CIS501 – Lecture 9

Woon Wei Lee
Fall 2013, 10:00am-11:15am,
Sundays and Wednesdays

For today:

- Administrative stuff
    - Quiz results review
- Decision trees
    - Golf example
    - Entropy and information gain
- Presentations
    - Bedoor Al Shebli
    - Ju Young Shin

Quiz results – quick review

- Top quartile - 21/30
- Median - 18/30
- Bottom quartile – 16/30

What does this mean for you...

egs of low scoring questions:

Q30: Data mining method are often intrinsically probabilistic. Why is this?

(B: Real world data is always noisy)

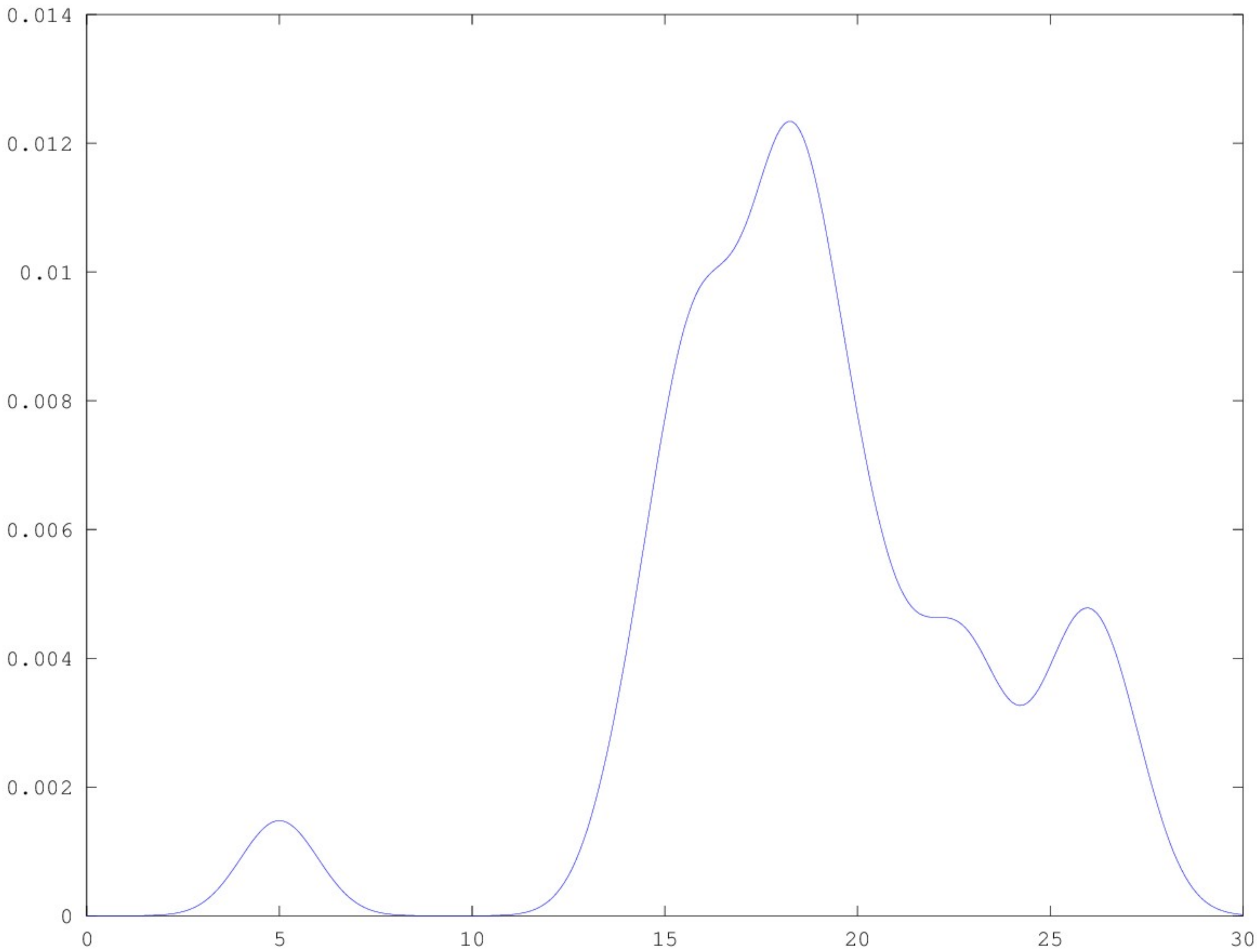
Q19: If you were to insist on sticking to the histogram approximation, which discretization technique could most help to solve the problem above:

(D: Non-disjoint)

Q5: Which of the following is a symptom of overfitting

(B: Low training errors but markedly higher validation errors)

Kernel density estimate



The problem..

- **Should I play golf today?**

- How would you decide?..
  - Depends... Is it raining? Is it windy? Will it be warm..?
- Can you build a classifier to “automate” this decision?
  - Rule-based:
    - IF raining → NO
    - IF NOT raining, NOT windy, NOT sunny → YES
    - IF NOT raining, IF windy → Maybe..

Cont'd

- Fictitious, but popular: “Quinlan golf dataset”
- Provides mix of different data types, and quite “tricky”

Outlook	Temp (°F)	Humidity (%)	Windy	Class
sunny	75	70	true	play
sunny	80	90	true	don’t play
sunny	85	85	false	don’t play
sunny	72	95	false	don’t play
sunny	69	70	false	play
overcast	72	90	true	play
overcast	83	78	false	play
overcast	64	65	true	play
overcast	81	75	false	play
rain	71	80	true	don’t play
rain	65	70	true	don’t play
rain	75	80	false	play
rain	68	80	false	play
rain	70	96	false	play

Classes
play, don’t play
Outlook
sunny, overcast, rain
Temperature
numerical value
Humidity
numerical value
Windy
true, false

Cont'd

- Focus on nominal categories for now..
  - Outlook → {Sunny, Rain, Overcast}
  - Windy → {True, False}
  - Target output → {Play, Don't Play}
- Ad-hoc rule formation:
- By “eye-balling”, we can form some simple rules:
  - IF Raining AND Windy → don't play
  - If Overcast → play!

- Also, there are some discrepancies:
  - (*Sunny,False*) occurs twice for “don't play”, and once for “play”
  - (*Sunny,True*) also appears once for “don't play” and once for “play”

(Note: Such data sets are said to have failed the *adequacy criterion*, and are hence inconsistent)

Outlook	Windy	Class
Sunny	TRUE	don’t play
Sunny	FALSE	don’t play
Sunny	FALSE	don’t play
Rain	TRUE	don’t play
Rain	TRUE	don’t play
Sunny	TRUE	play
Sunny	FALSE	play
Overcast	TRUE	play
Overcast	FALSE	play
Overcast	TRUE	play
Overcast	FALSE	play
Rain	FALSE	play
Rain	FALSE	play
Rain	FALSE	play

Cont'd

- Focus on nominal categories for now..
  - Outlook → {Sunny, Rain, Overcast}
  - Windy → {True, False}
  - Target output → {Play, Don't Play}
- Ad-hoc rule formation:
- By “eye-balling”, we can form some simple rules:
  - IF Raining AND Windy → don't play
  - If Overcast → play!

- Also, there are some discrepancies:
  - (*Sunny,False*) occurs twice for “don't play”, and once for “play”
  - (*Sunny,True*) also appears once for “don't play” and once for “play”

(Note: Such data sets are said to have failed the *adequacy criterion*, and are hence inconsistent)

Outlook	Windy	Class
Sunny	TRUE	don’t play
Sunny	FALSE	don’t play
Sunny	FALSE	don’t play
Rain	TRUE	don’t play
Rain	TRUE	don’t play
Sunny	TRUE	play
Sunny	FALSE	play
Overcast	TRUE	play
Overcast	FALSE	play
Overcast	TRUE	play
Overcast	FALSE	play
Rain	FALSE	play
Rain	FALSE	play
Rain	FALSE	play

Cont'd

- Focus on nominal categories for now..
  - Outlook → {Sunny, Rain, Overcast}
  - Windy → {True, False}
  - Target output → {Play, Don't Play}
- Ad-hoc rule formation:
- By “eye-balling”, we can form some simple rules:
  - IF Raining AND Windy → don't play
  - If Overcast → play!

- Also, there are some discrepancies:
  - (*Sunny,False*) occurs twice for “don't play”, and once for “play”
  - (*Sunny,True*) also appears once for “don't play” and once for “play”

(Note: Such data sets are said to have failed the *adequacy criterion*, and are hence inconsistent)

Outlook	Windy	Class
Sunny	TRUE	don’t play
Sunny	FALSE	don’t play
Sunny	FALSE	don’t play
Rain	TRUE	don’t play
Rain	TRUE	don’t play
Sunny	TRUE	play
Sunny	FALSE	play
Overcast	TRUE	play
Overcast	FALSE	play
Overcast	TRUE	play
Overcast	FALSE	play
Rain	FALSE	play
Rain	FALSE	play
Rain	FALSE	play

Masdar INSTITUTE

Decision tree induction..

- (Still looking at nominal categories)
- We want to build a particular class of decision tree, known as the “Top Down Inductive Decision Tree” (TDIDT)
- Simple rule induction algorithm:

<u>TDIDT Algorithm:</u>

IF all example are from same class:

- Tree is a leaf – label node with this class, and return

ELSE

- Select a feature to split on
- Sort examples into subsets based on values of feature (one for each value)
- Branch the tree by creating a new node (tree) for each subset
- Recurse..

Condition: No feature may be selected twice in a branch (obvious..)

Outlook	Windy	Class
Sunny	TRUE	don’t play
Sunny	TRUE	don’t play
Rain	TRUE	don’t play
Rain	TRUE	don’t play
Sunny	FALSE	play
Sunny	FALSE	play
Overcast	TRUE	play
Overcast	FALSE	play
Overcast	TRUE	play
Overcast	FALSE	play
Rain	FALSE	play
Rain	FALSE	play
Rain	FALSE	play

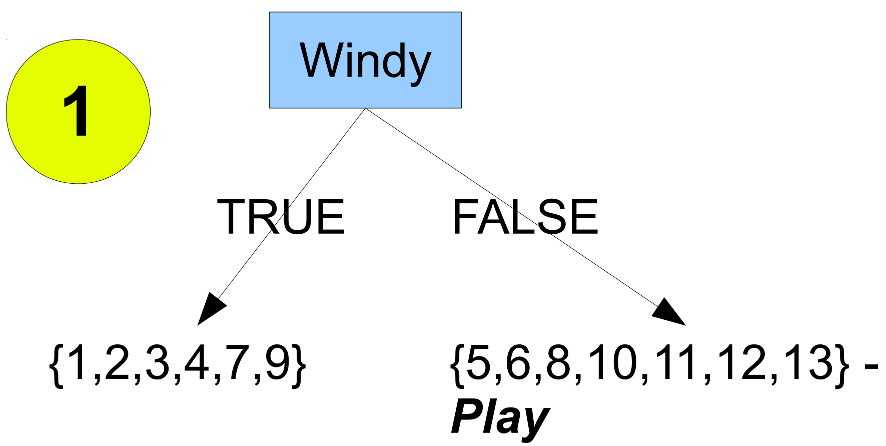
Note: Modified version of Quinlan Golf example

Masdar INSTITUTE

Decision tree induction..

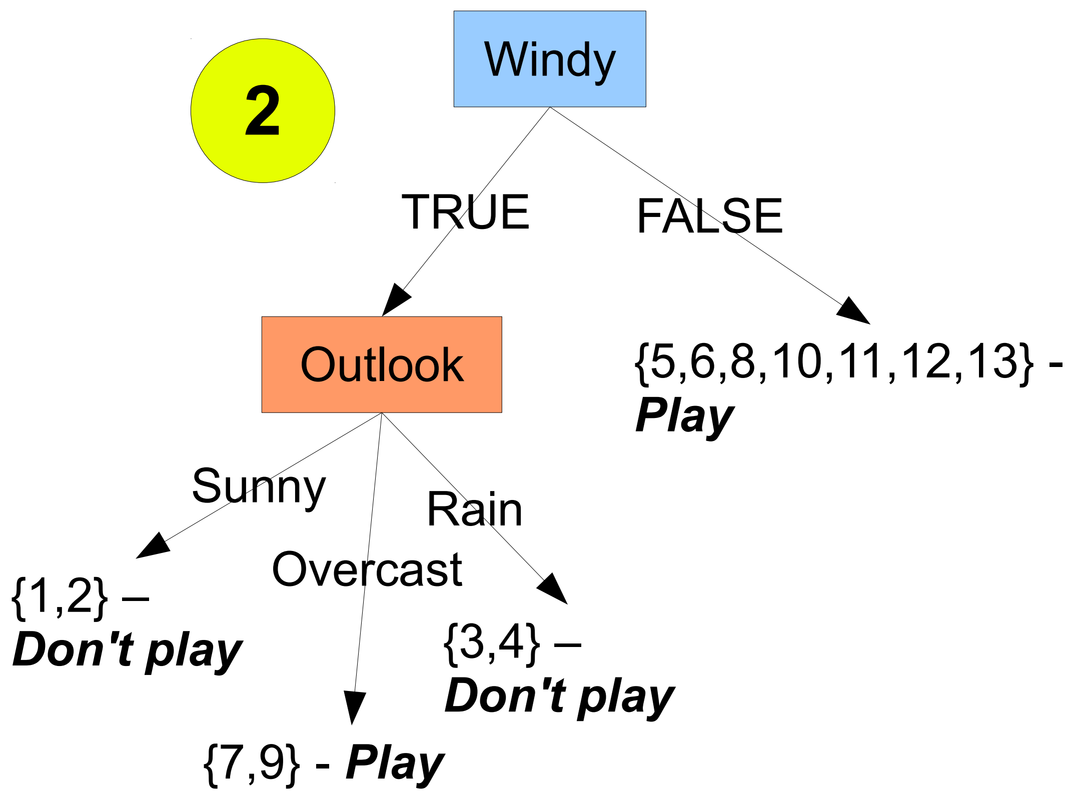
- Selection of attributes → three potential “strategies”:
  - *takefirst* - take the features in the order in which they appear in the training set, or according to alphabetical ordering, etc.
  - *takelast* – inverse of above..
  - *takerandom – select features at random*
- Let's start from *Windy*

Outlook	Windy	Class
1) Sunny	TRUE	don’t play
2) Sunny	TRUE	don’t play
3) Rain	TRUE	don’t play
4) Rain	TRUE	don’t play
5) Sunny	FALSE	play
6) Sunny	FALSE	play
7) Overcast	TRUE	play
8) Overcast	FALSE	play
9) Overcast	TRUE	play
10) Overcast	FALSE	play
11) Rain	FALSE	play
12) Rain	FALSE	play
13) Rain	FALSE	play



Decision tree induction..

- Selection of attributes → three potential "strategies":
  - *takefirst* - take the features in the order in which they appear in the training set, or according to alphabetical ordering, etc.
  - *takelast* – inverse of above..
  - *takerandom – select features at random*
- Let's start from *Windy*



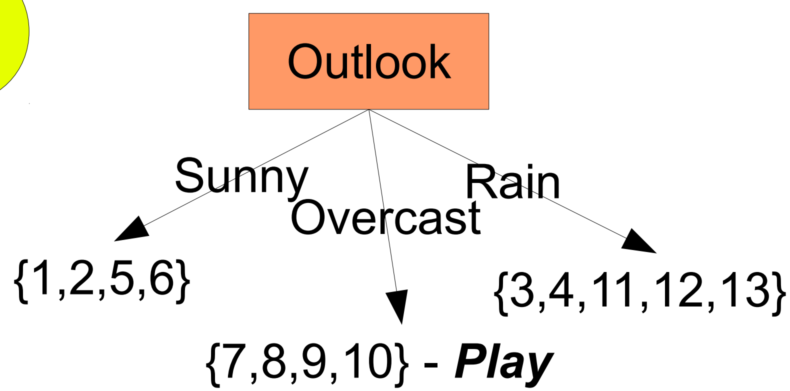
Outlook	Windy	Class
1) Sunny	TRUE	don't play
2) Sunny	TRUE	don't play
3) Rain	TRUE	don't play
4) Rain	TRUE	don't play
5) Sunny	FALSE	play
6) Sunny	FALSE	play
7) Overcast	TRUE	play
8) Overcast	FALSE	play
9) Overcast	TRUE	play
10) Overcast	FALSE	play
11) Rain	FALSE	play
12) Rain	FALSE	play
13) Rain	FALSE	play

Masdar INSTITUTE

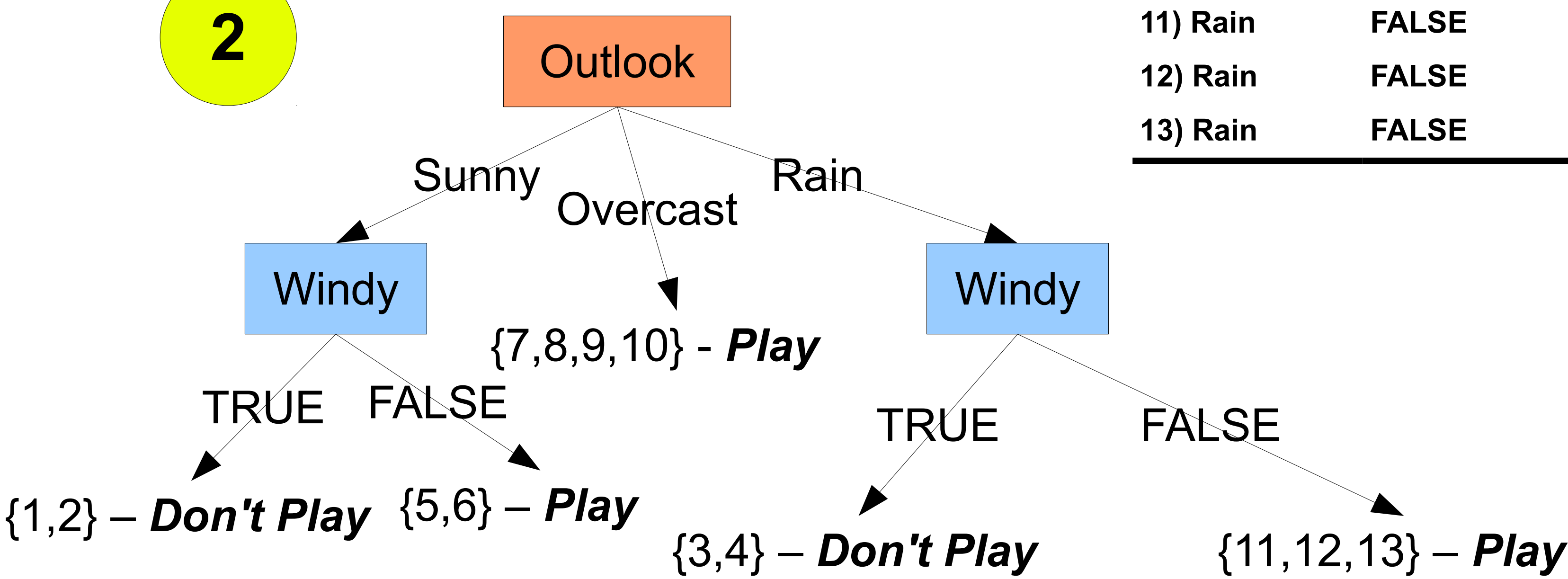
Cont'd

- Let's try starting from "Outlook"..

1



2



Outlook	Windy	Class
1) Sunny	TRUE	don't play
2) Sunny	TRUE	don't play
3) Rain	TRUE	don't play
4) Rain	TRUE	don't play
5) Sunny	FALSE	play
6) Sunny	FALSE	play
7) Overcast	TRUE	play
8) Overcast	FALSE	play
9) Overcast	TRUE	play
10) Overcast	FALSE	play
11) Rain	FALSE	play
12) Rain	FALSE	play
13) Rain	FALSE	play

Cont'd

- As can be seen, the choice of different starting conditions results in a different tree each time
  - If starting with *windy*, there is a (7/13) chance that the classification can be made in a single comparison
  - If starting with *outlook*, there is a (4/13) chance that the classification can be made in a single comparison/test
- Clearly, random selection of the attributes is not guaranteed to be optimal
  - The right choice of feature/attribute order can result in trees which are more efficient, or which are smaller in size.
  - In information retrieval or data mining terminology, this intuition can be expressed as some features being more “informative” than others
  - Numerical quantification of this informativeness can be obtained using information theory.

Entropy

- Central concept in both Physics and Mathematics → measure of uncertainty, disorder or randomness

  (Not really relevant, but for the sake of general knowledge: You may or may not have heard of the concept of "Heat Death")

- Information theory "variant" is known as *Shannon's Entropy*
  - Similar mathematical form to physical counterpart (discussed later)
- Can be interpreted in a number of ways:
  - Measure of uncertainty associated with a Random Variable
  - The information rate of that random variable
  - The best possible *lossless* compression rate for corresponding information
  - Etc..
- **Trivia**: Shannon completed his graduate education at MIT. At 21 his Master's thesis revolutionized the field of digital circuit design. Something to aspire to ☺

Claude Shannon (1916 – 2001)

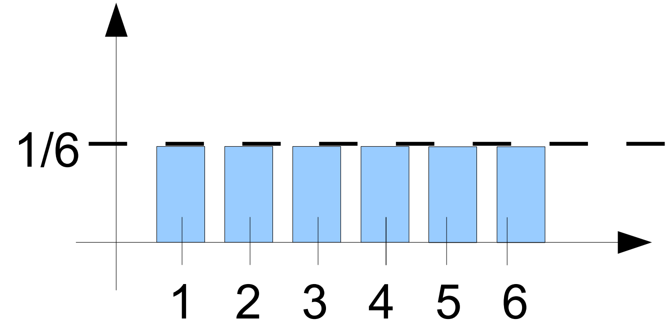
Entropy (Cont'd)

- Defined as:

$$H(x) = -E[\log_2(p(x))]$$
$$= -\sum_x p(x).\log_2(p(x))$$

- To understand how this works, consider the case of a die-throw..

- The PMF of a die-throw is:

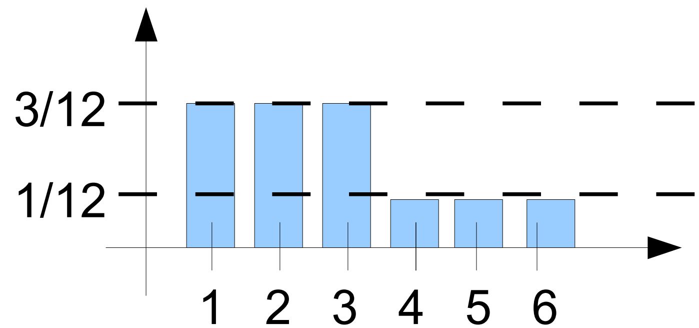


⇒ Entropy is given by:

$$H(x) = -\sum_x p(x).\log_2(p(x))$$
$$= -6\times[\frac{1}{6}\times\log_2(\frac{1}{6})] = 2.6$$

Cont'd

- Consider a “biased” die:

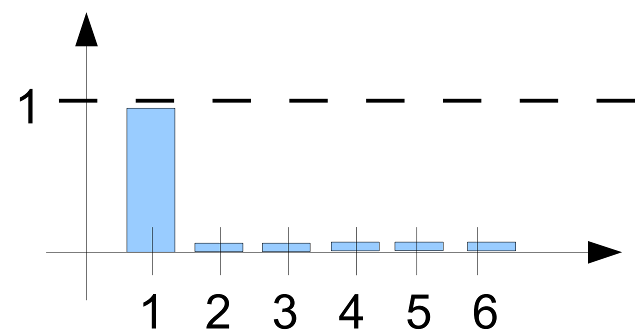


⇒ Entropy is given by:

$$H(x)=-\sum_{x} p(x).\log_2(p(x))$$

$$=-3\times[\frac{1}{4}\times\log_2(\frac{1}{4})]-3\times[\frac{1}{12}\times\log_2(\frac{1}{12})]=2.39$$

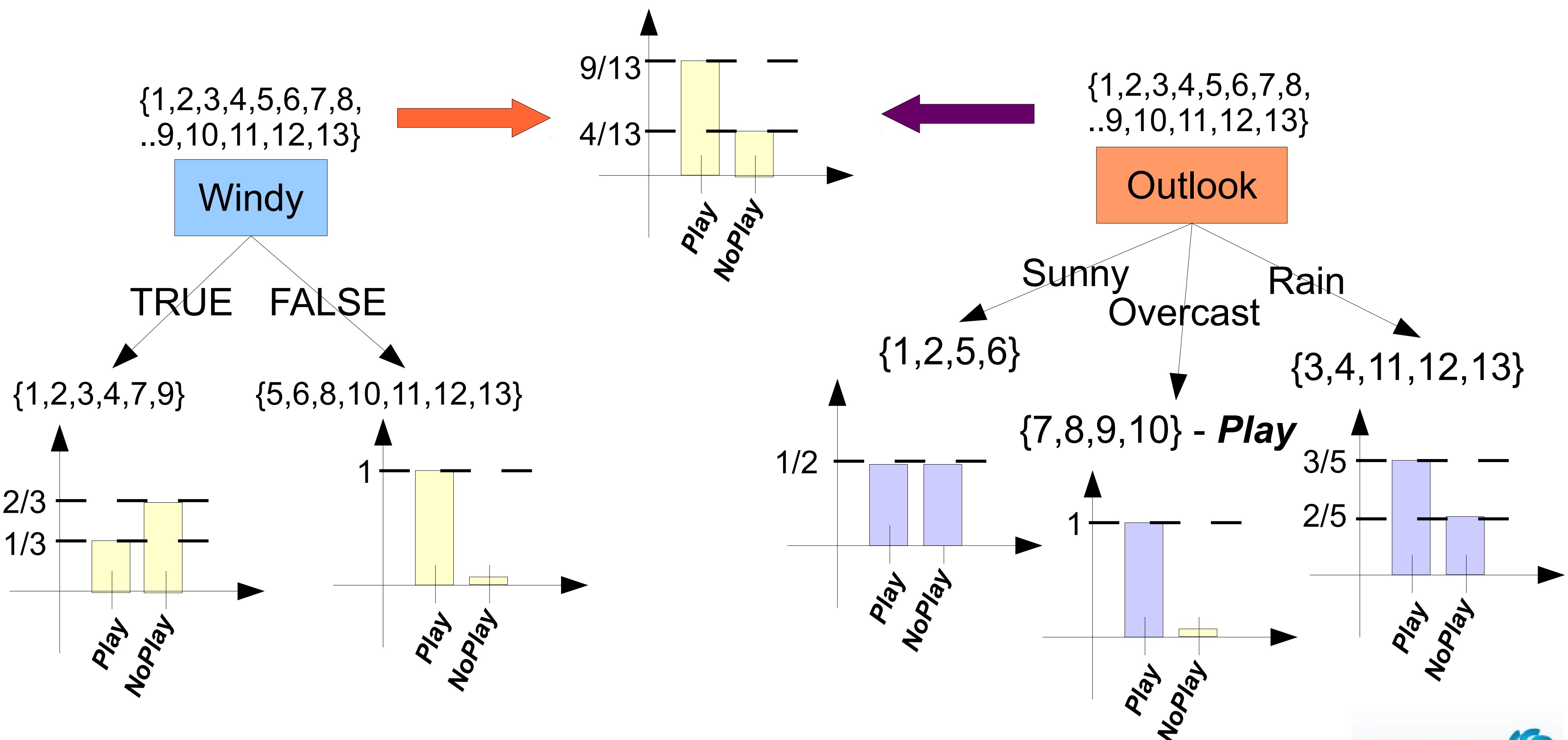
- An even more extreme case:



⇒ **Question:**
What is the Entropy in this case?

Information Gain

- How does this help us?
- Consider the entropy (uncertainty) in the outputs at each branch of the decision trees shown earlier:



Information Gain

- Examining the features individually:
  - For “Windy”, we see that:
    i. Branching on “true” results in a 1/3↔2/3 split.
    ii. Branching on “false” results in a 1↔0 split.
  - For “Outlook”, we see that:
    i. Branching on “sunny” results in a 1/2↔1/2 split.
    ii. Branching on “overcast” results in a 1↔0 split.
    iii. Branching on “rain” results in a 3/5↔2/5 split.
- i.e. Intuitively, branching on “Windy” seems to result in a greater reduction in the uncertainty or randomness of the data w.r.t. the class labels.
- This can be interpreted as a gain in informativeness → “Information Gain”!

Cont'd

- The concept of "Information Gain" formalizes this intuition using the concept of entropy
- Defined as:

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

- i.e. it is the difference in the entropy of the data set *X* before and after an attribute *a* is considered
- *H*(*X*|*a*) is calculated by taking the average entropy of the branches, weighted by the number of instances:

$$H(X) = -[\log(\frac{4}{13}) \times \log(\frac{4}{13}) + \log(\frac{9}{13}) \times \log(\frac{9}{13})] = 0.89$$

$$IG(windy) = 0.89 - \frac{6}{13} \times [\frac{1}{3} \times \log(\frac{1}{3}) + \frac{2}{3} \times \log(\frac{2}{3})] = 0.467$$

$$IG(outlook) = 0.89 - \frac{4}{13} \times [2 \times \frac{1}{2} \times \log(\frac{1}{2})] - \frac{5}{13} \times [\frac{3}{5} \times \log(\frac{3}{5}) + \frac{2}{5} \times \log(\frac{2}{5})] = 0.209$$

- i.e. IG(*windy)*>IG(*outlook*) ⇒ *windy* is a better feature!