School of Computing & IT

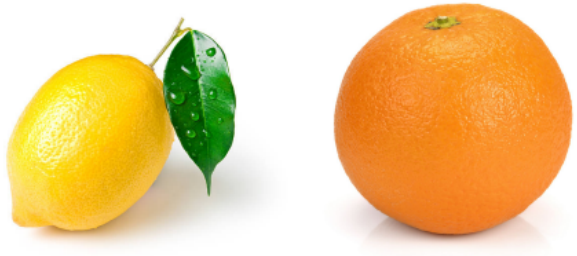
Non-Probabilistic Decision Trees

Lecture 6

CCS3440-Artificial Intelligence

Nuwan Vithanage

Decision Trees: Lemons or Oranges

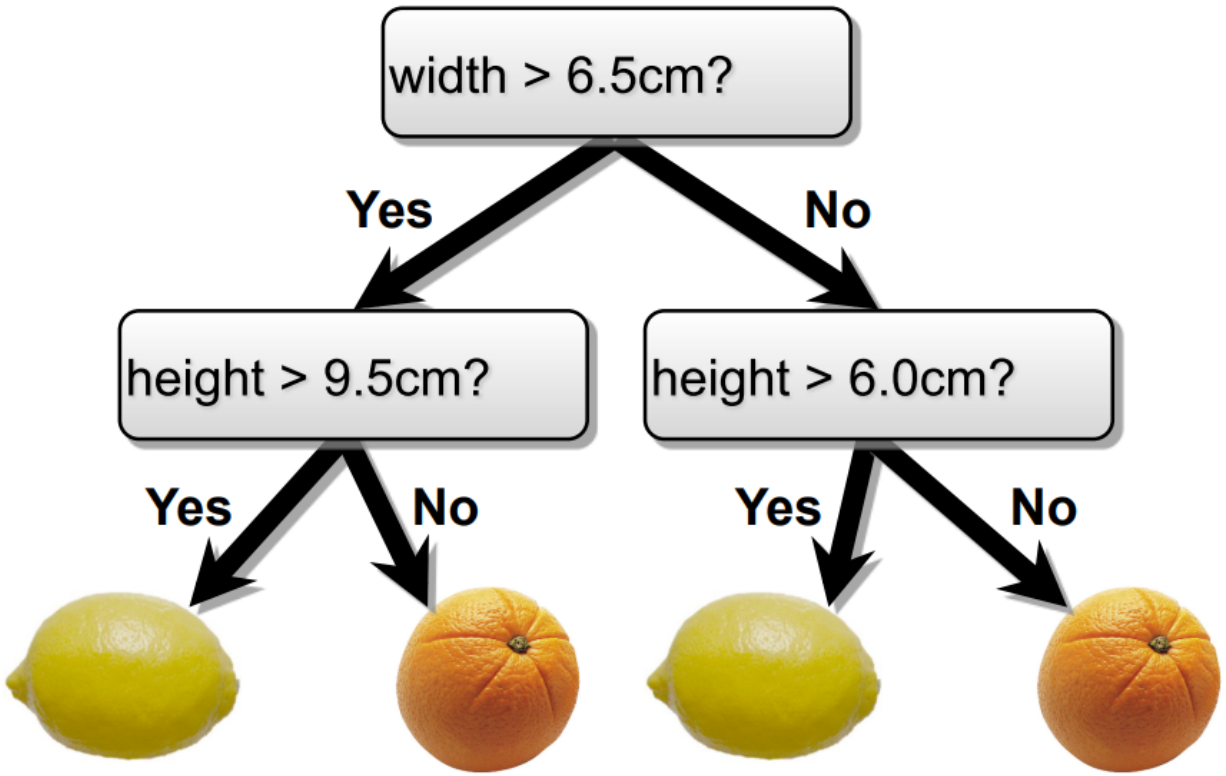


Scenario: You run a sorting facility for citrus fruits

- Binary classification: lemons or oranges
- Features measured by sensor on conveyor belt: height and width

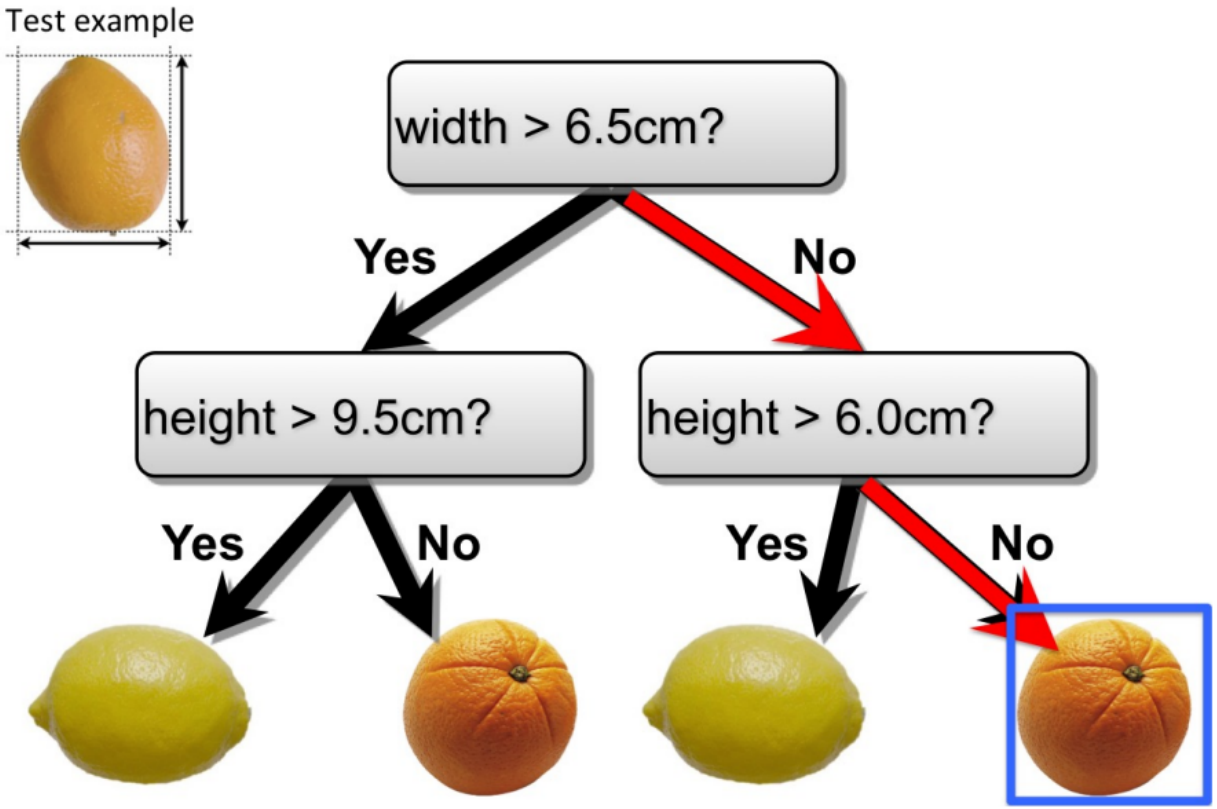
Decision Trees

- Make predictions by splitting on features according to a tree structure.



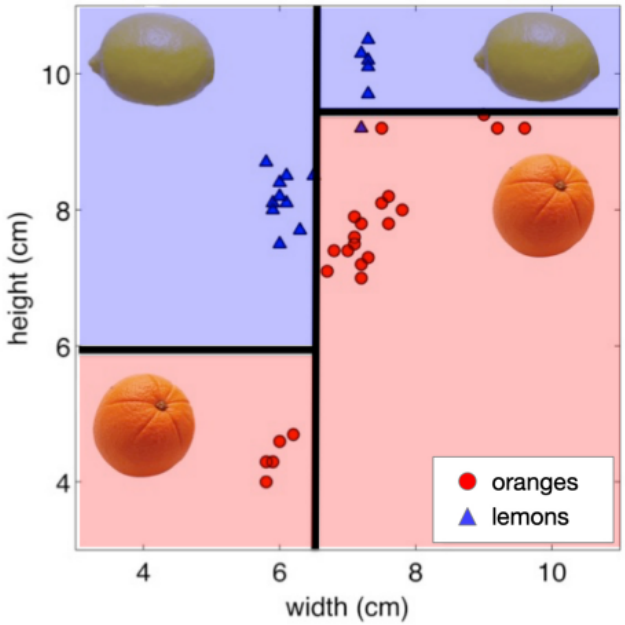
Decision Trees

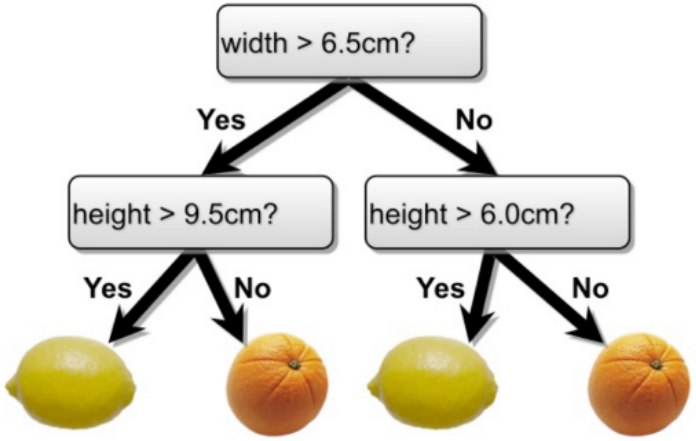
- Make predictions by splitting on features according to a tree structure.



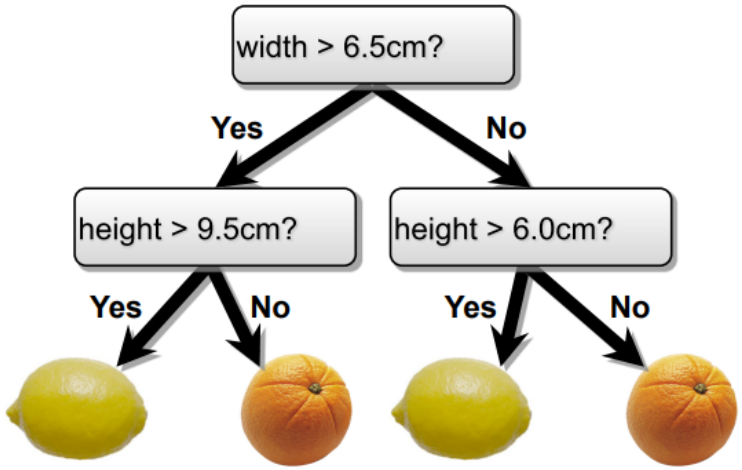
Decision Trees: Continuous Features

- Split *continuous features* by checking whether that feature is greater than or less than some threshold.
- Decision boundary is made up of axis-aligned planes.





Decision Trees



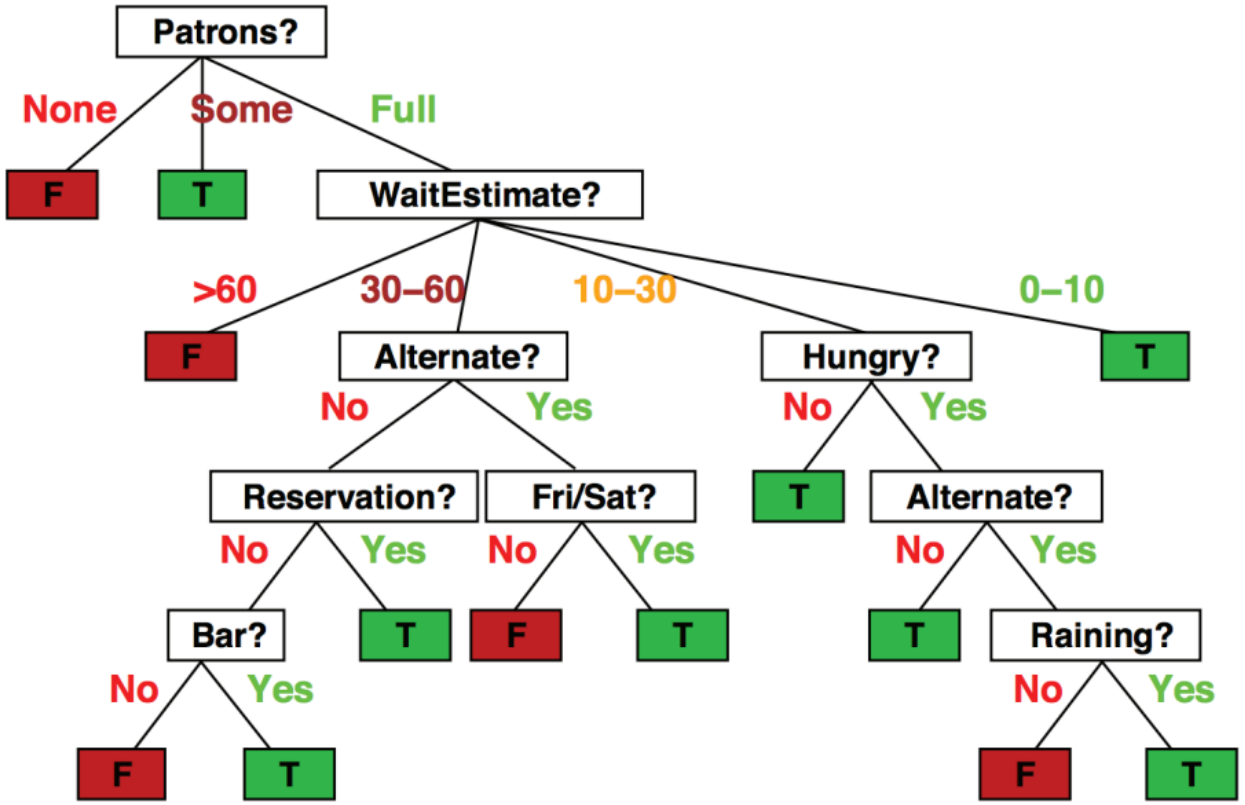
- Internal nodes test a feature
- Branching is determined by the feature value
- Leaf nodes are outputs (predictions)

Question: What are the hyperparameters of this model?

Decision Trees—Discrete Features

School of Computing & IT

- Will I eat at this restaurant?



Decision Trees—Discrete Features

School of Computing & IT

- Split *discrete features* into a partition of possible values.

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

Features:

1. Alternate: whether there is a suitable alternative restaurant nearby.
2. Bar: whether the restaurant has a comfortable bar area to wait in.
3. Fri/Sat: true on Fridays and Saturdays.
4. Hungry: whether we are hungry.
5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
6. Price: the restaurant's price range ($, $$, $$$).
7. Raining: whether it is raining outside.
8. Reservation: whether we made a reservation.
9. Type: the kind of restaurant (French, Italian, Thai or Burger).
10. WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

Decision Trees—Discrete Features

School of Computing & IT

- Decision trees are universal function approximators.
  - For any training set we can construct a decision tree that has exactly the one leaf for every training point, but it probably won't generalize.
  - Example - If all D features were binary, and we had $N = 2^D$ unique training examples, a **Full Binary Tree** would have one leaf per example.
- Finding the smallest decision tree that correctly classifies a training set is NP complete.

Learning Decision Trees

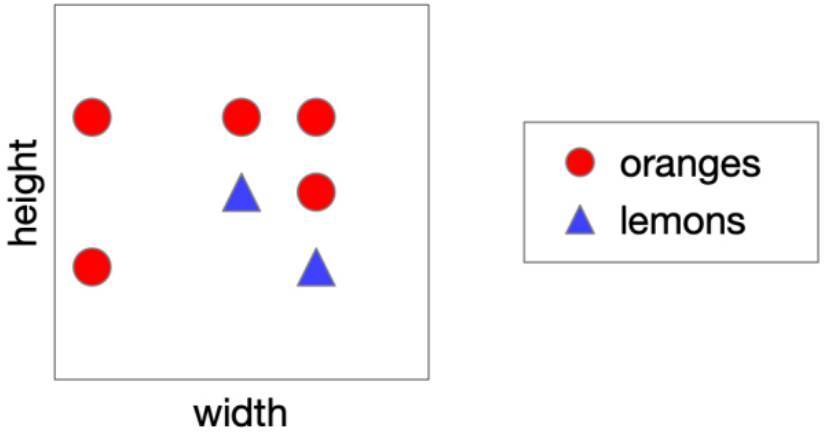
School of Computing & IT

- Resort to a greedy heuristic:
  - Start with the whole training set and an empty decision tree.
  - Pick a feature and candidate split that would most reduce a loss
  - Split on that feature and recurse on subpartitions.
- What is a loss?
  - When learning a model, we use a scalar number to assess whether we're on track
  - Scalar value: low is good, high is bad
- Which loss should we use?

Choosing a Good Split

School of Computing & IT

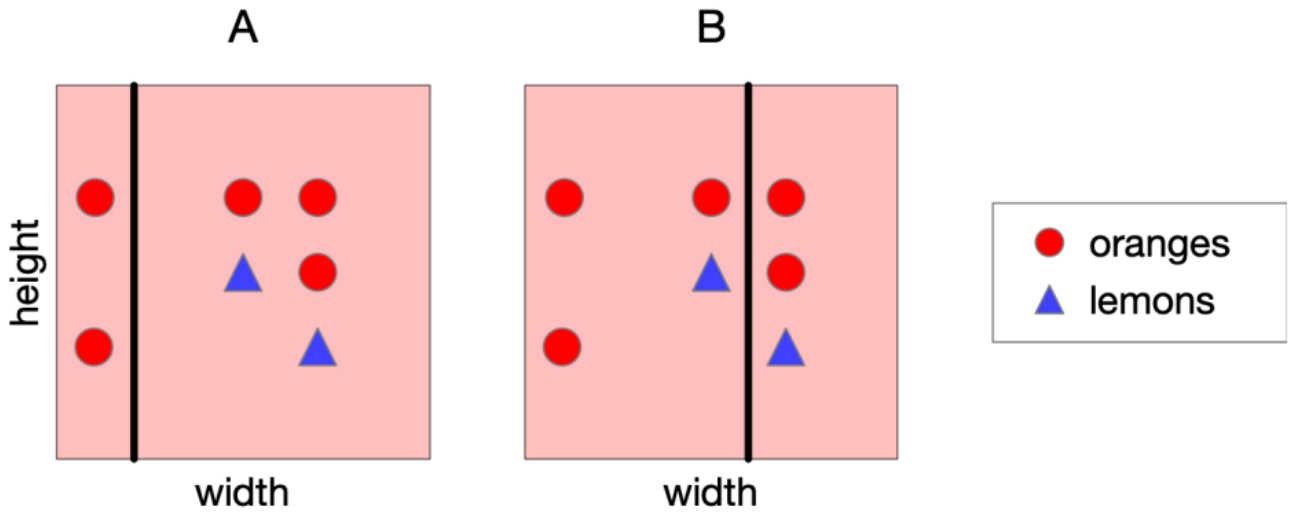
- Consider the following data. Let's split on width.
- Classify by majority.



Choosing a Good Split

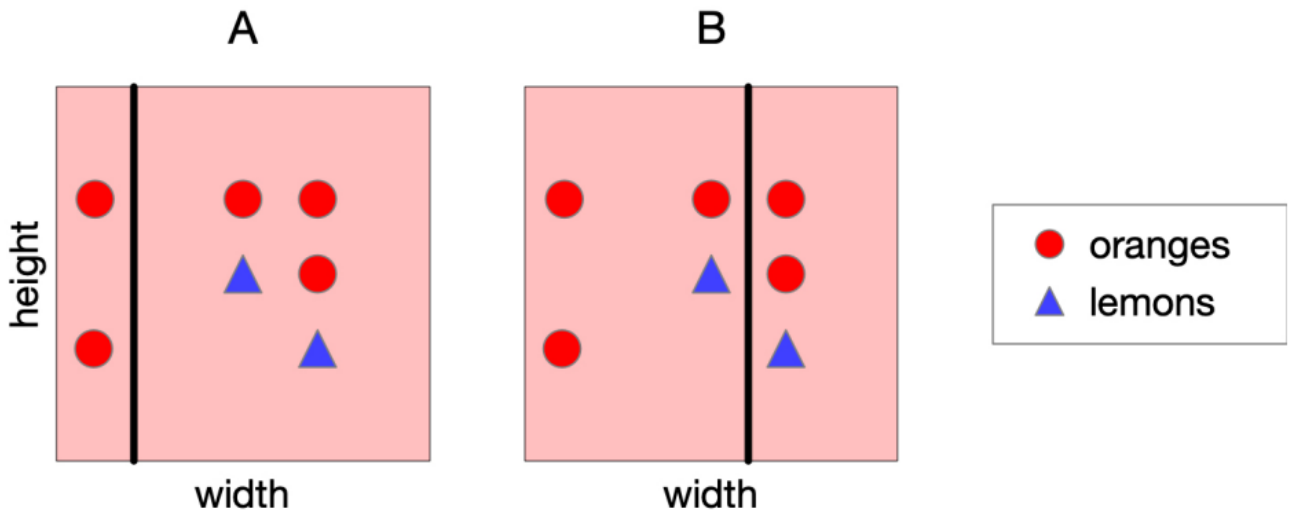
School of Computing & IT

- Which is the best split? Vote!



Choosing a Good Split

- A feels like a better split, because the left-hand region is very certain about whether the fruit is an orange.
- Can we quantify this?



Choosing a Good Split

School of Computing & IT

- How can we quantify uncertainty in prediction for a given leaf node?
  - If all examples in leaf have same class: good, low uncertainty
  - If each class has same amount of examples in leaf: bad, high uncertainty
- **Idea:** Use counts at leaves to define probability distributions; use a probabilistic notion of uncertainty to decide splits.
- There are different ways to evaluate a split. We will focus on a common way: **entropy**.
- A brief detour through information theory...

Entropy - Quantifying uncertainty

School of Computing & IT

- You may have encountered the term entropy quantifying the state of chaos in chemical and physical systems,
- In statistics, it is a property of a random variable,
- The entropy of a discrete random variable is a number that quantifies the uncertainty inherent in its possible outcomes.
- The mathematical definition of entropy that we give in a few slides may seem arbitrary, but it can be motivated axiomatically.
  - If you're interested, check: *Information Theory* by Robert Ash or Elements of Information Theory by Cover and Thomas.
- To explain entropy, consider flipping two different coins...

We Flip Two Different Coins

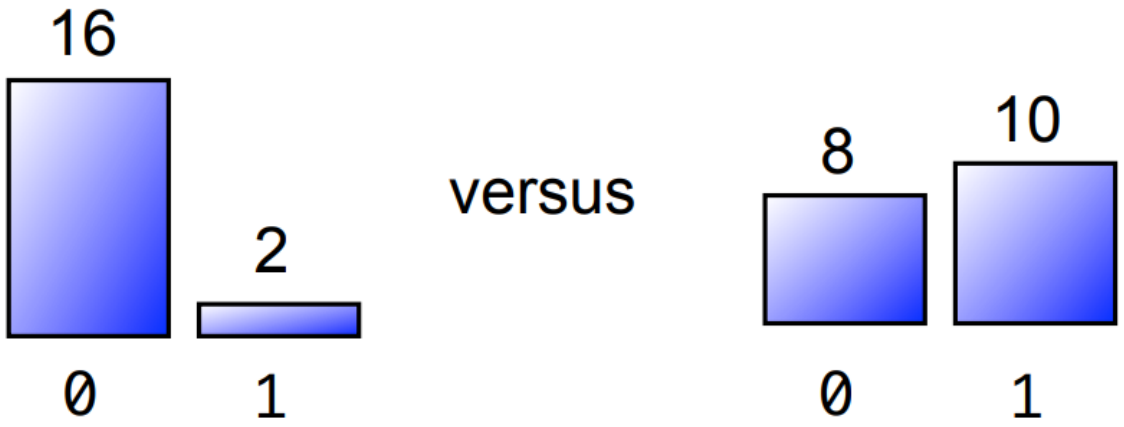
Each coin is a binary random variable with outcomes 1 or 0:

Sequence 1:

0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 ... ?

Sequence 2:

0 1 0 1 0 1 1 1 0 1 0 0 1 1 0 1 0 1 ... ?

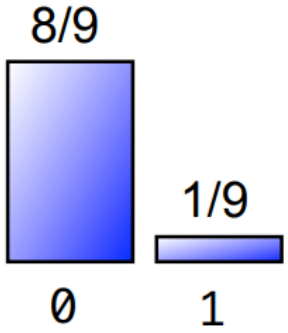


Quantifying Uncertainty

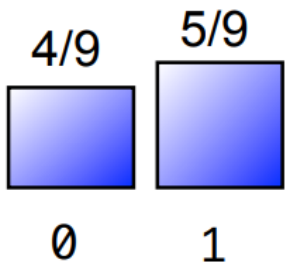
School of Computing & IT

- The entropy of a loaded coin with probability p of heads is given by

$$-p\log_2(p) - (1-p)\log_2(1-p)$$



$$-\frac{8}{9}\log_2\frac{8}{9} - \frac{1}{9}\log_2\frac{1}{9} \approx \frac{1}{2}$$



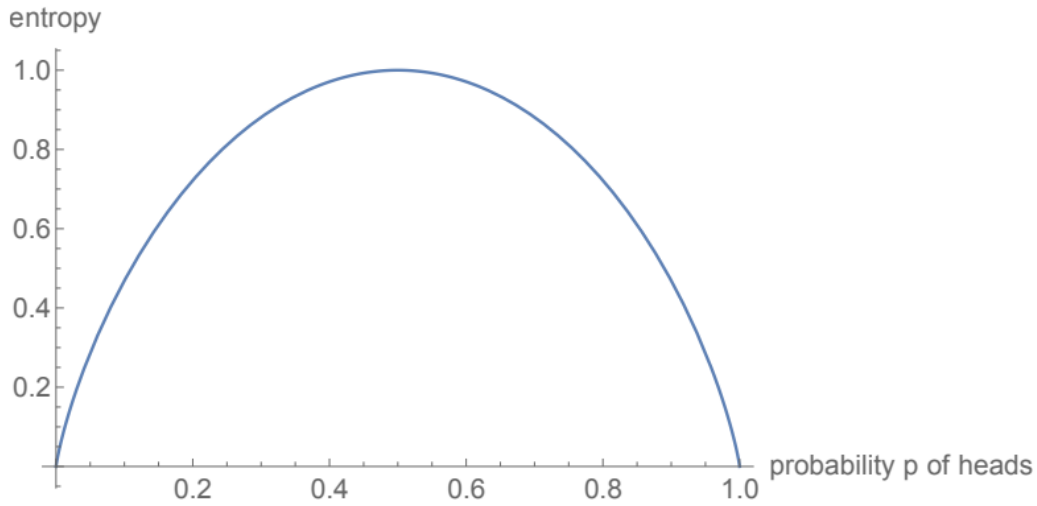
$$-\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} \approx 0.99$$

- Notice: the coin whose outcomes are more certain has a lower entropy.
- In the extreme case $p = 0$ or $p = 1$, we were certain of the outcome before observing. So, we gained no certainty by observing it, i.e., entropy is 0.

Quantifying Uncertainty

School of Computing & IT

- Can also think of entropy as the expected information content of a random draw from a probability distribution.



- Claude Shannon showed: you cannot store the outcome of a random draw using fewer expected bits than the entropy without losing information.
- So units of entropy are bits; a fair coin flip has 1 bit of entropy.

Entropy

School of Computing & IT

- More generally, the entropy of a discrete random variable Y is given by

$$H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y)$$

- **"High Entropy"**:
  - Variable has a uniform like distribution over many outcomes
  - Flat histogram
  - Values sampled from it are less predictable
- **"Low Entropy"**
  - Distribution is concentrated on only a few outcomes
  - Histogram is concentrated in a few areas
  - Values sampled from it are more predictable

Entropy

- Suppose we observe partial information X about a random variable Y
  - For example, $X = \text{sign}(Y)$.
- We want to work towards a definition of the expected amount of information that will be conveyed about Y by observing X.
  - Or equivalently, the expected reduction in our uncertainty about Y after observing X.

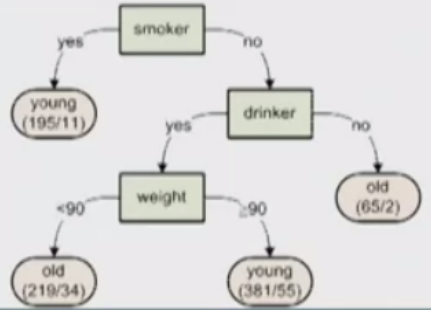
Decision Trees Classifier

- Key ideas
  - Iteratively split variables into groups
  - Evaluate “Homogeneity” within each group
  - Split again if necessary
- Pros:
  - Easy to interpret
  - Better performance in nonlinear settings
- Cons:
  - Without pruning/cross-validation can lead to overfitting
  - Harder to estimate uncertainty
  - Results maybe variable

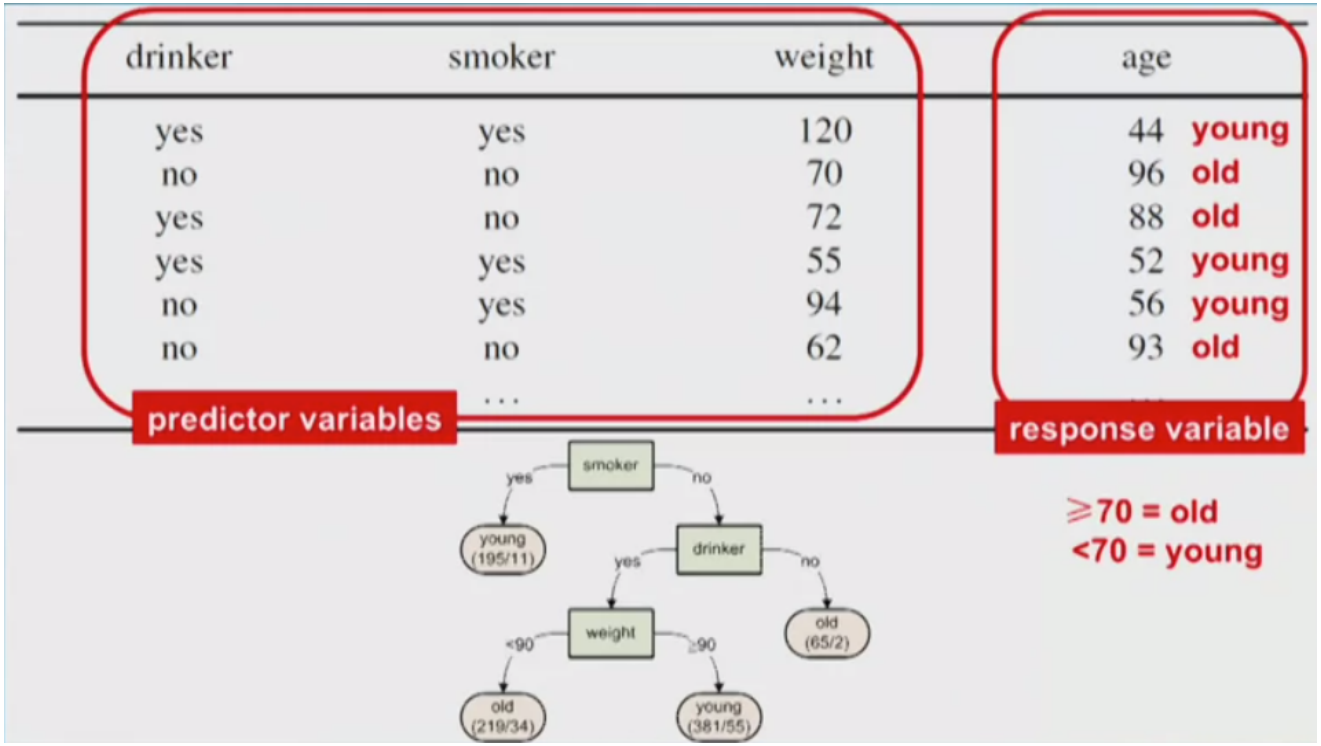
Decision Trees Classifier

- Data set 1: Effect of lifestyle on life expectancy

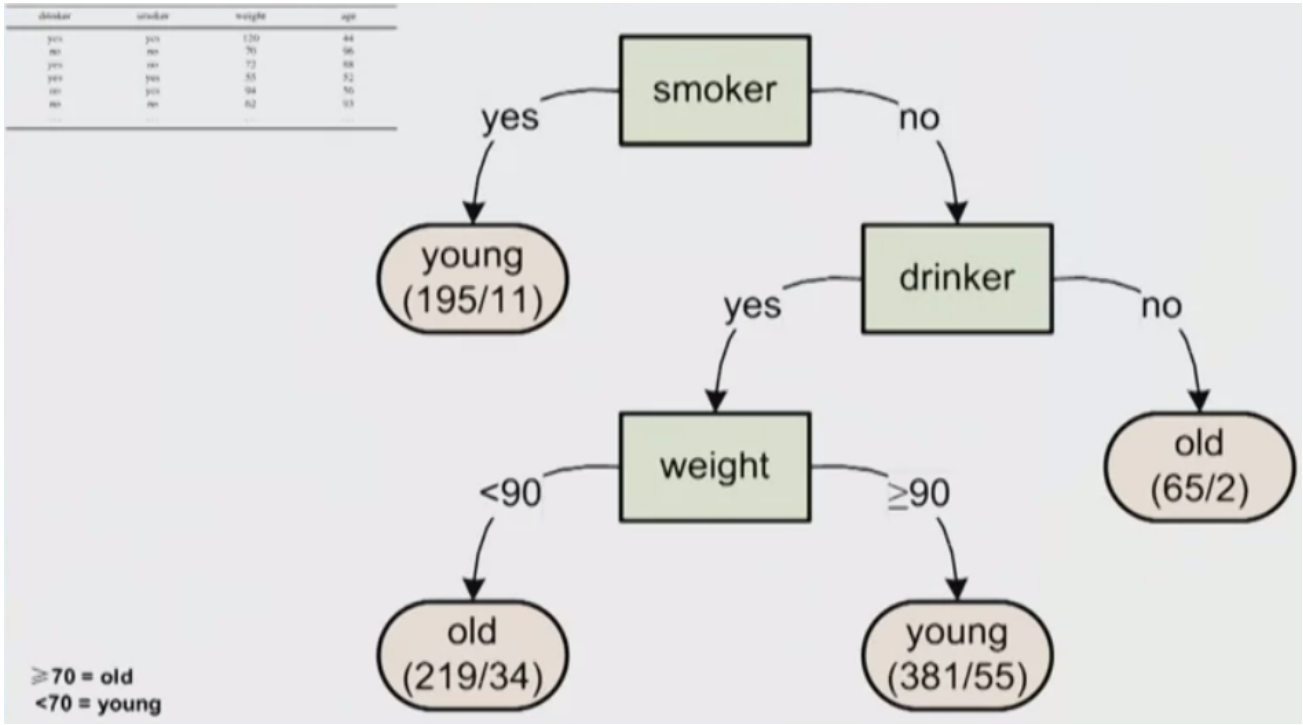
drinker	smoker	weight	age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	93
...	...	...	...



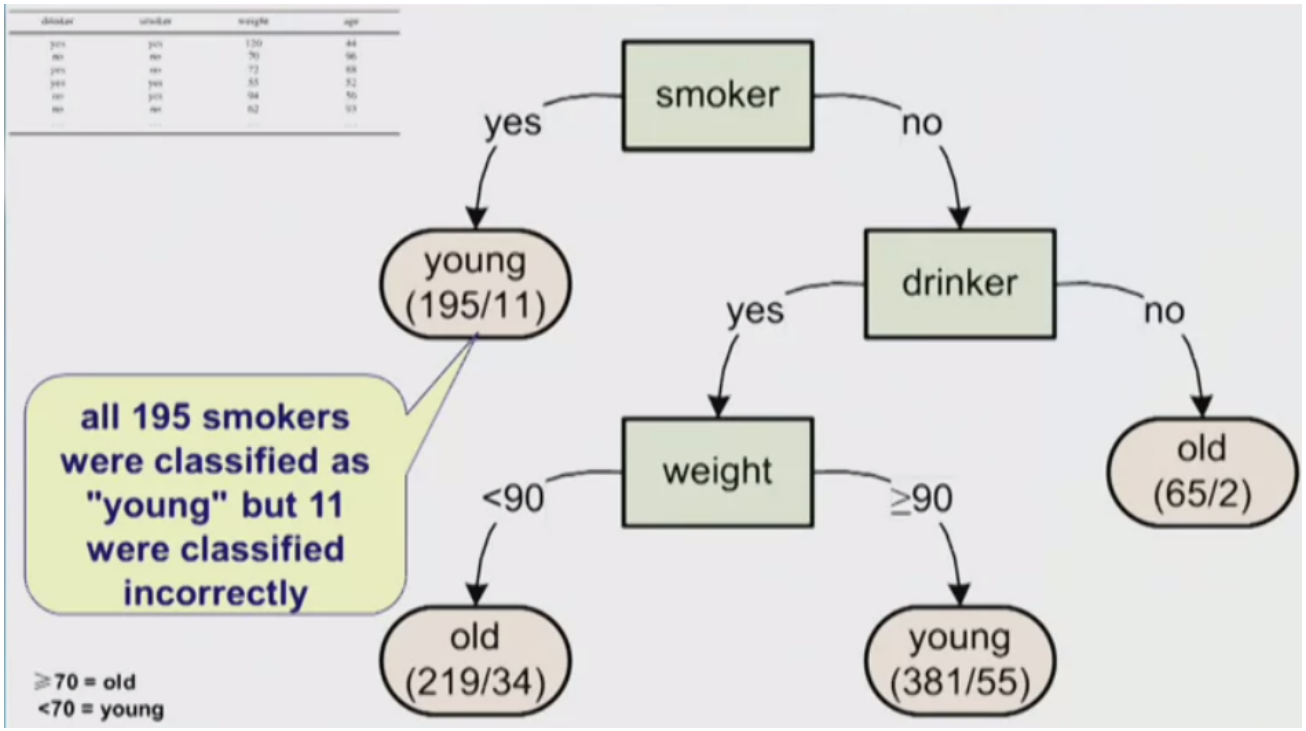
Decision Trees Classifier



Decision Trees Classifier

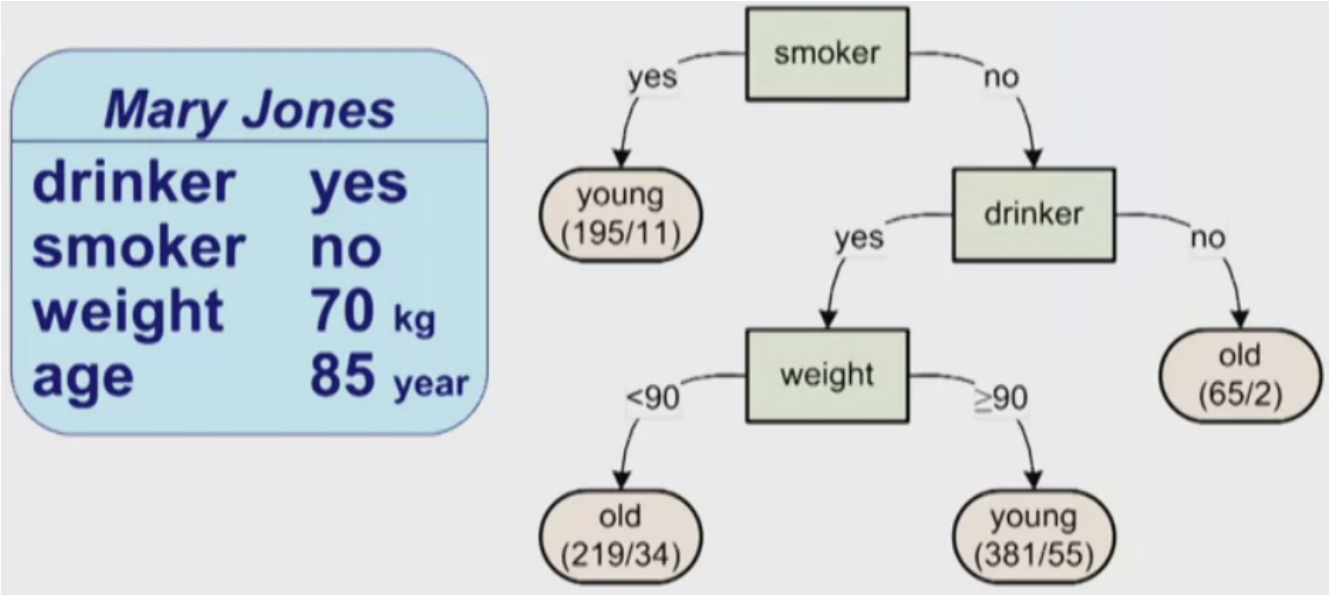


Decision Trees Classifier



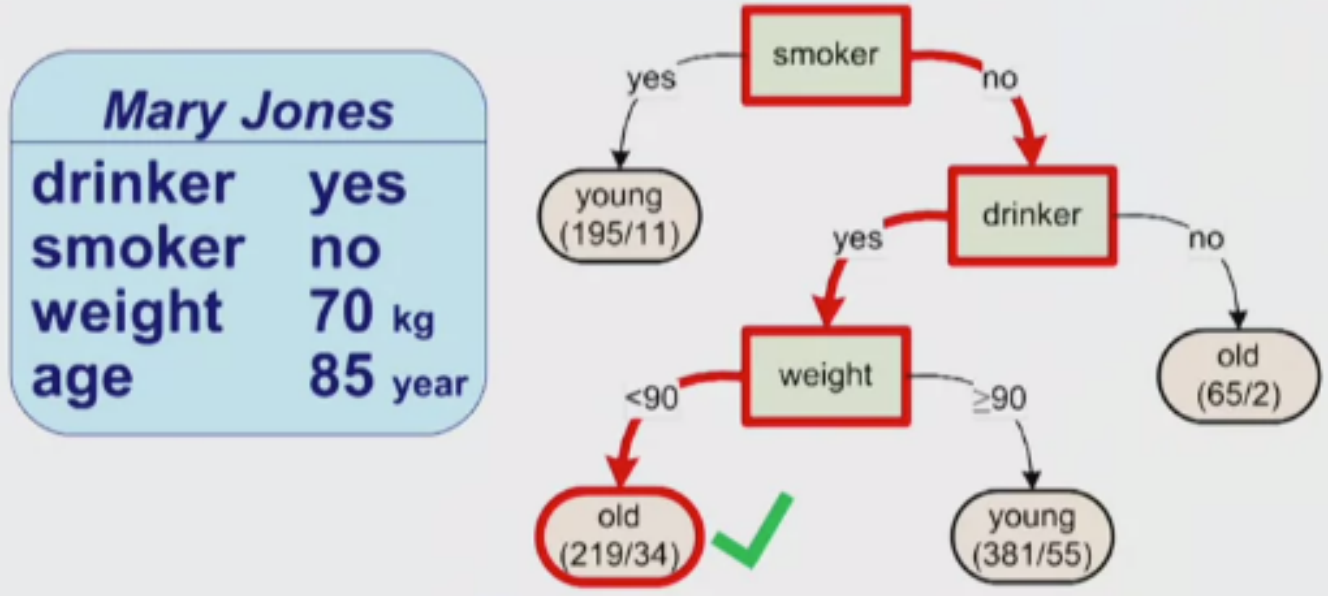
Decision Trees Classifier

- Question 1: Consider Mary, She did not smoke, did drink, had a weight of 70 kilogram, and died at the age of 85. Was she classified correctly?



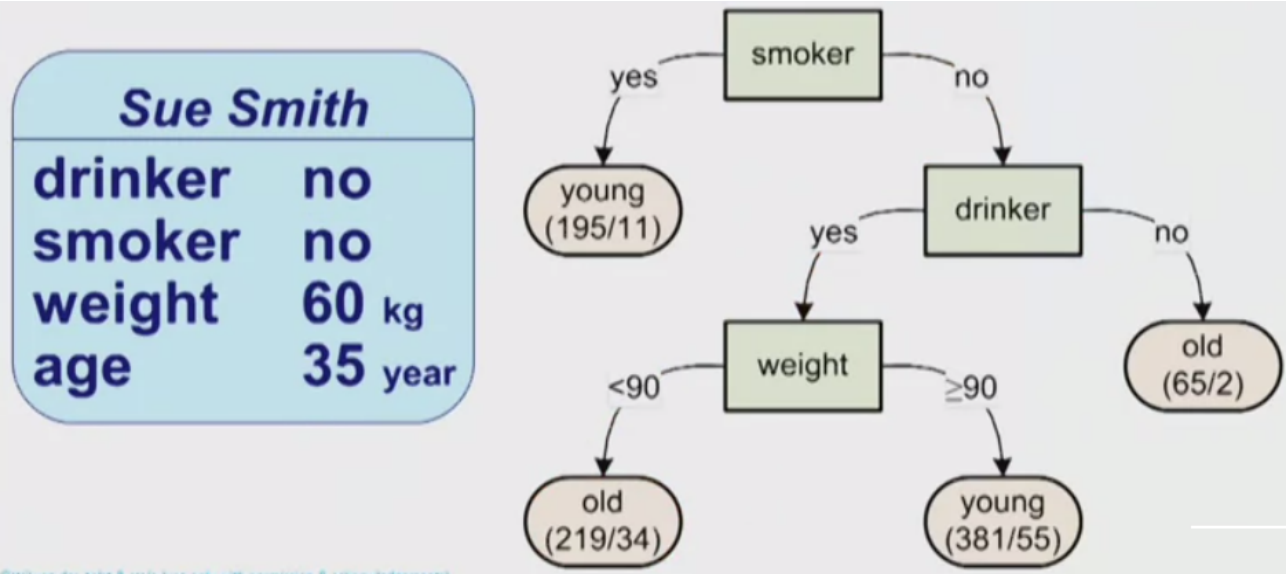
Decision Trees Classifier

- Answer: Yes



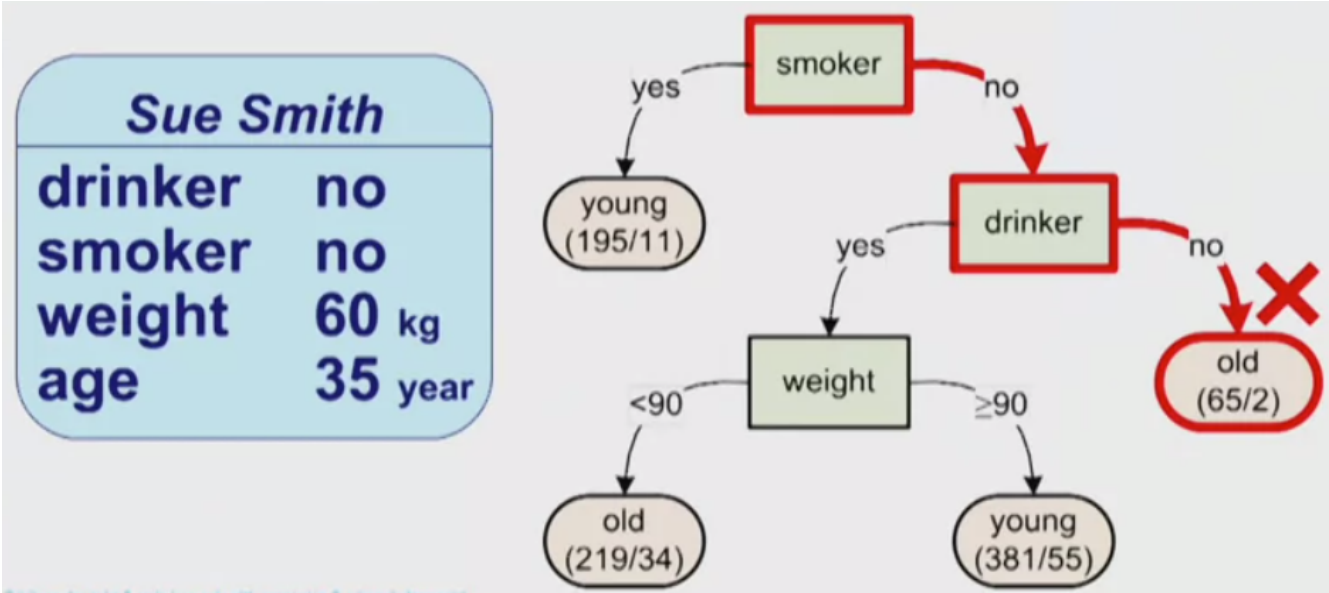
Decision Trees Classifier

- Question 2: Consider Sue, She did not smoke, did not drink, had a weight of 60 kilogram, and died at the age of 35. Was she classified correctly?



Decision Trees Classifier

- Answer: No



Decision Trees Classifier

School of Computing & IT

- Dataset 2: Effect of individual course results on graduation

linear algebra	logic	program-ming	operations research	workflow systems	...	duration	result
9	8	8	9	9	...	36	cum laude
7	6	-	8	8	...	42	passed
-	-	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	-	8	...	39	passed
9	9	9	9	8	...	38	cum laude
5	5	-	6	6	...	52	failed
...	...	...	...	...	...	...	...

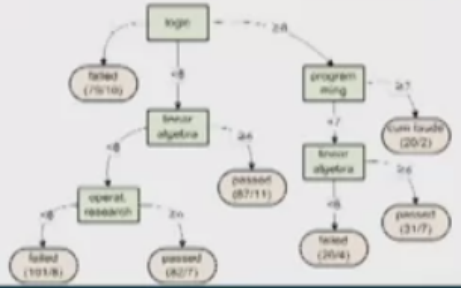
Decision Trees Classifier

School of Computing & IT

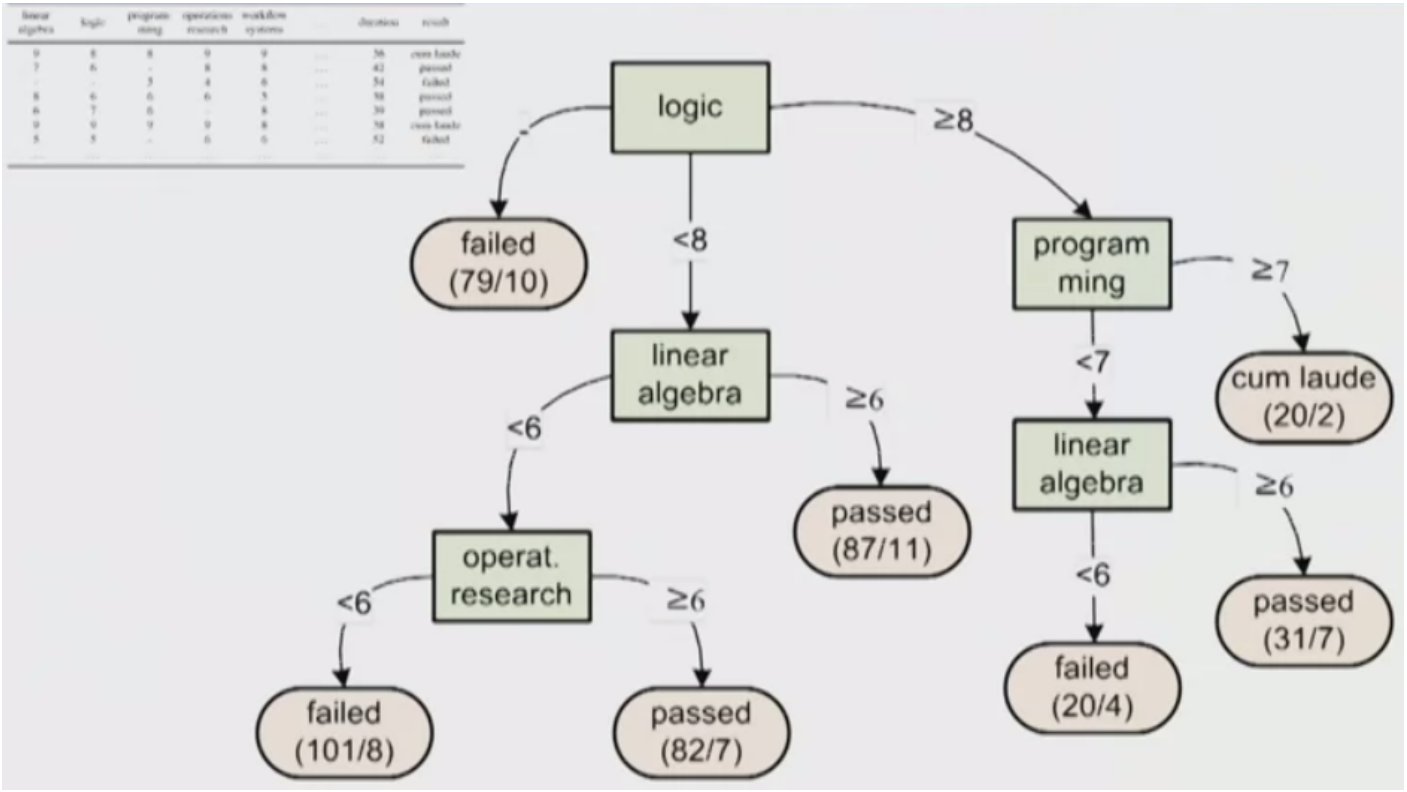
linear algebra	logic	program-ming	operations research	workflow systems	..	duration	result
9	8	8	9	9	..	36	cum laude
7	6	-	8	8	..	42	passed
-	-	5	4	6	..	54	failed
8	6	6	6	5	..	38	passed
6	7	6	-	8	..	39	passed
9	9	9	9	8	..	38	cum laude
5	5	-	6	6	..	52	failed
...	...	..	...	...	..	...	...

predictor variables

response variable



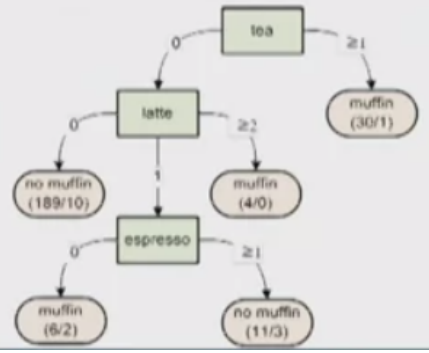
Decision Trees Classifier



Decision Trees Classifier

- Dataset 3: Muffin or no muffin

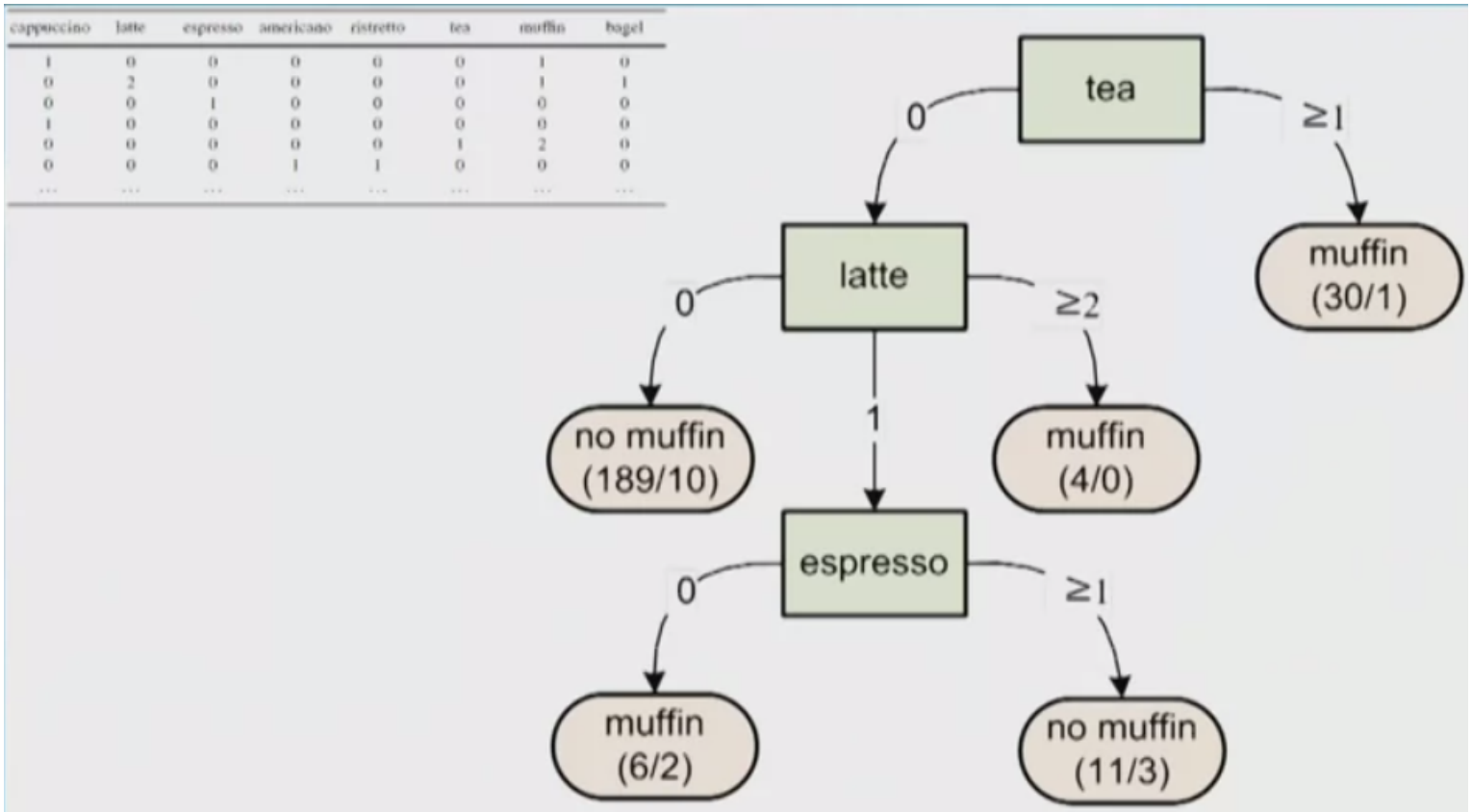
cappuccino	latte	espresso	americano	ristretto	tea	muffin	bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...



response variable

⩾1 = "muffin"
0 = "no muffin"

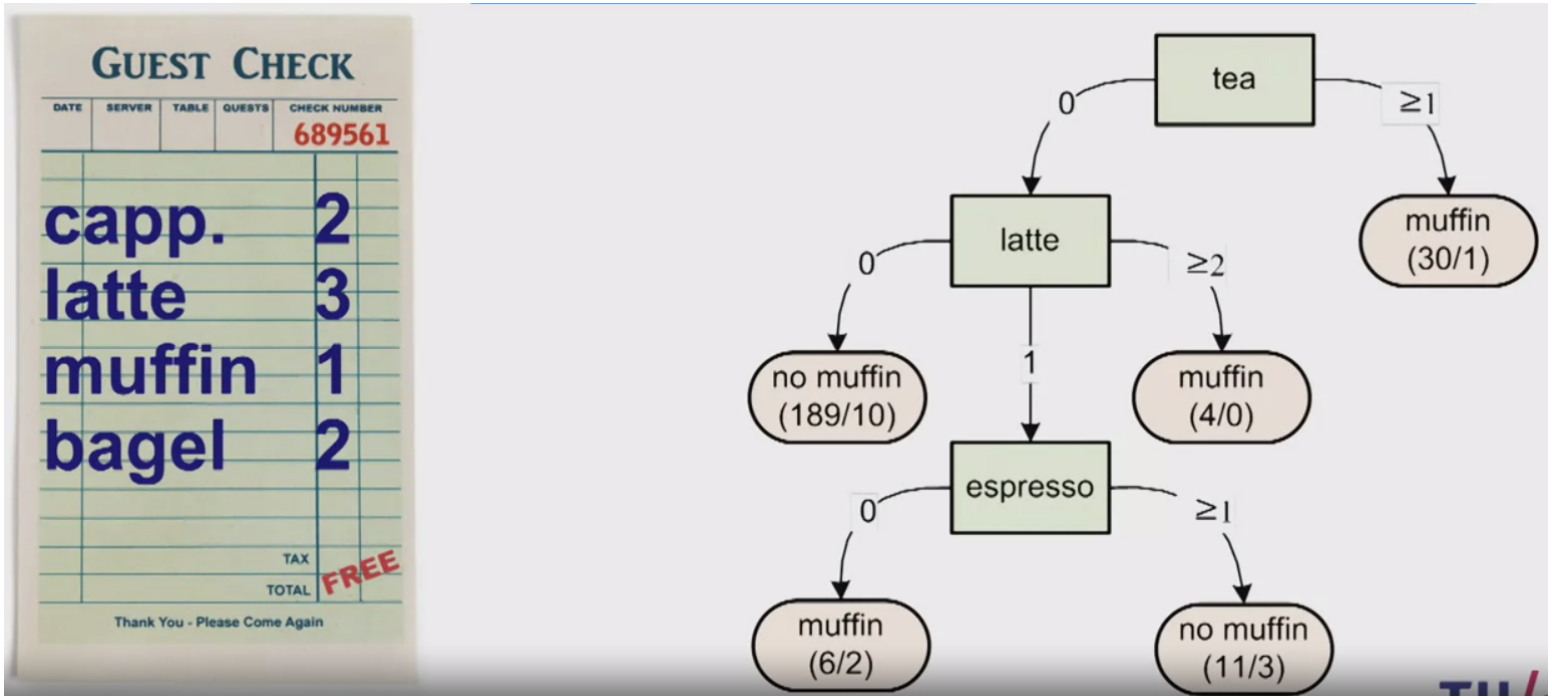
Decision Trees Classifier



cappuccino	latte	espresso	americano	ristretto	tea	muffin	bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

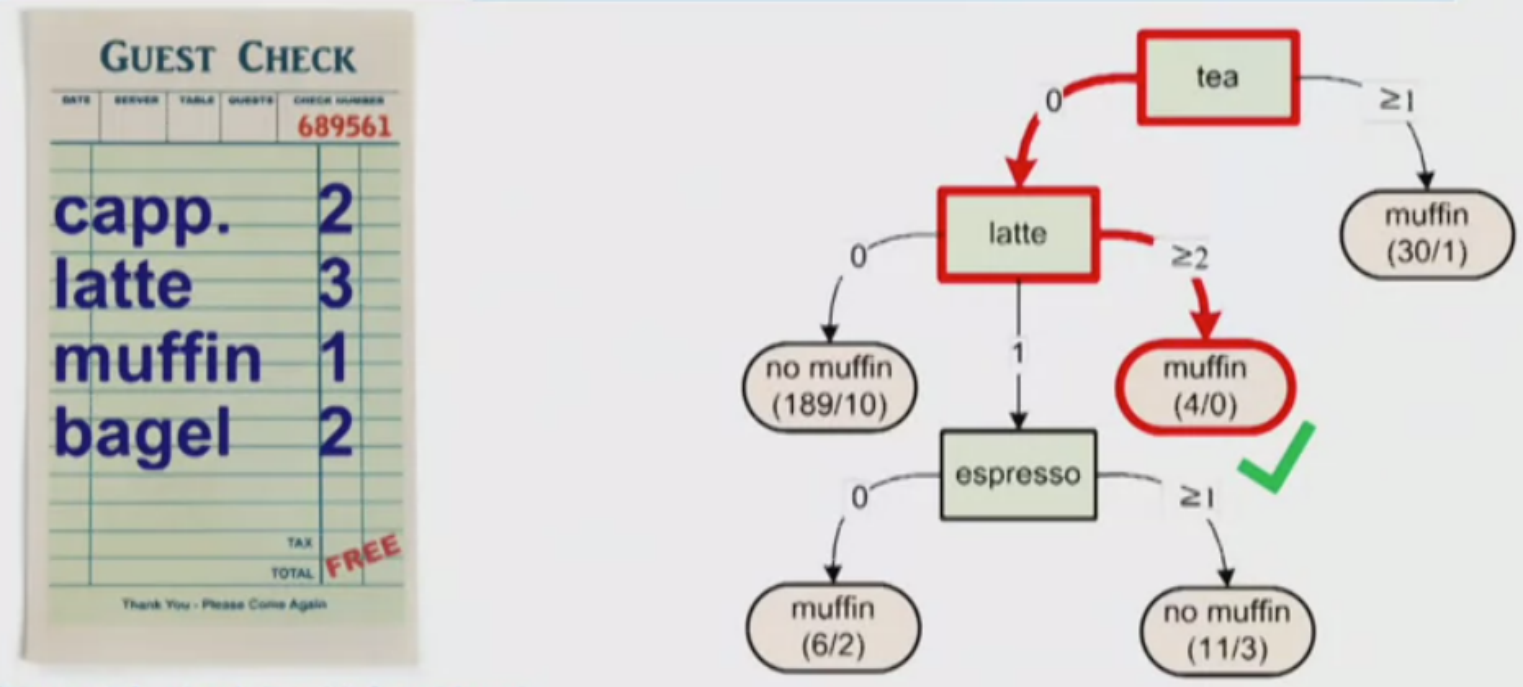
Decision Trees Classifier

- Question 4: Consider check (bill) of the visitor of the restaurant. Was it classified correctly?



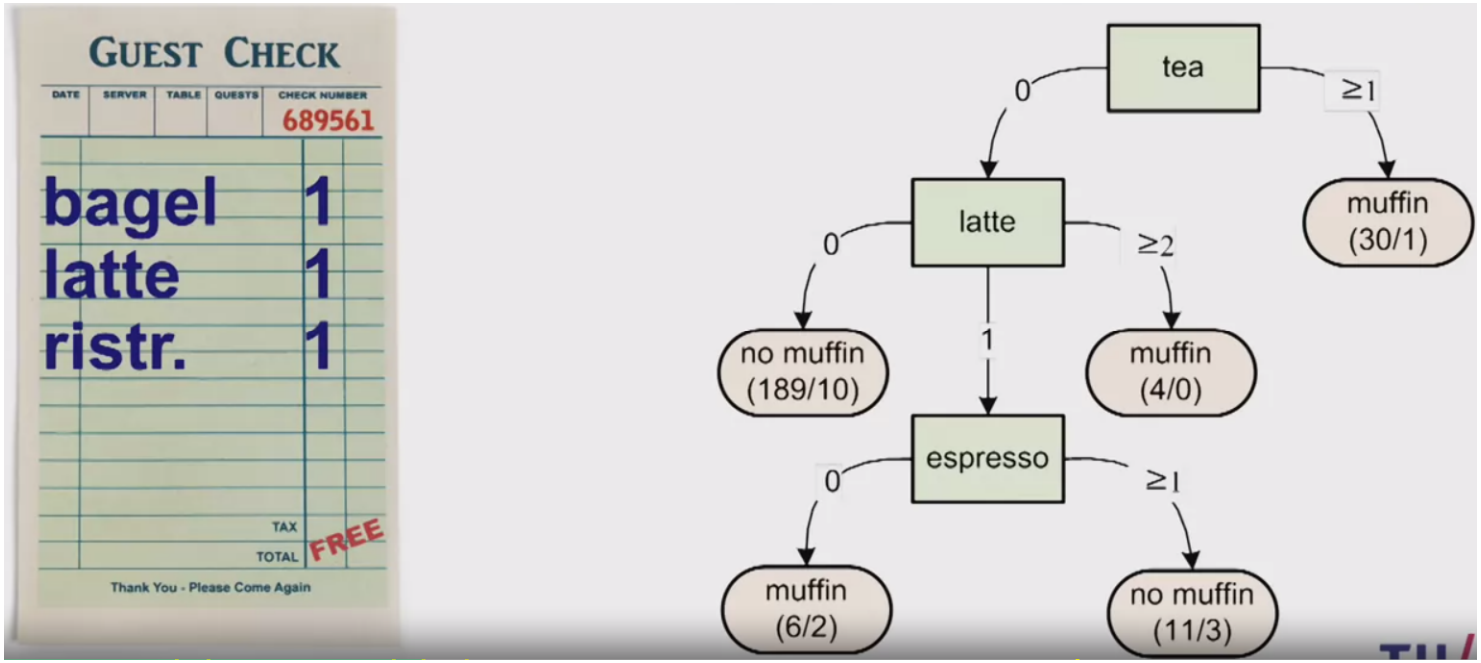
Decision Trees Classifier

- Answer: Yes



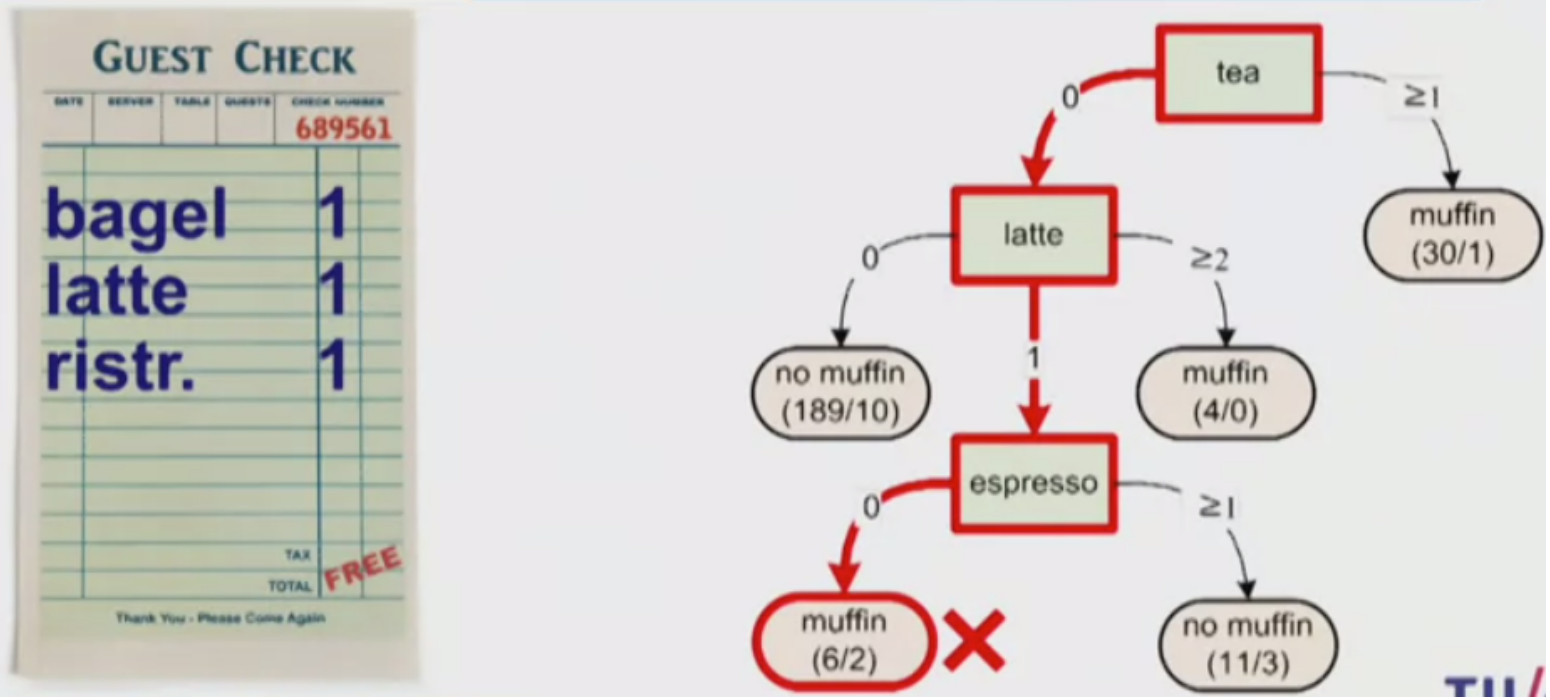
Decision Trees Classifier

- Question 5: Consider check (bill) of the visitor of the restaurant. A customer ordered a latte, a bagel, and a ristretto. Was the customer classified correctly?



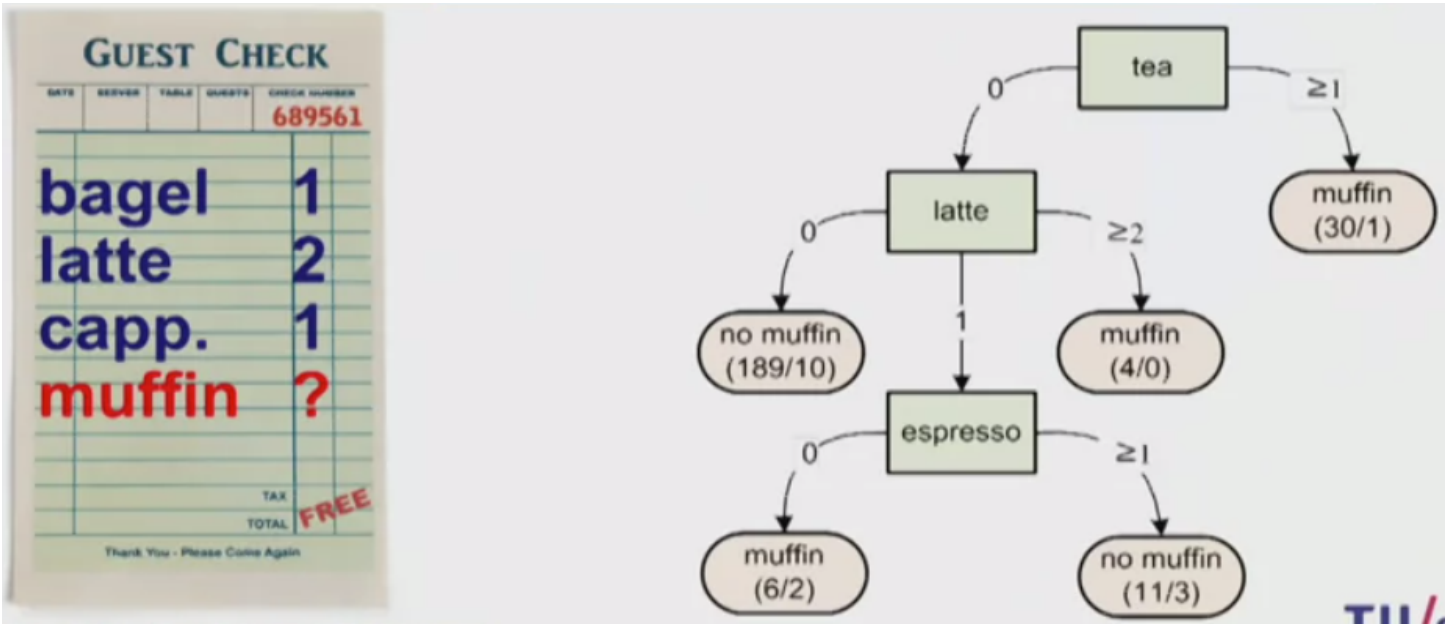
Decision Trees Classifier

- Answer: No



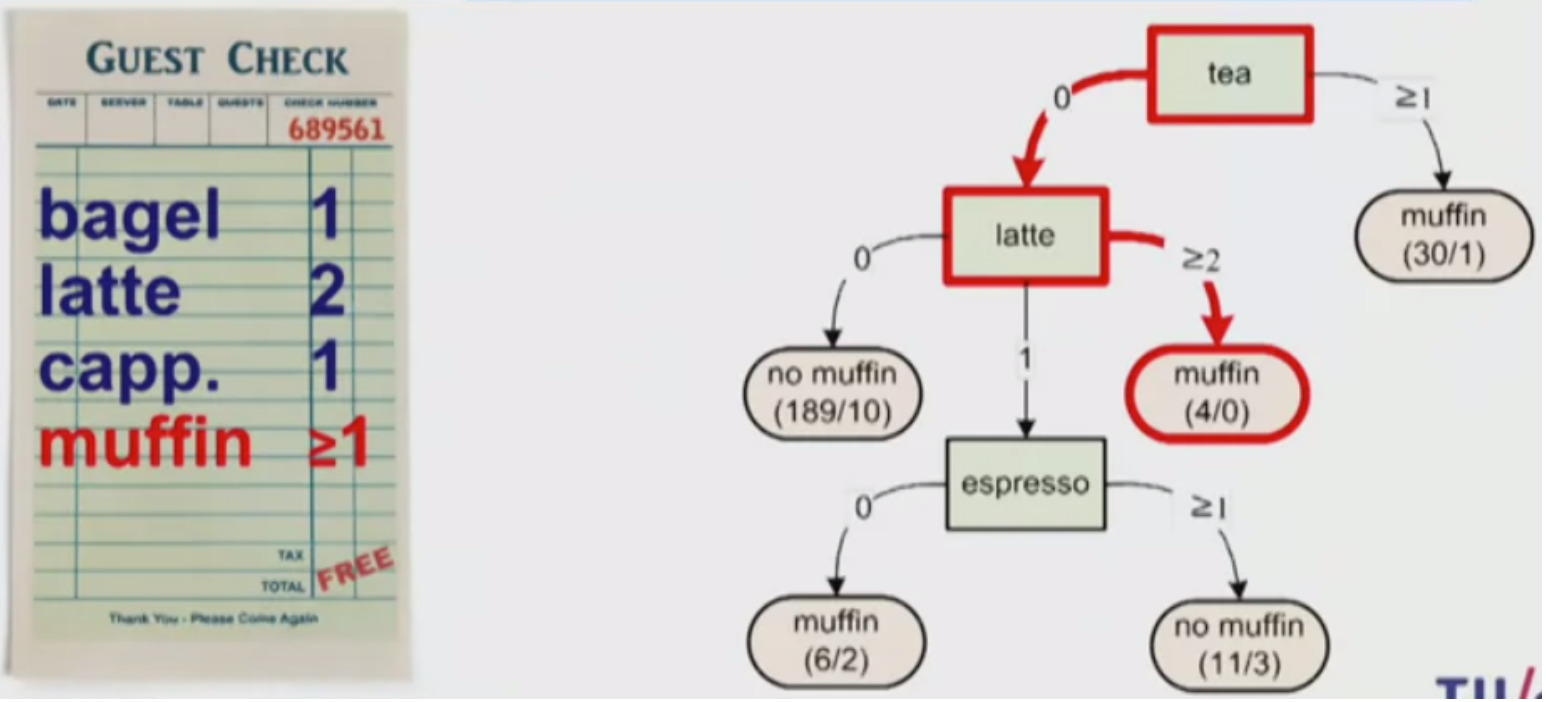
Decision Trees Classifier

- Question 6: Consider check (bill) of the visitor of the restaurant. A customer ordered two latte, a bagel, and a cappuccino. Did this person eat a muffin?



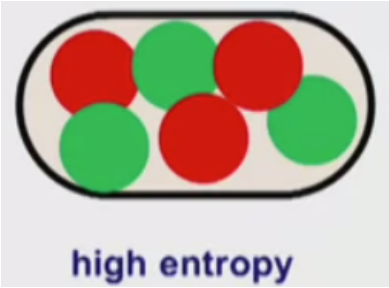
Decision Trees Classifier

- Answer: Yes



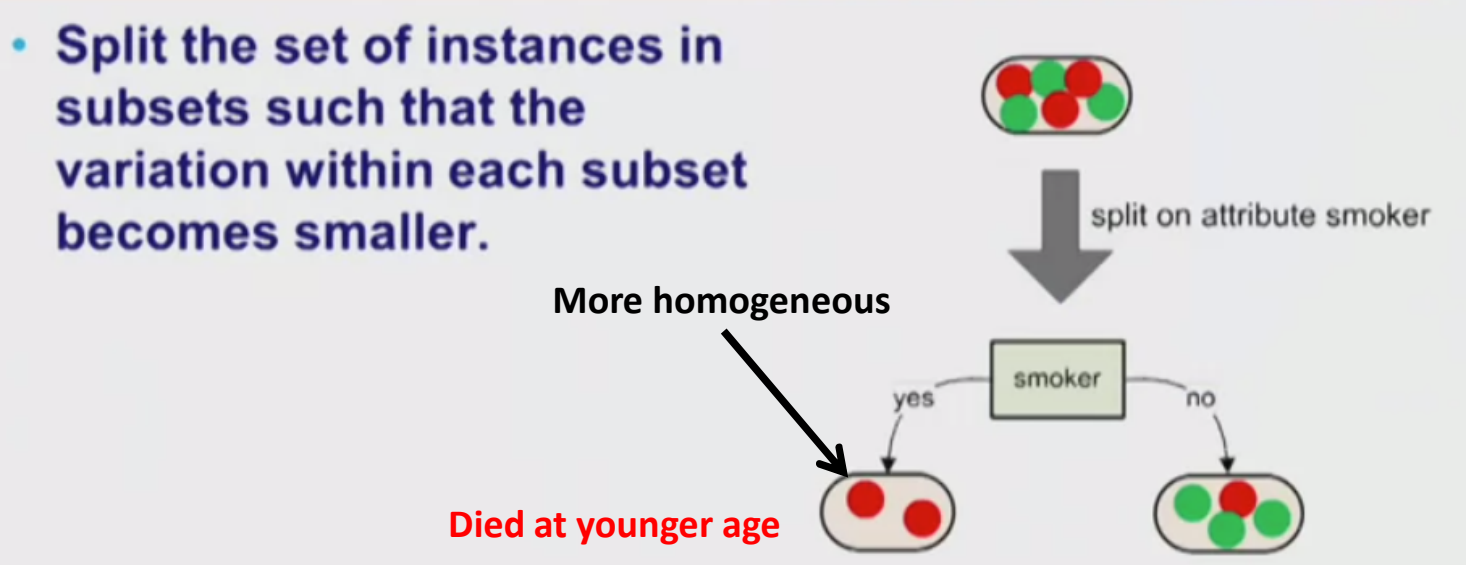
Decision Trees Classifier

- How does it work ? – Basic idea
- Decision trees can be used for understanding data and predicting data
- How can we automatically learn decision tree?
  - Basically done by splitting nodes to reduce the variability
- Example: We are very uncertain weather is should be green or red



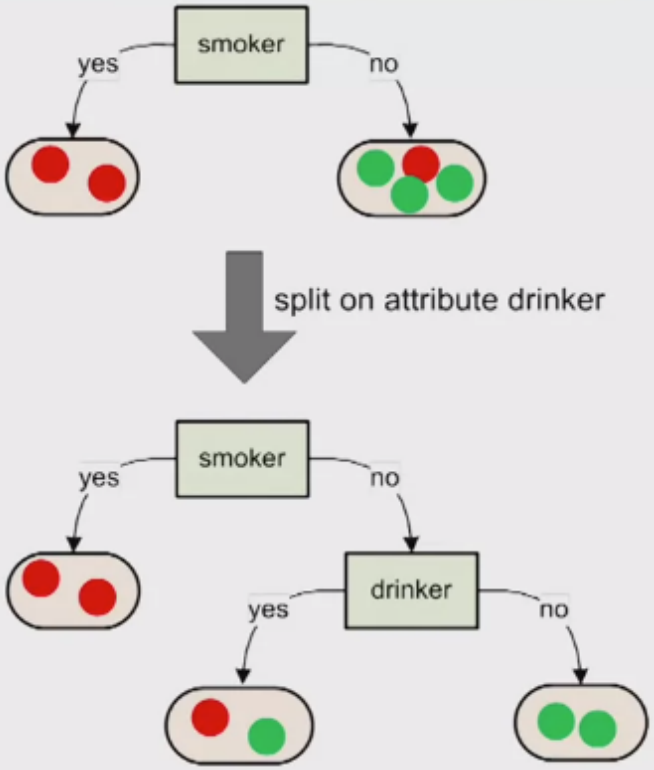
Decision Trees Classifier

- Example: If we split set of people into smokers and non-smokers, then we find two smokers died at younger age

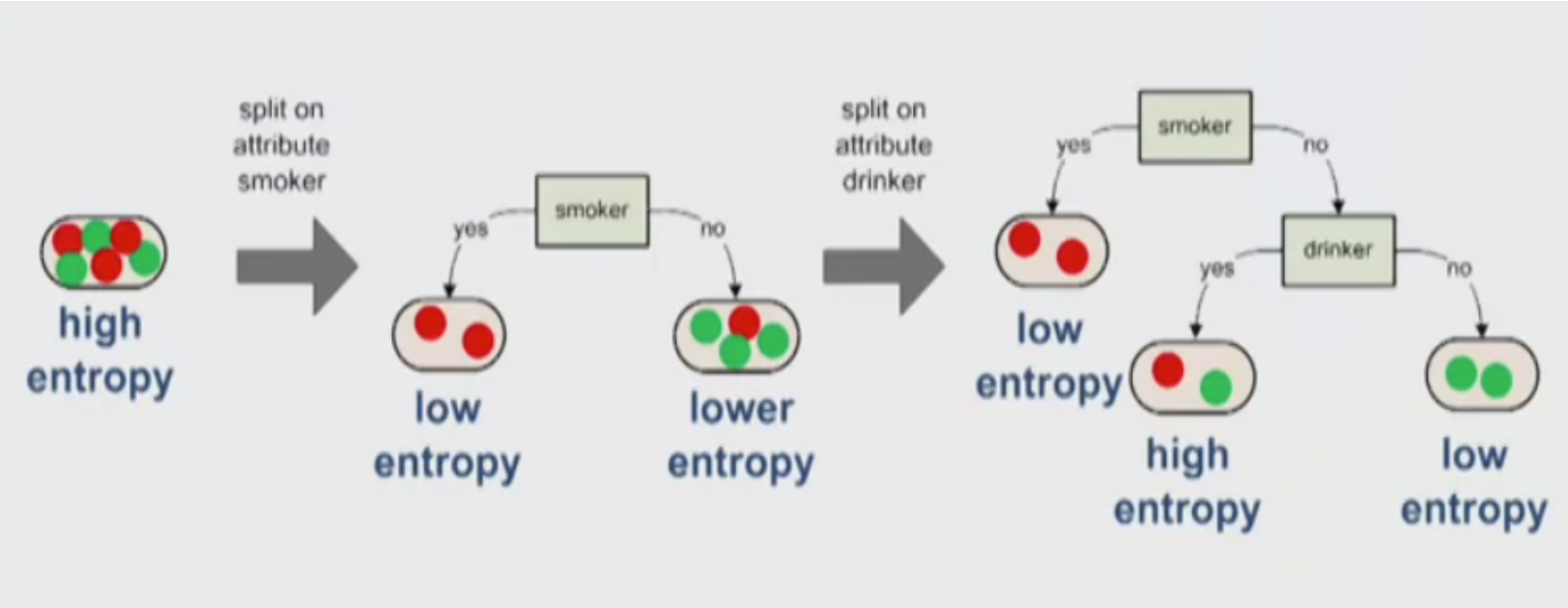


Decision Trees Classifier

- **Split the set of instances in subsets such that the variation within each subset becomes smaller.**



Decision Trees Classifier



Decision Trees Classifier

- **High Entropy**
- Degree of uncertainty
- Also can call as inverse of “compressibility”
- Goal: Reduce entropy in leaves of tree to improve predictability

Decision Trees Classifier

- Logarithms

$$\log_2(x) = y \Leftrightarrow 2^y = x$$

$$\log_2(2^n) = n \qquad \log_2(\frac{1}{2^n}) = -n$$

$$\log_2(1) = 0 \qquad \log_2(2) = 1$$

$$\log_2(8) = 3 \qquad \log_2(0.125) = -3$$

$$\log_2(1024) = 10 \qquad \log_2(0.75) = -0.415$$

Decision Trees Classifier

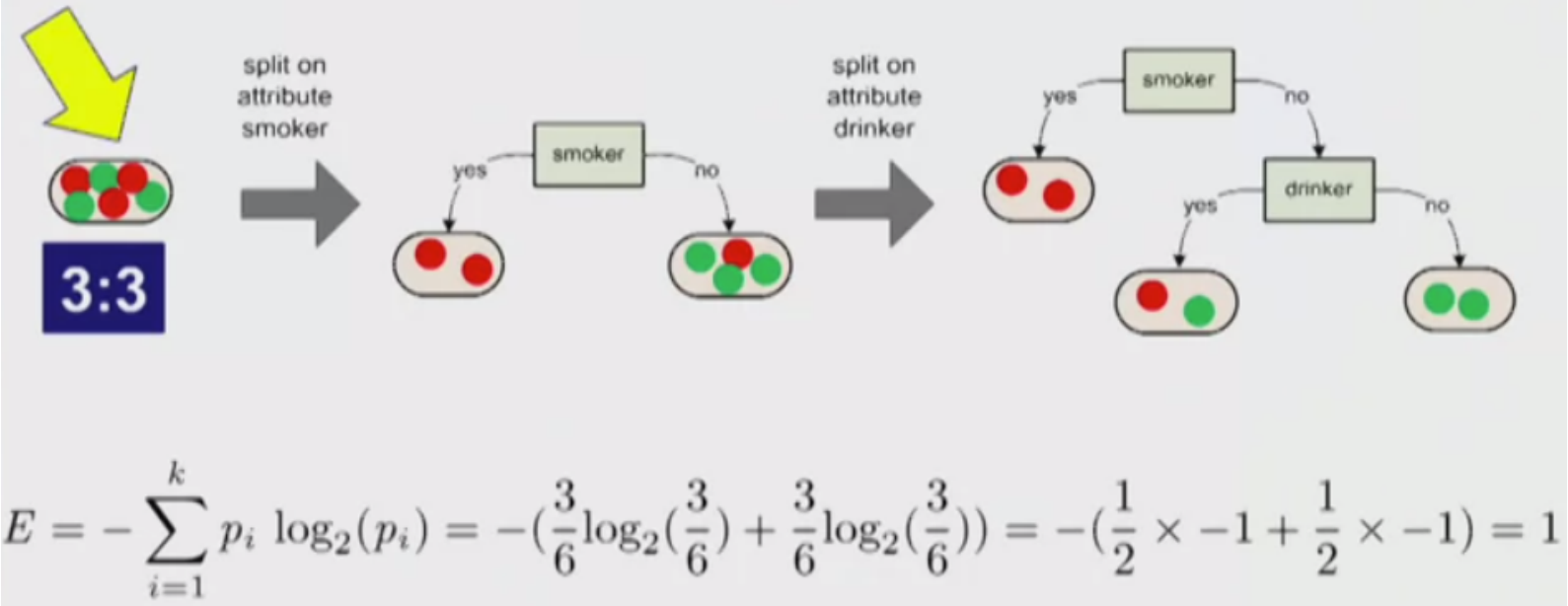
- Definition of Entropy

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

k possible values enumerated $1, 2, \ldots, k$
$p_i = \frac{c_i}{n}$ is the fraction of elements having value i
with $c_i \geq 1$ the number of i values and $n = \sum_{i=1}^{k} c_i$

Decision Trees Classifier

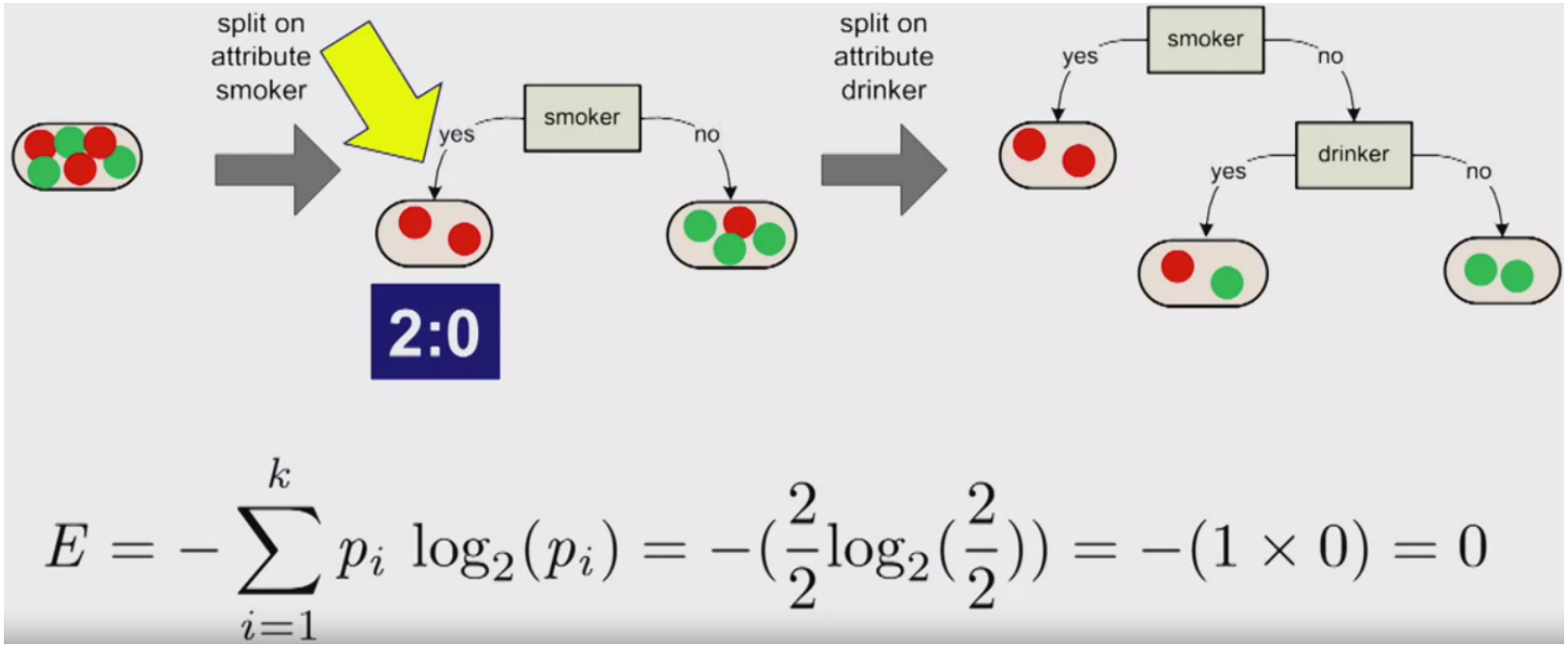
- Example E=1 (three red, three green)



$$E = -\sum_{i=1}^{k} p_i \log_2(p_i) = -(\frac{3}{6}\log_2(\frac{3}{6}) + \frac{3}{6}\log_2(\frac{3}{6})) = -(\frac{1}{2} \times -1 + \frac{1}{2} \times -1) = 1$$

Decision Trees Classifier

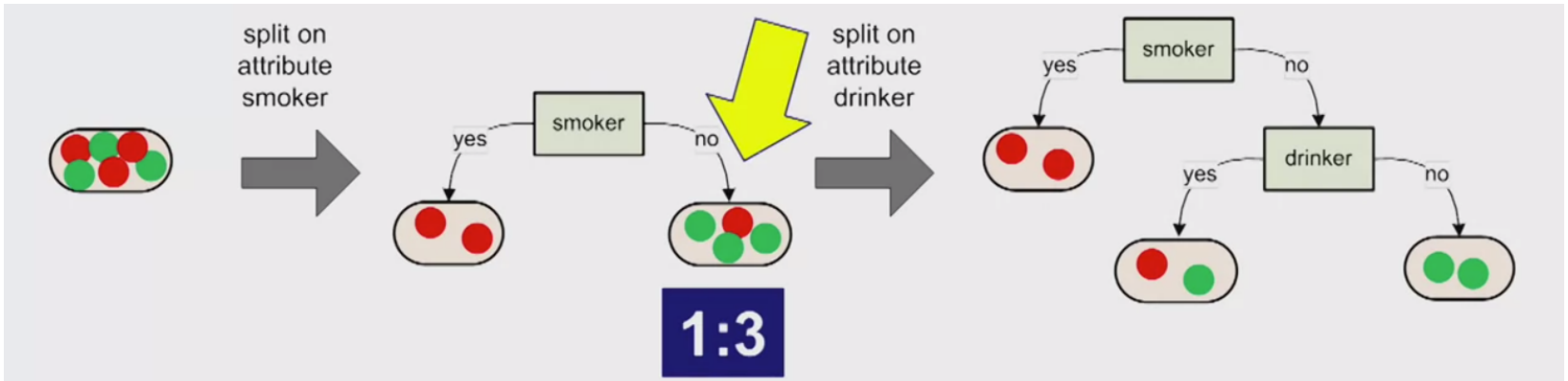
- Example E=0 (two red, no green)



$$E = -\sum_{i=1}^{k} p_i \log_2(p_i) = -(\frac{2}{2}\log_2(\frac{2}{2})) = -(1 \times 0) = 0$$

Decision Trees Classifier

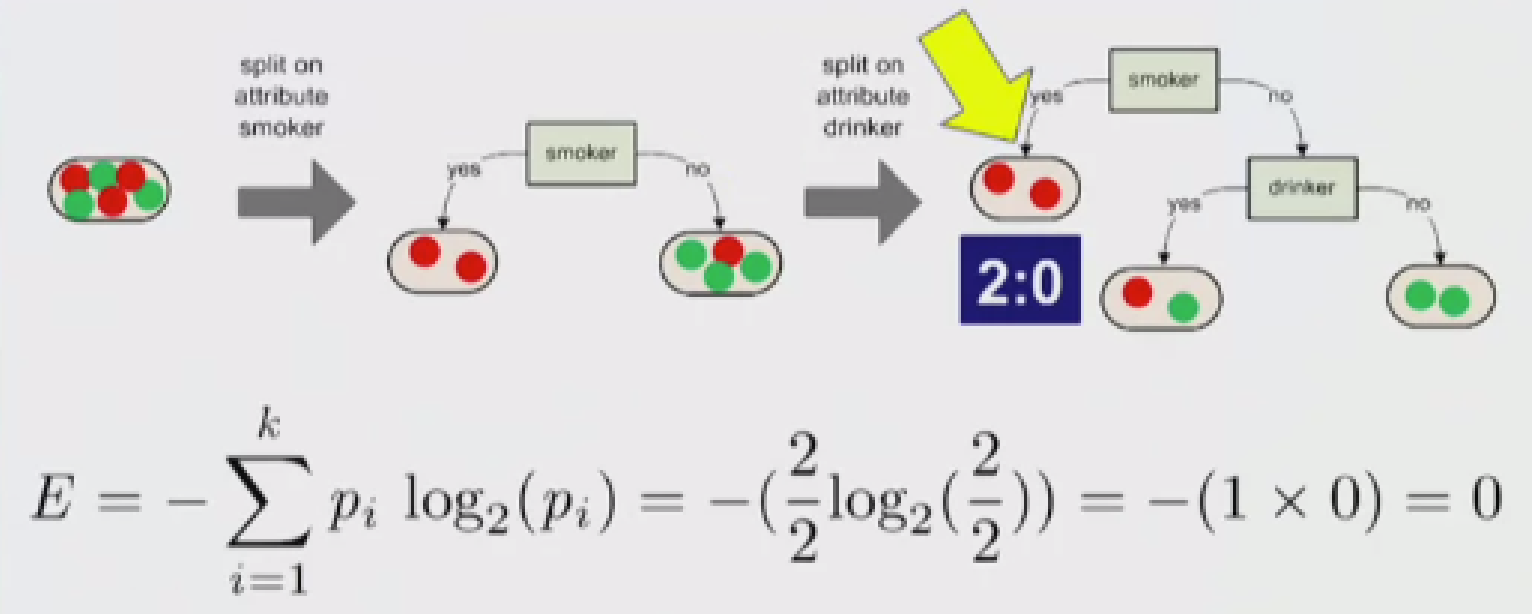
- Example E=0.811 (one red, three green)



$$E = -\sum_{i=1}^{k} p_i \log_2(p_i) = -(\frac{1}{4}\log_2(\frac{1}{4}) + \frac{3}{4}\log_2(\frac{3}{4})) = -(\frac{1}{4} \times -2 + \frac{3}{4} \times -0.415) = 0.811$$

Decision Trees Classifier

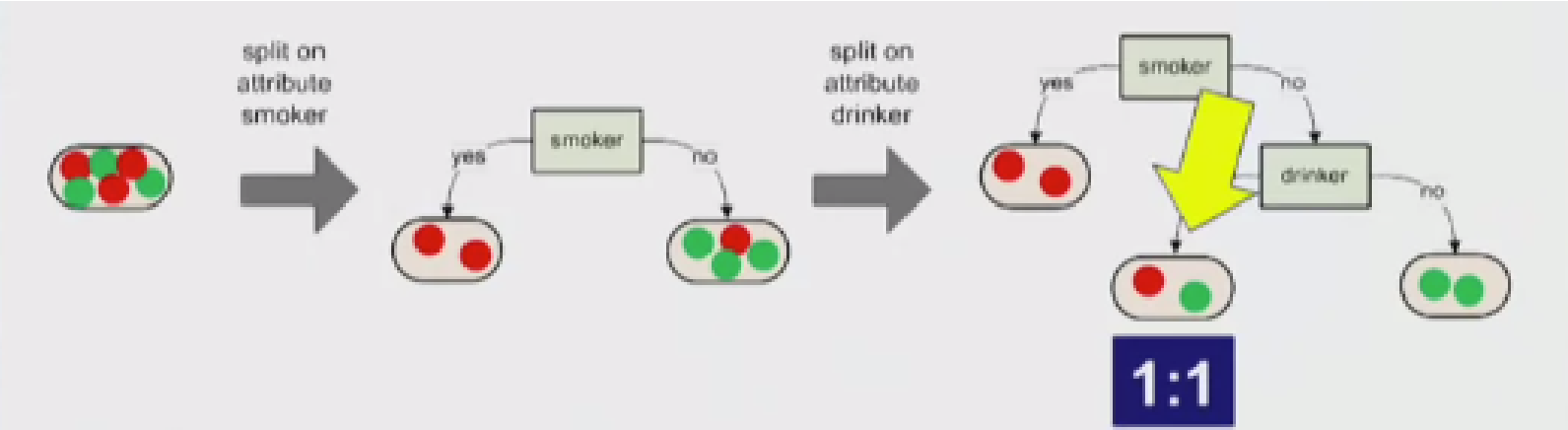
- Example E=0 (two red, no green)



$$E = -\sum_{i=1}^{k} p_i \log_2(p_i) = -(\frac{2}{2}\log_2(\frac{2}{2})) = -(1 \times 0) = 0$$

Decision Trees Classifier

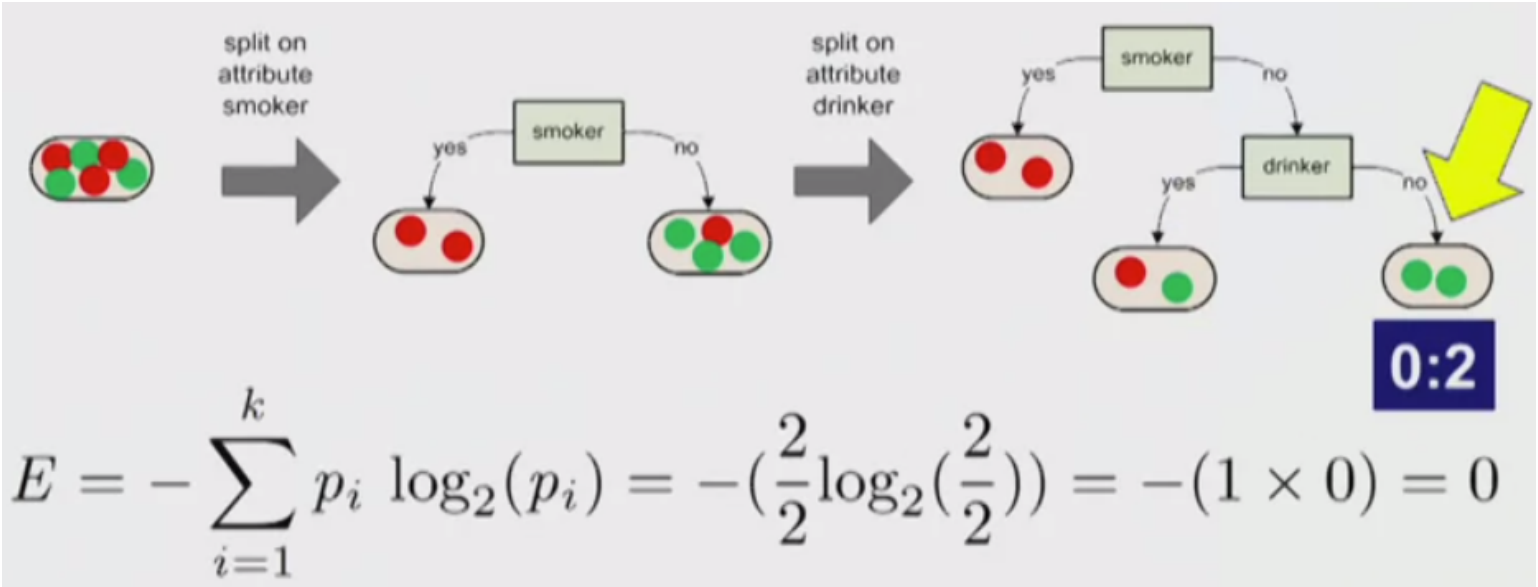
- Example E=1 (one red, two green)



$$E = -\sum_{i=1}^{k} p_i \log_2(p_i) = -(\frac{1}{2}\log_2(\frac{1}{2}) + \frac{1}{2}\log_2(\frac{1}{2})) = -(\frac{1}{2} \times -1 + \frac{1}{2} \times -1) = 1$$

Decision Trees Classifier

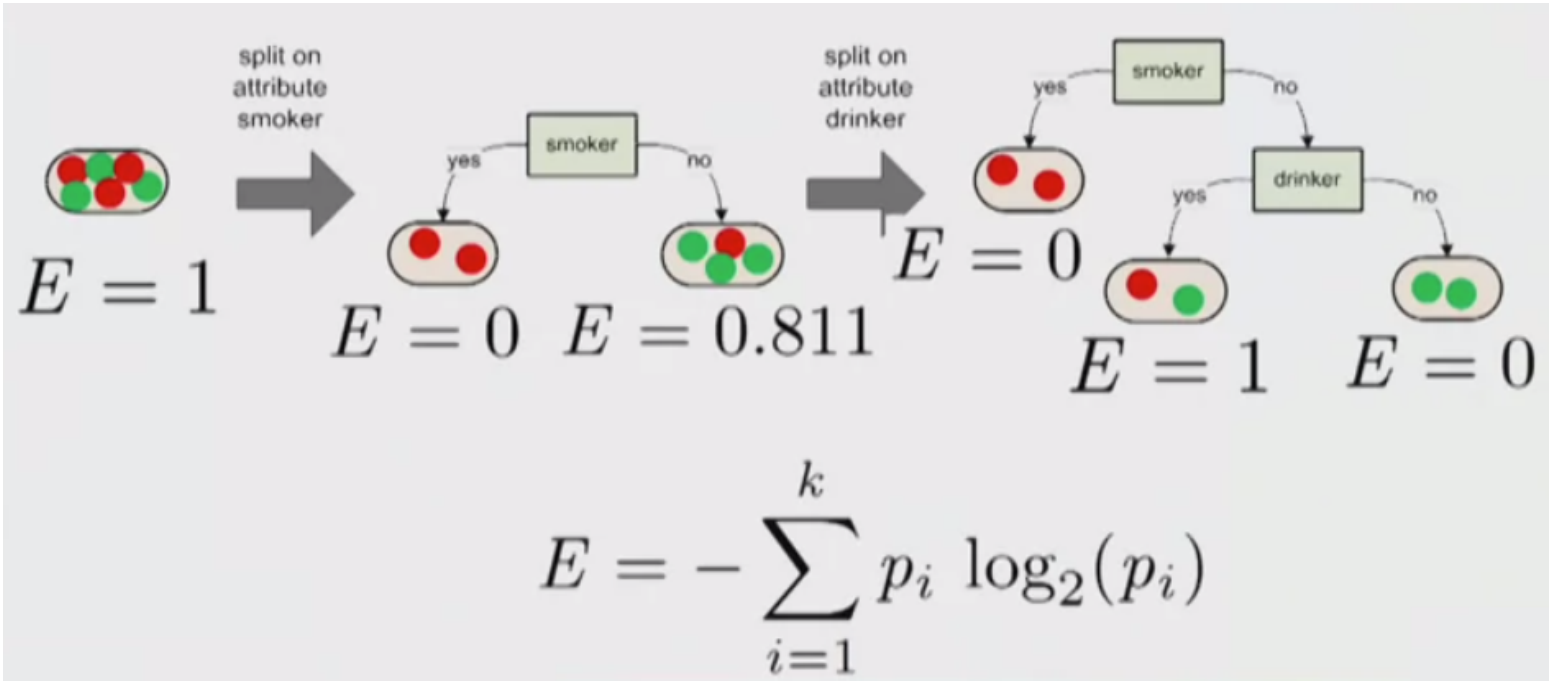
- Example E=1 (two red, no green)



$$E = -\sum_{i=1}^{k} p_i \log_2(p_i) = -(\frac{2}{2}\log_2(\frac{2}{2})) = -(1 \times 0) = 0$$

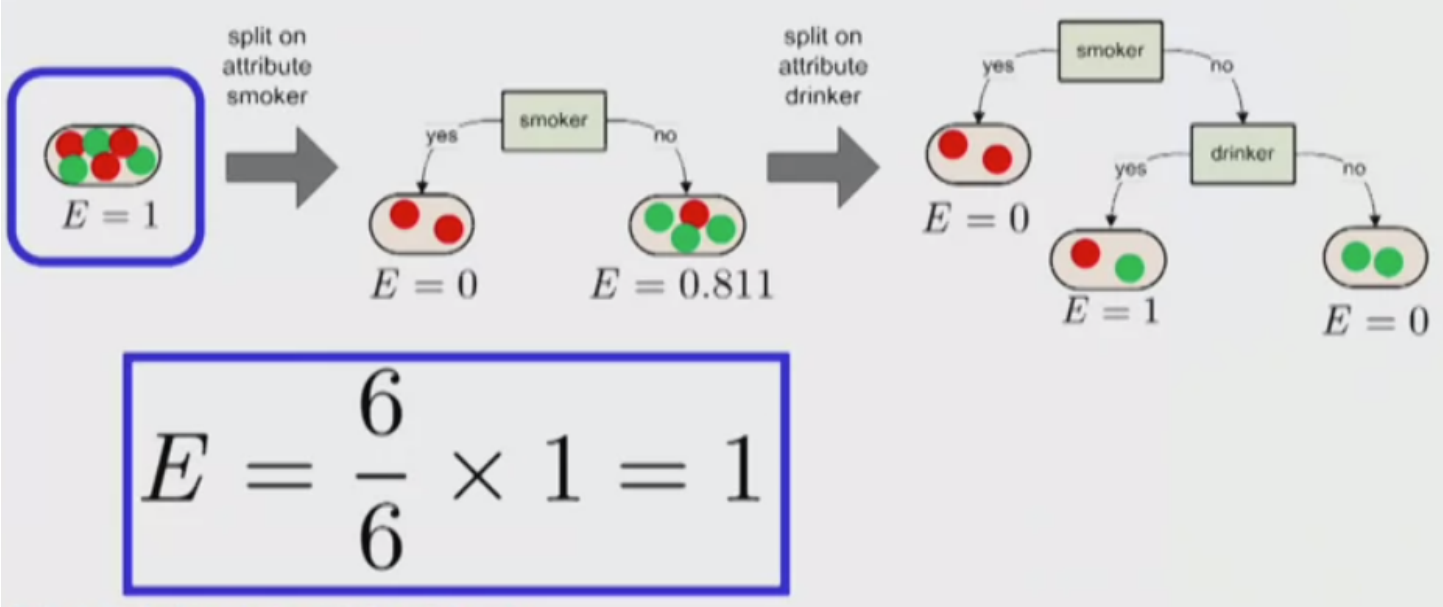
Decision Trees Classifier

- Entropy Values



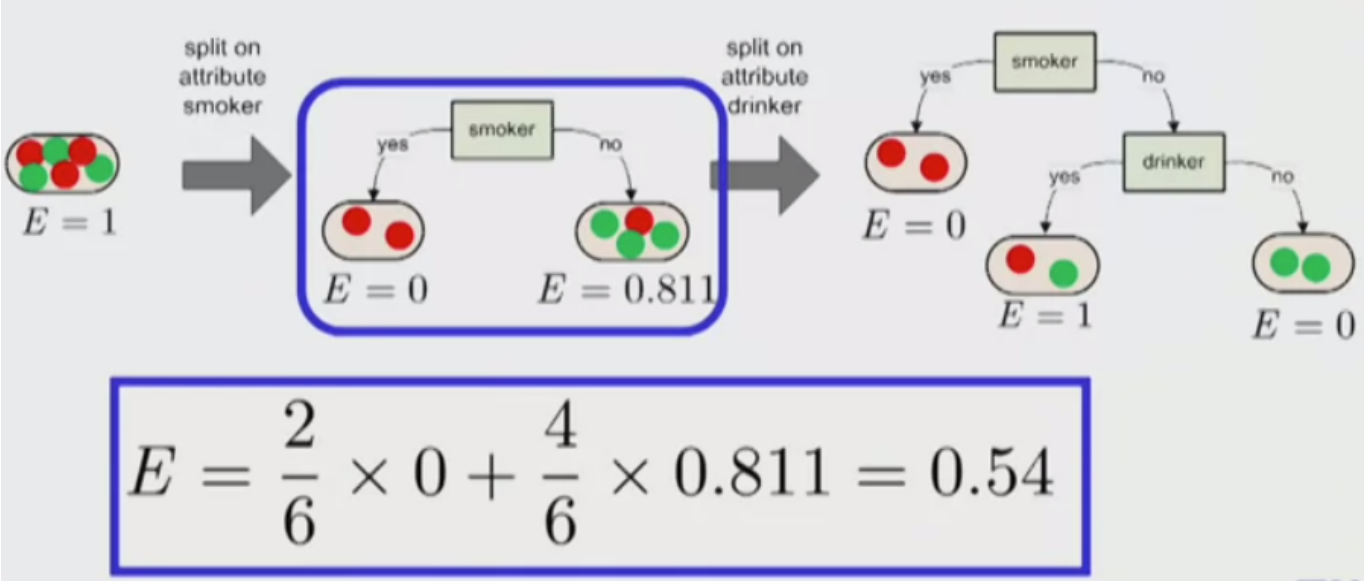
Decision Trees Classifier

- Weighted Average



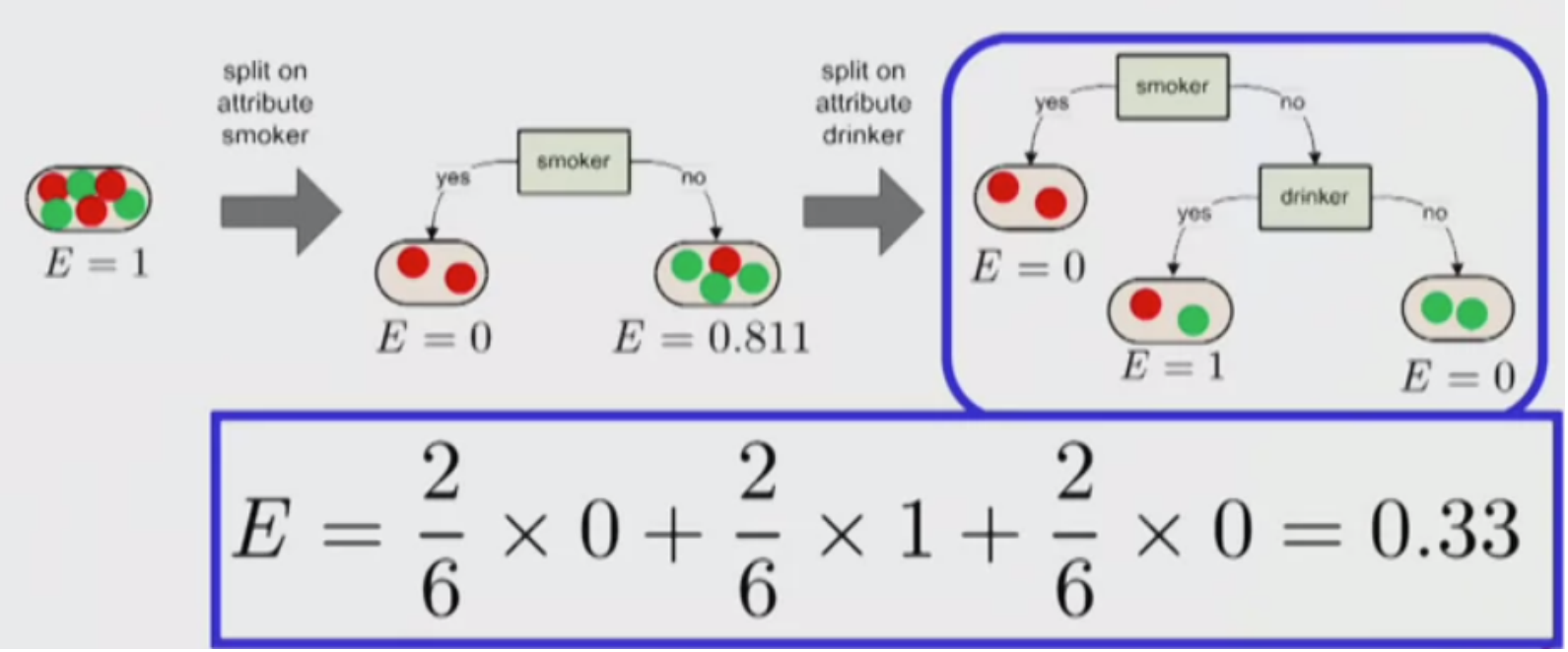
Decision Trees Classifier

- Weighted Average



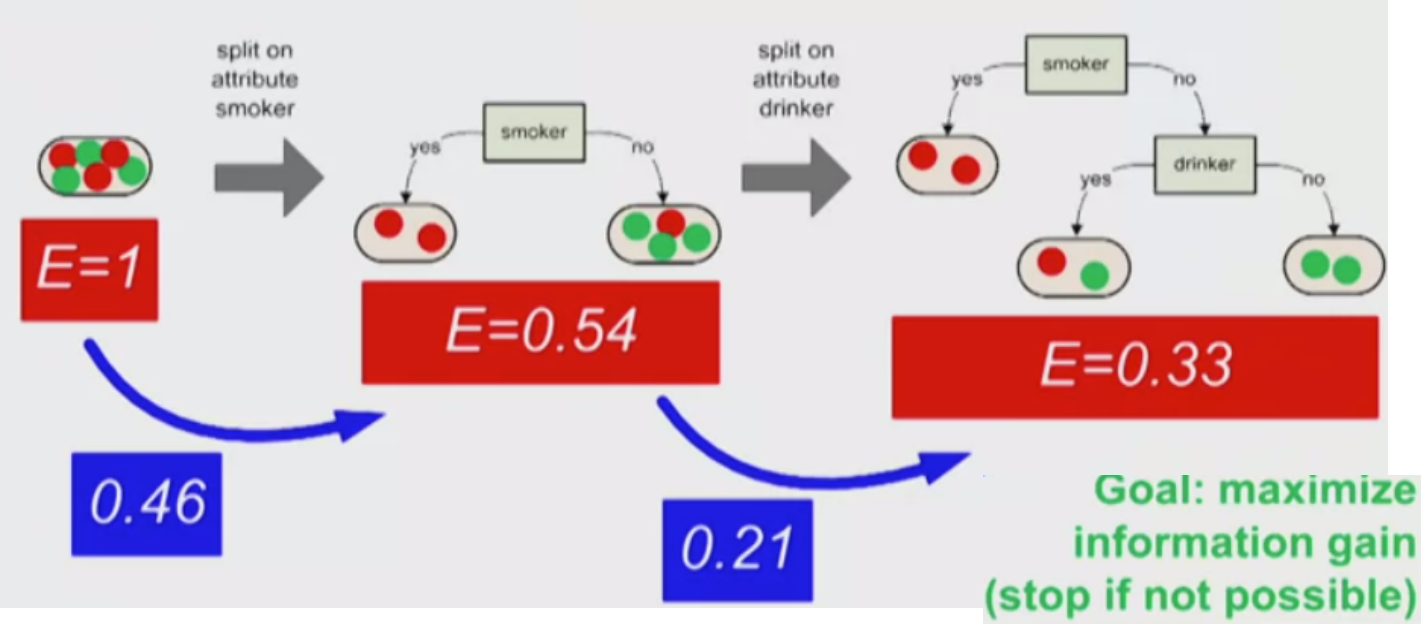
Decision Trees Classifier

- Weighted Average



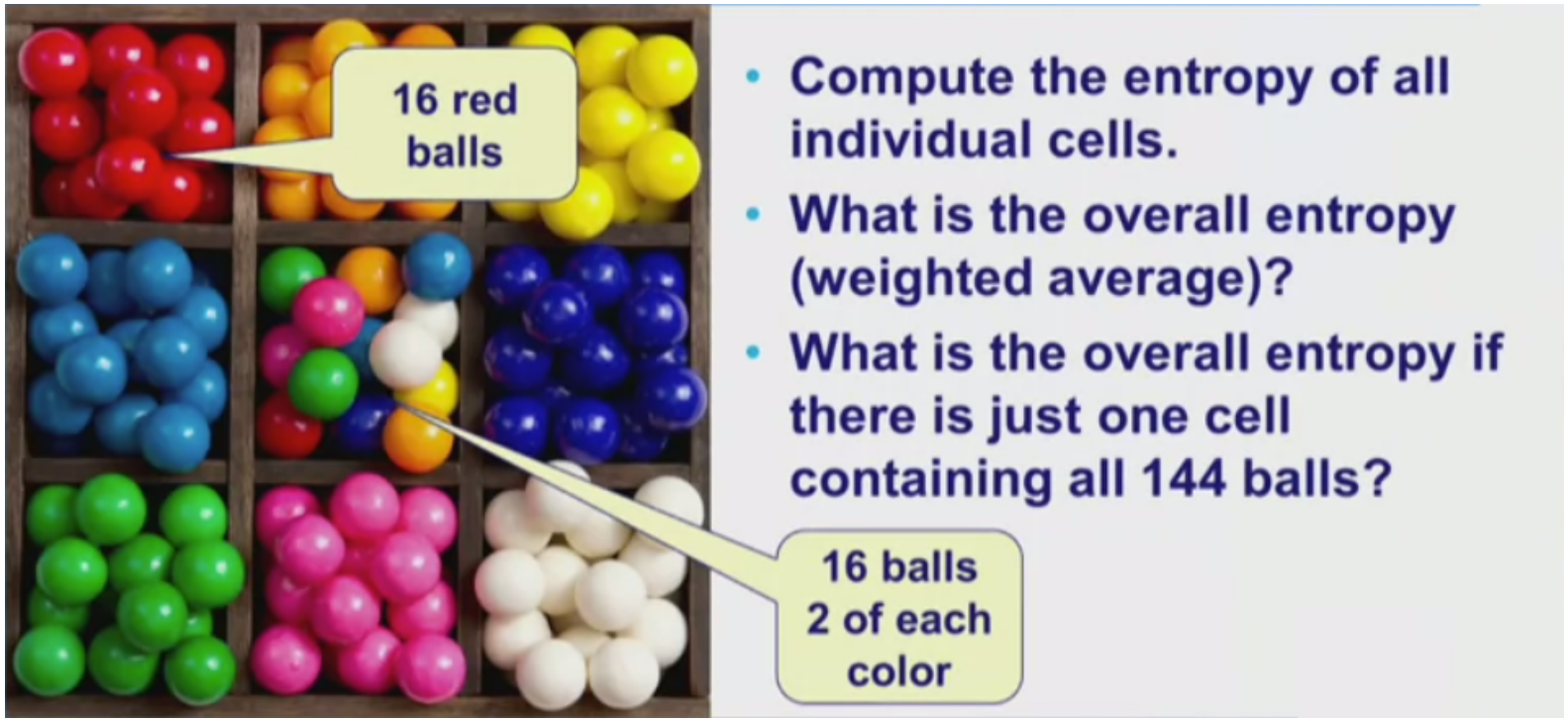
Decision Trees Classifier

- Information gain



Decision Trees Classifier

- Dataset 1:



Decision Trees Classifier

- Question 1: Consider the following distribution of 144 balls over 9 cells. One cell has 16 balls in eight different colors (two of each). The other eight cells each have 16 balls of the same color. What is the entropy of the different cells?

1. One cell has an entropy of 5 and the other cells have an entropy of 1
2. One cell has an entropy of 4 and the other cells have an entropy of 1
3. One cell has an entropy of 2 and the other cells have an entropy of 0
4. One cell has an entropy of 3 and the other cells have an entropy of 0

Decision Trees Classifier

- Answer 1: E=3 for cell in middle

- **Cell in the middle: 2+2+2+2+2+2+2+2 balls**

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

$$= -\sum_{i=1}^{8} \frac{2}{16} \log_2(\frac{2}{16})$$

$$= -8 \times \frac{1}{8} \times -3 = 3$$

Decision Trees Classifier

- Answer 1: E=0 for other cells

- **Other cells:**
  16+0+0+0+0+0+0+0 balls

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

$$= -\sum_{i=1}^{1} \frac{16}{16} \log_2(\frac{16}{16})$$

$$= -1 \times 0 = 0$$

Decision Trees Classifier

- Question 2: Consider the following distribution of 144 balls over 9 cells. One cell has 16 balls in eight different colors (two of each). The other eight cells each have 16 balls of the same color. What is the overall entropy (weighted average over the 9 cells)?

1. The entropy is 0.250
2. The entropy is 0.333
3. The entropy is 0.500
4. The entropy is 1.200

Decision Trees Classifier

- Answer 2: Overall entropy (weighted average): E=0.33

$$E = \frac{16}{144} \times 0 + \frac{16}{144} \times 0 + \ldots + \frac{16}{144} \times 3$$
$$= 8 \times \frac{16}{144} \times 0 + \frac{16}{144} \times 3$$
$$= \frac{1}{9} \times 3 = \frac{1}{3}$$

Decision Trees Classifier

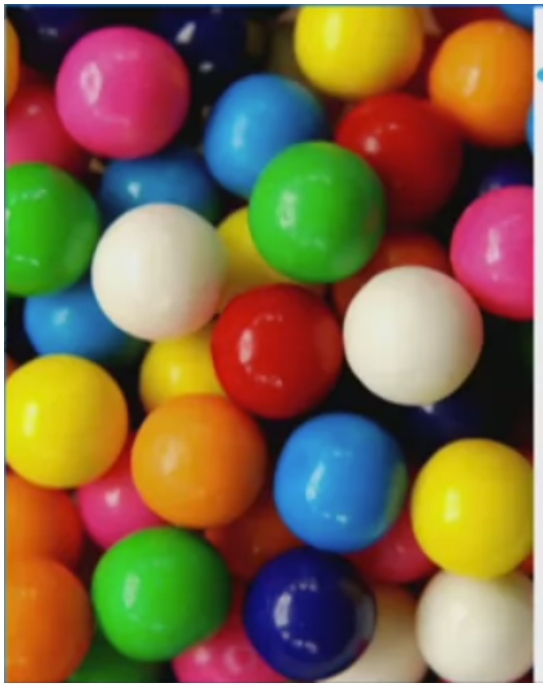
School of Computing & IT

- Question 3: 144 balls in eight different colors all put into the same bucket. There are 18 balls of each color. What is the overall entropy?

1. The entropy is 3.000
2. The entropy is 0.333
3. The entropy is 1.500
4. The entropy is 4.500

Decision Trees Classifier

- Answer 3: Entropy after mixing the 9 cells: E=3



- **144 balls having 8 different colors: 18:18:18:18:18:18:18:18**

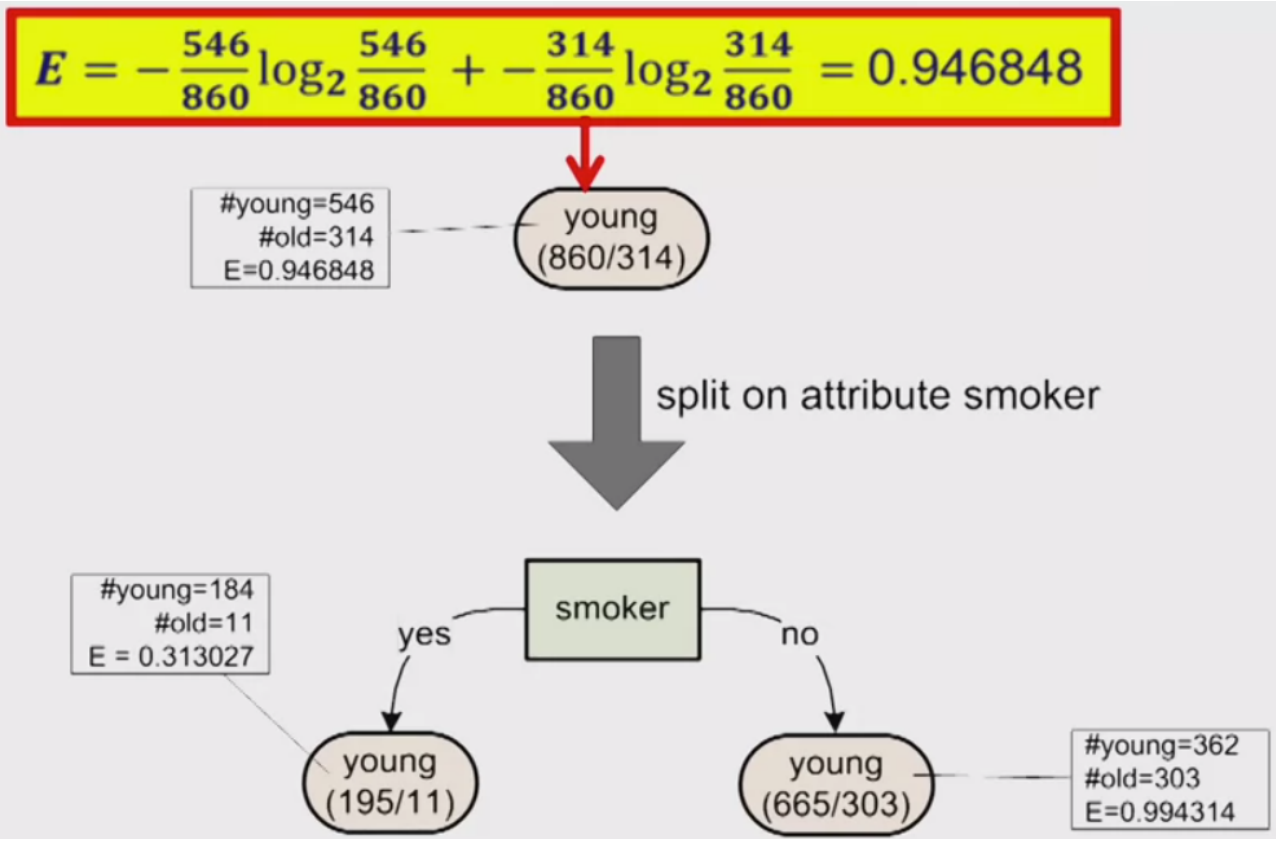
$$E = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

$$= -\sum_{i=1}^{8} \frac{18}{144} \log_2(\frac{18}{144})$$

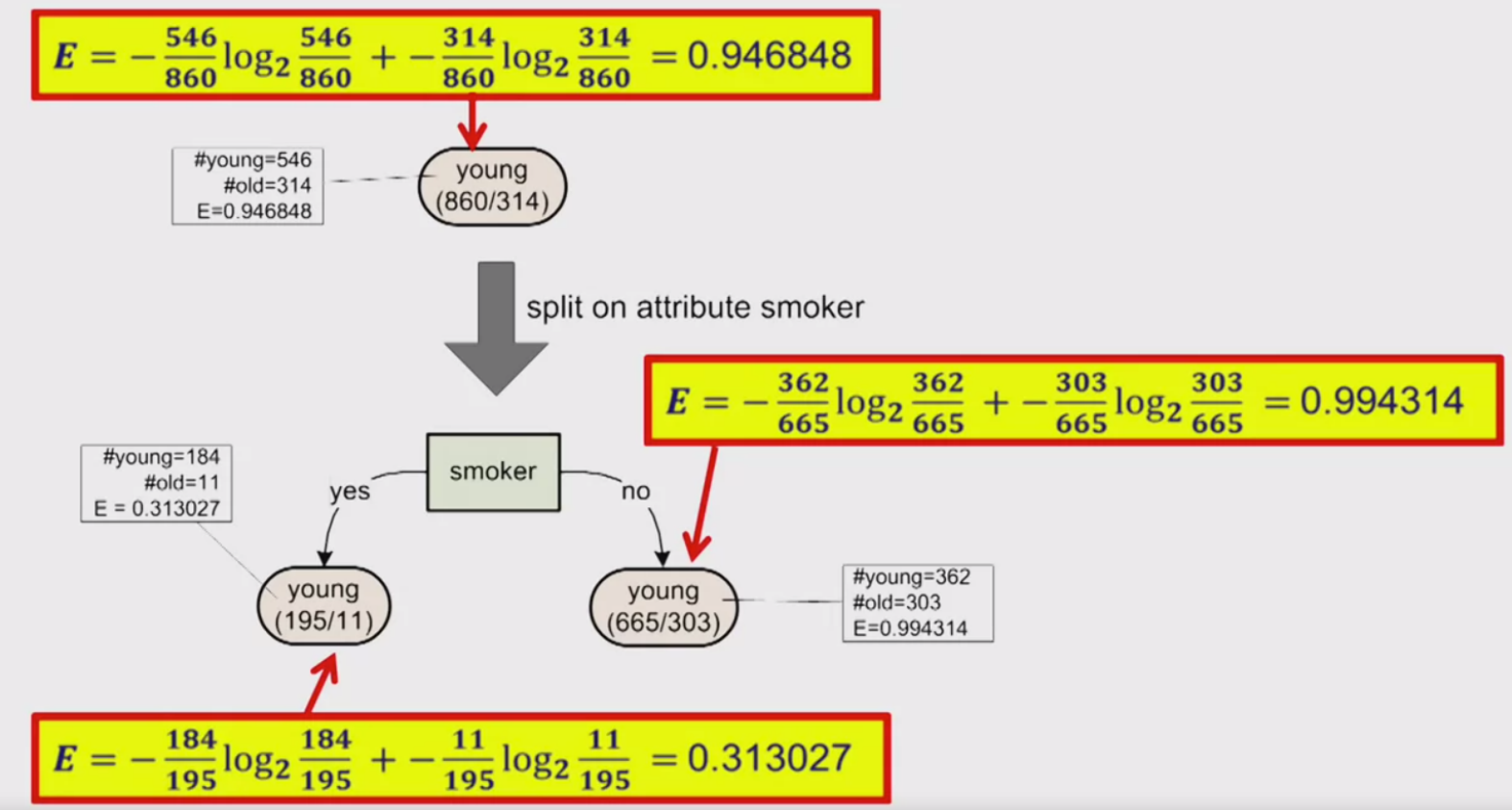
$$= -8 \times \frac{1}{8} \times -3 = 3$$

Decision Trees Classifier

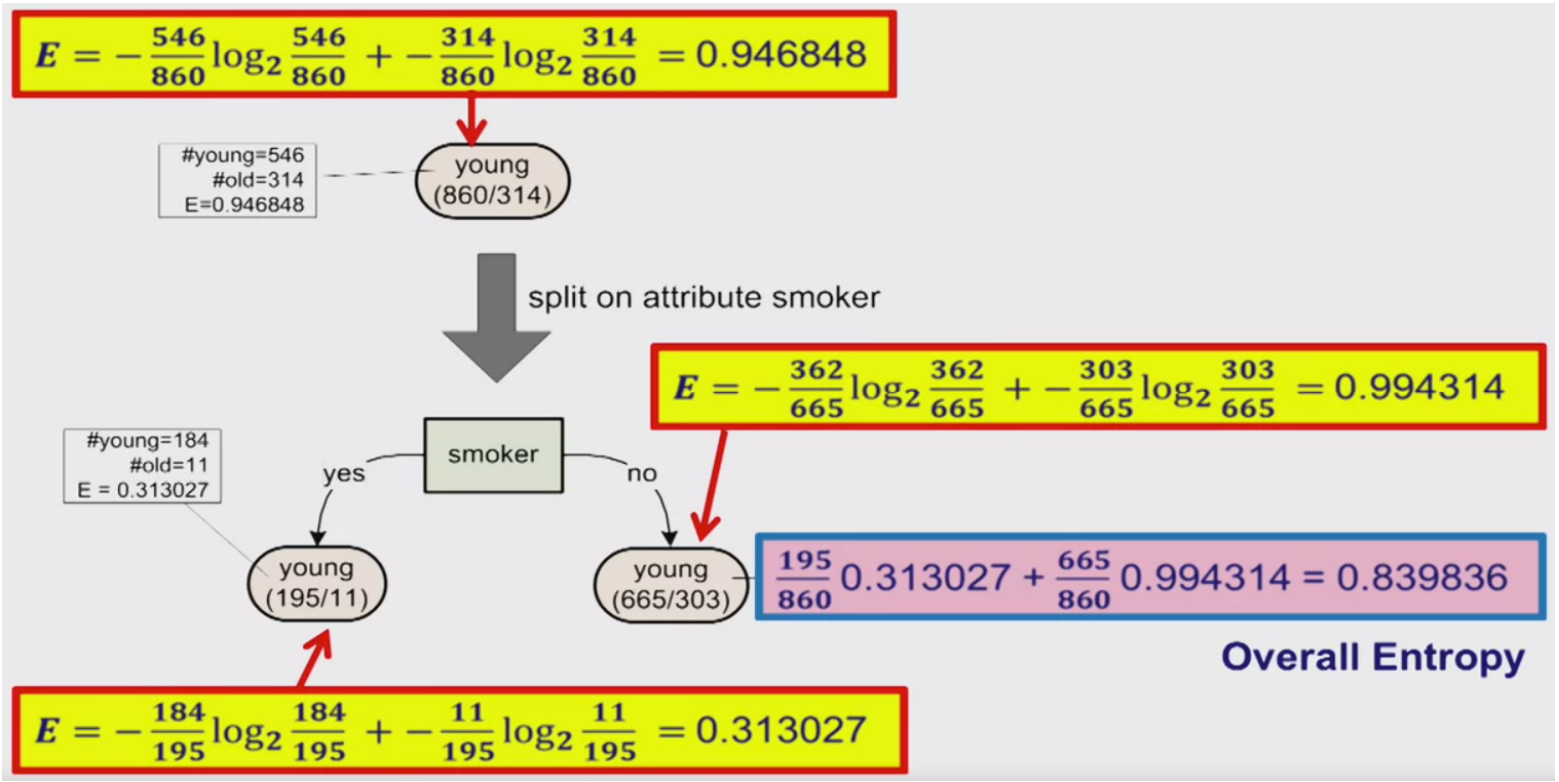
Decision Trees Classifier



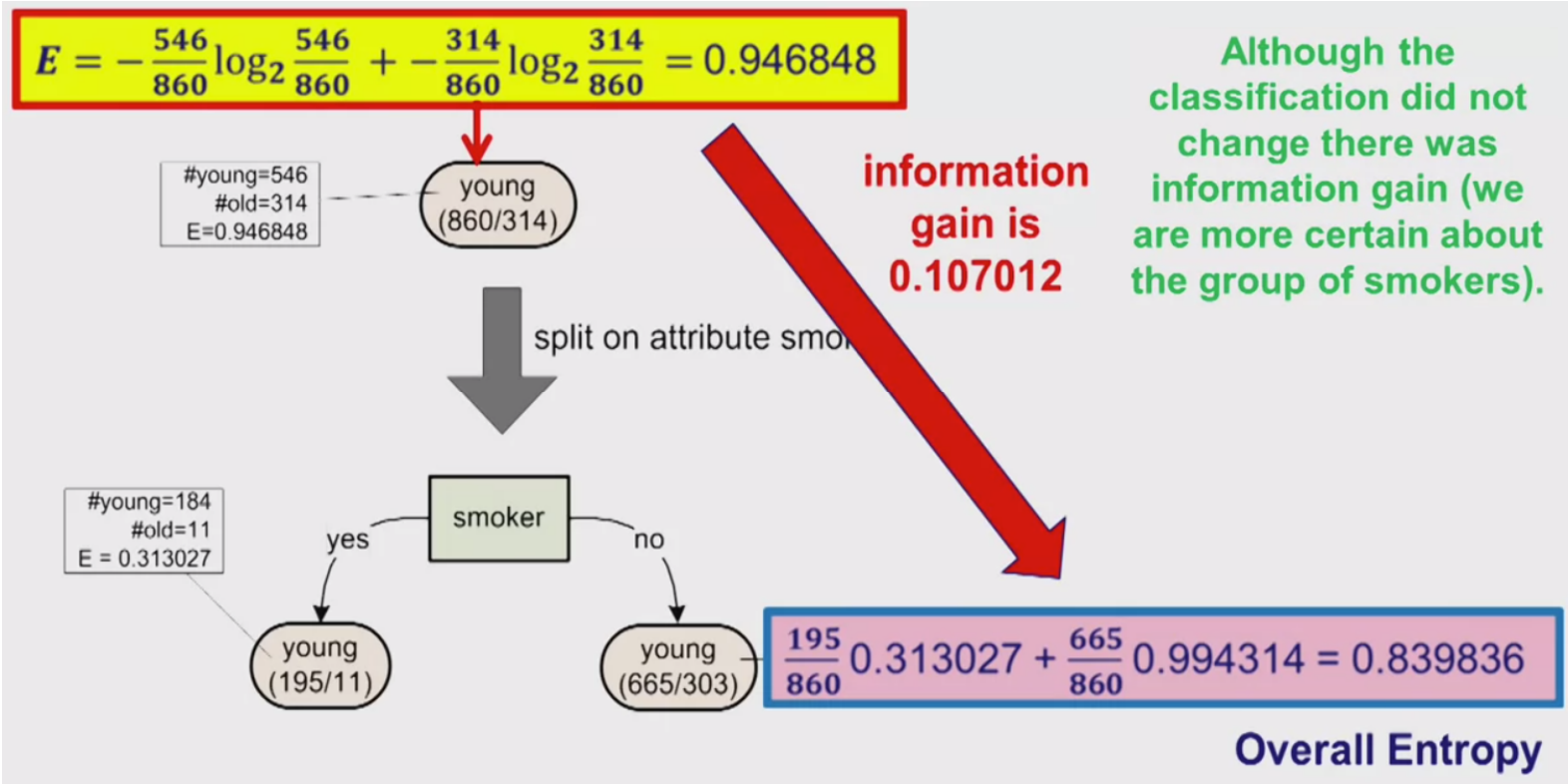
Decision Trees Classifier



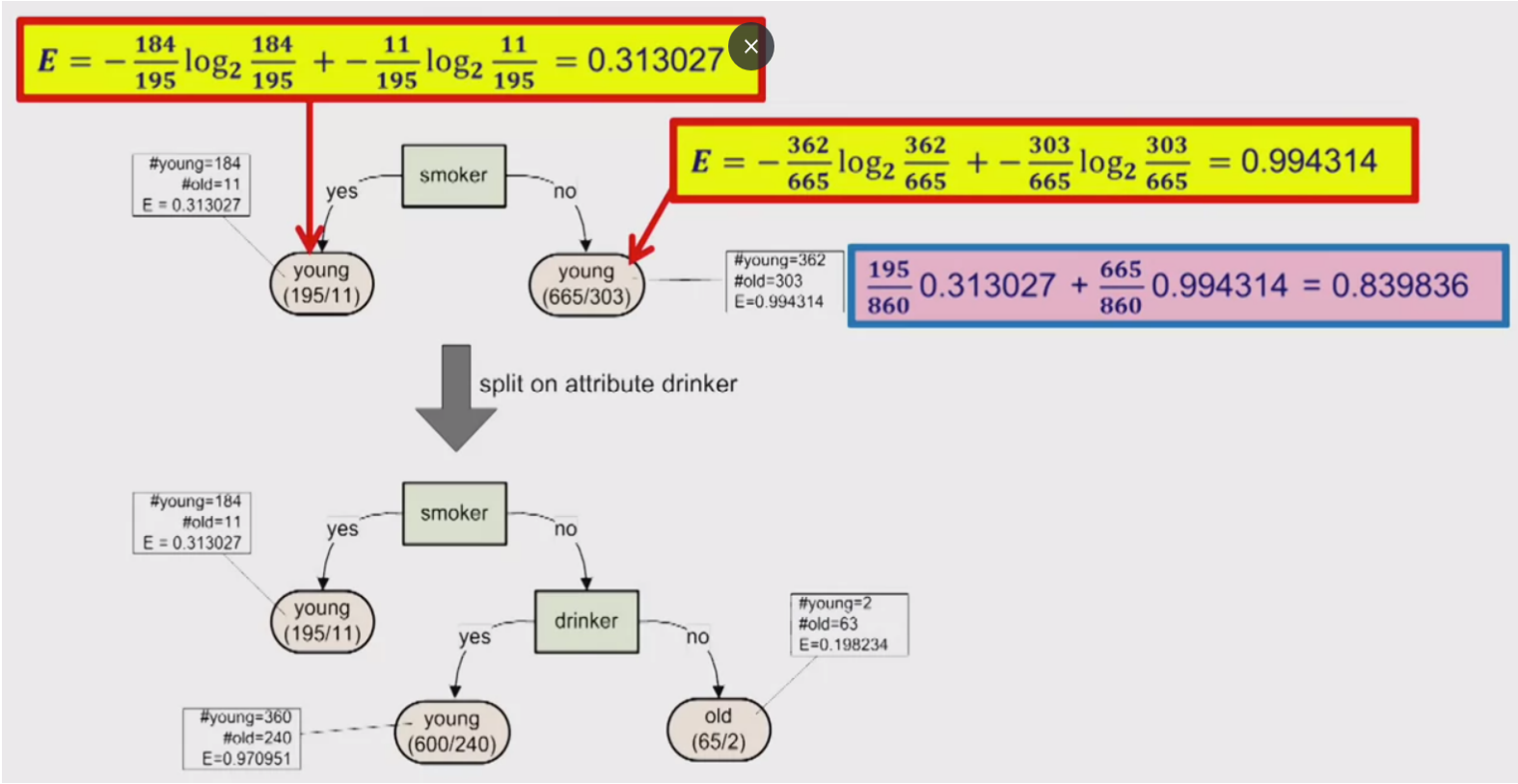
Decision Trees Classifier



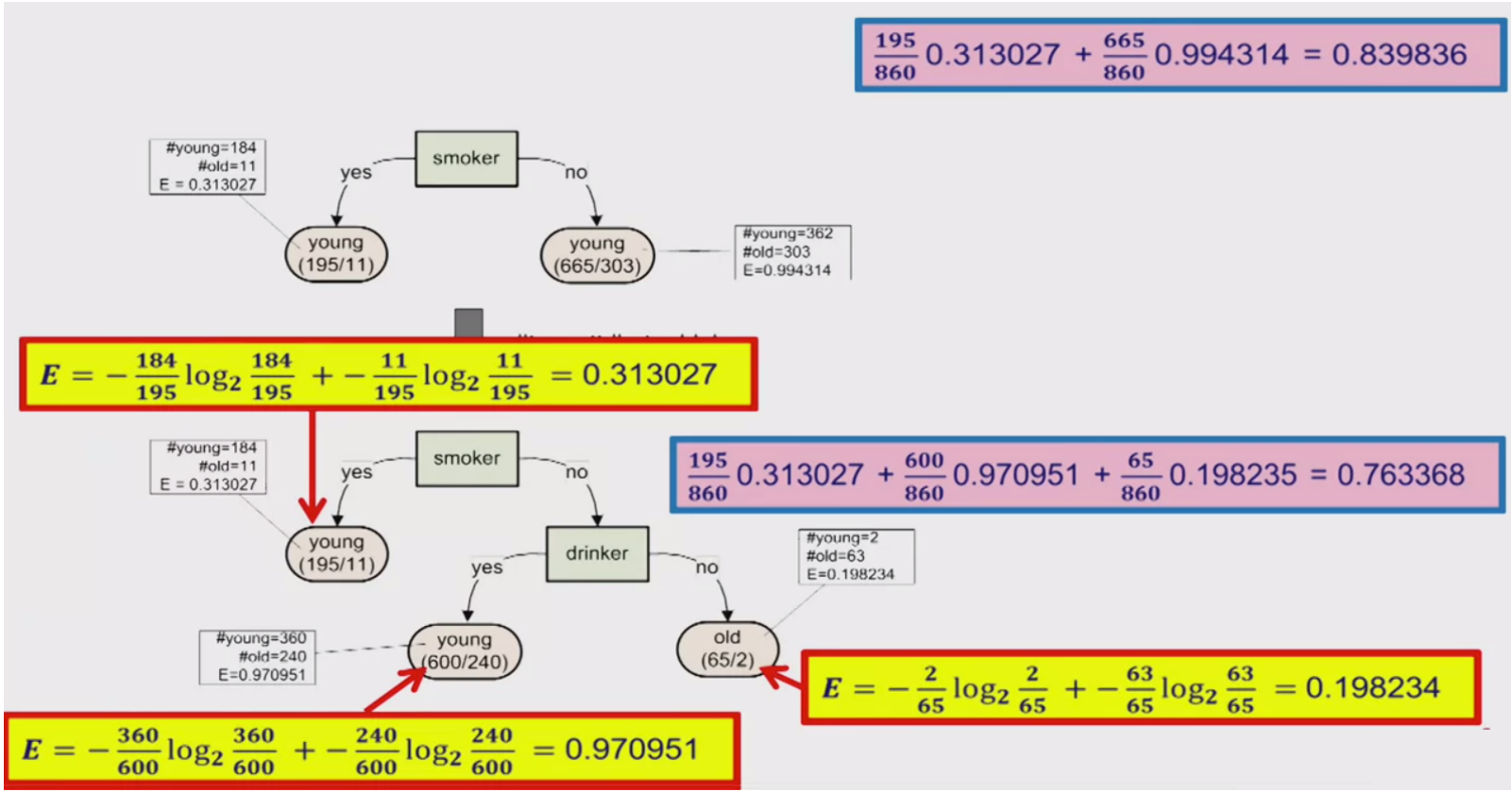
Decision Trees Classifier



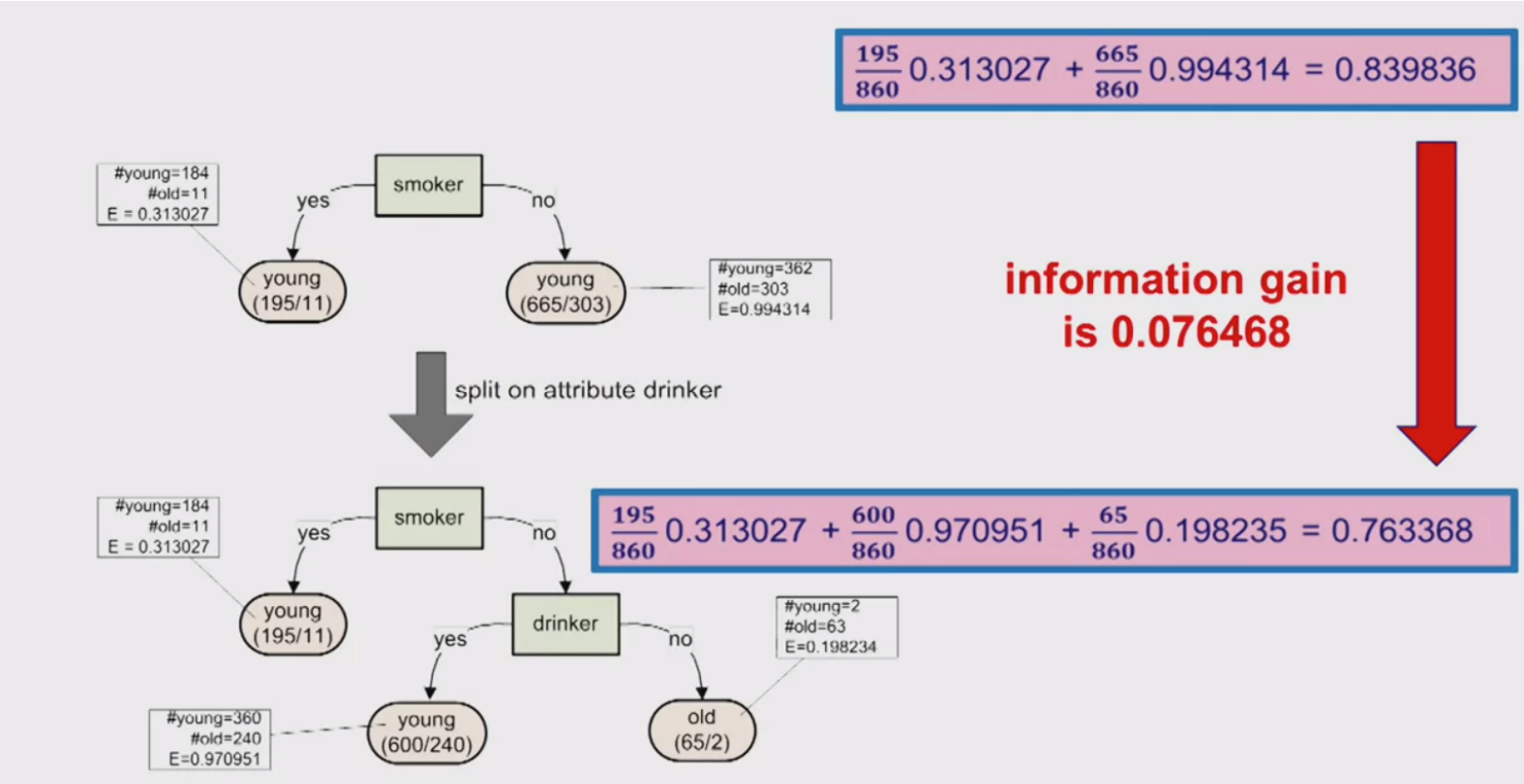
Decision Trees Classifier



Decision Trees Classifier



Decision Trees Classifier



Decision Tree Algorithm (sketch)

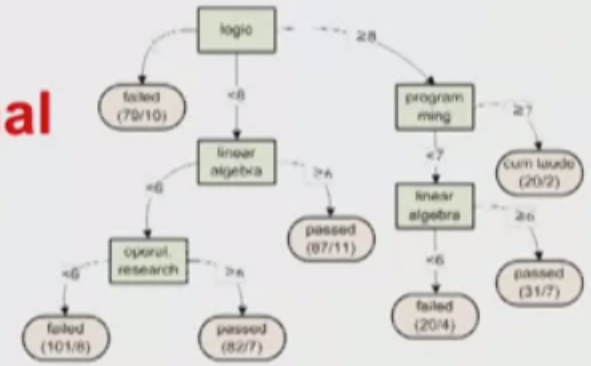
School of Computing & IT

- Start with root node corresponding to all instances.
- Iteratively traverse all nodes to see whether "information gain" (i.e., reduction of uncertainty) is possible.
- For each node and for every attribute, check what the effect of splitting the node is in terms of information gain.
- Select the attribute with the biggest information gain above a given threshold.
- Continue until no significant improvement is possible.
- Return the decision tree.

Decision Trees Classifier

- Many parameters/variations are possible

- **The minimal size of a node before or after splitting.**
- **Threshold setting the minimal gain (no split if information gain is too small).**
- **Maximal depth of the tree.**

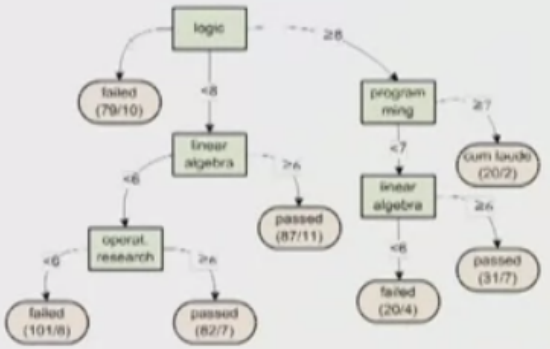


Decision Trees Classifier

- Many parameters/variations are possible

- **Alternatives to entropy, e.g., Gini index of diversity.**
- **Splitting the domain of a numerical variable.**
- **Post pruning: removing leaf nodes that do not significantly increase the discriminative power.**

$$G = \sum_{i=1}^{k} p_i(1-p_i) = 1 - \sum_{i=1}^{k} (p_i)^2 \text{ with } p_i = \frac{c_i}{n}$$

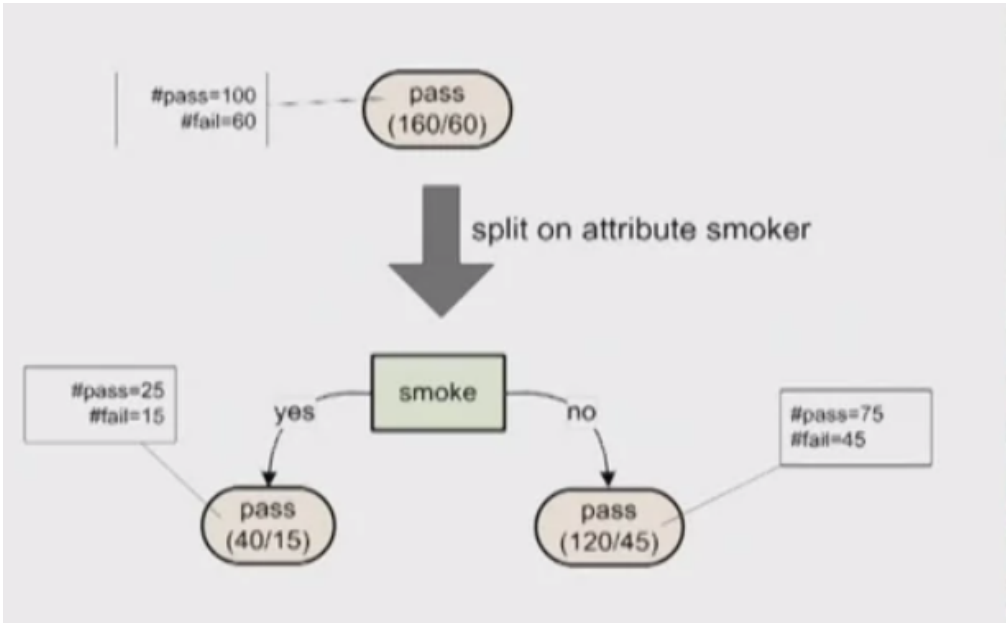


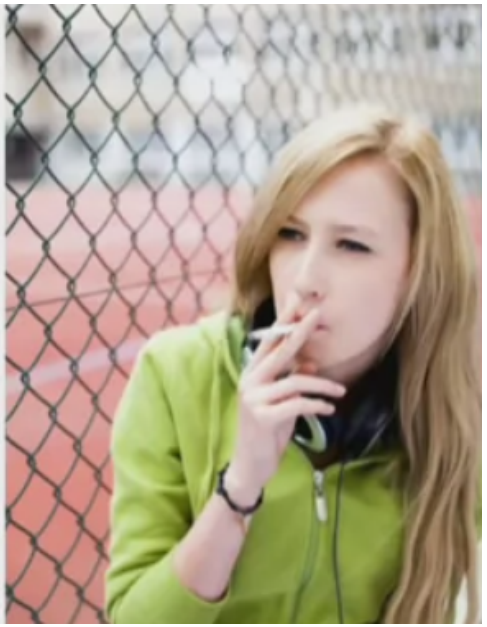
Decision Tree Learning

- Example: 160 students (100 pass, 60 fail)

Decision Tree Learning

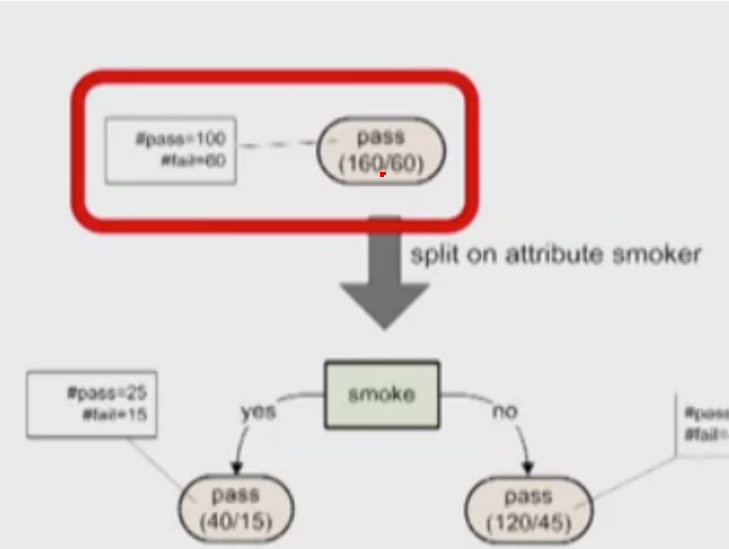
- Question 1: What is the information gain?





Decision Tree Learning

- Answer: Entropy of root node



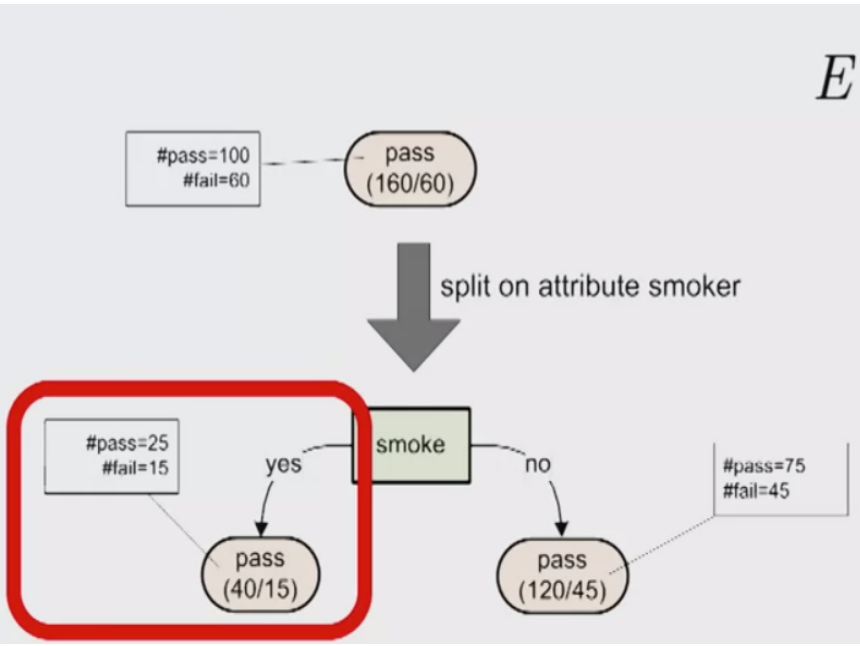
$$E = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

$$= -(\frac{100}{160}\log_2(\frac{100}{160}) + \frac{60}{160}\log_2(\frac{60}{160}))$$

$$= 0.9544$$

Decision Tree Learning

- Answer: Entropy of smokers



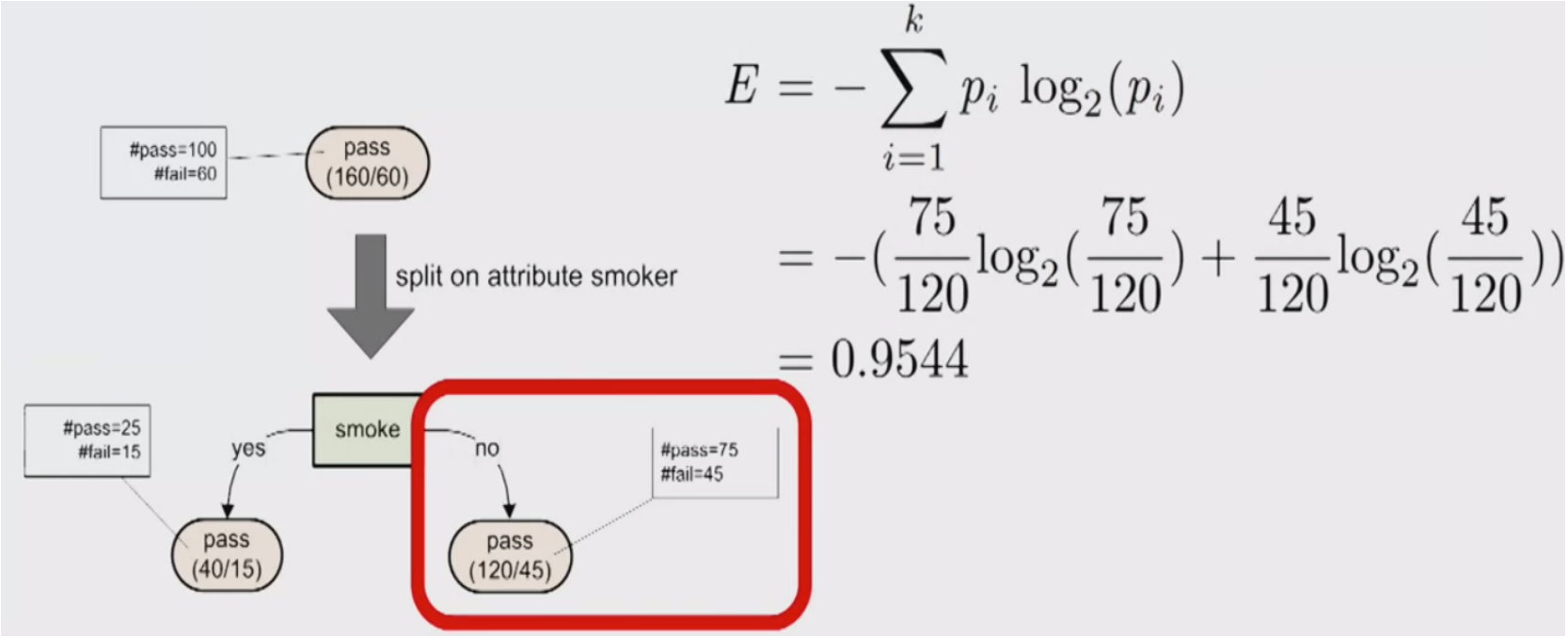
$$E = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

$$= -(\frac{25}{40}\log_2(\frac{25}{40}) + \frac{15}{40}\log_2(\frac{15}{40}))$$

$$= 0.9544$$

Decision Tree Learning

- Answer: Entropy of non-smokers



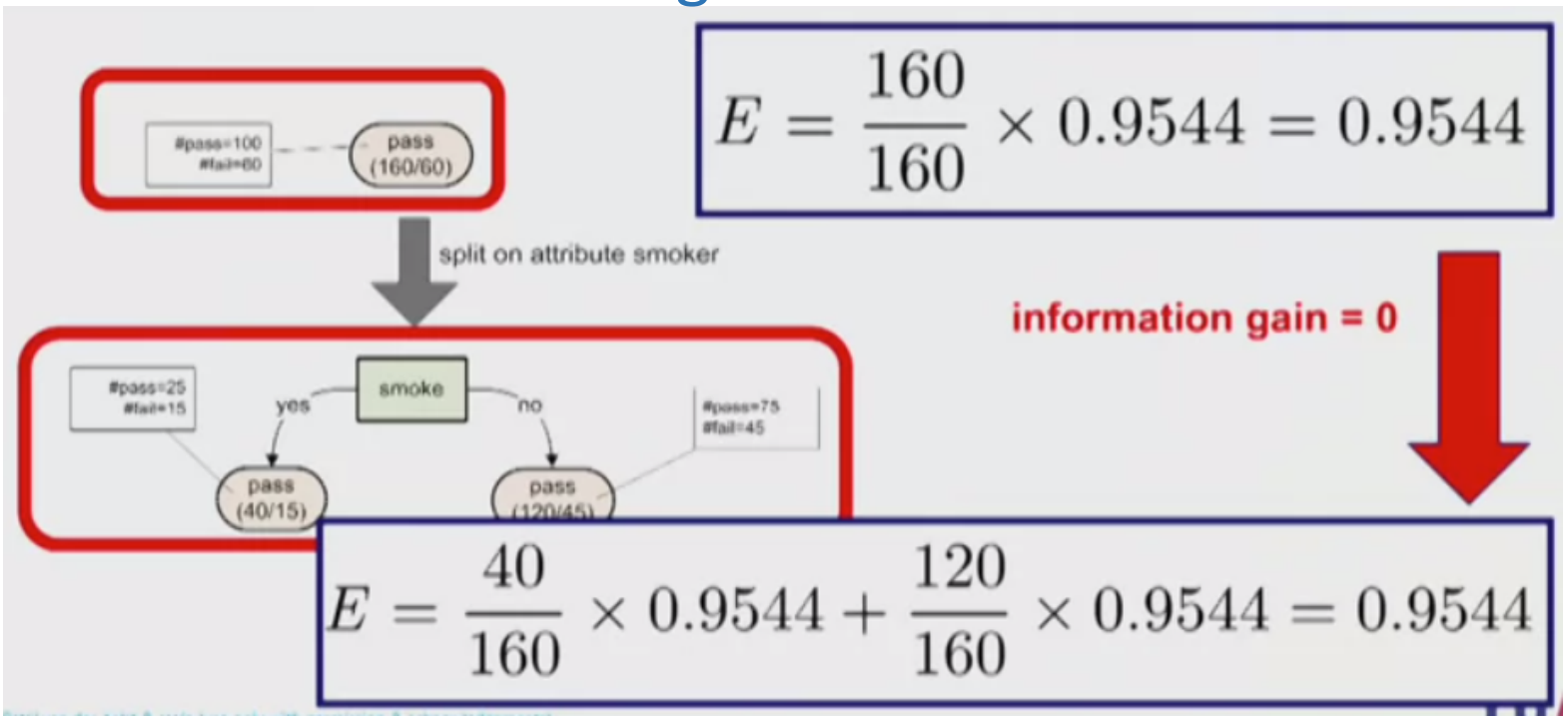
$$E = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

$$= -(\frac{75}{120}\log_2(\frac{75}{120}) + \frac{45}{120}\log_2(\frac{45}{120}))$$

$$= 0.9544$$

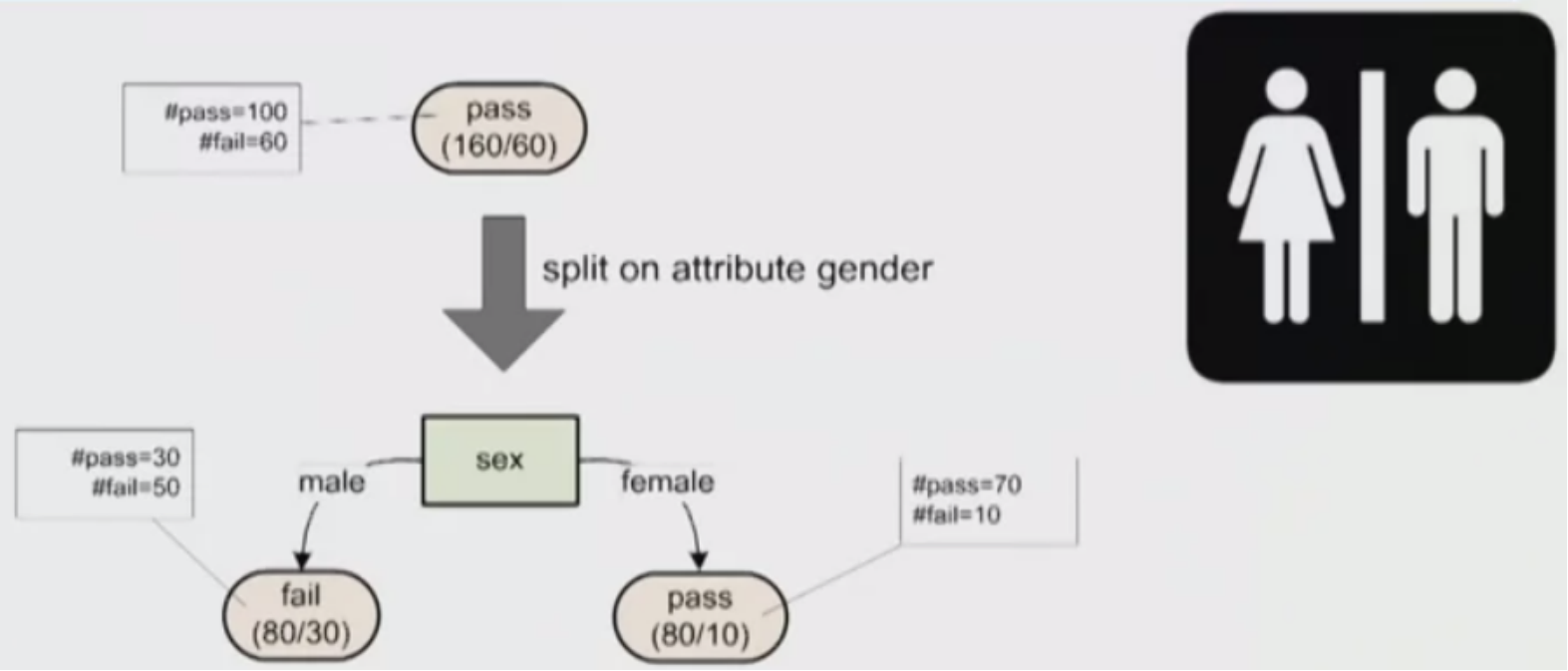
Decision Tree Learning

- Answer: No information gain



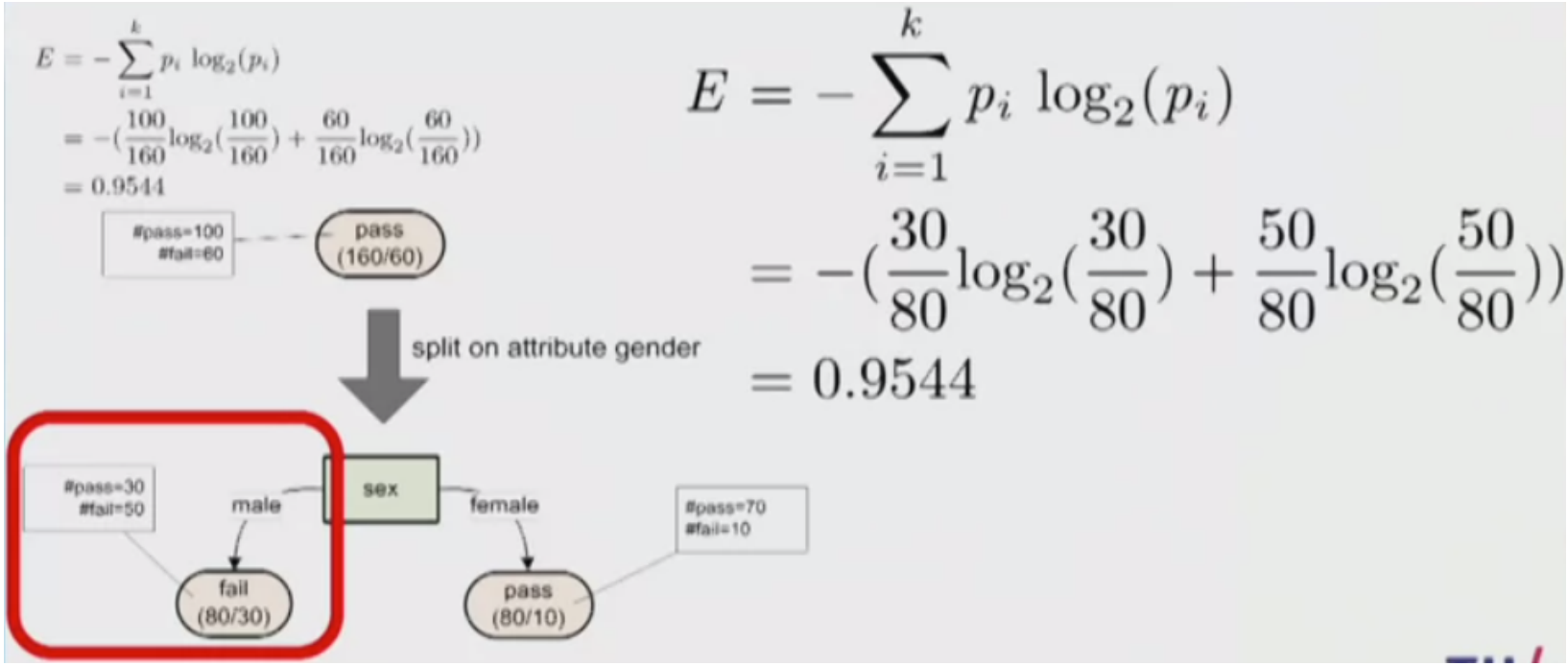
Decision Tree Learning

- Question 2: What is the information gain?



Decision Tree Learning

- Answer: Entropy for male students



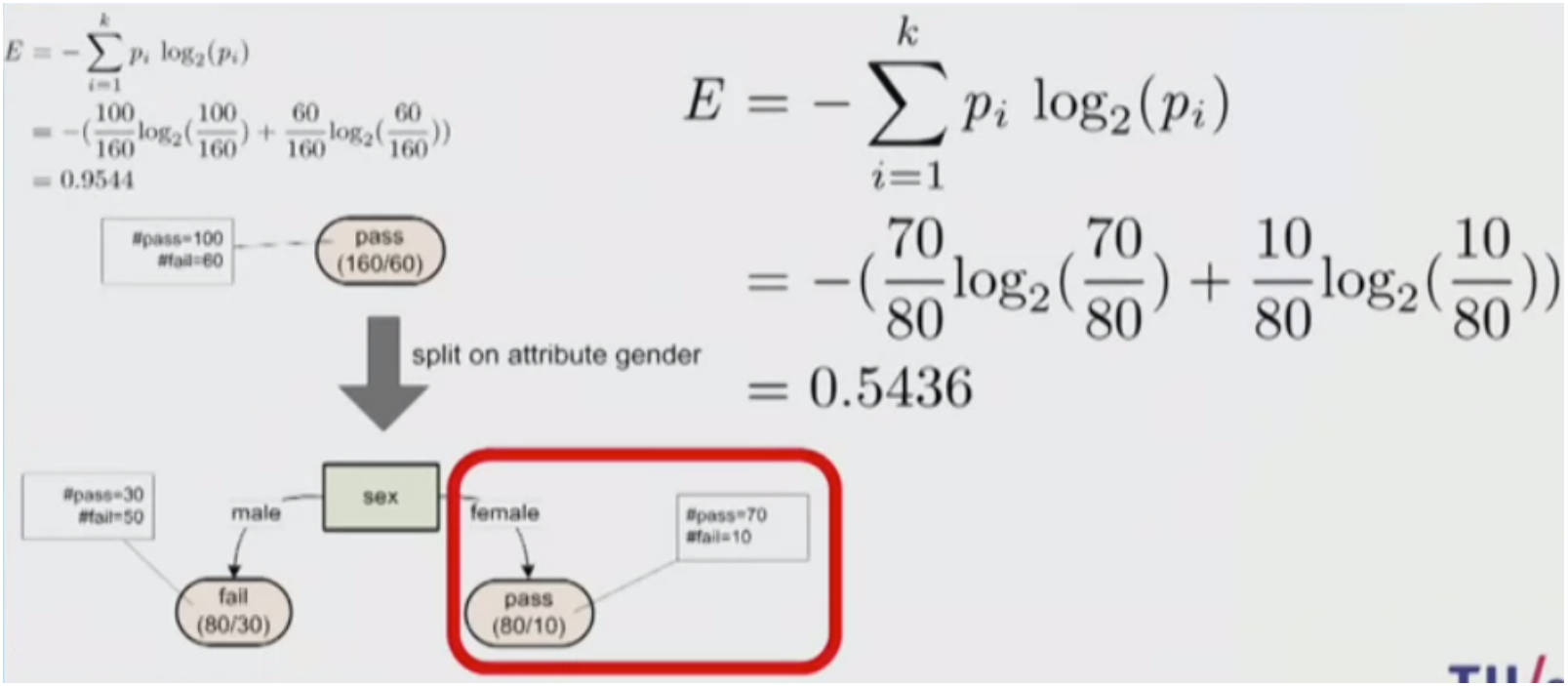
Decision Tree Learning

- Answer: Entropy for female students

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

$$= -(\frac{100}{160}\log_2(\frac{100}{160}) + \frac{60}{160}\log_2(\frac{60}{160}))$$

$$= 0.9544$$



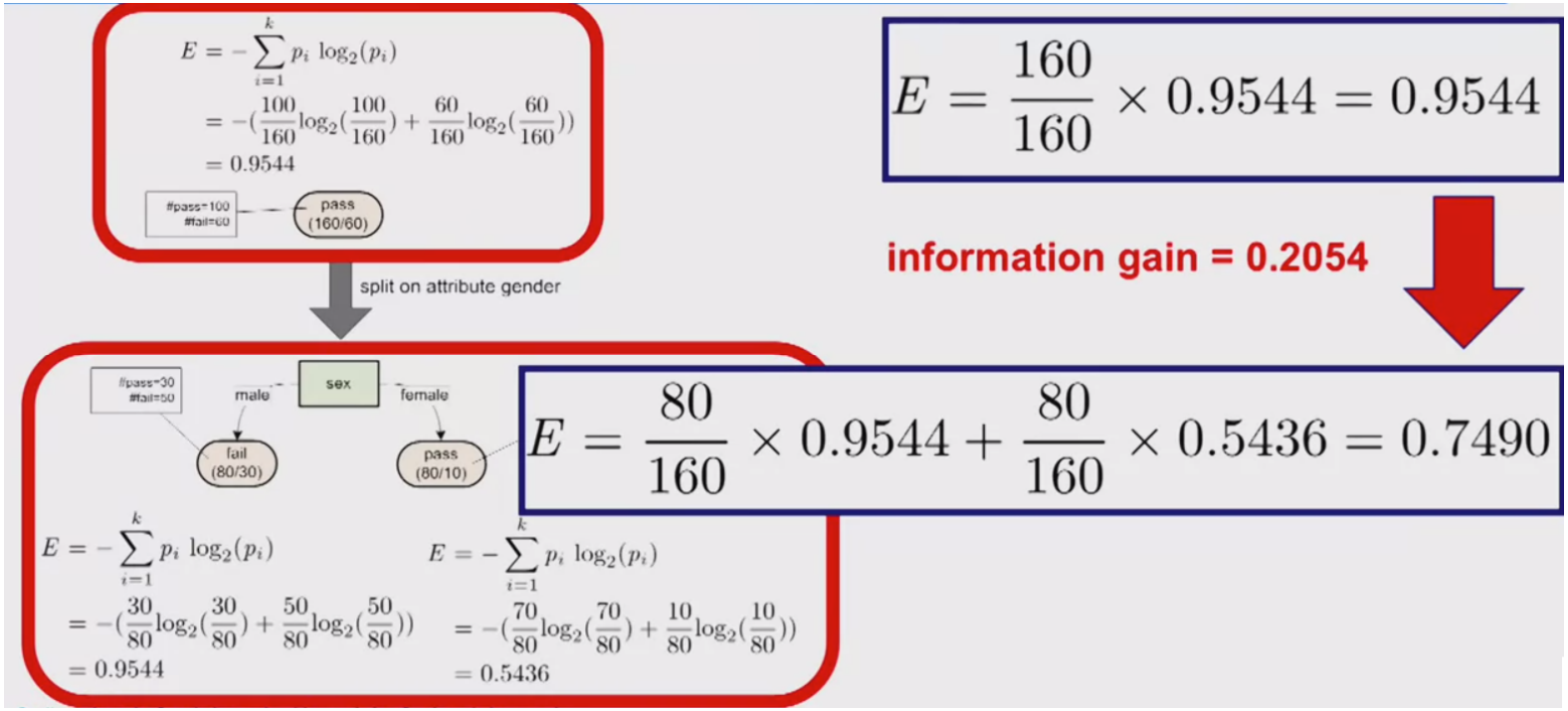
$$E = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

$$= -(\frac{70}{80}\log_2(\frac{70}{80}) + \frac{10}{80}\log_2(\frac{10}{80}))$$

$$= 0.5436$$

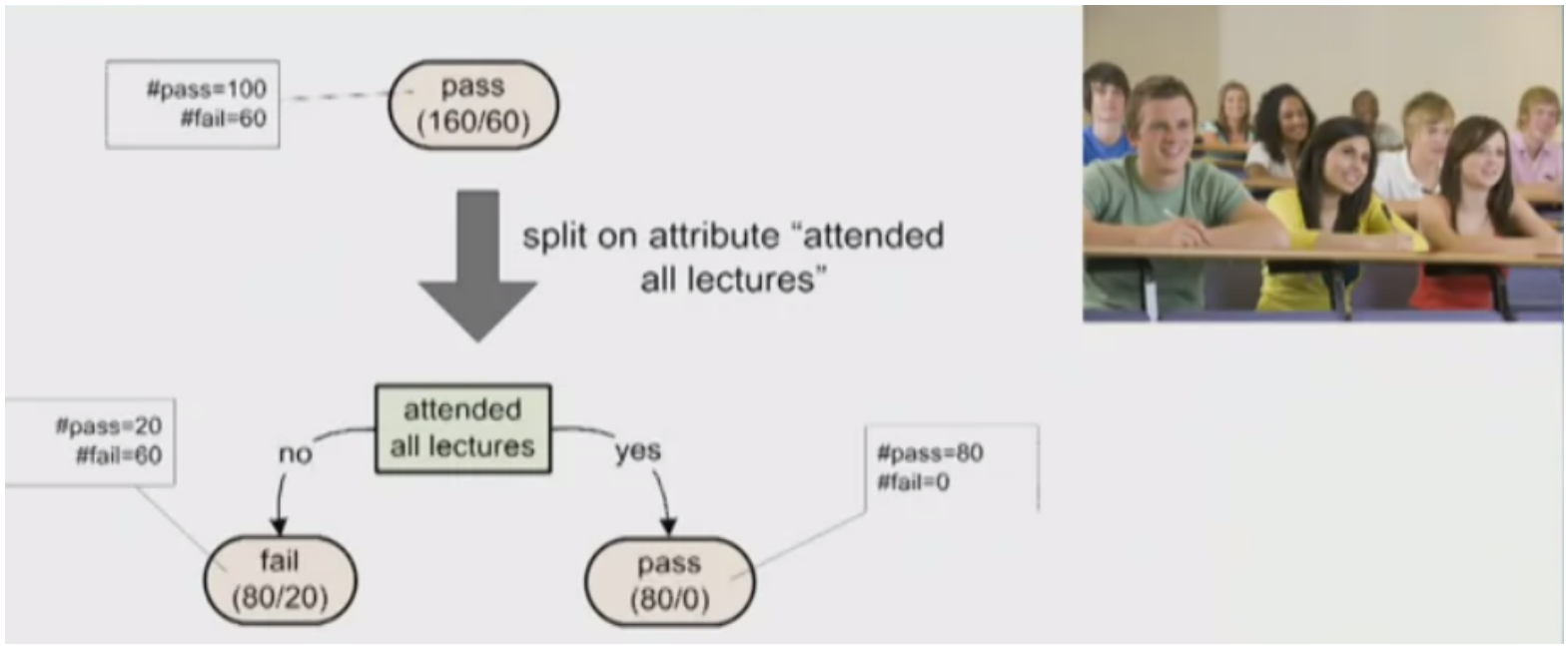
Decision Tree Learning

- Answer: Information gain



Decision Tree Learning

- Question 3: What is the information gain?



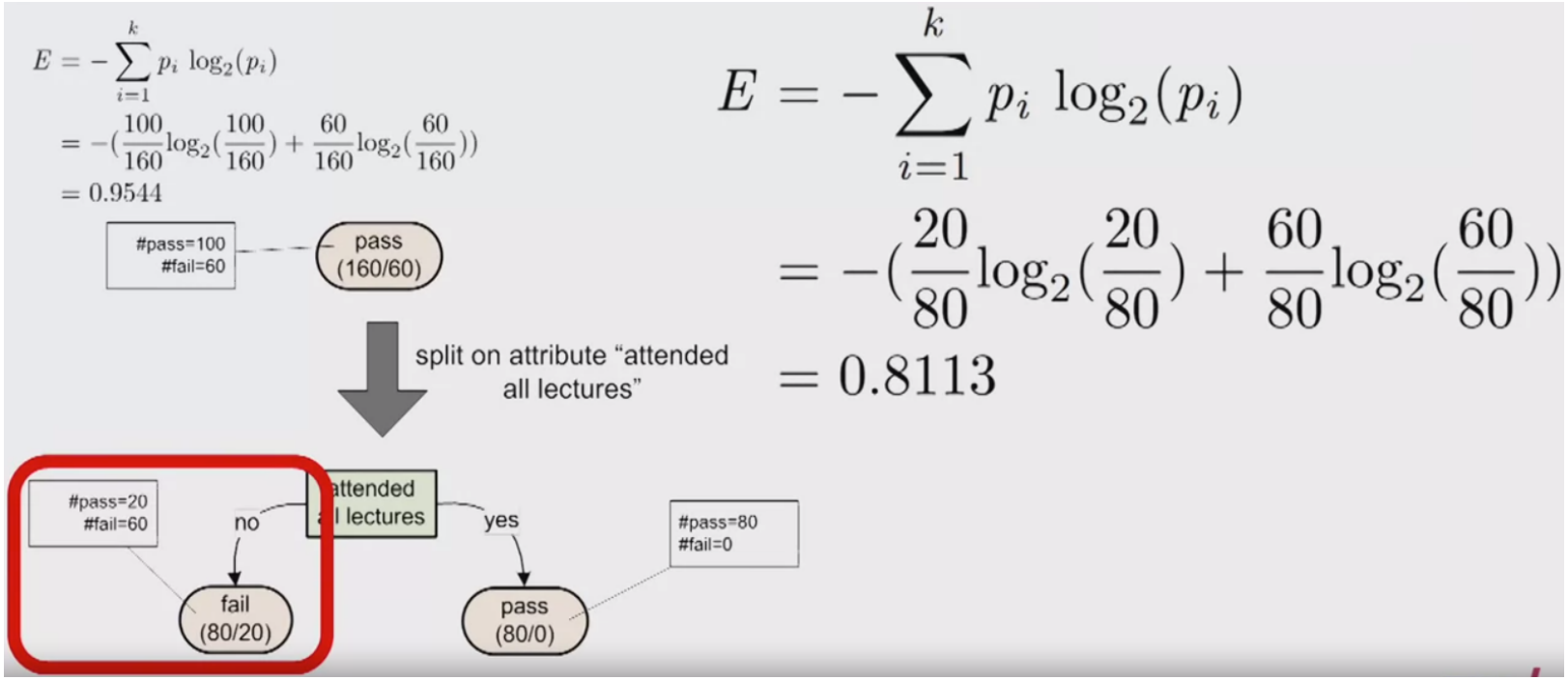
Decision Tree Learning

- Answer: Entropy of attending students

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

$$= -(\frac{100}{160}\log_2(\frac{100}{160}) + \frac{60}{160}\log_2(\frac{60}{160}))$$

$$= 0.9544$$



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

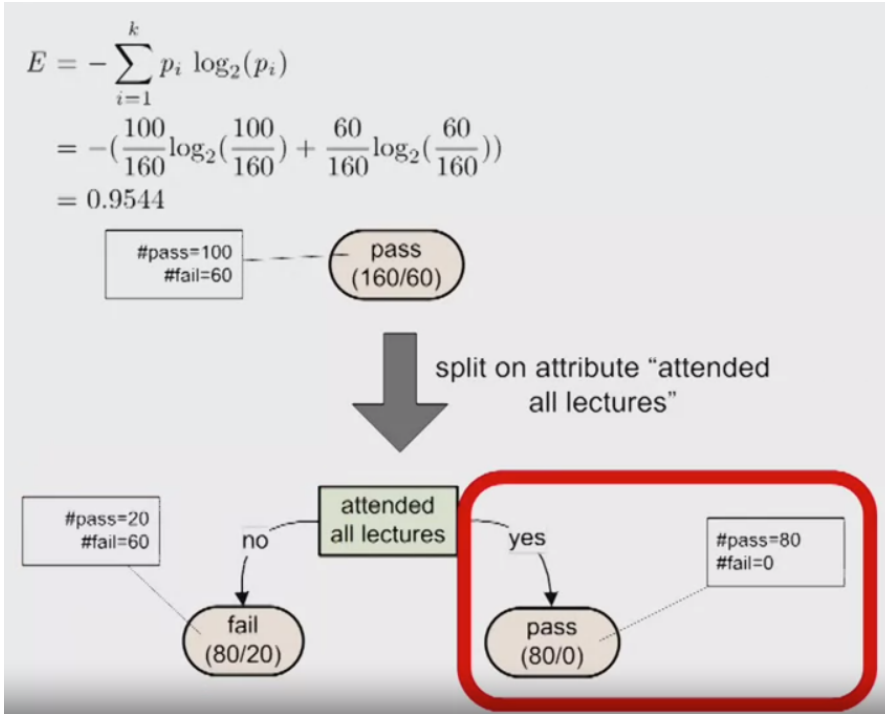
$$= -(\frac{20}{80}\log_2(\frac{20}{80}) + \frac{60}{80}\log_2(\frac{60}{80}))$$

$$= 0.8113$$

Decision Tree Learning

- Answer: Entropy of missing students

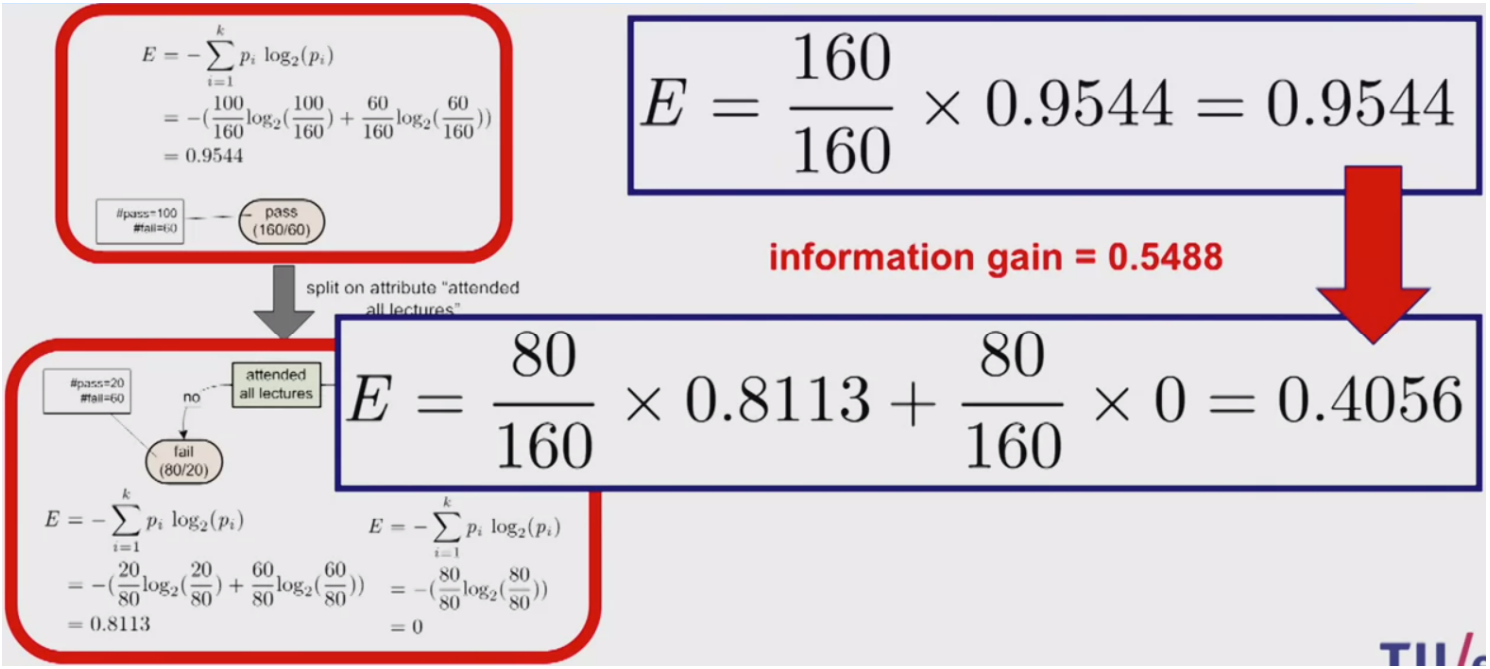
$$E = -\sum_{i=1}^{k} p_i \log_2(p_i)$$
$$= -(\frac{100}{160}\log_2(\frac{100}{160}) + \frac{60}{160}\log_2(\frac{60}{160}))$$
$$= 0.9544$$



$$E = -\sum_{i=1}^{k} p_i \log_2(p_i)$$
$$= -(\frac{80}{80}\log_2(\frac{80}{80}))$$
$$= 0$$

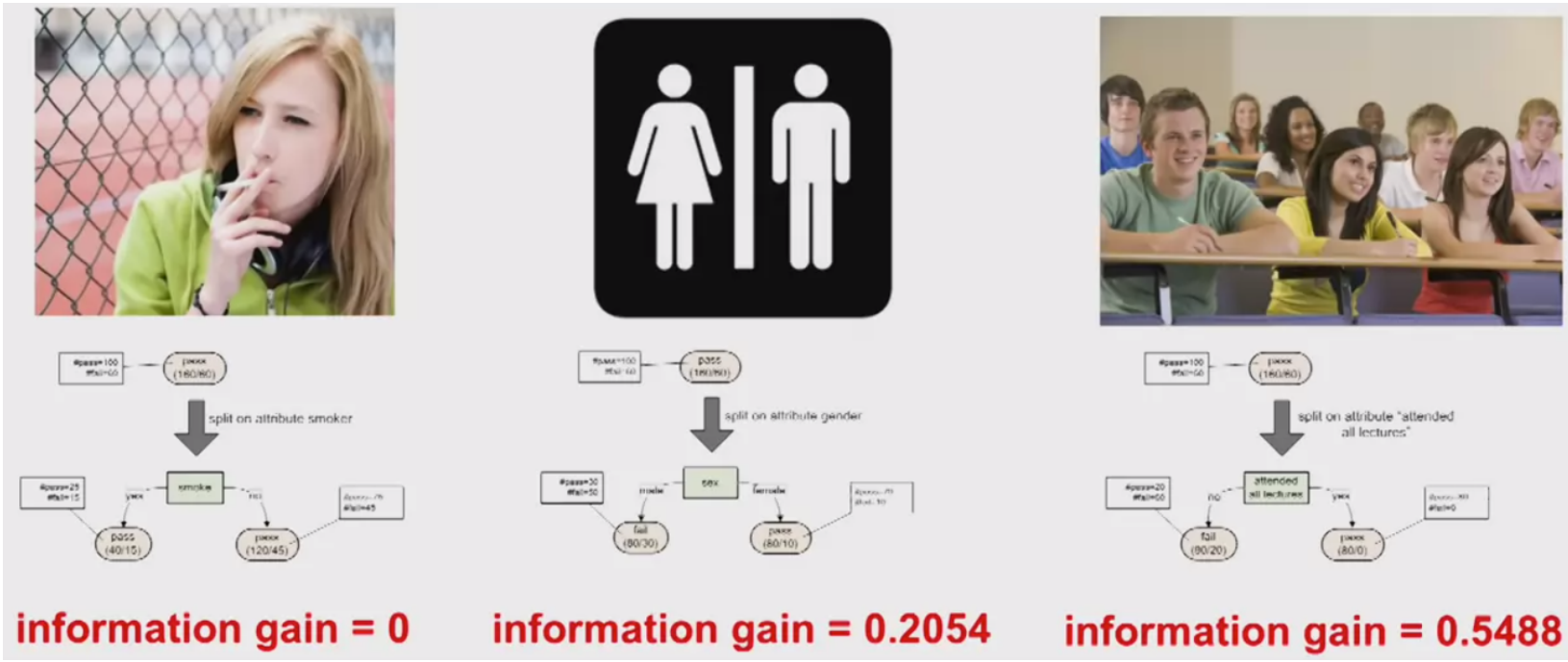
Decision Tree Learning

- Answer: Information gain



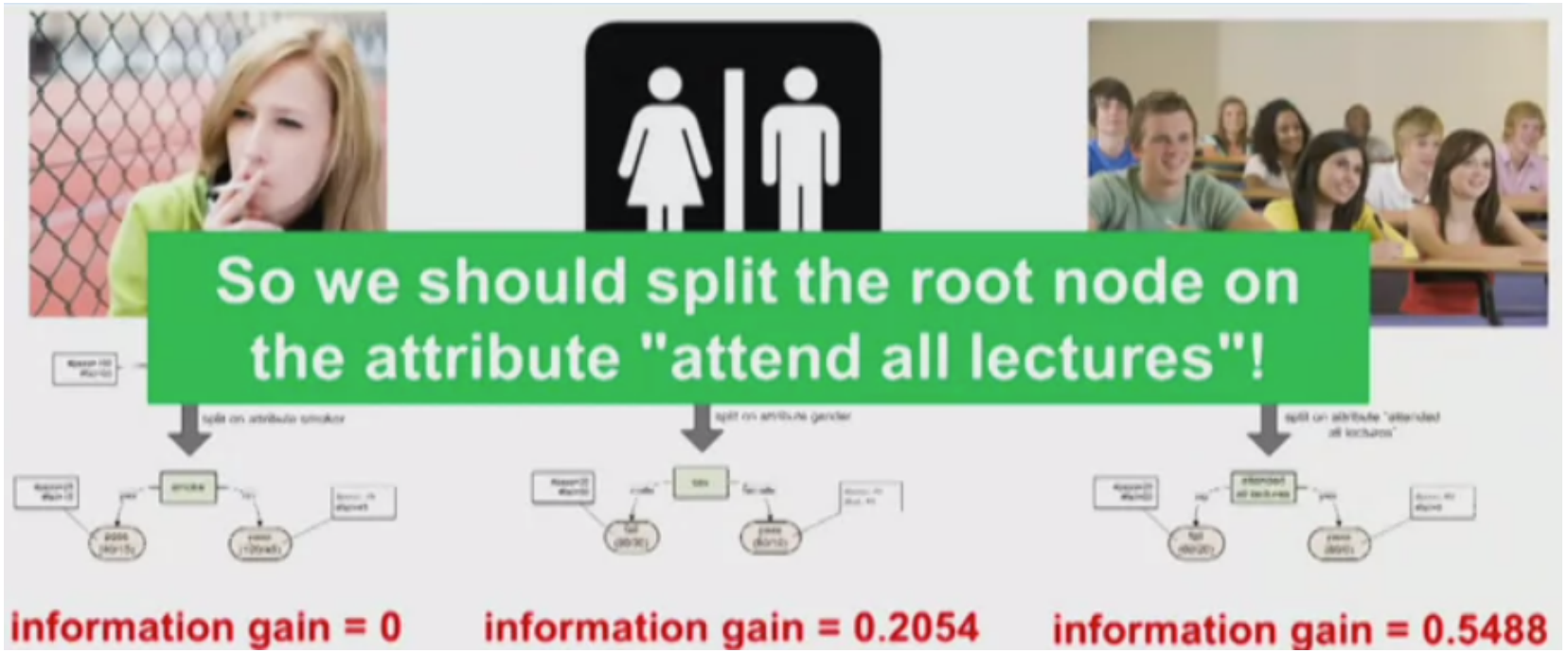
Decision Tree Learning

- Comparing information gain



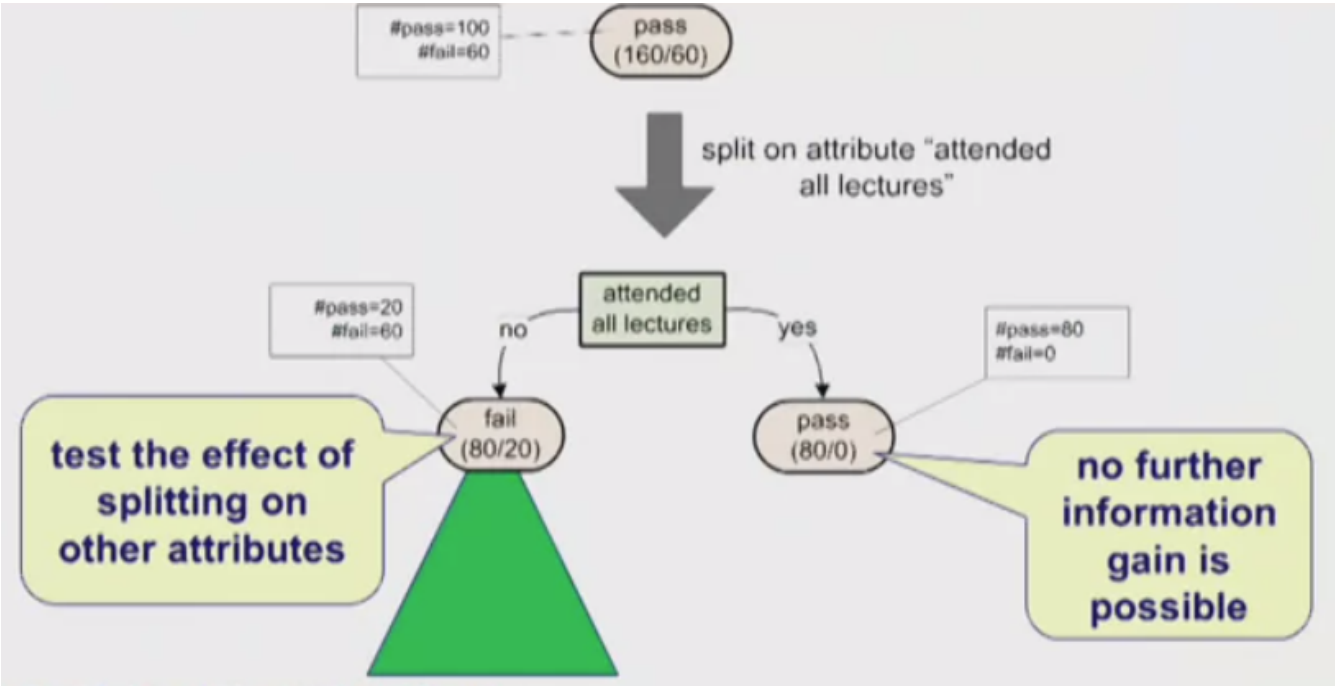
Decision Tree Learning

- Comparing information gain



Decision Tree Learning

- Iterate until no significant information gain



Entropy of a Joint Distribution

School of Computing & IT

- Example: $X = \{\text{Raining, Not raining}\}$, $Y = \{\text{Cloudy, Not cloudy}\}$

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

$$
\begin{aligned}
H(X,Y) &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(x,y) \\
&= -\frac{24}{100}\log_2\frac{24}{100} - \frac{1}{100}\log_2\frac{1}{100} - \frac{25}{100}\log_2\frac{25}{100} - \frac{50}{100}\log_2\frac{50}{100} \\
&\approx 1.56\text{bits}
\end{aligned}
$$

Conditional Entropy

- Example: $X = \{\text{Raining, Not raining}\}$, $Y = \{\text{Cloudy, Not cloudy}\}$

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- What is the entropy of cloudiness Y, **given that it is raining**?

$$\begin{aligned} H(Y|X=x) &= -\sum_{y\in Y} p(y|x)\log_2 p(y|x) \\ &= -\frac{24}{25}\log_2\frac{24}{25} - \frac{1}{25}\log_2\frac{1}{25} \\ &\approx 0.24\text{bits} \end{aligned}$$

- We used: $p(y|x) = \frac{p(x,y)}{p(x)}$, and $p(x) = \sum_y p(x,y)$ (sum in a row)

Conditional Entropy

School of Computing & IT

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- The expected conditional entropy:

$$
\begin{aligned}
H(Y|X) &= \mathbb{E}_x[H[Y|x]] \\
&= \sum_{x \in X} p(x) H(Y|X = x) \\
&= -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(y|x)
\end{aligned}
$$

Conditional Entropy

School of Computing & IT

- Example: $X = \{\text{Raining, Not raining}\}$, $Y = \{\text{Cloudy, Not cloudy}\}$

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- What is the entropy of cloudiness, given the knowledge of whether or not it is raining?

$$\begin{aligned} H(Y|X) &= \sum_{x \in X} p(x) H(Y|X = x) \\ &= \frac{1}{4} H(\text{cloudy}|\text{is raining}) + \frac{3}{4} H(\text{cloudy}|\text{not raining}) \\ &\approx 0.75 \text{ bits} \end{aligned}$$

Conditional Entropy

School of Computing & IT

- Some useful properties:
  - H is always non-negative
  - Chain rule: $H(X,Y) = H(X|Y) + H(Y) = H(Y|X) + H(X)$
  - If X and Y independent, then X does not affect our uncertainty about Y: $H(Y|X) = H(Y)$
  - But knowing Y makes our knowledge of Y certain: $H(Y|Y) = 0$
  - By knowing X, we can only decrease uncertainty about Y: $H(Y|X) \leq H(Y)$

Information Gain

School of Computing & IT

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- How much more certain am I about whether it's cloudy if I'm told whether it is raining? My uncertainty in Y minus my expected uncertainty that would remain in Y after seeing X.
- This is called the information gain $IG(Y|X)$ in Y due to X, or the mutual information of Y and X

$$IG(Y|X) = H(Y) - H(Y|X) \tag{1}$$

- If X is completely uninformative about Y: $IG(Y|X) = 0$
- If X is completely informative about Y: $IG(Y|X) = H(Y)$

Revisiting Our Original Example

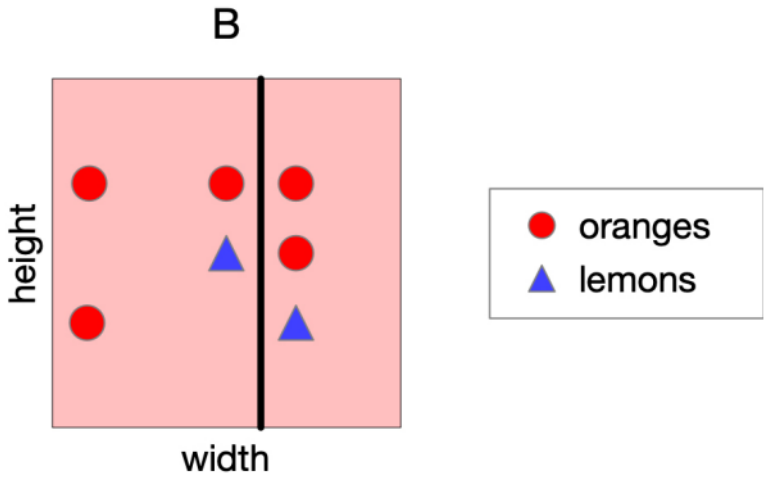
School of Computing & IT

- Information gain measures the informativeness of a variable, which is exactly what we desire in a decision tree split!
- The information gain of a split: how much information (over the training set) about the class label Y is gained by knowing which side of a split you're on.

Information Gain of Split B

School of Computing & IT

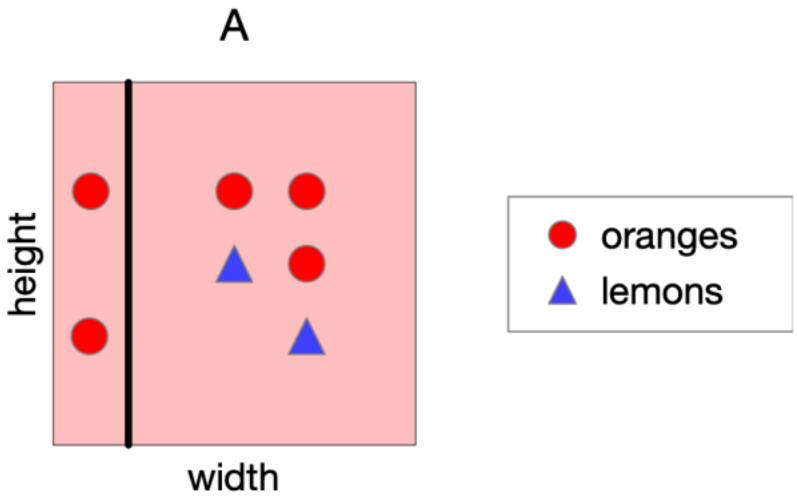
- What is the information gain of split B? Not terribly informative...



- Entropy of class outcome before split:
  $H(Y) = -\frac{2}{7}\log_2(\frac{2}{7}) - \frac{5}{7}\log_2(\frac{5}{7}) \approx 0.86$
- Conditional entropy of class outcome after split:
  $H(Y|left) \approx 0.81$, $H(Y|right) \approx 0.92$
- $IG(split) \approx 0.86 - (\frac{4}{7} \cdot 0.81 + \frac{3}{7} \cdot 0.92) \approx 0.006$

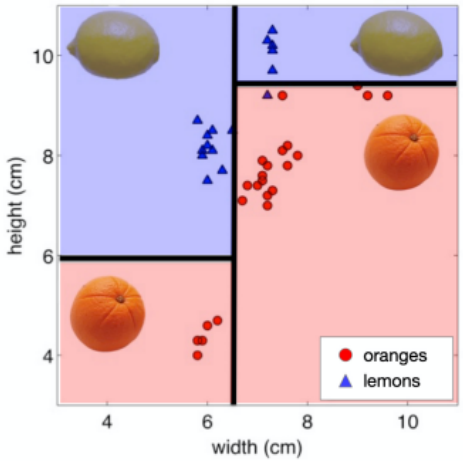
Information Gain of Split A

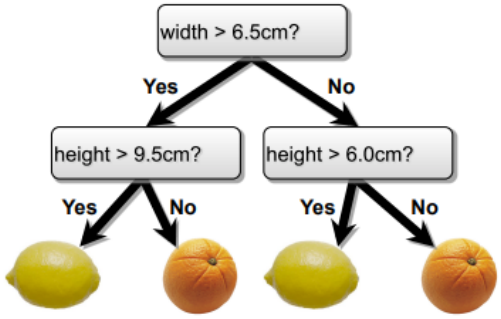
- What is the information gain of split A? Very informative!



- Entropy of class outcome before split:
  $H(Y) = -\frac{2}{7}\log_2(\frac{2}{7}) - \frac{5}{7}\log_2(\frac{5}{7}) \approx 0.86$
- Conditional entropy of class outcome after split:
  $H(Y|left) = 0,\ H(Y|right) \approx 0.97$
- $IG(split) \approx 0.86 - (\frac{2}{7} \cdot 0 + \frac{5}{7} \cdot 0.97) \approx 0.17$!!

Constructing Decision Trees





- At each level, one must choose:
  1. Which feature to split.
  2. Possibly where to split it.
- Choose them based on how much information we would gain from the decision! (choose feature that gives the highest gain)

Decision Tree Construction Algorithm

School of Computing & IT

- Simple, greedy, recursive approach, builds up tree node-by-node
  1. pick a feature to split at a non-terminal node
  2. split examples into groups based on feature value
  3. for each group:
     - if no examples – return majority from parent
     - else if all examples in same class – return class
     - else loop to step 1
- Terminates when all leaves contain only examples in the same class or are empty.
- Questions for discussion:
  - How do you choose the feature to split on?
  - How do you choose the threshold for each feature?

Back to Our Example

School of Computing & IT

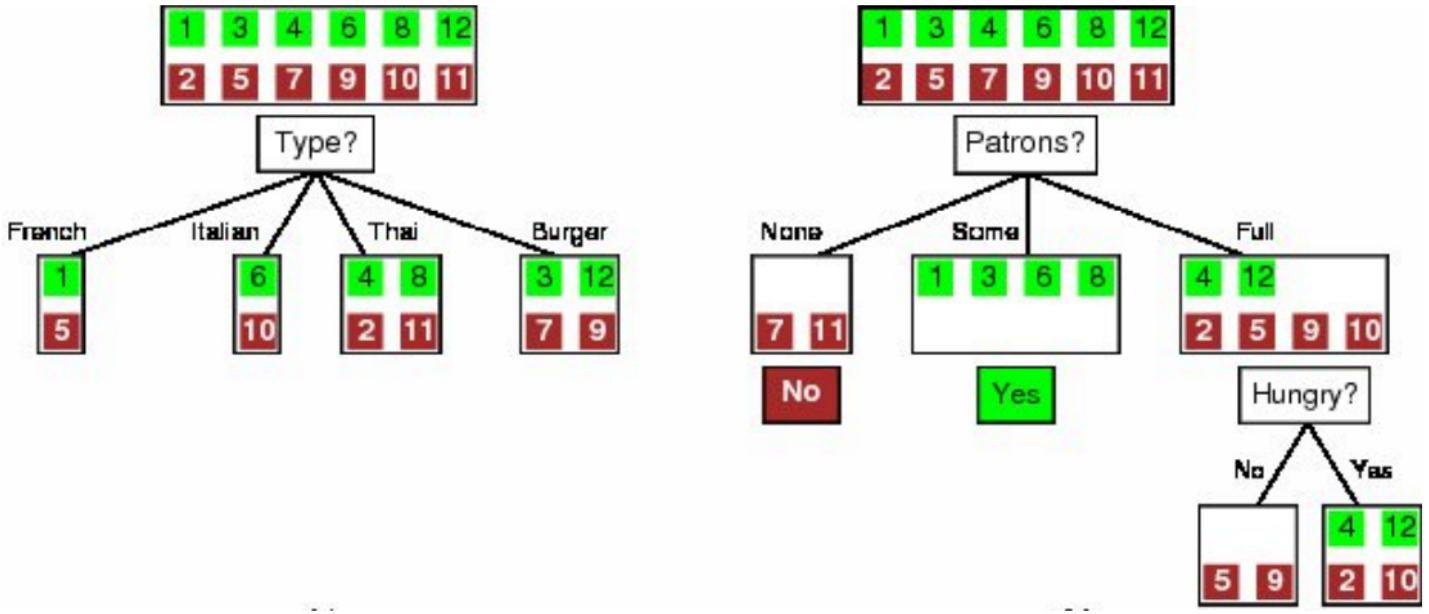
Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 =$ *Yes*
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 =$ *No*
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 =$ *Yes*
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 =$ *Yes*
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 =$ *No*
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 =$ *Yes*
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 =$ *No*
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 =$ *Yes*
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 =$ *No*
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} =$ *No*
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} =$ *No*
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} =$ *Yes*

Features:

1. Alternate: whether there is a suitable alternative restaurant nearby.
2. Bar: whether the restaurant has a comfortable bar area to wait in.
3. Fri/Sat: true on Fridays and Saturdays.
4. Hungry: whether we are hungry.
5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
6. Price: the restaurant's price range ($, $$, $$$).
7. Raining: whether it is raining outside.
8. Reservation: whether we made a reservation.
9. Type: the kind of restaurant (French, Italian, Thai or Burger).
10. WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

[from: Russell & Norvig]

Feature Selection



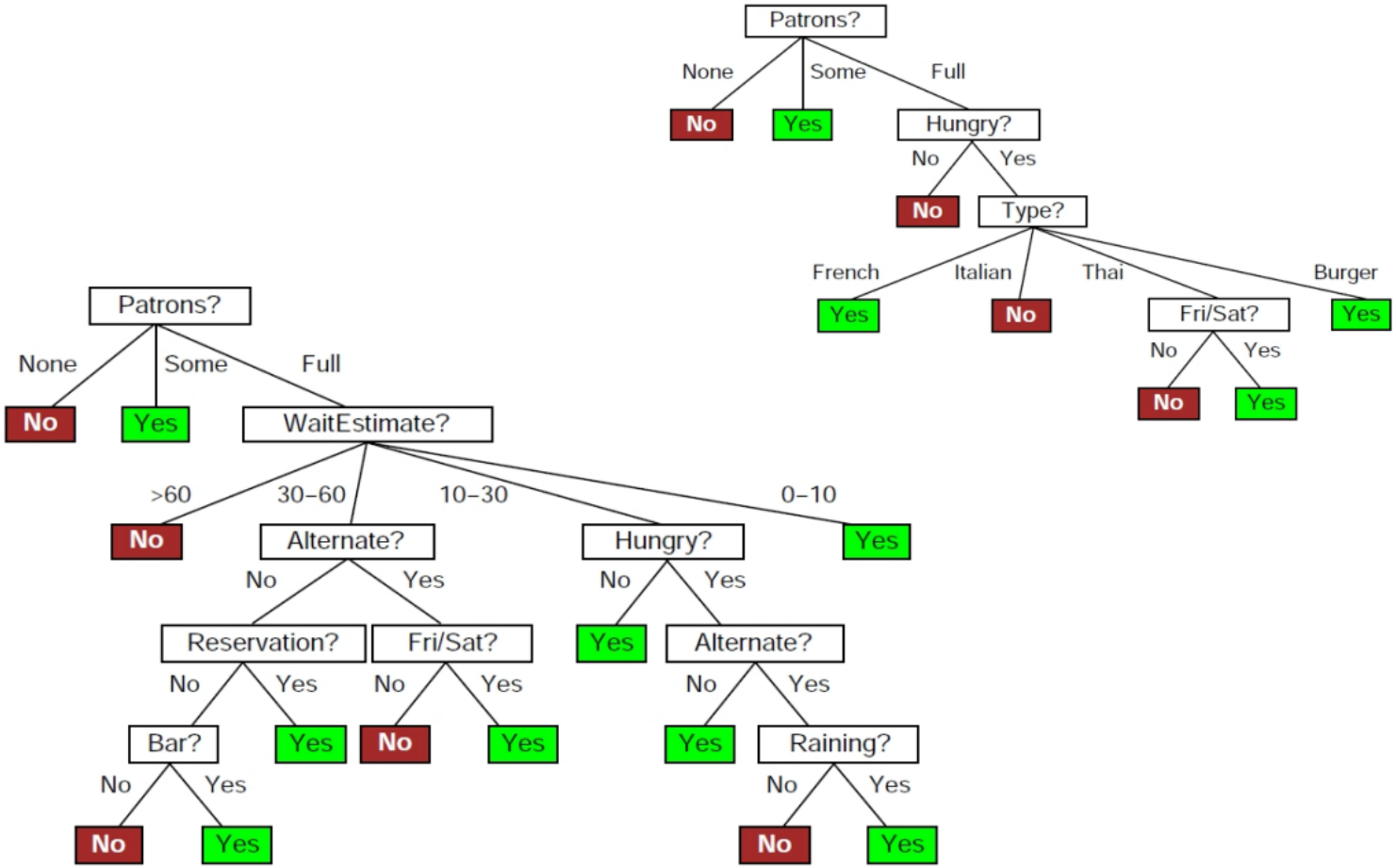
$$IG(Y) = H(Y) - H(Y|X)$$

$$IG(type) = 1 - \left[\frac{2}{12}H(Y|\text{Fr.}) + \frac{2}{12}H(Y|\text{It.}) + \frac{4}{12}H(Y|\text{Thai}) + \frac{4}{12}H(Y|\text{Bur.})\right] = 0$$

$$IG(Patrons) = 1 - \left[\frac{2}{12}H(0,1) + \frac{4}{12}H(1,0) + \frac{6}{12}H(\frac{2}{6},\frac{4}{6})\right] \approx 0.541$$

Which Tree is Better? Vote!

School of Computing & IT



What Makes a Good Tree?

School of Computing & IT

- Not too small: need to handle important but possibly subtle distinctions in data
- Not too big:
  - Computational efficiency (avoid redundant, spurious attributes)
  - Avoid over-fitting training examples
  - Human interpretability
- "Occam's Razor": find the simplest hypothesis that fits the observations
  - Useful principle, but hard to formalize (how to define simplicity?)
  - See Domingos, 1999, "The role of Occam's razor in knowledge discovery"
- We desire small trees with informative nodes near the root

Decision Tree Miscellany

School of Computing & IT

- Problems:
  - You have exponentially less data at lower levels
  - Too big of a tree can overfit the data
  - Greedy algorithms don't necessarily yield the global optimum
- Handling continuous attributes
  - Split based on a threshold, chosen to maximize information gain
- Decision trees can also be used for regression on real-valued outputs. Choose splits to minimize squared error, rather than maximize information gain.

School of
Computing & IT