Introduction to Data Science

(Lecture 6)

Dr. Mohammad Pourhomayoun
Associate Professor
Computer Science Department
California State University, Los Angeles

Decision Tree Classification

Review: Titanic Disaster

- Let's start this topic with a famous problem/competition from kaggle website: Predicting **survival on the Titanic!**

[1]: Ref: www.kaggle.com.

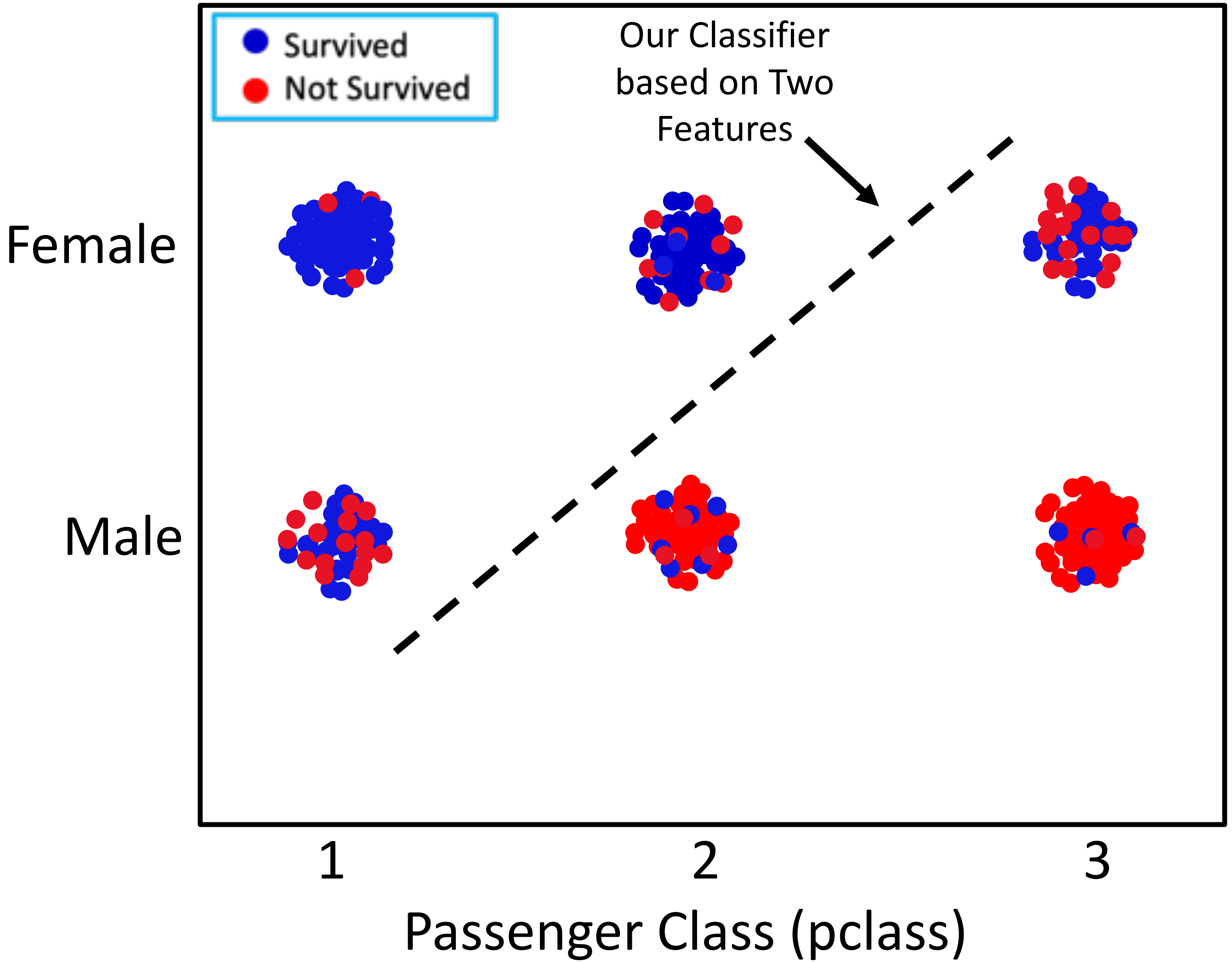
Predict survival on the Titanic

- Passenger list on Titanic[1]:

pclass	age	gender	sibsp	parch	fare	Survival
1	54	male	0	0	52	0
3	2	male	3	1	21	0
3	27	female	0	2	11	1
2	14	female	1	1	30	1
3	4	female	1	2	16	1
3	38	male	0	0	7	0
1	24	female	0	0	71	1
3	22	female	0	0	8	1
1	38	female	1	0	53	1
3	26	male	0	1	8	0
3	???	male	1	2	8	0

[1]: Ref: Kaggle website, and Bill Howe, University of Washington.

Predict survival on the Titanic



An Improvement on the Classifier

- Making decision based on **three features**.
  - Example: Based on “gender”, “pclass”, and “age”.

IF (Sex=‘female’):

IF (pclass=‘1’) OR (pclass=‘2’) THEN: Survive ← Yes

ELSE IF (pclass=‘3’):

IF (age<4) THEN: Survive ← Yes

ELSE IF (age>4) THEN: Survive ← No

ELSE IF (Sex=‘male’):

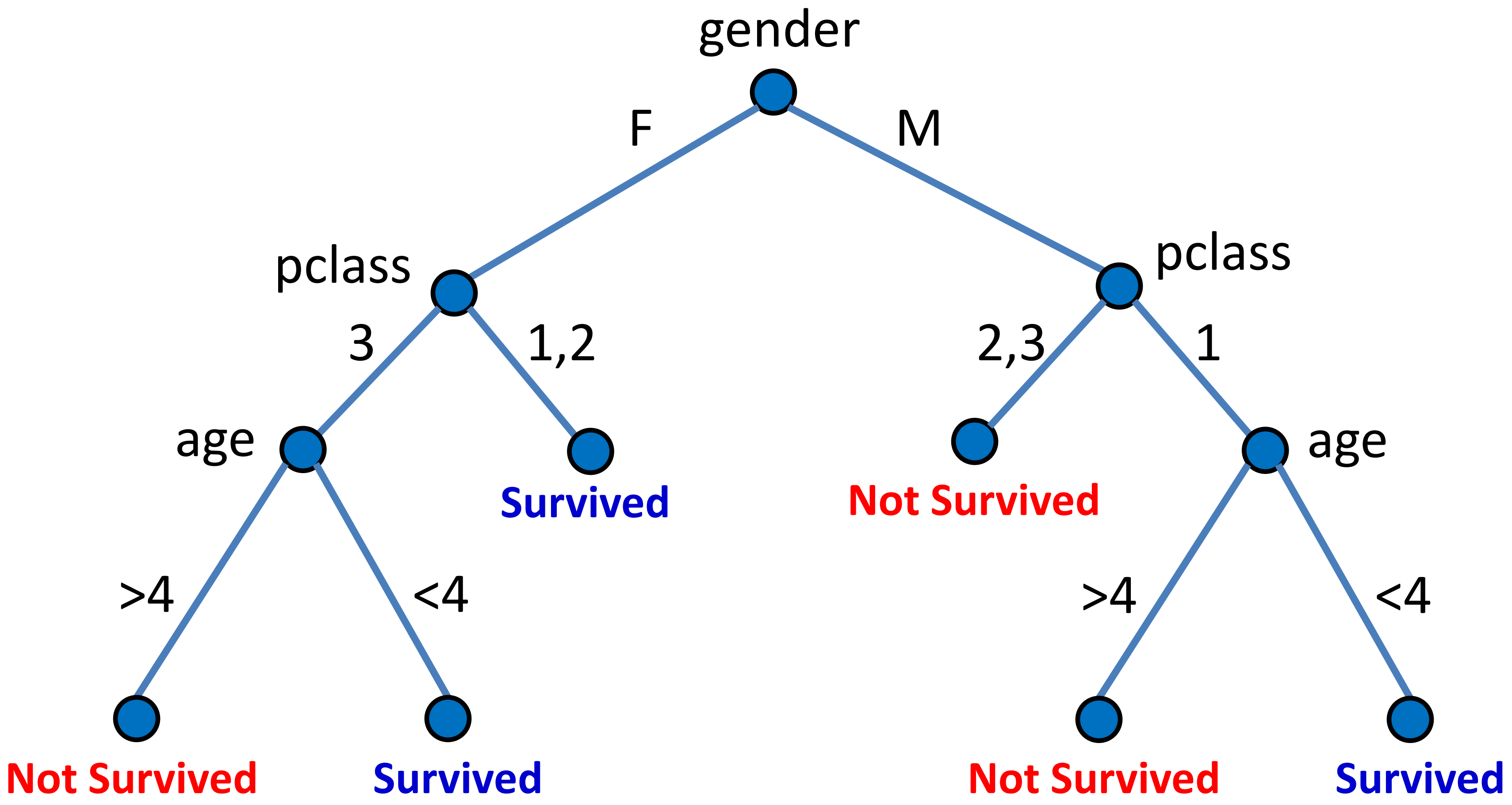
IF (pclass=‘2’) OR (pclass=‘3’) THEN: Survive ← No

ELSE IF (pclass=‘1’):

IF (age<4) THEN: Survive ← Yes

ELSE IF (age>4) THEN: Survive ← No

Decision Tree

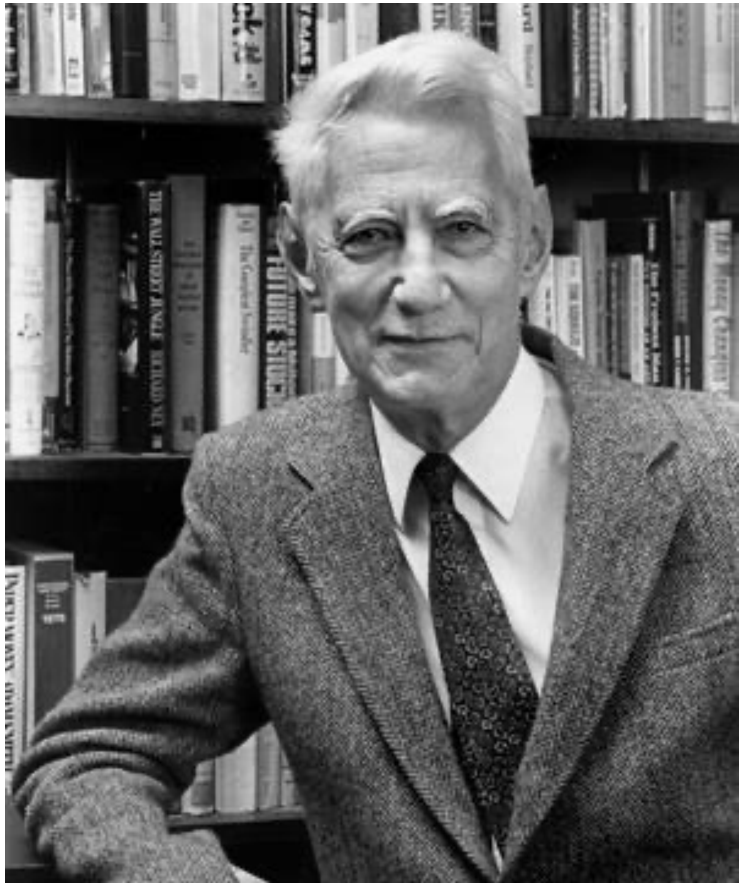


Training a Decision Tree Model

- **Question**: How to select the first feature at the top of the tree: The feature that can split (classify) the data samples best.
- **Idea**: The best feature is the one that provides the ***most amount of information*** about the label.
- So, we need a ***metric*** to measure **information**.

Claude E. Shannon (1916 – 2001)
The Father of Information Theory

- Shannon is noted for having founded ***information theory*** with a landmark paper, ”***A Mathematical Theory of Communication”***, that he published in 1948.
- He is, perhaps, equally well known for founding ***digital circuit design theory*** in 1937, when—as a 21-year-old master's degree student at MIT [1].



[1]: wikipedia

Two Important Concepts about Measuring the Information

1. The amount of information about an <u>event x</u> has **<u>inverse relationship</u>** to the **<u>probability</u>** of that event.

- Example:
  - *"The sun will rise tomorrow morning"*
    - This sentence provides very Low amount of information because it talks about a common (very likely) event.
  - *"An Eclipse occurs tomorrow"*
    - This sentence provides High amount of information because it is an unlikely event.

$$I(X) \sim \frac{1}{p(x)}$$

The amount of information about event x → $I(X)$

$p(x)$ ← The Probability of event x

Two Important Concepts about Measuring the Information

2. When two independent events happen, the joint probability of them is the **multiplication** of the two probabilities. However, the total information about two independent events should be the **summation** of the two piece of information.

- Example: Flipping a Coin twice: H,T
  - Prob(two independent events) = prob(event1) * prob(event2)
  - info(two independent events) = info(event1) + info(event2)

Two Important Concepts about Measuring the Information

- When two independent events happen, the joint probability is the multiplication of the two probabilities. However, in this case, the total information about them should be the summation of the two pieces of information.

- So, the "**Information function**" should have this property:
  Information (p_1,p_2) = Information(p_1) + Information(p_2)

Two Important Concepts about Measuring the Information

Question: What function has this property?

$$f(xy) = f(x) + f(y)$$

Answer: Log!!!

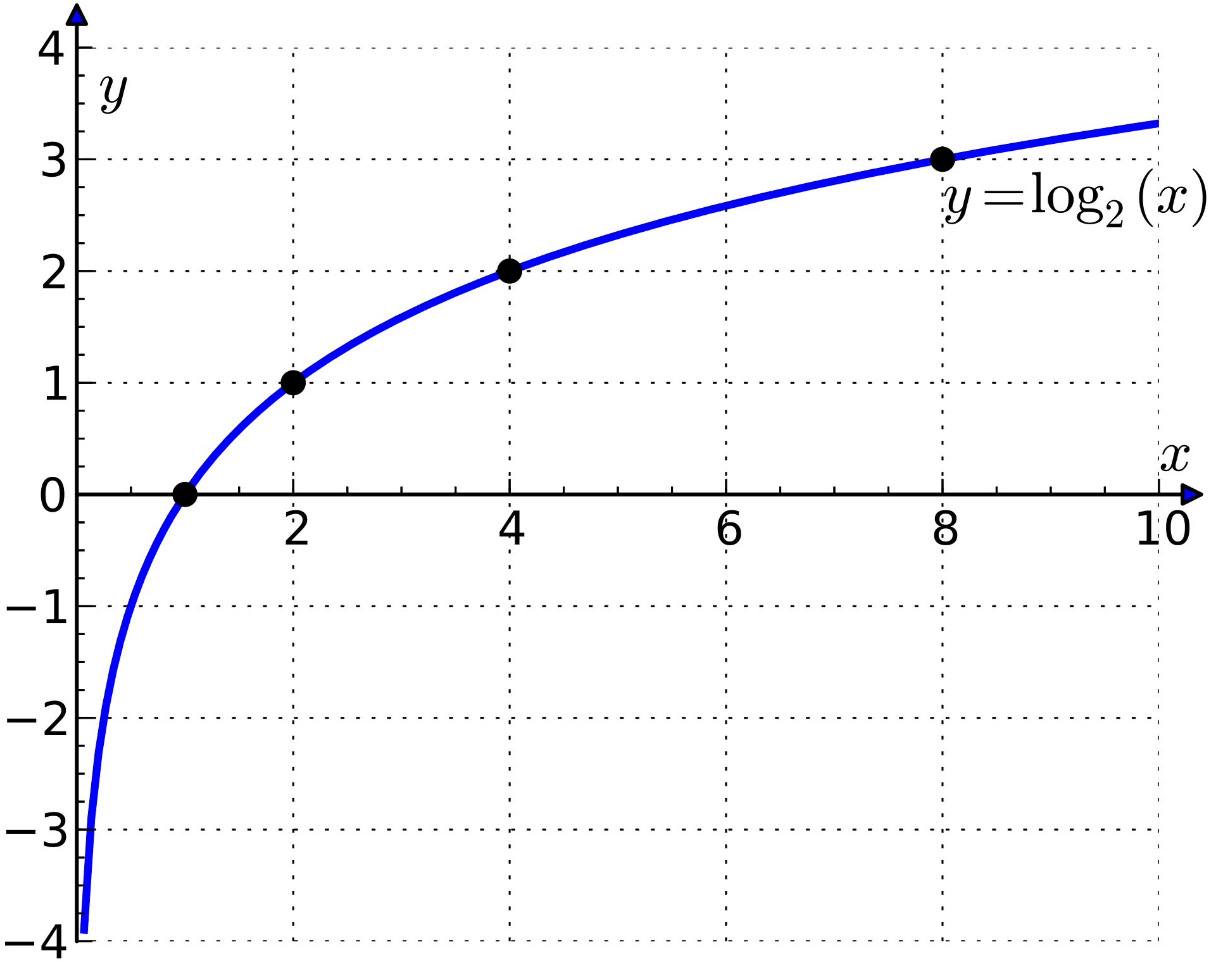
$$\log(xy) = \log(x) + \log(y)$$

More properties:

$$\log(1/x) = -\log(x)$$

$$\log(x^n) = n.\log(x)$$

$\text{Log}_2(x)$



Two Important Concepts about Measuring the Information

1. The information about an event x has inverse relationship to the probability of that event.
2. When two independent events happens, the total information about them should be the summation of the two pieces of information.

➔ Thus, **information metric** can be defined as:

$$I(X) = \log_2\left(\frac{1}{p(x)}\right) = -\log_2(p(x))$$

- Note: It is common to use log based 2, and then the unit of information is in ***bit***.

ENTROPY

- Entropy measures the amount of "**Uncertainty**" or "**Unpredictability**".
- In other word, Entropy is the "**expected information**".
- If a random variable X has K different **possible** values $x_1, x_2, \ldots, x_K$, the **entropy** is defined as (E is the *Expected Value*):

$$H(X) = E\big(I(X)\big)$$

$$= \sum_{k=1}^{K} p(X = x_k) I(X = x_k)$$

$$= -\sum_{k=1}^{K} p(X = x_k) \log_2 p(X = x_k)$$

Example: Flipping a Coin

- **Question**: We have a Fair Coin and an Unfair Coin. If we flip both coins, Which one is more predictable (H = Head, T = Tail)?

- Fair Coin: $p(\mathrm{H}) = p(\mathrm{T}) = 0.5$

$$H(X) = -\sum_{k=1}^{K} p(X = x_k)\log_2 p(X = x_k)$$

$$= -(0.5\log_2 0.5 + 0.5\log_2 0.5) = 1 \text{ bit}$$

- Unfair Coin: $p(\mathrm{H}) = 0.7$, $p(\mathrm{T}) = 0.3$

$$H(X) = -\sum_{k=1}^{K} p(X = x_k)\log_2 p(X = x_k)$$

$$= -(0.7\log_2 0.7 + 0.3\log_2 0.3) = 0.88 \text{ bit}$$

- The unfair coin is **less unpredictable** (**more predictable**) than a fair coin.

Example: Rolling a Fair Die

- Fair Die: $p(1) = p(2) = p(3) = p(4) = p(5) = p(6) = 1/6$

$$H(X) = -\sum^{K} p(X = x_k) \log_2 p(X = x_k)$$

$$= -6 \times \left(\frac{1}{6} \log_2 \frac{1}{6} \right) = 2.58 \text{ bit}$$

Example: Rolling an Unfair Die

- Unfair Die: $p(1) = p(2) = p(3) = p(4) = p(5) = 0.1,\ \ p(6) = 0.5$

$$H(X) = -\sum_{k=1}^{K} p(X = x_k)\log_2 p(X = x_k)$$

$$= -5 \times (0.1 \log_2 0.1) - 0.5 \log_2 0.5$$

$$= 2.16 \text{ bit}$$

- The unfair die is less unpredictable (more predictable) than a fair coin.

Example: Titanic

- 1500 died, 724 survivors, out of 2224 passengers in titanic,
  – Thus, the probability of survival: 724/2224
  – The probability of not survival: 1500/2224

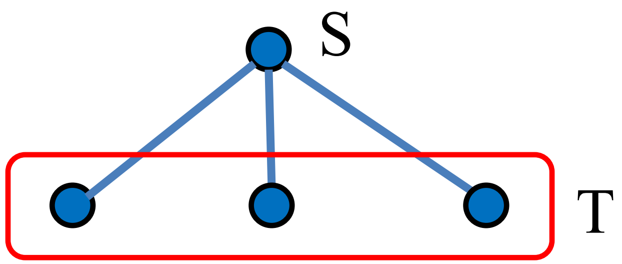
$$H(X) = -\sum_{k=1}^{K} p(X = x_k)\log_2 p(X = x_k)$$

$$= -\left((\frac{1500}{2224})\log_2(\frac{1500}{2224}) + (\frac{724}{2224})\log_2(\frac{724}{2224}) \right) = 0.91\ bit$$

- If the log base is 2, the unit of the entropy is called “bit”

Information Gain

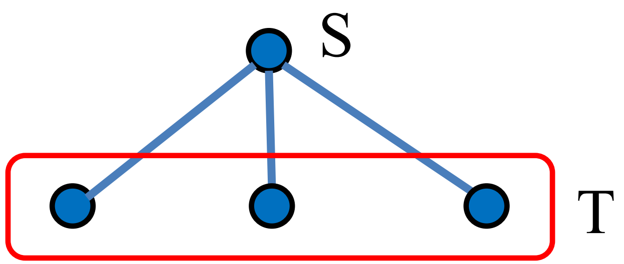
- **Reminder**: Entropy measures the uncertainty.
- **Idea**: Gaining Information reduces uncertainty.
- In case of decision tree, we define Information Gain as the ***reduction in the Entropy*** from ***before to after*** the dataset is split on a feature.



Information Gain

- Reminder: Entropy measures the uncertainty
- Idea: Gaining Information reduces uncertainty
- In case of decision tree, we define Information Gain as the reduction in the Entropy from before to after the dataset is split on an attribute:

$$IG = H(S) - \sum_{t \in T} p(t)H(t)$$



- $H(S)$: Entropy of dataset before splitting
- T: The set of subsets created after splitting the dataset
- $P(t)$: The proportion of the number of elements in each subset *t* after splitting
- $H(t)$: Entropy of the subset *t*

Splitting in Decision Trees

- **Question**: Which feature do we choose at each level of the tree to split data samples?

- **Answer**: The one with the **largest information gain**.
  - The one that reduces the entropy (i.e. unpredictability, uncertainty) the most.

Training a Decision Tree Classifier

Temp	Humidity	Windy	Label
high	low	Yes	Sunny
low	high	Yes	Rainy
high	low	No	Sunny
high	high	Yes	Rainy
mild	mild	No	Sunny
mild	high	No	Rainy
low	mild	Yes	Rainy

Training Data

Example: Training (building) a Decision Tree Classifier for Weather Forecasting, based on Temperature, Humidity, and Wind information of the past 7 days.

Before Splitting: 7 data samples, 3 Sunny, 4 Rainy

- Probability of sunny: 3/7
- Probability of rainy: 4/7

Temp	Humidity	Windy	Label
high	low	Yes	Sunny
low	high	Yes	Rainy
high	low	No	Sunny
high	high	Yes	Rainy
mild	mild	No	Sunny
mild	high	No	Rainy
low	mild	Yes	Rainy

Before Splitting: 7 data samples, 3 Sunny, 4 Rainy

- Probability of sunny: 3/7
- Probability of rainy: 4/7

Entropy Before Splitting:

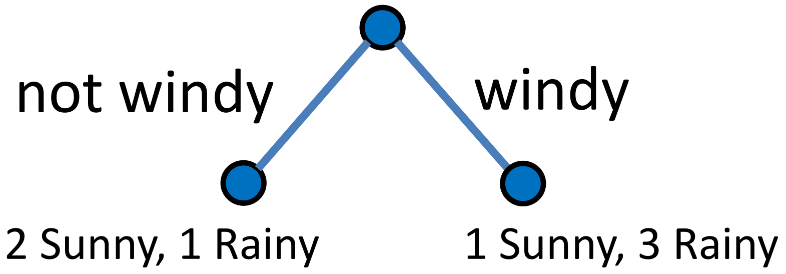
$$H(X) = -\sum_{k=1}^{K} p(X = x_k)\log_2 p(X = x_k)$$

$$= -\left((\frac{4}{7})\log_2(\frac{4}{7}) + (\frac{3}{7})\log_2(\frac{3}{7}) \right) = 0.98\ bit$$

Entropy Before Splitting

Temp	Humidity	Windy	Label
high	low	Yes	Sunny
low	high	Yes	Rainy
high	low	No	Sunny
high	high	Yes	Rainy
mild	mild	No	Sunny
mild	high	No	Rainy
low	mild	Yes	Rainy

- **Split based on Wind (2 branches):**
  - Windy: 4 samples (4/7): 1 Sunny, 3 Rainy
  - Not Windy: 3 samples (3/7): 2 Sunny, 1 Rainy



Windy: $$H(X) = -\left((\frac{1}{4})\log_2(\frac{1}{4}) + (\frac{3}{4})\log_2(\frac{3}{4}) \right) = 0.81\ bit$$

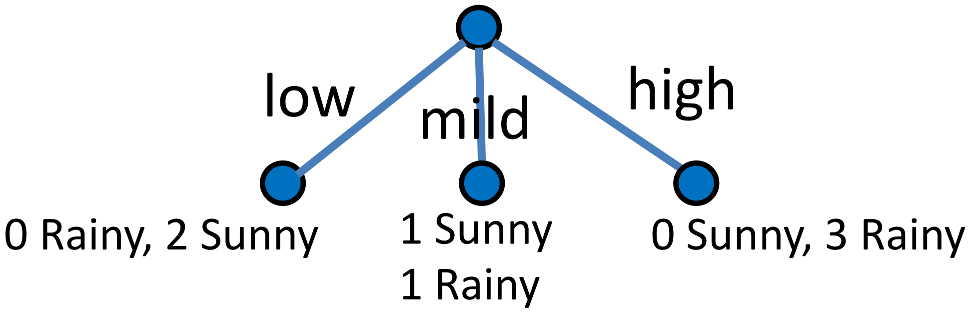
Not Windy: $$H(X) = -\left((\frac{2}{3})\log_2(\frac{2}{3}) + (\frac{1}{3})\log_2(\frac{1}{3}) \right) = 0.91\ bit$$

Average Entropy After Splitting on Wind

Weighted Average: $$E(H(X)) = \left(\frac{4}{7} \times 0.81 + + \frac{3}{7} \times 0.91 \right) = 0.85\ bit$$

Temp	Humidity	Windy	Label
high	low	Yes	Sunny
low	high	Yes	Rainy
high	low	No	Sunny
high	high	Yes	Rainy
mild	mild	No	Sunny
mild	high	No	Rainy
low	mild	Yes	Rainy

- **Split based on Humidity (3 branches):**
  - high: 3 samples (3/7): 0 Sunny, 3 Rainy
  - mild: 2 samples (2/7): 1 Sunny, 1 Rainy
  - low: 2 samples (2/7): 0 Rainy, 2 Sunny



High: $$H(X) = -\left((\frac{3}{3})\log_2(\frac{3}{3}) + (\frac{0}{3})\log_2(\frac{0}{3}) \right) = 0\ bit$$

Mild: $$H(X) = -\left((\frac{1}{2})\log_2(\frac{1}{2}) + (\frac{1}{2})\log_2(\frac{1}{2}) \right) = 1\ bit$$

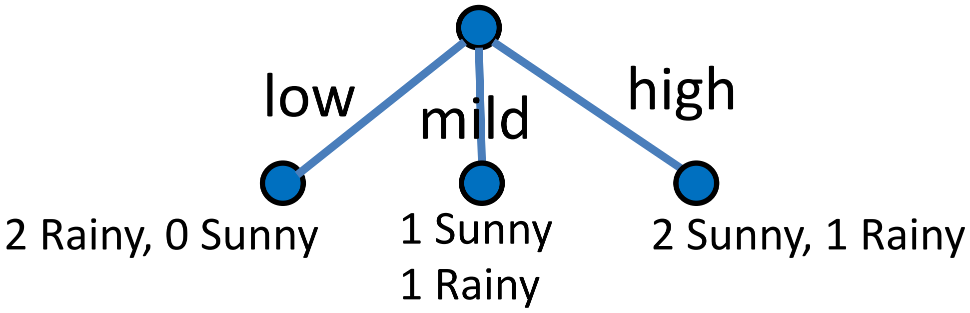
Low: $$H(X) = -\left((\frac{2}{2})\log_2(\frac{2}{2}) + (\frac{0}{2})\log_2(\frac{0}{2}) \right) = 0\ bit$$

Average Entropy
After Splitting on Humidity

Weighted Average: $$E(H(X)) = \left(\frac{3}{7}\times 0 + \frac{2}{7}\times 1 + \frac{2}{7}\times 0 \right) = 0.28\ bit$$

Temp	Humidity	Windy	Label
high	low	Yes	Sunny
low	high	Yes	Rainy
high	low	No	Sunny
high	high	Yes	Rainy
mild	mild	No	Sunny
mild	high	No	Rainy
low	mild	Yes	Rainy

- **Split based on Temp (3 branches):**
  - high: 3 samples (3/7): 2 Sunny, 1 Rainy
  - mild: 2 samples (2/7): 1 Sunny, 1 Rainy
  - low: 2 samples (2/7): 2 Rainy, 0 Sunny



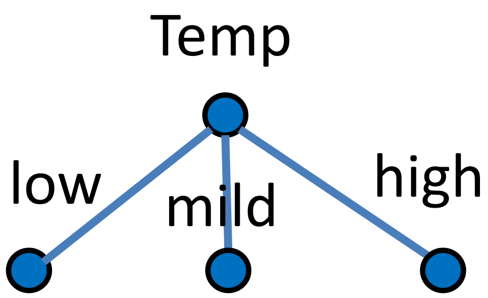
High: $$H(X) = -\left((\frac{2}{3})\log_2(\frac{2}{3}) + (\frac{1}{3})\log_2(\frac{1}{3}) \right) = 0.91\ bit$$

Mild: $$H(X) = -\left((\frac{1}{2})\log_2(\frac{1}{2}) + (\frac{1}{2})\log_2(\frac{1}{2}) \right) = 1\ bit$$

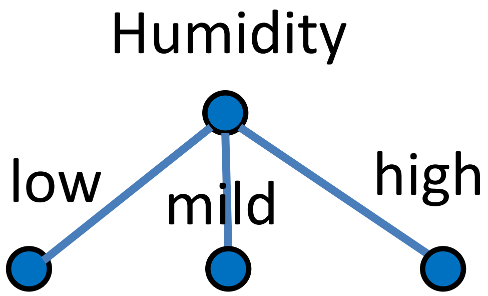
Low: $$H(X) = -\left((\frac{2}{2})\log_2(\frac{2}{2}) + (\frac{0}{2})\log_2(\frac{0}{2}) \right) = 0\ bit$$

Average Entropy After Splitting on Temp

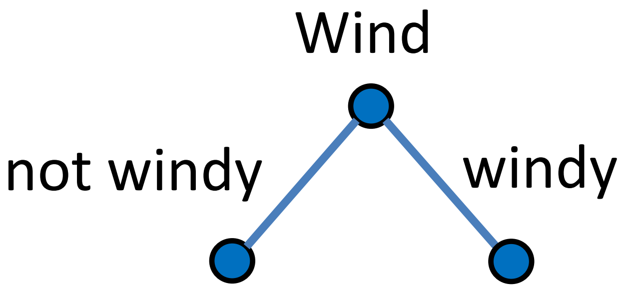
Weighted Average: $$E(H(X)) = \left(\frac{3}{7} \times 0.91 + \frac{2}{7} \times 1 + \frac{2}{7} \times 0 \right) = 0.67\ bit$$



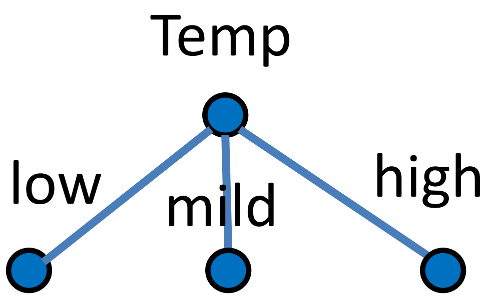
Entropy Before Split: 0.98
Entropy After Split on Temp: 0.67
Information Gain: 0.98 – 0.67 = 0.31



Entropy Before Split: 0.98
Entropy After Split on Humidity: 0.28
Information Gain: 0.98 – 0.28 = 0.70



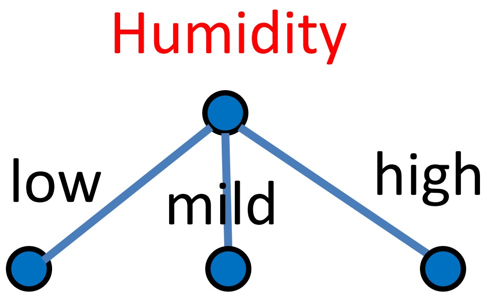
Entropy Before Split: 0.98
Entropy After Split Wind: 0.85
Information Gain: 0.98 – 0.85 = 0.13



Entropy Before Split: 0.98

Entropy After Split on Temp: 0.67

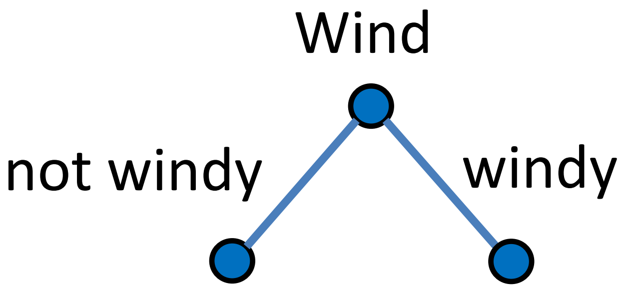
Information Gain: 0.98 – 0.67 = 0.31



Entropy Before Split: 0.98

Entropy After Split on Humidity: 0.28

Information Gain: 0.98 – 0.28 = 0.70



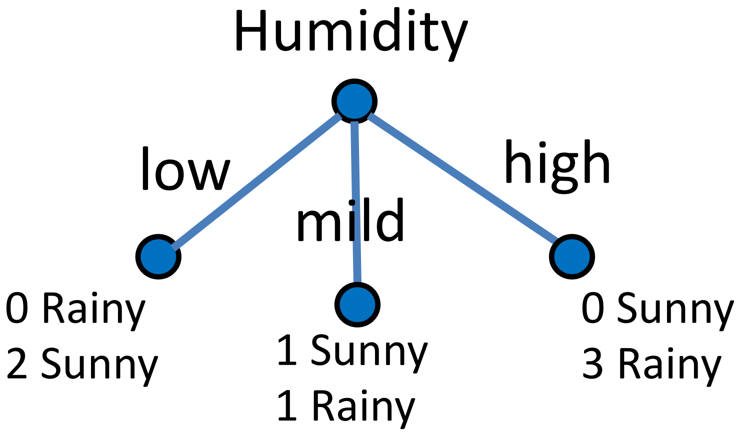
Entropy Before Split: 0.98

Entropy After Split Wind: 0.85

Information Gain: 0.98 – 0.85 = 0.13

The Best Feature

- In this example, Splitting the data samples based on **Humidity** will **minimize the entropy** (unpredictability) compare to other features.
- In other word, splitting based on Humidity provides the **maximum amount of Information Gain** at this level.
- So, the **best feature** to split the data samples at the top of the decision tree is Humidity.



Building a Decision Tree (ID3 Algorithm)

1. Calculate the entropy after splitting the dataset based on every feature.
2. Select the feature for which the entropy is minimum (information gain is maximum).
3. Split the dataset into subsets using that feature, and make a decision tree node for that.
4. Repeat with remaining features.
5. Stop splitting a branch if:
   - No features left, or
   - All samples assigned the same label

- **Note**: In ID3, we assume that features are discrete. Thus, we need to Discretize continuous attributes.

Discretizing Numerical Attributes

Temp	Humidity	Windy	Label
90	low	Yes	Sunny
60	high	Yes	Rainy
92	low	No	Sunny
89	high	Yes	Sunny
70	mild	No	Sunny
73	high	No	Rainy
61	mild	Yes	Rainy

- In general, features can be continuous-valued (numeric)
- We need to define intervals/thresholds to discretize continuous features.

Discretizing Numerical Attributes

Temp	Humidity	Windy	Label
90	low	Yes	Sunny
60	high	Yes	Rainy
92	low	No	Sunny
89	high	Yes	Sunny
70	mild	No	Sunny
73	high	No	Rainy
61	mild	Yes	Rainy

Sort

Temp	Humidity	Windy	Label
92	low	No	Sunny
90	low	Yes	Sunny
89	high	Yes	Sunny
73	high	No	Rainy
70	mild	No	Sunny
61	mild	Yes	Rainy
60	high	Yes	Rainy

- We need to define intervals/thresholds to discretize continuous features.

Discretizing Numerical Attributes

Temp	Humidity	Windy	Label
90	low	Yes	Sunny
60	high	Yes	Rainy
92	low	No	Sunny
89	high	Yes	Sunny
70	mild	No	Sunny
73	high	No	Rainy
61	mild	Yes	Rainy

Sort

Temp	Humidity	Windy	Label
92	low	No	Sunny
90	low	Yes	Sunny
89	high	Yes	Sunny
73	high	No	Rainy
70	mild	No	Sunny
61	mild	Yes	Rainy
60	high	Yes	Rainy

- General Approach (Brute Force): We have to try every possible split to see which one minimizes the entropy.

Discretizing Numerical Attributes

Temp	Humidity	Windy	Label
92	low	No	Sunny
90	low	Yes	Sunny
89	high	Yes	Sunny
73	high	No	Rainy
70	mild	No	Sunny
61	mild	Yes	Rainy
60	high	Yes	Rainy

Any smarter way to do it?
(rather than trying every possible split point)

- General Approach (Brute Force): We have to try every possible split to see which one minimizes the entropy.

Discretizing Numerical Attributes

Temp	Humidity	Windy	Label
92	low	No	Sunny
90	low	Yes	Sunny
89	high	Yes	Sunny
73	high	No	Rainy
70	mild	No	Sunny
61	mild	Yes	Rainy
60	high	Yes	Rainy

Good Splitting point: Splitting at this point can distinguish the labels (Sunny vs. Rainy) with pretty good accuracy!

- General Approach (Brute Force): We have to try every possible split to see which one minimizes the entropy.

Discretizing Numerical Attributes

Temp	Humidity	Windy	Label
92	low	No	Sunny
90	low	Yes	Sunny
89	high	Yes	Sunny
73	high	No	Rainy
70	mild	No	Sunny
61	mild	Yes	Rainy
60	high	Yes	Rainy

Another Good Splitting point: Splitting at this point can distinguish the labels (Sunny vs. Rainy) with pretty good accuracy!

- General Approach (Brute Force): We have to try every possible split to see which one minimizes the entropy.

Discretizing Numerical Attributes

Temp	Humidity	Windy	Label
92	low	No	Sunny
90	low	Yes	Sunny
89	high	Yes	Sunny
73	high	No	Rainy
70	mild	No	Sunny
61	mild	Yes	Rainy
60	high	Yes	Rainy

Discretizing

Temp	Humidity	Windy	Label
Hi	low	No	Sunny
Hi	low	Yes	Sunny
Hi	high	Yes	Sunny
Mild	high	No	Rainy
Mild	mild	No	Sunny
Low	mild	Yes	Rainy
Low	high	Yes	Rainy

- General Approach (Brute Force): We have to try every possible split to see which one minimizes the entropy.
- But, the above choice of threshold might be a good approach!

Advantages and Disadvantages

- **Advantages of using decision tree classifier:**
  - Easily **interpretable** by human
  - Handles both numerical and categorical data
  - It is a parametric algorithm: unlike KNN, we do not need to carry our training dataset around

- **Disadvantages:**
  - Very prone to Overfitting (more on this later).
  - Heuristic training techniques (brute force, trial & error)

Thank You!

Questions?