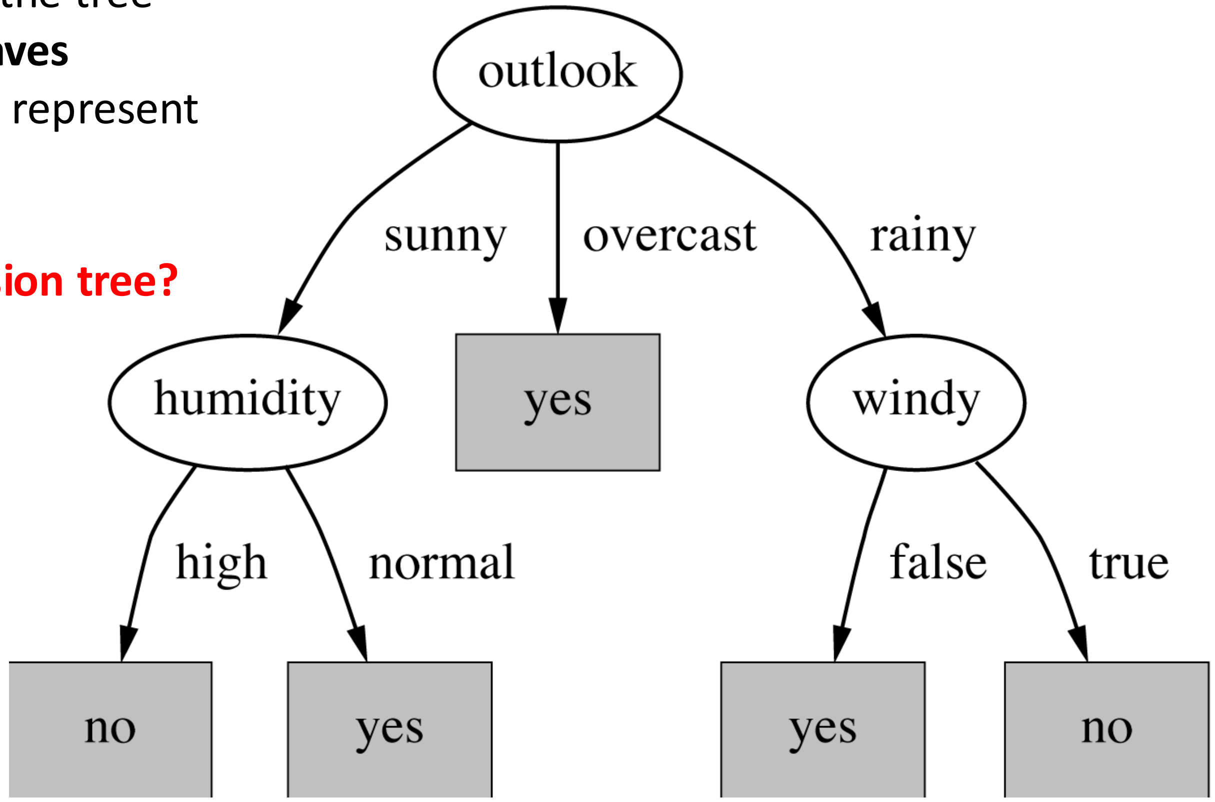
Classification

Decision trees

What is a decision tree?

- "First" node is the **root** of the tree
- "Last" nodes are called **leaves**
- The edges between nodes represent **branches**

- **How do we "read" a decision tree?**



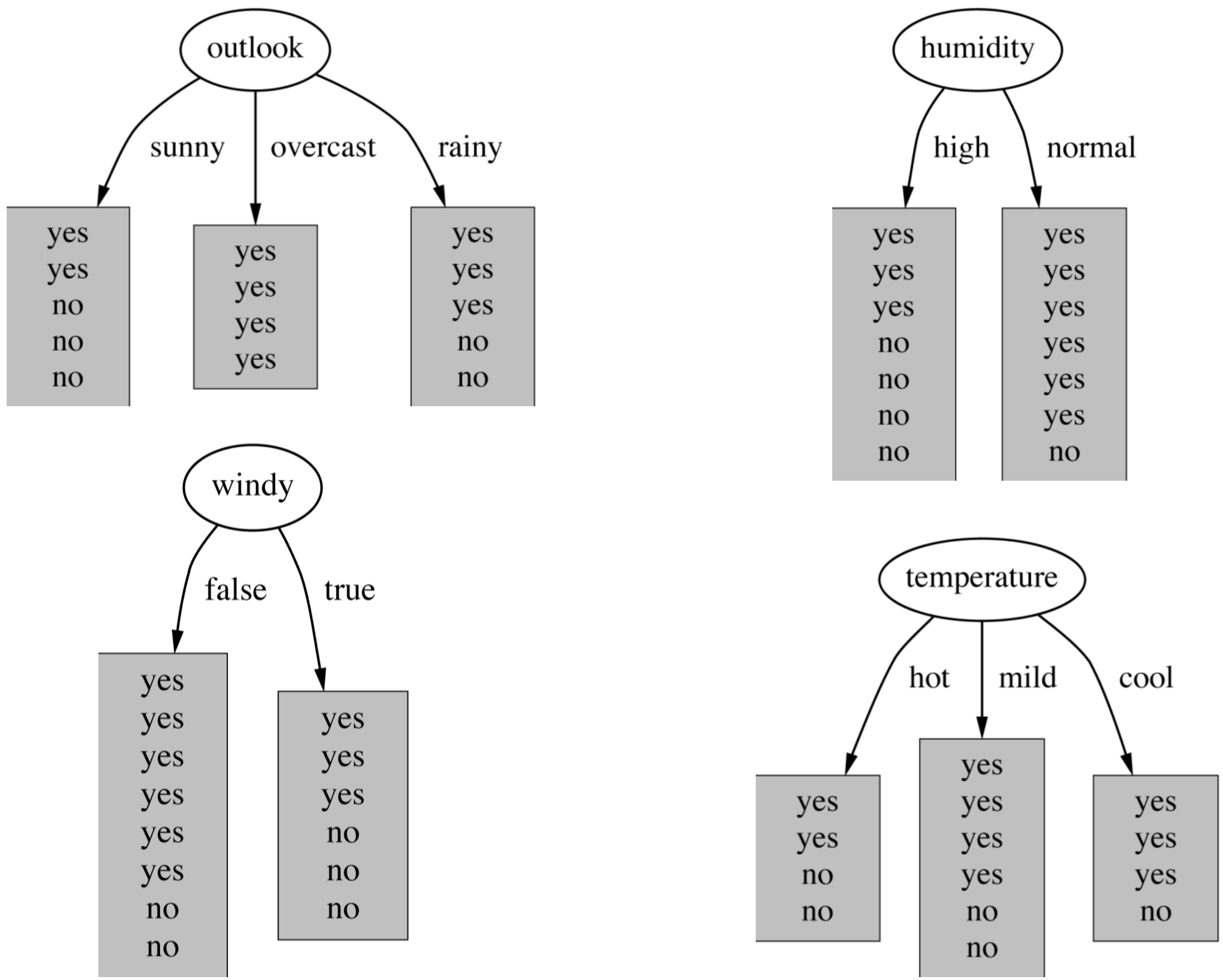
How to "build" a decision tree?

- Procedure (recursive):
  - Choose the "best" attribute
  - Split the data into subsets according to the values of the "best" attribute
  - Repeat the procedure in each of the subsets
- Stop when (the stopping criterion):
  - out of attributes,
  - all leaves are "pure" (all examples in a leaf belong to the same class),
  - the (sub)set is empty.

Choosing the "best" attribute

- Which attribute is "the best"?
  - The one, yielding the smaller tree
  - The one, yielding the highest classification accuracy
  - Heuristic: choose attribute, yielding "purest" nodes
- Let's look at some "(im)purity" measures:
  - Information gain
  - Information gain ration
  - Gini index

Which attribute to choose? – example



Information gain

- Measuring the information as <u>entropy</u>:
  - given the probability distribution of some events, entropy measures the information that is needed to encode every possible outcome of these events
  - entropy is measured in bits
- Formula for computing the entropy:

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_n \log_2 p_n$$

- where p_1, p_2, …, p_n are the probabilities of given events

Computing information – "before"

- If a set ***T*** contains examples with ***n*** classes, ***info(T)*** is defined as:

$$\mathrm{info}(T) = \sum_{j=1}^{n} -p_j \log_2(p_j)$$

- where $\boldsymbol{p_j}$ is the relative frequency (probability) of class ***j*** in ***T***.

Computing information – "after"

- We split the set $\boldsymbol{T}$ containing $\boldsymbol{N}$ examples into subsets $\boldsymbol{T_1}$, $\boldsymbol{T_2}$, …, $\boldsymbol{T_k}$ containing $\boldsymbol{N_1}$, $\boldsymbol{N_2}$, …, $\boldsymbol{N_k}$ examples, respectively. The information of this split is defined as:

$$info_{split}(T) = info(T_1, T_2, \ldots, T_k) = \sum_{i=1}^{k} \frac{N_i}{N} info(T_i)$$

Information gain – definition

$$\boldsymbol{InfoGain(T) = info(T) - info(T_1, T_2, \ldots, T_k)}$$

- **Information gain =**
  = information "before split" – information "after split"

- **InfoGain = IBS – IAS**

InfoGain – the "weather" example

Outlook	**Temperature**	**Humidity**	**Windy**	**Play**
Sunny	Hot	High	False	**No**
Sunny	Hot	High	True	**No**
Overcast	Hot	High	False	**Yes**
Rainy	Mild	High	False	**Yes**
Rainy	Cool	Normal	False	**Yes**
Rainy	Cool	Normal	True	**No**
Overcast	Cool	Normal	True	**Yes**
Sunny	Mild	High	False	**No**
Sunny	Cool	Normal	False	**Yes**
Rainy	Mild	Normal	False	**Yes**
Sunny	Mild	Normal	True	**Yes**
Overcast	Mild	High	True	**Yes**
Overcast	Hot	Normal	False	**Yes**
Rainy	Mild	High	True	**No**

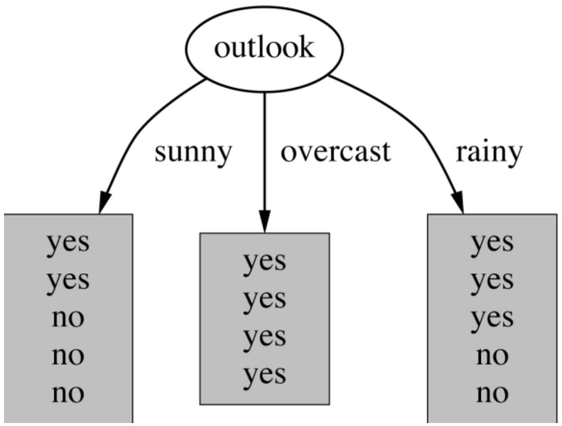
Computing the IBS

- Information "before the split":
  - **Yes**: 9 examples
  - **No**: 5 examples
  - Info([9,5]) = entropy(5/14, 9/14) =

    $= -\ 5/14 * \log_2(5/14) - 9/14 * \log_2(9/14) =$ **0.940**

InfoGain("outlook")

Outlook	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
Sunny	2	3	2/5	3/5	**0.971**	5/14
Overcast	4	0	4/4	0/4	**0**	4/14
Rainy	3	2	3/5	2/5	**0.971**	5/14



Entropy:

Sunny: Info([2,3]) = entropy(2/5,3/5) = $-2/5 * \log_2(2/5) - 3/5 * \log_2(3/5)$ = **0.971**

Overcast: Info([4,0]) = entropy(1,0) = $-1 * \log_2(1) - 0 * \log_2(0)$ = **0**

Rainy: Info([3,2]) = entropy(3/5,2/5) = $-3/5 * \log_2(3/5) - 2/5 * \log_2(2/5)$ = **0.971**

Info("outlook") =

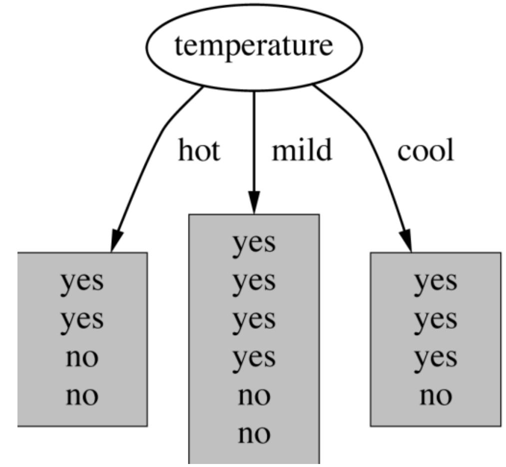
Info([2,3],[4,0],[3,2]) = (5/14) * 0.971 + (4/14) * 0 + (5/14) * 0.971 = **0.693**

InfoGain("outlook") = IBS – Info("outlook")

InfoGain("outlook") = 0.940 – 0.693 = 0.247

InfoGain("temperature")

Temperature	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
Hot	2	2	2/4	2/4	**1**	4/14
Mild	4	2	4/6	2/6	**0.918**	6/14
Cool	3	1	3/4	1/4	**0.811**	4/14



Entropy:

Hot: Info([2,2]) = entropy(2/4,2/4) = – 2/4 * $\log_2(2/4)$ – 2/4 * $\log_2(2/4)$ = **1**

Mild: Info([4,2]) = entropy(4/6,2/6) = – 4/6 * $\log_2(4/6)$ – 2/6 * $\log_2(2/6)$ = **0.918**

Cool: Info([3,1]) = entropy(3/4,1/4) = – 3/4 * $\log_2(3/4)$ – 1/4 * $\log_2(1/4)$ = **0.811**

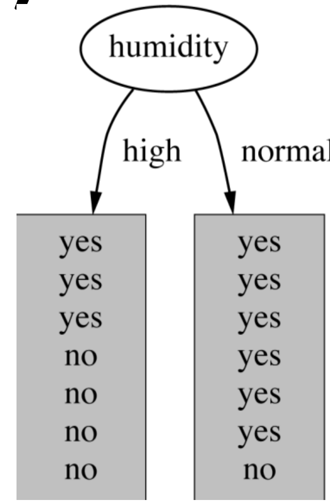
Info("temperature") =

Info([2,2],[4,2],[3,1]) = (4/14) * 1 + (6/14) * 0.918 + (4/14) * 0.811 = **0.911**

InfoGain("temperature") = 0.940 – 0.911 = 0.029

InfoGain("humidity")

Humidity	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
High	3	4	3/7	4/7	**0.985**	7/14
Normal	6	1	6/7	1/7	**0.592**	7/14



Entropy:

High: Info([3,4]) = entropy(3/7,4/7) = – 3/7 * $\log_2(3/7)$ – 4/7 * $\log_2(4/7)$ = **0.985**

Normal: Info([6,1]) = entropy(6/7,1/7) = – 6/7 * $\log_2(6/7)$ – 1/7 * $\log_2(1/7)$ = **0.592**

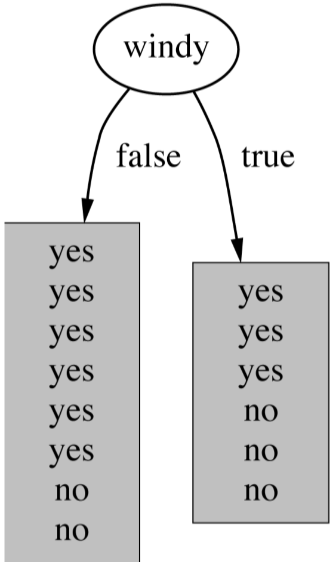
Info("humidity") =

Info([3,4],[6,1]) = (7/14) * 0.985 + (7/14) * 0.592 = **0.789**

InfoGain("humidity") = 0.940 – 0.789 = 0.151

InfoGain("windy")

Windy	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
True	6	2	6/8	2/8	**0.811**	8/14
False	3	3	3/6	3/6	**1**	6/14



Entropy:

True: Info([6,2]) = entropy(6/8,2/8) = $-$ 6/8 * $\log_2(6/8)$ $-$ 2/8 * $\log_2(2/8)$ = **0.811**

False: Info([3,3]) = entropy(3/6,3/6) = $-$ 3/6 * $\log_2(3/6)$ $-$ 3/6 * $\log_2(3/6)$ = **1**

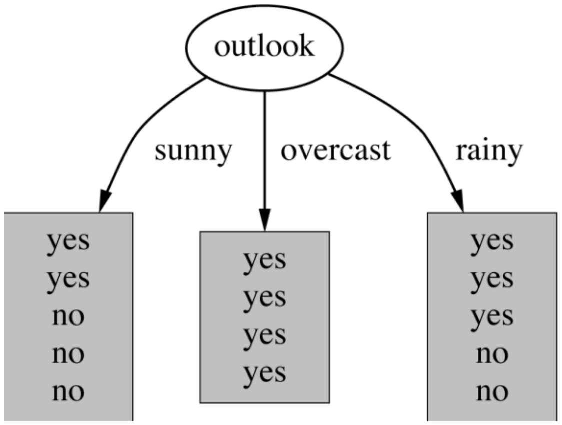
Info("windy") =

Info([6,2],[3,3]) = (8/14) * 0.811 + (6/14) * 1 = **0.892**

InfoGain("windy") = 0.940 – 0.892 = 0.048

Which attribute is "best"?

- The one with the highest information gain (**InfoGain**):
  - **Outlook: 0.247 bits**
  - Temperature: 0.029 bits
  - Humidity: 0.152 bits
  - Wind: 0.048 bits



Step 2 = recursion

Outlook = Sunny

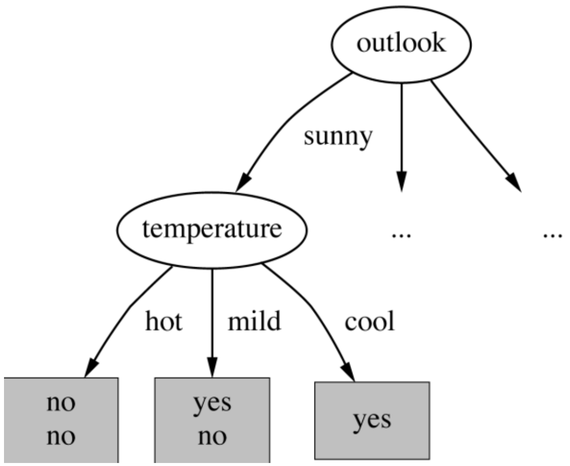
Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Next level in the tree ...

Outlook = Sunny

Attribute **Temperature**

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Sunny	Mild	Normal	True	Yes



Temperature	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
Hot	0	2	0/2	2/2	**0**	2/5
Mild	1	1	1/2	1/2	**1**	2/5
Cool	1	0	1/1	0/1	**0**	1/5

Hot: info([0,2]) = entropy(0,1) = **0**

Mild: info([1,1]) = entropy(1/2,1/2) = **1**

Cool: info([1,0]) = entropy(1,0) = **0**

Info("Temperature") = info([0,2],[1,1,],[1,0]) = 2/5*0 + 2/5*1 + 1/5*0 = **0.4**

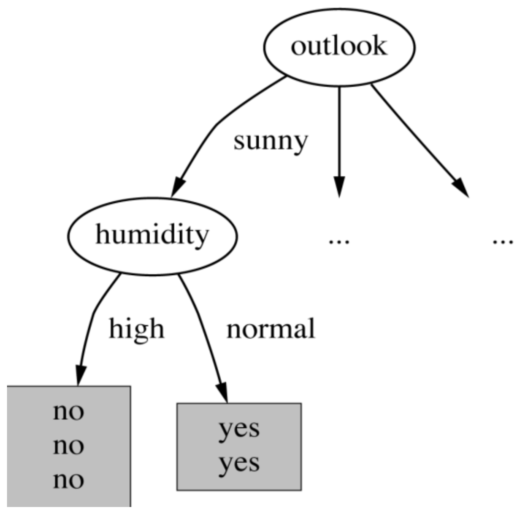
InfoGain("Temperature") = info([2,3]) – info([0,2],[1,1,],[1,0]) = 0.971 – 0.4 = **0.571**

Next level in the tree …

Outlook = Sunny

Attribute **Humidity**

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Sunny	Mild	Normal	True	Yes



Humidity	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
High	0	3	0/3	3/3	**0**	3/5
Normal	2	0	2/2	0/2	**0**	2/5

High: info([0,3]) = entropy(0,1) = **0**

Normal: info([2,0]) = entropy(1,0) = **0**

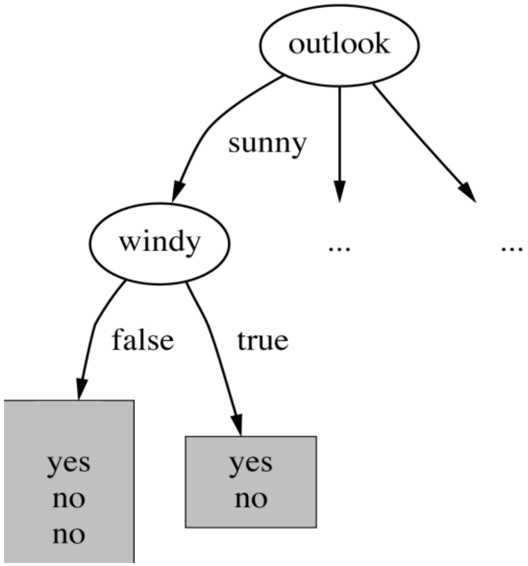
Info("Humidity") = info([0,3],[2,0]) = 2/5*0 + 3/5*0 = **0**

InfoGain("Humidity") = info([2,3]) – info([0,3],[2,0]) = 0.971 – 0 = **0.971**

Next level in the tree …

Outlook = Sunny
Attribute **Windy**

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Sunny	Mild	Normal	True	Yes



Windy	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
True	1	2	1/3	2/3	**0.918**	3/5
False	1	1	1/2	1/2	**1**	2/5

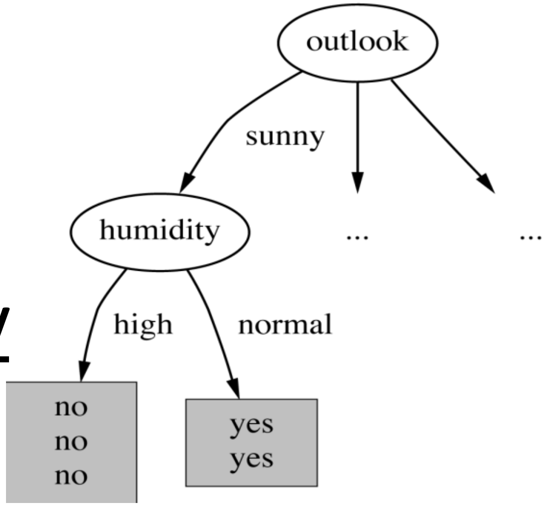
True: info([1,2]) = entropy(1/3,2/3) = $-1/3*\log_2(1/3) - 2/3*\log_2(2/3)$ = **0.918**
False: info([1,1]) = entropy(1/2,1/2) = $-1/2*\log_2(1/2) - 1/2*\log_2(1/2)$ = **1**

Info("Windy") = info([1,2],[1,1]) = 3/5* 0.918 + 2/5*1 = **0.951**

InfoGain("Windy") = info([2,3]) – info([1,2],[1,1,]) = 0.971 – 0.951 = **0.02**

Choosing the "best" attribute

- Information gain (**InfoGain**):
  - Temperature: 0.571
  - **Humidity: 0.971**
  - Windy: 0.02
- Choose the attribute Humidity
  - The leaves are pure → no need to continue the procedure



Step 2 = recursion

Outlook = Overcast

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Next level in the tree …

Outlook = Overcast

Outlook	**Temp**	**Humidity**	**Windy**	**Play**
Overcast	Hot	High	False	**Yes**
Overcast	Cool	Normal	True	**Yes**
Overcast	Mild	High	True	**Yes**
Overcast	Hot	Normal	False	**Yes**

- No need to further split ← the node is "pure" (contains just examples of class "**Yes**")

Step 2 = recursion

Outlook = Rainy

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Next level in the tree …

Outlook = Rainy

Outlook	Temperature	Humidity	Windy	Play
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Rainy	Mild	Normal	False	Yes
Rainy	Mild	High	True	No

Temperature	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
Mild	2	1	2/3	1/3	**0.918**	3/5
Cool	1	1	1/2	1/2	**1**	2/5

Mild: info([1,2]) = entropy(1/3,2/3) = $-1/3*\log_2(1/3) - 2/3*\log_2(2/3)$ = **0.918**
Cool: info([1,1]) = entropy(1/2,1/2) = $-1/2*\log_2(1/2) - 1/2*\log_2(1/2)$ = **1**

Info("Temperature") = info([2,1],[1,1]) = 3/5*0.918 + 2/5*1 = **0.951**

InfoGain("Temperature") = info([2,3]) – info([1,2],[1,1,]) = 0.971 – 0.951 = **0.02**

Next level in the tree …

Outlook = Rainy

Outlook	Temp	Humidity	Windy	Play
Rainy	Mild	High	False	**Yes**
Rainy	Cool	Normal	False	**Yes**
Rainy	Cool	Normal	True	**No**
Rainy	Mild	Normal	False	**Yes**
Rainy	Mild	High	True	**No**

Humidity	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
High	1	1	1/2	1/2	**1**	2/5
Normal	2	1	2/3	1/3	**0.918**	3/5

High: info([1,1]) = entropy(1/2,1/2) = $-1/2*\log_2(1/2) - 1/2*\log_2(1/2)$ = **1**

Normal: info([1,2]) = entropy(1/3,2/3) = $-1/3*\log_2(1/3) - 2/3*\log_2(2/3)$ = **0.918**

Info("Humidity") = info([2,1],[1,1]) = 3/5*0.918 + 2/5*1 = **0.951**

InfoGain("Humidity") = info([2,3]) – info([1,2],[1,1,]) = 0.971 – 0.951 = **0.02**

Next level in the tree …

Outlook = Rainy

Outlook	Temp	Humidity	Windy	Play
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Rainy	Mild	Normal	False	Yes
Rainy	Mild	High	True	No

Windy	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
True	0	2	0/2	2/2	**0**	2/5
False	3	0	3/3	0/3	**0**	3/5

High: info([0,2]) = **0**
Normal: info([3,0]) = **0**

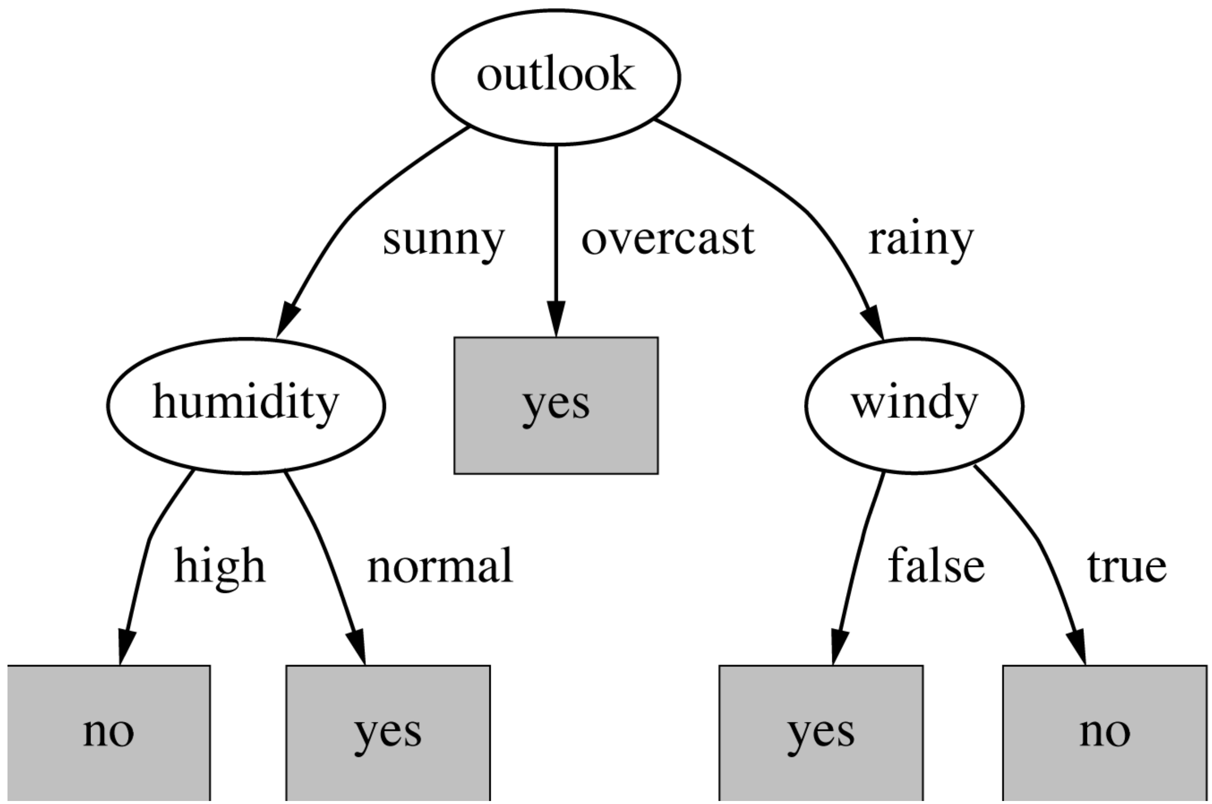
Info ("Windy") = info([2,1],[1,1]) = 3/5*0 + 2/5*0 = **0**

InfoGain("Windy") = info([2,3]) – info([1,2],[1,1,]) = 0.971 – 0 = **0.971**

Choosing the "best" attribute

- Informatiom gain (**InfoGain**):
  - Temperature: 0.02
  - Humidity: 0.02
  - **Windy: 0.971**
- Choose the attribute <u>Windy</u>
  - The leaves are pure →
    no need to continue the procedure

The "final" decision tree

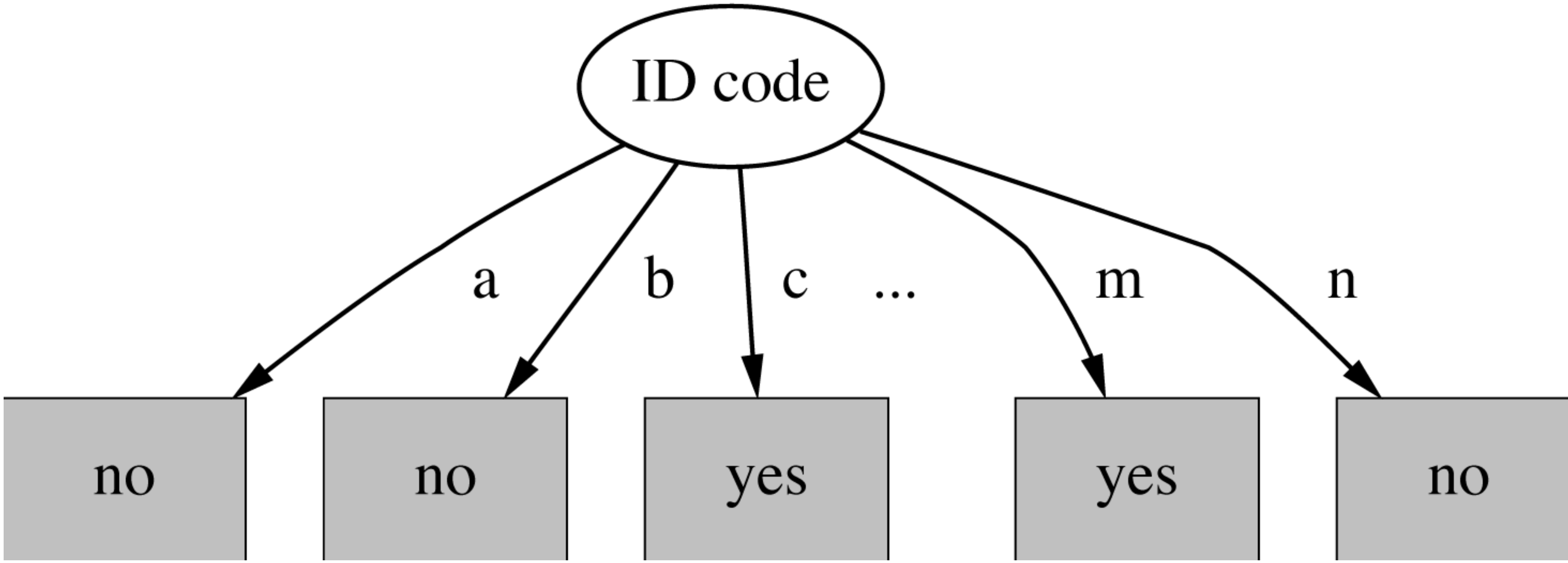


Information gain: problems

- If an attribute has many values, there is a higher probability that the subsets will be "pure";

- Consequence → InfoGain "perfers" attributes with larger number of values.

Example: attribute "ID code"



- The entropy of the split = **0**
  (each leaf is "pure" containing just a single example);

- Information gain of attribute "ID code" is maximal
  = extreme case

Information gain ratio (GainRatio)

- Intrinsic/split information = entropy of the split

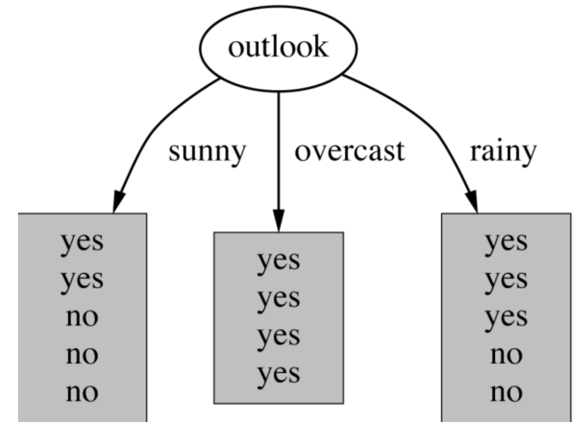
$$IntrinsicInfo(S,A) = -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}.$$

- Information gain ratio (GainRatio):

$$\boldsymbol{GainRatio(S,A) = \frac{InfoGain(S,A)}{IntrinsicInfo(S,A)}}.$$

GainRatio("outlook")

Outlook	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
Sunny	2	3	2/5	3/5	**0.971**	5/14
Overcast	4	0	4/4	0/4	**0**	4/14
Rainy	3	2	3/5	2/5	**0.971**	5/14



InfoGain("outlook") = 0.940 – 0.693 = **0.247**

SplitInfo("outlook") = info([5,4,5]) = entropy(5/14, 4/14, 5/14) =

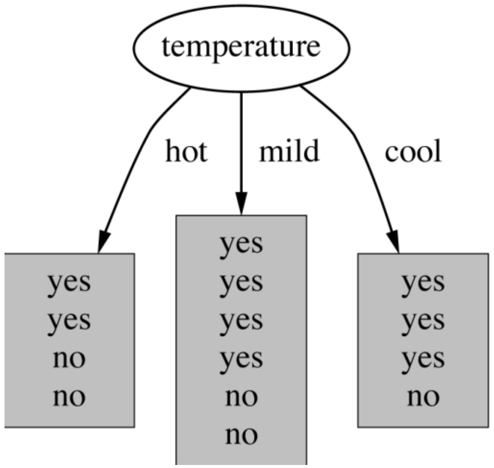
$= – 5/14*\log_2(5/14) – 4/14*\log_2(4/14) – 5/14*\log_2(5/14) = \mathbf{1.577}$

GainRatio("outlook") = InfoGain("outlook") / SplitInfo("outlook") =

= 0.247 / 1.577 = **0.156**

GainRatio("temperature")

Temperature	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
Hot	2	2	2/4	2/4	**1**	4/14
Mild	4	2	4/6	2/6	**0.918**	6/14
Cool	3	1	3/4	1/4	**0.811**	4/14



InfoGain("temperature") = 0.940 – 0.911 = **0.029**

SplitInfo("temperature") = info([4,6,4]) = entropy(4/14, 6/14, 4/14) =

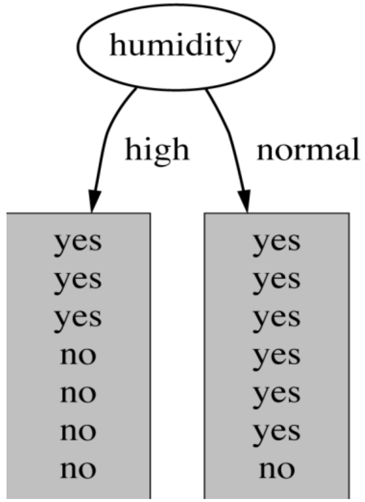
= – 4/14*$\log_2$(4/14) – 6/14*$\log_2$(6/14) – 4/14*$\log_2$(4/14) = **1.362**

GainRatio("temperature") = InfoGain("temperature") / SplitInfo("temperature") =

= 0.029 / 1.362 = **0.021**

GainRatio("humidity")

Humidity	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
Hot	3	4	3/7	4/7	**0.985**	7/14
Mild	6	1	6/7	1/7	**0.592**	7/14



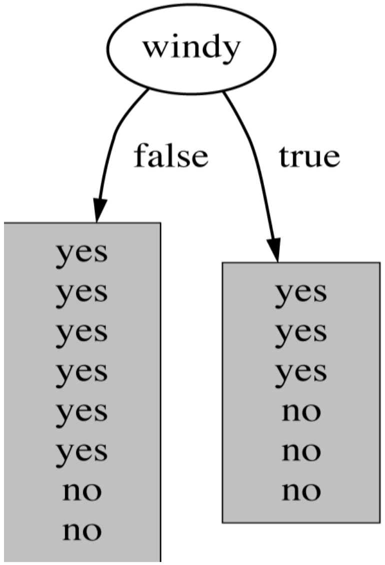
InfoGain("humidity") = 0.940 – 0.789 = **0.151**

SplitInfo("humidity") = info([7,7]) = entropy(7/14, 7/14) =
= – 7/14*$\log_2$(7/14) – 7/14*$\log_2$(7/14) = **1**

GainRatio("humidity") = InfoGain("humidity") / SplitInfo("humidity") =
= 0.151 / 1 = **0.151**

GainRatio("windy")

Windy	Yes	No	P(Yes)	P(No)	Entropy (bits)	Probability
True	6	2	6/8	2/8	**0.811**	8/14
False	3	3	3/3	3/3	**1**	6/14



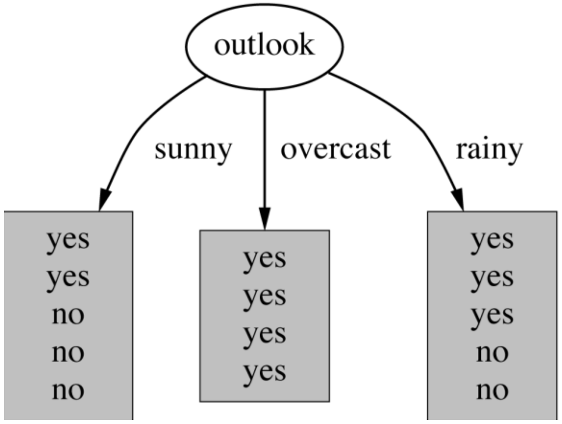
InfoGain("windy") = 0.940 – 0.892 = **0.048**

SplitInfo("windy") = info([8,6]) = entropy(8/14, 6/14) =
= – $8/14*\log_2(8/14) - 6/14*\log_2(6/14)$ = **0.985**

GainRatio("windy") = InfoGain("windy") / SplitInfo("windy") =
= 0.048 / 0.985 = **0.049**

Choosing the "best" attribute

- Information gain ratio (GainRatio):
  - **Outlook: 0.156 bits**
  - Temperature: 0.021 bits
  - Humidity: 0.152 bits
  - Windy: 0.049 bits



Gini index – "before split"

- If a set ***T*** contains examples with ***n*** classes, ***gini(T)*** is defined as:

$$gini(T) = 1 - \sum_{j=1}^{n} {p_j}^2$$

- where p_j is the relative frequency (probability) of class ***j*** in ***T***.

Gini index – "after split"

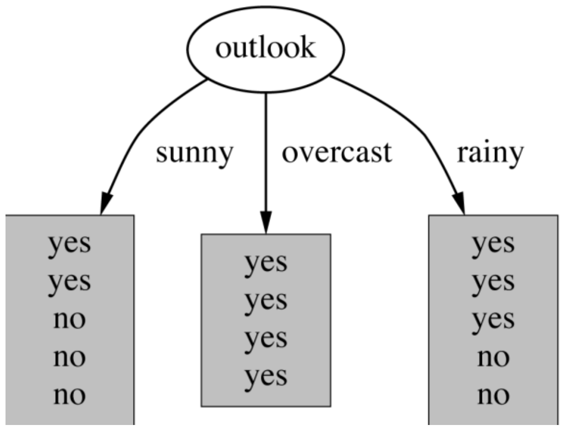
- We split the set ***T*** containing ***N*** examples into subsets $T_1, T_2, \ldots, T_k$ containing $N_1, N_2, \ldots, N_k$ examples, respectively. The information of this split is defined as:

$$gini_{split}(T) = \sum_{i=1}^{k} \frac{N_i}{N} gini(T_i)$$

- The attribute with the lowest $gini_{split}(T)$ is chosen as the "best" attribute.

Gini("outlook")

Outlook	Yes	No	P(Yes)	P(No)	Gini	Probability
Sunny	2	3	2/5	3/5	**0.48**	5/14
Overcast	4	0	4/4	0/4	**0**	4/14
Rainy	3	2	3/5	2/5	**0.48**	5/14



Gini index:

Sunny: Gini([2/5,3/5]) = $1 - ((2/5)^2 + (3/5)^2)$ = **0.48**

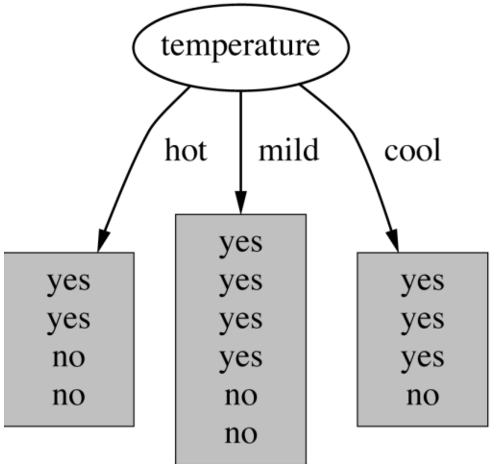
Overcast: Gini([4/4,0/4]) = $1 - ((4/4)^2 + (0/4)^2)$ = **0**

Rainy: Gini([3/5,2/5]) = $1 - ((3/5)^2 + (2/5)^2)$ = **0.48**

Gini("outlook") = (5/14)*0.48 + (4/14)*0 + (5/14)*0.48 = **0.342**

Gini("temperature")

Temperature	Yes	No	P(Yes)	P(No)	Gini	Probability
Hot	2	2	2/4	2/4	**0.5**	4/14
Mild	4	2	4/6	2/6	**0.444**	6/14
Cool	3	1	3/4	1/4	**0.375**	4/14



Gini index:

Hot: Gini(2/4,2/4) = $1 - ((2/4)^2 + (2/4)^2)$ = **0.5**

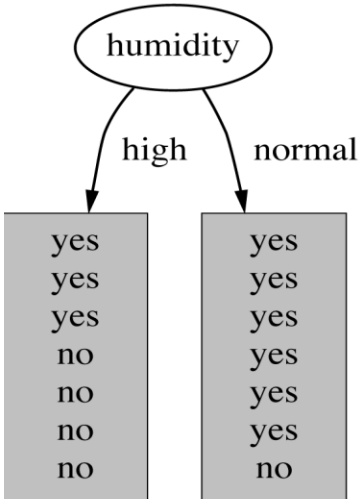
Mild: Gini(4/6,2/6) = $1 - ((4/6)^2 + (2/6)^2)$ = **0.444**

Cool: Gini(3/4,1/4) = $1 - ((3/4)^2 + (1/4)^2)$ = **0.375**

Gini("temperature") = (4/14)*0.5 + (6/14)*0.444 + (4/14)*0.375 = **0.440**

Gini("humidity")

Humidity	Yes	No	P(Yes)	P(No)	Gini	Probability
High	3	4	3/7	4/7	**0.490**	7/14
Normal	6	1	6/7	1/7	**0.245**	7/14



Gini index:

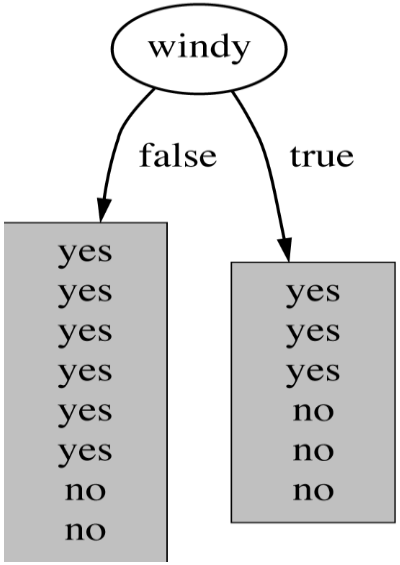
High: Gini(3/7,4/7) = 1 – $((3/7)^2 + (4/7)^2)$ = **0.490**

Normal: Gini(6/7,1/7) = 1 – $((6/7)^2 + (1/7)^2)$ = **0.245**

Gini("humidity") = (7/14)*0.490 + (7/14)*0.245 = **0.368**

Gini("windy")

Windy	Yes	No	P(Yes)	P(No)	Gini	Probability
True	6	2	6/8	3/8	**0.375**	8/14
False	3	3	3/3	3/3	**0.5**	6/14



Gini index:

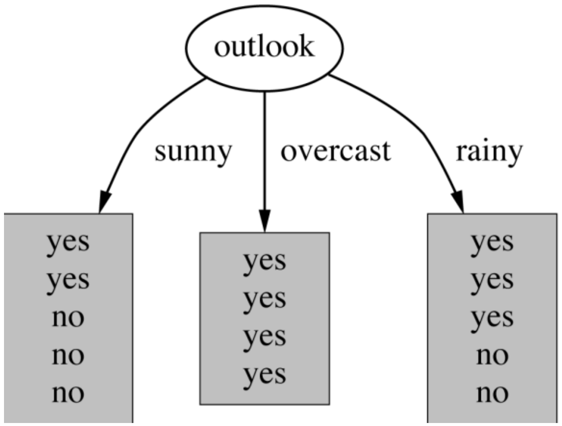
True: Gini(6/8,2/8) = $1 - ((6/8)^2 + (2/8)^2)$ = **0.375**

False: Gini(3/6,3/6) = $1 - ((3/6)^2 + (3/6)^2)$ = **0.5**

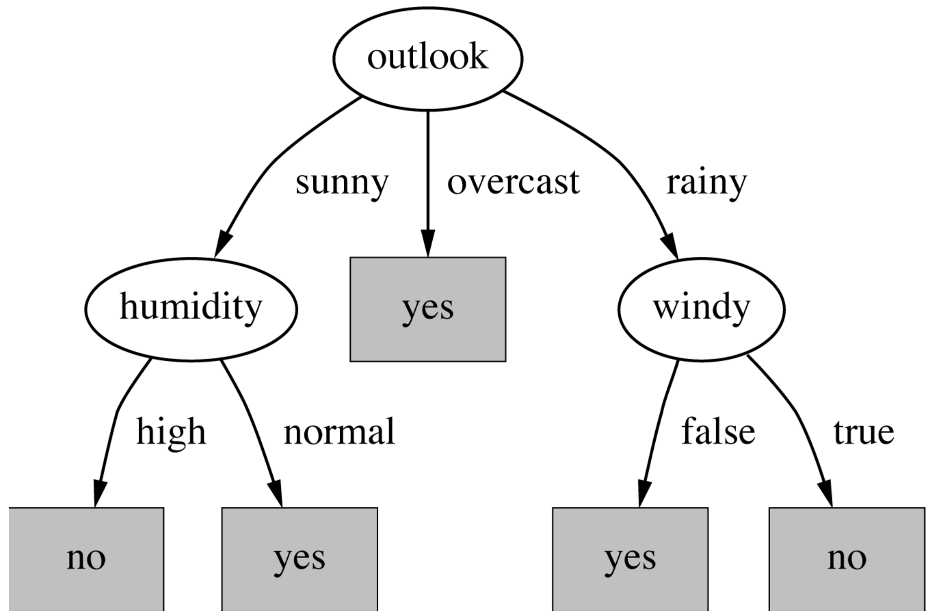
Gini("windy") = (8/14)*0.375 + (6/14)*0.5 = **0.429**

Choosing the "best" attribute

- Gini index:
  - **Outlook: 0.342**
  - Temperature: 0.440
  - Humidity: 0.368
  - Windy: 0.429



How to classify new examples?



Outlook	Temperature	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Sunny	Cool	High	True	No
Rainy	Mild	High	False	Yes

And for slightly different data?

I	D	A	B	E	F	C
438	**12.03.2040**	5	3.49	14	good	**y**
450	24.04.1934	3	58.48	32	bad	**z**
461	05.01.1989	5	47.23	12	bad	**y**
466	07.08.1945	1	31.40	21	good	**y**
467	**21.07.2028**	5	79.60	20	bad	**y**
469	30.04.1966	3	19.88	3	bad	**w**
485	28.02.2015	5	59.13	4	bad	**w**
514	19.03.2033	3	27.05	2	bad	**x**
522	13.03.2022	2	80.14	16	good	**y**
529	28.07.2037	4	65.02	20	bad	**z**
534	05.10.1986	2	99.17	13	good	**z**

InfoGain("A")

A	w	x	y	z	P(w)	P(x)	P(y)	P(z)	entropy	probabilities
1	0	0	**1**	0	0/1	0/1	1/1	0/1	0	1/11
2	0	0	**1**	1	0/2	0/2	1/2	1/2	1	2/11
3	**1**	1	0	1	1/3	1/3	0/3	1/3	1.585	3/11
4	0	0	0	**1**	0/1	0/1	0/1	1/1	0	1/11
5	1	0	**3**	0	1/4	0/4	3/4	0/4	0.811	4/11

IBS = info([2,1,5,3]) = entropy(2/11,1/11,5/11,3/11) =
$= -2/11*\log_2(2/11) - 1/11*\log_2(1/11) - 5/11*\log_2(5/11) - 3/11*\log_2(3/11) =$ **1.79**

1: info([0,0,1,0]) = entropy(0,0,1,0) $= -3*0/1*\log_2(0/1) - 1/1*\log_2(1/1)$ = **0**
2: info([0,0,1,1]) = entropy(0,0,1/2,1/2) $= -2*0/2*\log_2(0/2) - 2*1/2*\log_2(1/2)$ = **1**
3: info([1,1,0,1]) = entropy(1/3,1/3,0,1/3) $= -0/3*\log_2(0/3) - 3*1/3*\log_2(1/3)$ = **1.585**
4: info([0,0,0,1]) = entropy(0,0,0,1) $= -3*0/1*\log_2(0/1) - 1/1*\log_2(1/1)$ = **0**
5: info([1,0,3,0]) = entropy(1/4,0,3/4,0) $= -2*0/4*\log_2(0/4) - 1/4*\log_2(1/4) - 3/4*\log_2(3/4)$ = **0.811**

Info("A") = 1/11*0 + 2/11*1 + 3/11*1.585 + 1/11*0 + 4/11*0.811 = **0.909**

InfoGain("A") = 1.79 – 0.909 = **0.881**

The "rest" … for homework ☺