DECISION TREE

Dr. Hilman F. Pardede

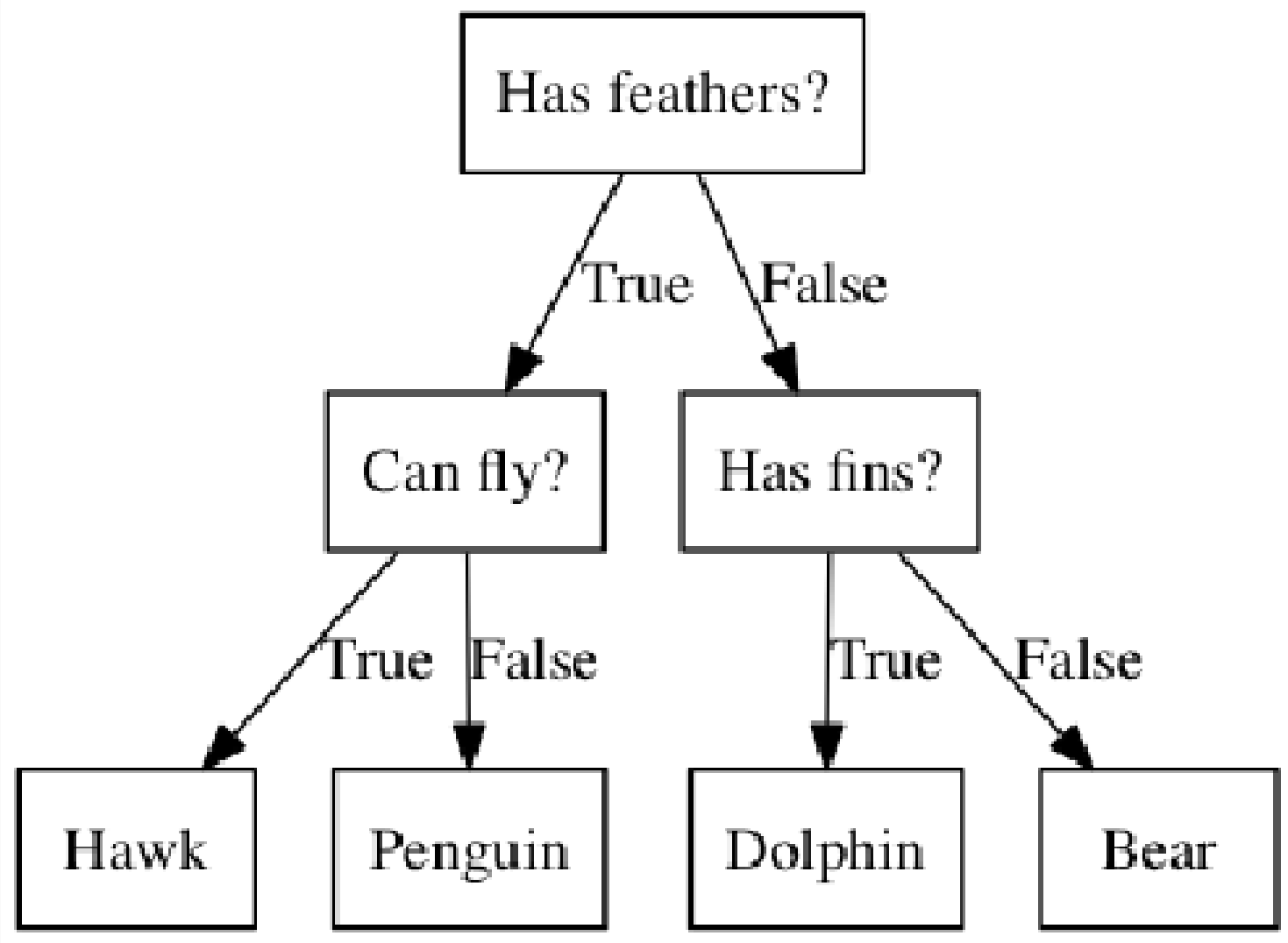
Research Center for Informatics
Indonesian Institute of Sciences

The materials are compiled from the following resources:

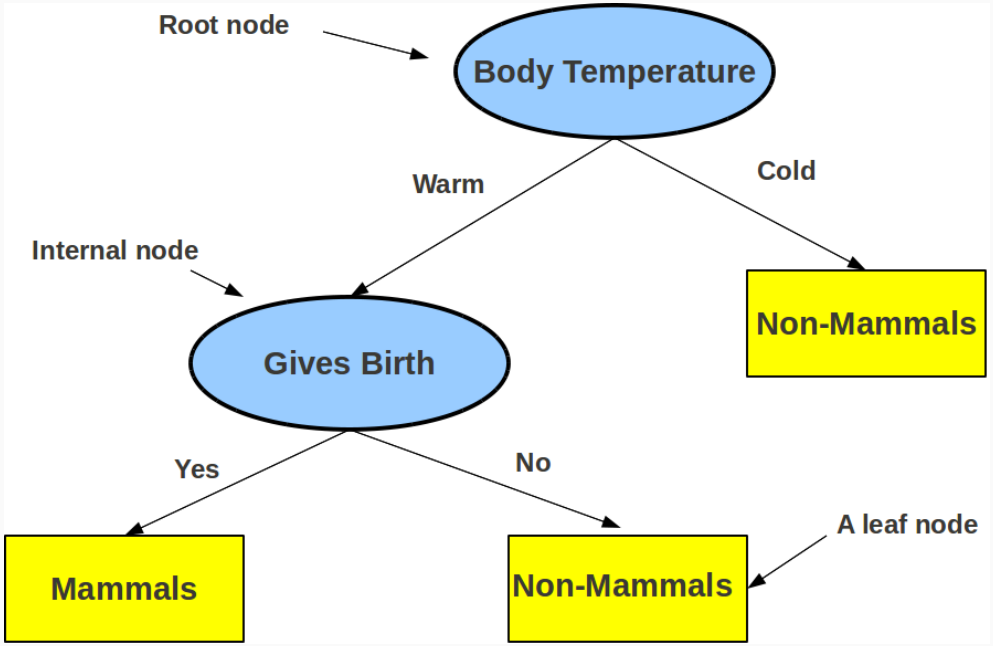
- https://github.com/joaquinvanschoren/ML-course
- https://www.cse.iitk.ac.in/users/piyush/courses/ml_autumn16/ML.html
- http://sli.ics.uci.edu/Classes/2015W-273a

DECISION TREE BASICS

DECISION TREE



- Defined by a hierarchy of rules (in form of a tree)



- Rules form the internal nodes of the tree (topmost internal node = root)
- Each internal node tests the value of some feature and "splits" data across the outgoing branches
- Note: The tree need not be a binary tree
- (Labeled) Training data is used to construct the Decision Tree (DT)
- The DT can then be used to predict label y of a test example x

BUILDING DECISION TREES

- Split the data in two (or more) parts
- Search over all possible splits and choose the one that is most informative
- Repeat recursive partitioning until stopping conditions are satisfied

DECISION TREE: AN EXAMPLE

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

- Deciding whether to play or not to play Tennis on a Saturday

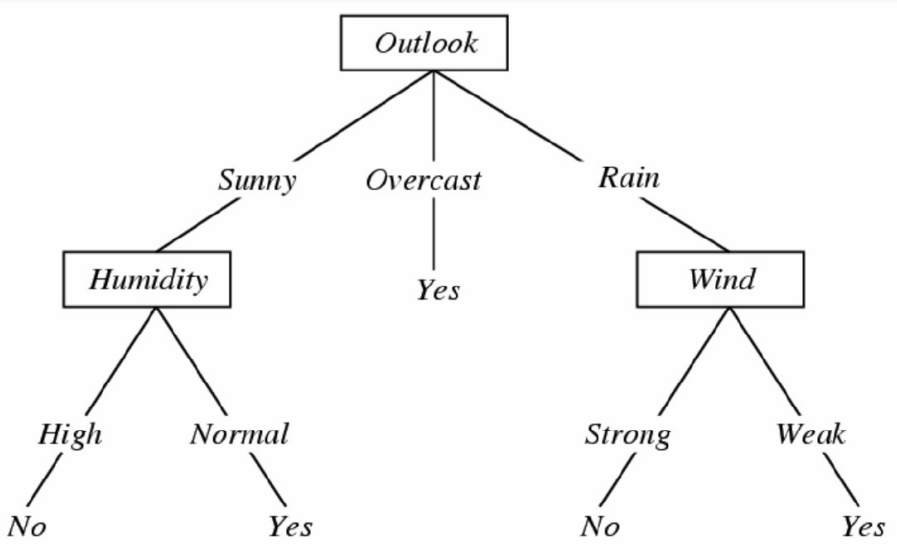
DECISION TREE: AN EXAMPLE

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

- Deciding whether to play or not to play Tennis on a Saturday
- If we would like to make a rule, which attribute to be the first we split? Why

DECISION TREE: AN EXAMPLE

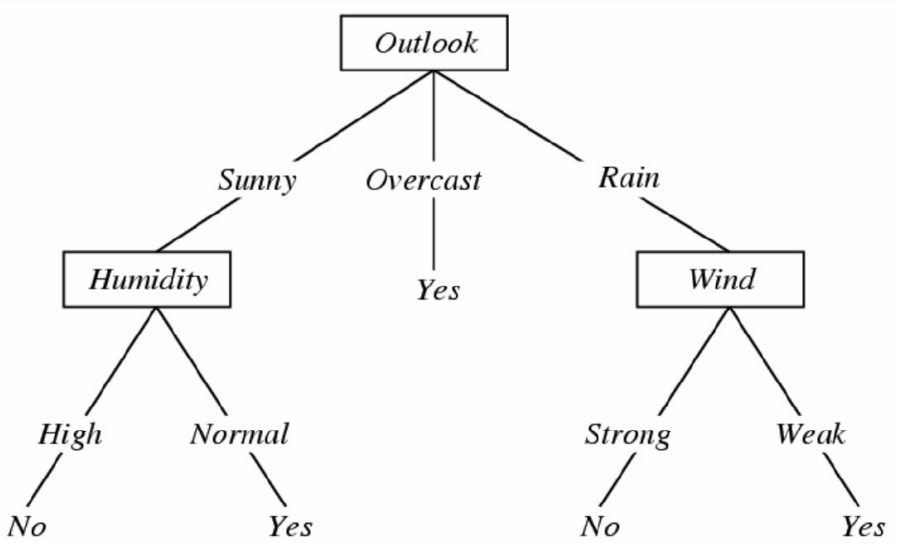
day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



- Deciding whether to play or not to play Tennis on a Saturday
- If we would like to make a rule, which attribute to be the first we split? Why

DECISION TREE: AN EXAMPLE

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



- Deciding whether to play or not to play Tennis on a Saturday
- If we would like to make a rule, which attribute to be the first we split? Why
- How to quantify "the informativeness"

LEARNING IN DT

- Entropy is a measure of randomness/uncertainty of a set

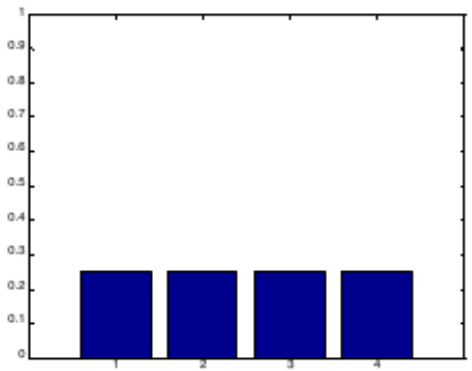
$$E(X) = -\sum_{k=1}^{K} p_k(x) \log_2 p_k(x) \tag{1}$$

with p_k = the relative frequency of class k in the leaf node

- Entropy is a measure of randomness/uncertainty of a set

$$E(X) = -\sum_{k=1}^{K} p_k(x) \log_2 p_k(x) \tag{1}$$

with p_k = the relative frequency of class k in the leaf node

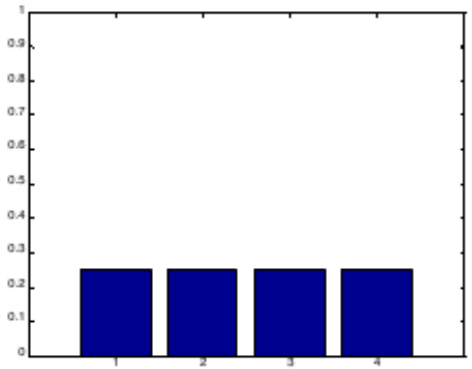


H(x) = .25 log 4 + .25 log 4 + .25 log 4 + .25 log 4 = log 4 = 2 bits

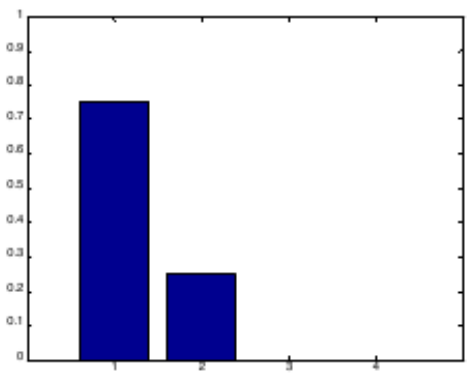
- Entropy is a measure of randomness/uncertainty of a set

$$E(X) = -\sum_{k=1}^{K} p_k(x) \log_2 p_k(x) \tag{1}$$

with p_k = the relative frequency of class k in the leaf node



H(x) = .25 log 4 + .25 log 4 + .25 log 4 + .25 log 4 = log 4 = 2 bits

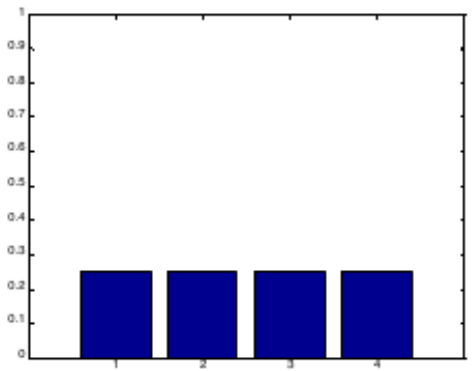


H(x) = .75 log 4/3 + .25 log 4 ≈ .8133 bits

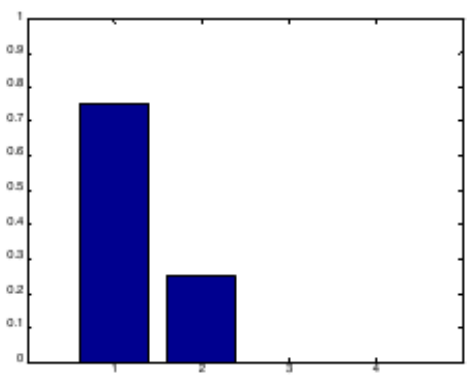
- Entropy is a measure of randomness/uncertainty of a set

$$E(X) = -\sum_{k=1}^{K} p_k(x) \log_2 p_k(x) \tag{1}$$

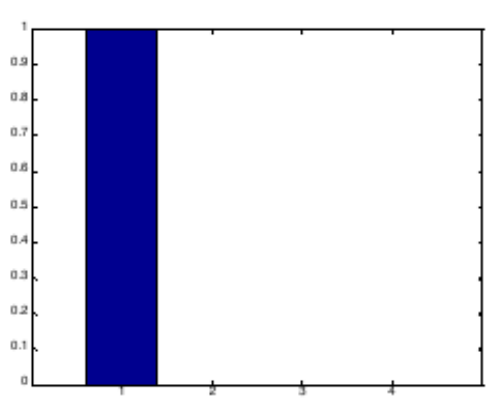
with p_k = the relative frequency of class k in the leaf node



H(x) = .25 log 4 + .25 log 4 + .25 log 4 + .25 log 4 = log 4 = 2 bits



H(x) = .75 log 4/3 + .25 log 4 ≈ .8133 bits



H(x) = 1 log 1 = 0 bits

-
- We can assess informativeness of each feature by looking at how much it reduces the entropy of the class distribution
- Information Gain (IG) on knowing the value of some feature V

$$G(X, V) = E(X) - \sum_{v=1}^{V} \frac{|X_v|}{|X|} E(X_v) \tag{2}$$

with p_k = the relative frequency of class k in the leaf node, X = the training set, X_v denotes the subset of elements of X for which feature V has value v

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

- Coming back to playing tennis..
- Let's begin with the root node of the DT and compute IG of each feature
- Consider feature "wind" $\in$ {weak,strong} and its IG w.r.t. the root note:

- Root node: S = [9+, 5−] (all training data: 9 play, 5 no-play)
- Entropy: $E(S) = (9/14)\log_2(9/14)(5/14)\log_2(5/14) = 0.94$
- $S_{weak} = [6+, 2] \Rightarrow E(S_weak) = 0.811$
- $S_{strong} = [3+, 3] \Rightarrow E(S_{strong}) = 1$

$$IG(S, wind) = E(S) - \frac{|S_{weak}|}{|S|}E(S_weak) - \frac{|S_{strong}|}{|S|}E(S_{strong})$$
$$= 0.948/140.8116/141$$
$$= 0.048$$

- At the root node, the information gains are:
- $IG(S, wind) = 0.048$ (we already saw)
- $IG(S, outlook) = 0.246$
- $IG(S, humidity) = 0.151$
- $IG(S, temperature) = 0.029$
- "outlook" has the maximum $IG \Rightarrow$ chosen as the root node
- Iterate - for each child node, select the feature with the highest IG
- Other criteria for judging feature informativeness: Gini-index, misclassification rate

AVOIDING OVERFITTING IN DT

OVERFITTING: CONTROLLING COMPLEXITY OF DECISION TREES

- Decision trees can very easily overfit the data. Regularization strategies:
- Pre-pruning: stop creation of new leafs at some point
  - Limiting the depth of the tree, or the number of leafs
  - Requiring a minimal leaf size (number of instances)
- Post-pruning: build full tree, then prune (join) leafs
  - Reduced error pruning: evaluate against held-out data
  - Many other strategies exist.
  - scikit-learn supports none of them (yet)

Some key strengths:

- Simple and each to interpret
- Do not make any assumption about distribution of data
- Easily handle different types of features (real, categorical/nominal, etc.)
- Very fast at test time (just need to check the features, starting the root node and following the DT until you reach a leaf node)
- Multiple DTs can be combined via ensemble methods (e.g., Decision/random Forest)
- Each DT can be constructed using a (random) small subset of features

Some key weaknesses:

- Can be unstable if some labeled examples are noisy
- Can sometimes become very complex unless some pruning is applied