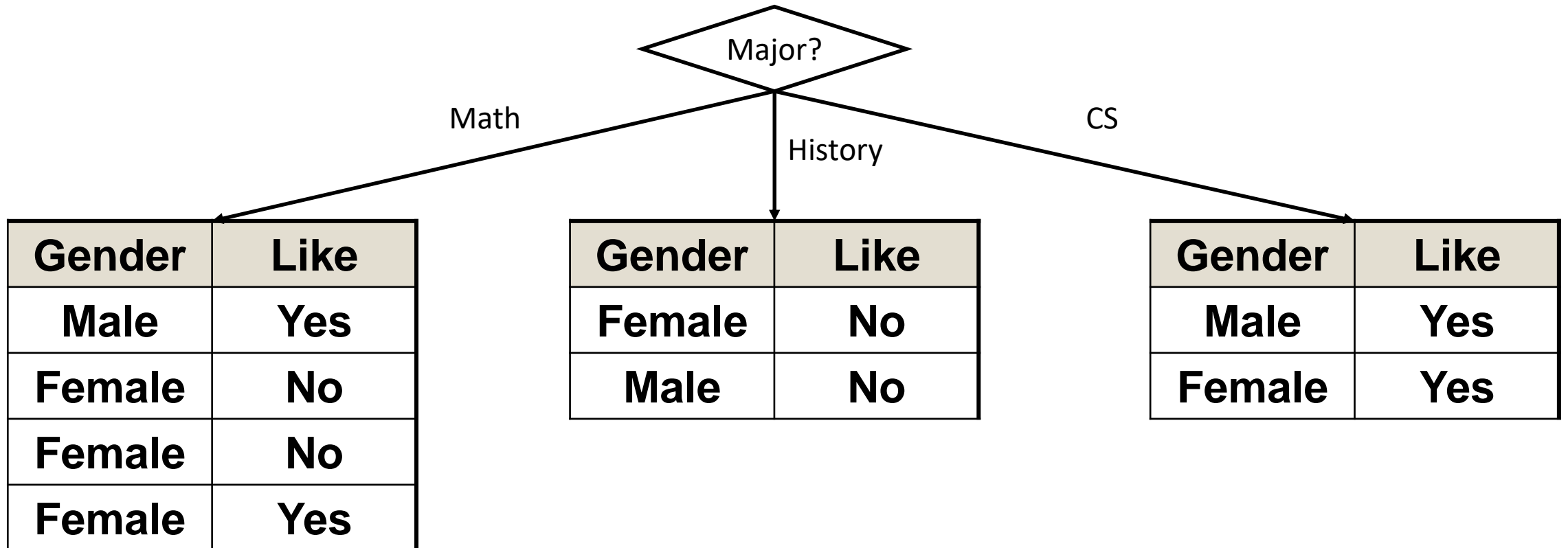
# Example (cont.)

- Compare gender and major
- As we have computed
  - $I(\text{Like}, \text{Gender}) = E(\text{Like}) - E(\text{Like}|\text{Gender}) = 1 - 0.81 = 0.19$
  - $I(\text{Like}, \text{Major}) = E(\text{Like}) - E(\text{Like}|\text{Major}) = 1 - 0.5 = 0.5$
- **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

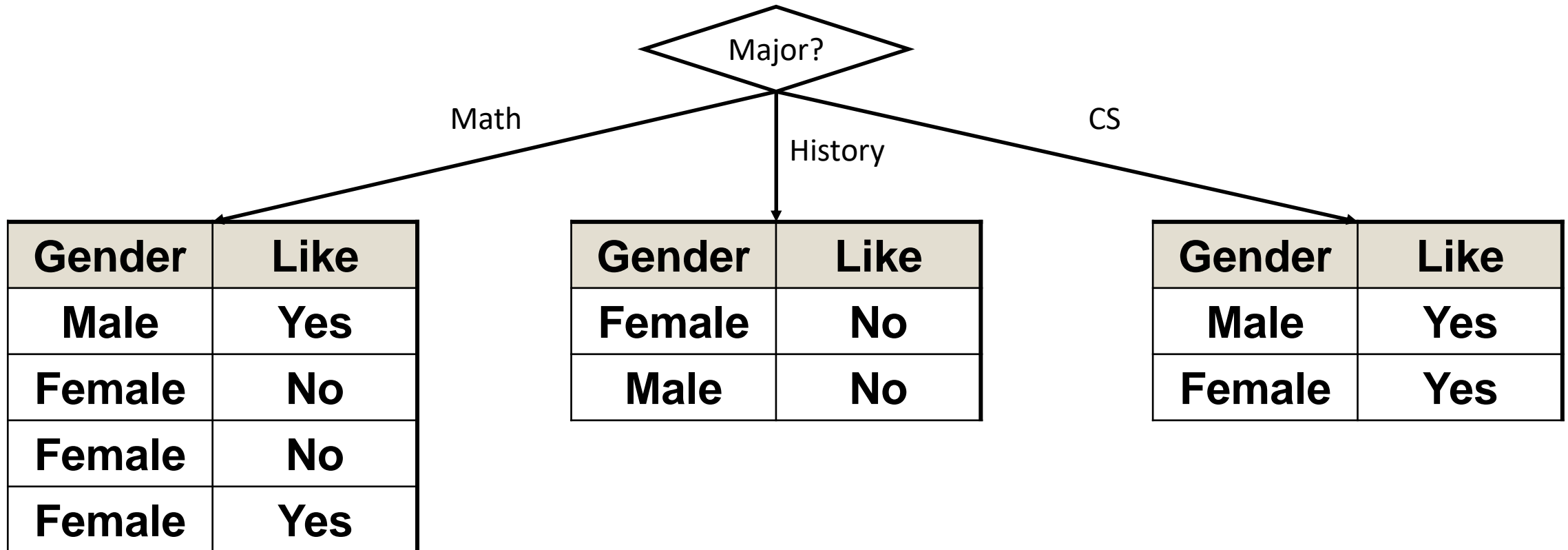
# Example (cont.)

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



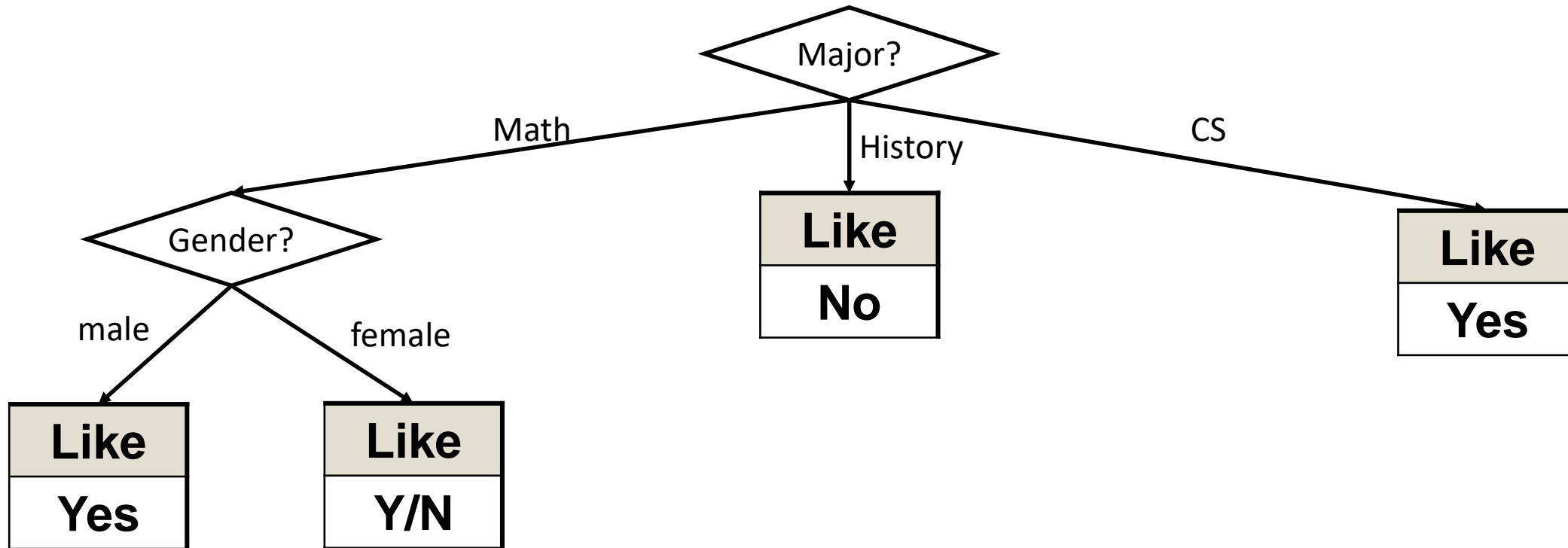
# Example (cont.)

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



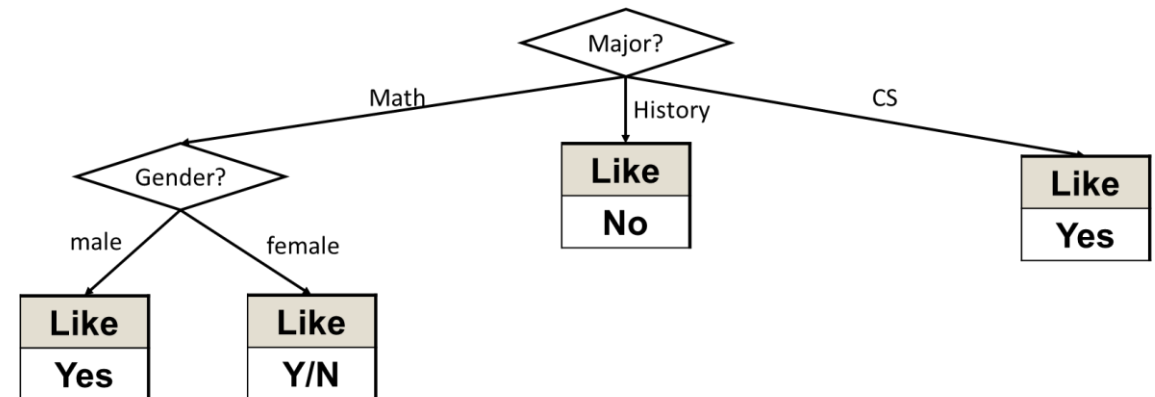


# Example (cont.)



## 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?



# Decision tree building: ID3 algorithm

- Algorithm framework
  - Start from the root node with all data
  - For each node, calculate the information gain of all possible features
  - Choose the feature with the highest information gain
  - Split the data of the node according to the feature
  - Do the above recursively for each leaf node, until
    - There is no information gain for the leaf node
    - Or there is no feature to select
- Testing
  - Pass the example through the tree to the leaf node for a label

# Continuous Labels

# 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](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

- Count =  $n = 14$
- Average =  $\bar{x} = 39.8$
- Standard deviation  $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} = 9.32$
- Coefficient of variation (CV) =  $\frac{\sigma}{\bar{x}} = \frac{9.32}{39.8} = 23\%$

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

# Standard deviation example

- With additional outlook label

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

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



$$\begin{aligned} S(\text{Hours}, \text{Outlook}) &= P(\text{Sunny}) * S(\text{Sunny}) + P(\text{Overcast}) * S(\text{Overcast}) + P(\text{Rainy}) * S(\text{Rainy}) \\ &= (4/14) * 3.49 + (5/14) * 7.78 + (5/14) * 10.87 \\ &= 7.66 \end{aligned}$$

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



# 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
- So 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		

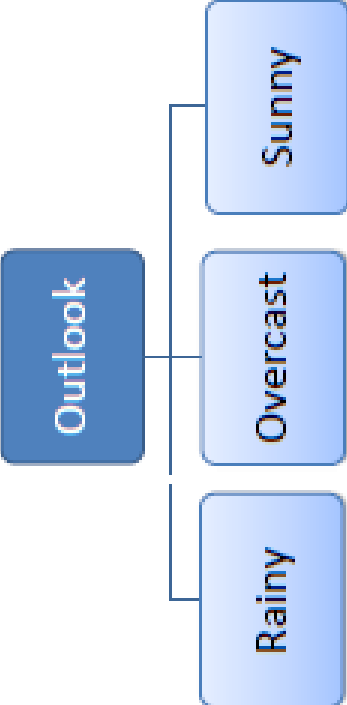
# Largest SDR

- 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



A decision tree diagram illustrating the recursive splitting of a dataset based on the 'Outlook' attribute. The root node is 'Outlook', which branches into three categories: 'Sunny', 'Overcast', and 'Rainy'. Each category is associated with a table of data points, where the first column represents the 'Outlook' value, and the subsequent columns represent 'Temp', 'Humidity', 'Windy', and 'Hours Played'.

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

Outlook	Temp	Humidity	Windy	Hours Played
Overcast	Hot	High	FALSE	46
Overcast	Cool	Normal	TRUE	43
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44

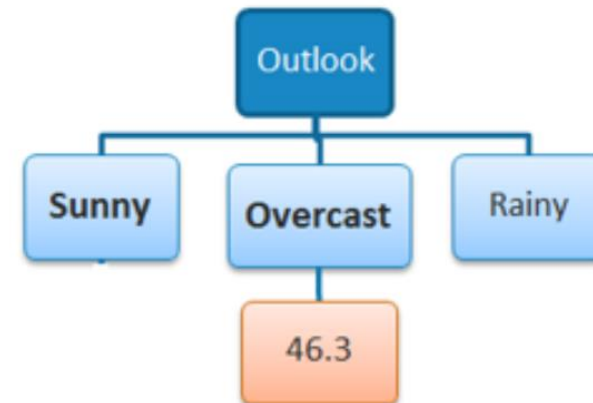
Outlook	Temp	Humidity	Windy	Hours Played
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Rainy	Mild	Normal	TRUE	48

# Stopping criteria

- The split of the dataset is stopped until the coefficient of variation (CV) 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

