Decision tree

Akito Sakurai

Today's topic

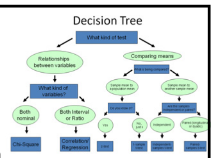
- How to construct a decision tree
  - A greedy-type algorithm
    - To select an attribute on node: information gain
      - Sometimes "information gain ratio" is in need
    - What is information
  - Can do regression (regression tree)
  - In R, tree and rpart are easy to use

skip

Contents

- Decision tree
  - what is it?
  - Construction of DT – flow, a node, greedy alg. –
  - How to select an attribute – categorical, numerical–
  - Entropy
  - Information gain and information gain ratio
  - Construction of DT, again
  - A problem –an attribute with many values–
  - Numerical attribute
  - GINI
  - Regression tree
  - R

Decision tree

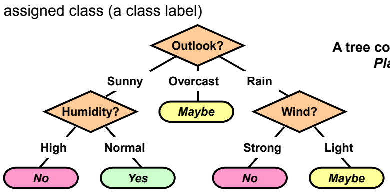


- DT is different from many others
  - The boundary defined by DT is not smooth
  - The purpose of construction is not clear
  - Construction alg. is not an OR-like
- But, DT is easy to understand by end users and the accuracy is not bad (sometimes performs very well).
- There were many ideas to improve DT.
  - Don't make light of it
- Constructed DT is easy to understand (therefore it is used quite often). But construction is a bit difficult

Decision Trees

- It is a classifier.
  - samples: a vector of attribute (feature) values
- Internal Nodes: will test attribute value
  - typical: equality test (e.g., "Wind = ?")
  - Some other tests such as inequality or combination of them are possible
- Branches: correspond to one (or a set of) attribute values; or True/False
  - One-to-one correspondence (e.g., "Wind = Strong", "Wind = Light")
- Leaves: assigned class (a class label)

For the moment, we consider only "one attribute value" case.



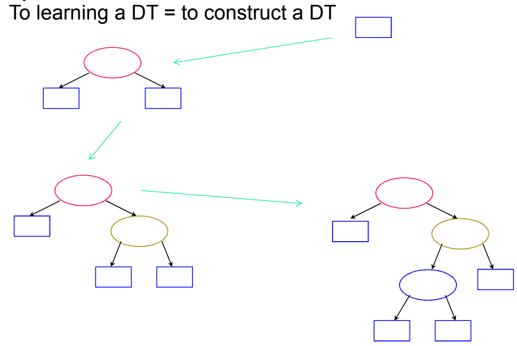
A tree corresponding to *PlayTennis*

Contents

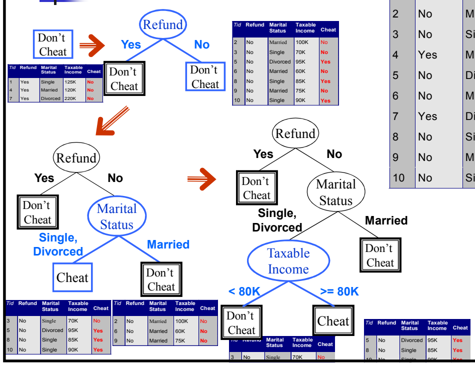
- Decision tree
  - what is it?
  - Construction of DT – flow, a node, greedy alg. –
  - How to select an attribute – categorical, numerical–
  - Entropy
  - Information gain and information gain ratio
  - Construction of DT, again
  - A problem –an attribute with many values–
  - Numerical attribute
  - GINI
  - Regression tree
  - R

DT construction – a flow –

To learning a DT = to construct a DT

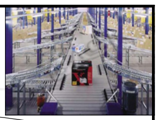


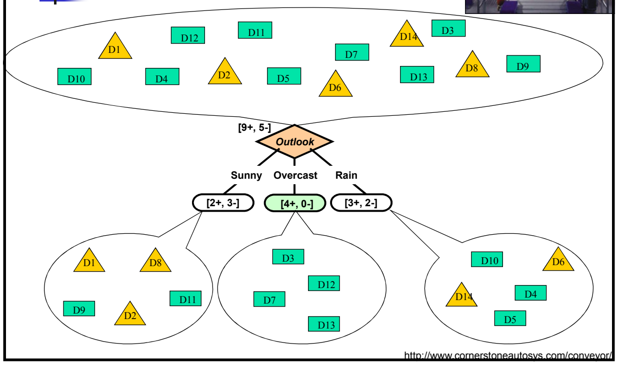
DT construction



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

DT construction – 1 node –





Contents

- Decision tree
  - what is it?
  - Construction of DT – flow, a node, greedy alg. –
  - How to select an attribute – categorical, numerical–
  - Entropy
  - Information gain and information gain ratio
  - Construction of DT, again
  - A problem –an attribute with many values–
  - Numerical attribute
  - GINI
  - Regression tree
  - R

DT construction – greedy algorithm

- Greedy policy.
  - In general: if decision is made, will not retract it
    - In a maze, will not backtrack. Once grab it, will not release it
    - The result is not optimal, but fast
  - Decision tree: split the training data in a way to optimize some criterion by using some attribute.
    - When braches are made by splitting, will never retract it

Sub-contents

- Problem
  - How to split the training data?
    - How to define attribute test method
    - How to define a best split
  - How to define when to stop

DT construction – Greedy algorithm –

- Greedy policy.
  - In general: if decision is made, will not retract it
    - In a maze, will not backtrack. Once grab it, will not release it
    - The result is not optimal, but fast
  - Decision tree: split the training data in a way to optimize some criterion by using some attribute.
    - When braches are made by splitting, will never retract it

Sub-contents

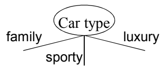
- Problem
  - How to split the training data
    - How to define attribute test method
    - How to define a best split
  - How to define when to stop

How to find a best attribute

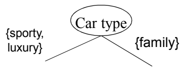
- May depend on attribute types
  - Categorical variables
  - Ordinal variables
  - Numerical variables
- May depend on the number of splits
  - 2 splits
  - More-than-two splits

Categorical variables

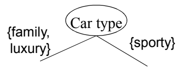
- More than 2: the number of values of the variable.



- 2: split the attribute values into 2 subsets.
  Need to find optimal split.



or



Numerical variables

- Height, weight, blood pressure, cholesterol,,,
  - Too many splits if we categorize it with multiples of unit
  - discretization: set thresholds to split the values
- A few ways
  - Treat it as ordinal attributes after discretization
    - static – discretize it only once at the start
    - dynamic – equal width, equal frequency (percentile), clustering
  - Binary discrimination: (A < v) or (A ≥ v)
    - By considering all splits and select the best one
    - May require much calculation

DT construction – Greedy algorithm –

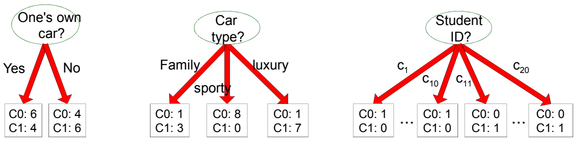
- Greedy policy.
  - In general: if decision is made, will not retract it
    - In a maze, will not backtrack. Once grab it, will not release it
    - The result is not optimal, but fast
  - Decision tree: split the training data in a way to optimize some criterion by using some attribute.
    - When braches are made by splitting, will never retract it

Sub-contnts

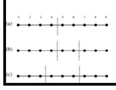
- Problem
  - How to split the training data
    - How to define attribute test method
    - How to define a best split
  - How to define when to stop

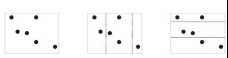
How to find a best split?

Before split: 10 samples in C0 (class 0),
10 samples in C1 (class 1)



Which is the best?



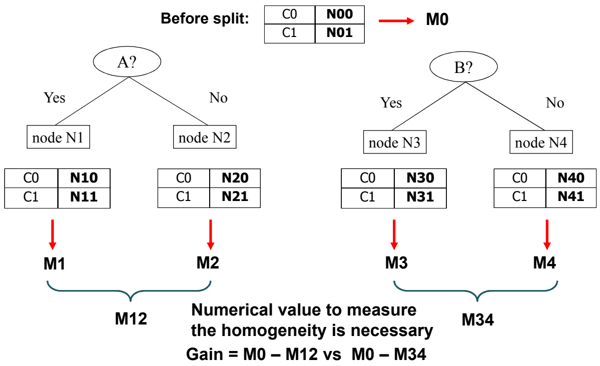


How to find a best split?

- Greedy policy:
  - A split which causes more homogeneity is better
  - If a class is dominant on a node, claiming the class without hesitation on the node is highly acceptable
- For just one node, simple calculation suffices:

C0: 5 C1: 5	C0: 9 C1: 1
heterogeneous, non-pure	**homogeneous, pure**
High non-purrity	**Low non-purity**

How to find a best split



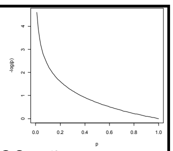
Measure for inpurity/purity

- Entropy
- Gini Index
- Error rate

Contents

- Decision tree
  - what is it?
  - Construction of DT – flow, a node, greedy alg. –
  - How to select an attribute – categorical, numerical–
  - Entropy
  - Information gain and information gain ratio
  - Construction of DT, again
  - A problem –an attribute with many values–
  - Numerical attribute
  - GINI
  - Regression tree
  - R

Entropy



- Average information content

$$H(p_1,\ldots,p_m) = -p_1\log_2 p_1 - \ldots - p_m \log_2 p_m$$
$$= p_1(-\log_2 p_1) + \ldots + p_m(-\log_2 p_m)$$

- (For comparison) average number of spots of a die when cast

$$+\, p_1 * 1 + p_2 * 2 + \ldots + p_6 * 6$$

- That is, if entropy is an average information content $-\log_2 p_i$ is the information content

Negative sign "–" is for changing the sign, because log p <0 always hold since p<0 and is cumbersome for manipulation.

Information content

Information content for an stochastic event, is a value to know that the event happens when no others do not know it

Suppose that the event is "the coin lands heads up" (its probability is 1/2)

Suppose a is the value to know that the event happens.

To know the two events "coin1 lands heads up" and "coin2 lands heads up" happen, is clearly $a + a = 2a$. (because we can imagine these two cases happen consecutively)

The probability that these two events happen is ½ * ½ =1/4 .

Suppose that the event is "the die shows one pip" (probability 1/6).

Suppose that b is the value of knowing the event happens.

Knowing the two events "die1 shows one" and "die2 shows one" happen, is clearly $b + b = 2b$ as above.

The probability that these two events happen is 1/6*1/6=1/36

Therefore, if the probability squared, the value to know the event happen doubles.

A function to represent information

The probability that the event happens is squared, then the value to know the event happens is doubled

The probability p that the event happens becomes p^2 , the value v becomes 2v.

Suppose that you are a predictor. You tell a prediction and get a reward. Again you tell a prediction and get a reward. The probability that your prediction is correct is their product; and the reward is the sum/

The probability becomes from p to p^2 , then the value v(p) should $v(p^2)$ = 2v(p).

The function that shows the charateristic is just the log function.

Let the base be 2 and change the sign to get positive numbers (or just the base set to be ½). The value knowing that the event with probability p happens is – log p

$$\mathrm{InformationContent}(p) = -\log_2 p = \log_{1/2} p$$

Possibly unfair coin

- Suppose that the probability to land heads up is p and heads down is 1-p.
- What is the value of knowing that the coin lands heads up or down?
- The value of knowing "heads up" is –log p, "heads down" –log(1– p) .
- Because the probability of "heads up" is p and "heads down" is 1–p, the average value is:

$$H(p,1-p) = p(-\log_2 p) + (1-p)(-\log_2 (1-p))$$
$$= -p\log_2 p - (1-p)\log_2 (1-p)$$

Possibly unfair die

- Suppose that the probability that "a die shows i" is p_i .
- What is the value to know that the event happens?
- Clearly the value to know the event is –log p_i .
- The average value of "information" is:

$$H(p_1, p_2, \ldots, p_6)$$
$$= p_1(-\log_2 p_1) + p_2(-\log_2 p_2) + \cdots + p_6(-\log_2 p_6)$$
$$= -p_1\log_2 p_1 - p_2\log_2 p_2 - \cdots - p_6\log_2 p_6$$

Contents

- Decision tree
  - what is it?
  - Construction of DT – flow, a node, greedy alg. –
  - How to select an attribute – categorical, numerical–
  - Entropy
  - Information gain and information gain ratio
  - Construction of DT, again
  - A problem –an attribute with many values–
  - Numerical attribute
  - GINI
  - Regression tree
  - R

Information content to measure impurity

- Impurity
  - Suppose that two kind of entities are mixed. Le the proportion be p and 1–p. Because the state that p=0 or p=1 is purest, impurity is 0 for the case. When p=1/2, it is the most impure and therefore the impurity is 1. Clearly the entropy function satisfies this condition.
  - How about the situation when n kinds of entities are mixed. Let the proportion be $p_1,\ldots p_n$ .When any of p_i is 1 whereas the others are 0 is the purest state. Conversely the state when all of the p_i are equal (i.e., are 1/n) the purity is the lowest. Clearly the entropy function satisfies this property.
  - How about the situation we have d datasets $\{D_1,..,D_d\}$ where each set have different number of entities and different impurities (entropies). What should be the average impurity (entropy)?
    Since the entropy is defined as average "information content," it means the entropy is per one entity.
    Therefore

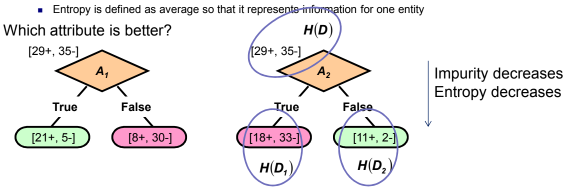
$$\textit{Information}(D) \equiv H(D) = \sum_{c=1}^{d}\left[-\frac{|D_c|}{|D|}\cdot\log\frac{|D_c|}{|D|}\right]$$

Information gain

- Definition
  - Information gain for a dataset D when an attribute A is used is, the amount of information that decreases when a split using A is selected:

$$\textit{InformationGain}(D,A) \equiv H(D) - \sum_{v\in values(A)}\left[\frac{|D_v|}{|D|}\cdot H(D_v)\right] = \frac{1}{|D|}\left(|D|\cdot H(D) - \sum_{v\in values(A)}|D_v|\cdot H(D_v)\right)$$

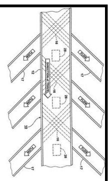
    where D_v is $\{x \in D \mid x.(A) = v\}$, i.e., a subset of D consisting of elements whose attribute value of A is v.
  - Note: as explained in the previous slide, entropy for a group of sets is defined to be not just sum of the entropies but a kind of size-adjusted sum of entropies
    - Entropy is defined as average so that it represents information for one entity
- Which attribute is better?



Contents

- Decision tree
  - what is it?
  - Construction of DT – flow, a node, greedy alg. –
  - How to select an attribute – categorical, numerical–
  - Entropy
  - Information gain and information gain ratio
  - Construction of DT, again
  - A problem –an attribute with many values–
  - Numerical attribute
  - GINI
  - Regression tree
  - R

Again: DT construction



- Standard procedure: from top to bottom (from root to leaves) and be greedy、 Recursive and divide-and-conquer
  - First: select one attribute and make a root. One branch and leaf is for one attribute value.
  - Then: split the training data for the node into subsets and assign it to each leaf (one subset for one leaf/branch)
  - Repeat: Apply the same procedure to the leaves. The training dataset is the one that has been split and assigned
  - Halt when all the elements in the assigned subsets have the same class labels.

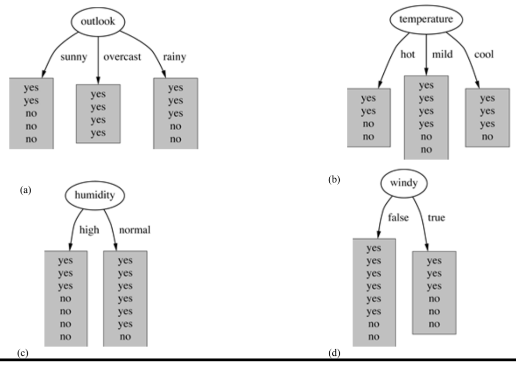
http://www.freepatentsonline.com/7086519.html

Play tennis or not (again)

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

From Tom Mitchell's "Machine Learning"

Which attribute is better?

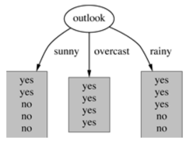


Calculation: "Outlook"

"Outlook" = "Sunny":
info([2,3]) = entropy(2/5,3/5) = – (2/5)log(2/5) – (3/5)log(3/5) = 0.971
"Outlook" = "Overcast":
info([4,0]) = entropy(1,0) = – 1 log(1) – 0 log(0) = 0
"Outlook" = "Rainy":
info([3,2]) = entropy(3/5/,2/5) = – (3/5)log(3/5) – (2/5)log(2/5) = 0.971

Information content when this attribute is used:
info([3,2],[4,0],[3,2]) = (5/14) × 0.971 + (4/14) × 0 + (5/14) × 0.971
= 0.693 bits

$$InformationGain(D,A) = H(D) - \sum_{v \in values(A)} \left[\frac{|D_v|}{|D|} \cdot H(D_v) \right] = \frac{1}{|D|}\left(|D| \cdot H(D) - \sum_{v \in values(A)} |D_v| \cdot H(D_v) \right)$$



http://www.space.com/4700-weather-outlook-improves-thursday-shuttle-launch.html

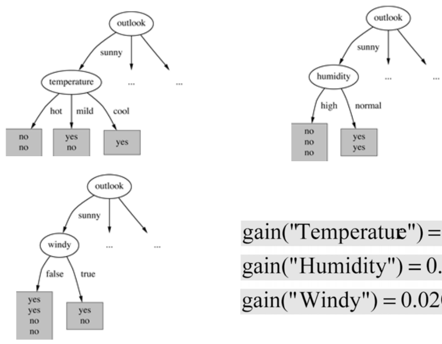
Calculation: Information gain

- We then calculate the information gain.

Information gain: information before the split – information after the split
gain("Outlook") = info([9,5]) – info([2,3],[4,0],[3,2]) = 0.940 – 0.693
= 0.247 bits
In a similar way, we get
gain("Outlook") = 0.247
gain("Temperature") = 0.029
gain("Humidity") = 0.152
gain("Windy") = 0.048

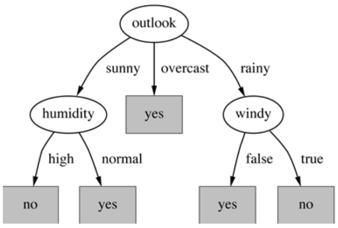
- The larger the information gain is, the purer it is. Therefore we select "Outlook."

Let us continue



$$\text{gain("Temperature")} = 0.571 \text{ bits}$$
$$\text{gain("Humidity")} = 0.971 \text{ bits}$$
$$\text{gain("Windy")} = 0.020 \text{ bits}$$

DT obtained



- Note: All the leaves are not necessary the purest, because there might be some noise (error of class labels).

⇒ if further split makes the tree worse, halt the calculation.

http://www.mochadad.com/2011/01/the-importance-of-family-goal-setting

Contents

- Decision tree
  - what is it?
  - Construction of DT – flow, a node, greedy alg. –
  - How to select an attribute – categorical, numerical–
  - Entropy
  - Information gain and information gain ratio
  - Construction of DT, again
  - A problem –an attribute with many values–
  - Numerical attribute
  - GINI
  - Regression tree
  - R

A problem

- An attribute (e.g. ID) with too many attribute values cause some troubles
  - An attribute with many values tend to be selected
  - But the result is misery

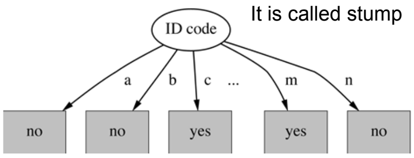
If resulted branches are large in number,

- Let us assign IDs.

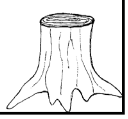
ID code	Outlook	Temp.	Humidity	Windy	Play
A	Sunny	Hot	High	False	No
B	Sunny	Hot	High	True	No
C	Overcast	Hot	High	False	Yes
D	Rainy	Mild	High	False	Yes
E	Rainy	Cool	Normal	False	Yes
F	Rainy	Cool	Normal	True	No
G	Overcast	Cool	Normal	True	Yes
H	Sunny	Mild	High	False	No
I	Sunny	Cool	Normal	False	Yes
J	Rainy	Mild	Normal	False	Yes
K	Sunny	Mild	Normal	True	Yes
L	Overcast	Mild	High	True	Yes
M	Overcast	Hot	Normal	False	Yes
N	Rainy	Mild	High	True	No

When ID comes to root

It is called stump



The entropy of this split is:
info("IDcode") = info([0,1]) + info([0,1]) + … + info([0,1]) = 0 bits
⇒ Information gain is the largest (0.940 bits in this case))



http://www.extension.umn.edu/yardandgarden/ygbriefs/h446removetree.html

Attribute with many values

Clearly

- if an attribute has many values, the subset split by the attribute tend to be very pure
  - Information gain is biased to an attribute with many values
  - This causes overfitting (In a sense of fitting to old data, it behaves nicely, but for a prediction of unseen data, it behaves miserably)

One possible solution: gain ratio

- Gain ratio: decrease the bias that the information gain has
- Gain ratio is defined so that the number of branches and the size of training datasets to the branch are considered.

Calculation of gain ratio

Example: Split information of ID
info([1,1,…,1]) = 14 × (- (1/14) log(1/14)) = 3.807 bits

The definition of gain ratio
gain_ratio(“Attribute”) = gain(“Attribute”) / split_info(“Attribute”)
example:
gain_ratio(“IDcode”) = 0.940 bits / 3.807 bits = 0.246

$$\boldsymbol{InformationGain(D, A)} = \boldsymbol{H(D)} - \sum_{v \in values(A)} \left[\frac{|D_v|}{|D|} \cdot H(D_v) \right]$$

$$\boldsymbol{SplitInformation(D, A)} = - \sum_{v \in values(A)} \left[\frac{|D_v|}{|D|} \cdot \log\left(\frac{|D_v|}{|D|} \right) \right] = H\left(\frac{|D_1|}{|D|}, \frac{|D_2|}{|D|}, \ldots, \frac{|D_{|A|}|}{|D|} \right)$$

Gain ratio of other attributes

Outlook		Temperature	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.247	Gain: 0.940-0.911	0.029
Split info: info([5,4,5])	1.577	Split info: info([4,6,4])	1.362
Gain ratio: 0.247/1.577	0.156	Gain ratio: 0.029/1.362	0.021
Humidity		Windy	
Info:	0.788	Info:	0.892
Gain: 0.940-0.788	0.152	Gain: 0.940-0.892	0.048
Split info: info([7,7])	1.000	Split info: info([8,6])	0.985
Gain ratio: 0.152/1	0.152	Gain ratio: 0.048/0.985	0.049

ID	
Info:	0
Gain:	0.940
Split info:	3.807
Gain ratio:	0.246

Is the problem solved?

- “Outlook” comes to the top, but this time “Humidity” comes second, because “Humidity” causes split into two and therefore gain ration becomes relatively better.
- Still gain ratio of “ID” is the largest (advantage, though, decreases).
- A problem of gain ratio: there are cases of over-compensation (contrary to the above failure)
  - Possibility of selecting inappropriate attribute because split information is relatively small
  - A simple remedy: although attribute with the largest gain ratio is selected, the attribute should be the one whose information gain is larger than their average among all the attributes.

Note

- A top-down algorithm of constructing DT (“ID3”) was developed by Ross Quinlan (University of Sydney Australia)
- Gain ration is an improvement of the basic algorithm
  - Further improvement is done in C4.5. Numerical attributes, missing values, and noisy data are treated in this version.
- Some other methods exist for attribute selection! (but not so impressive difference exist)

Summary

- DT and its construction
  - DT is easy to understand and use. Accuracy is relatively low.
  - Greedy algorithm is used for construction.
    - Attribute on nodes are selected based on information gain
      - Sometimes "information gain ratio" works better

Exercise

- The data is the same as for Naïve Bayes.
- Select the attribute for the root as is explained.

snow	weather	season	physical condition	go skiing
sticky	foggy	low	rested	no
fresh	sunny	low	injured	no
fresh	sunny	low	rested	yes
fresh	sunny	high	rested	yes
fresh	sunny	mid	rested	yes
frosted	windy	high	tired	no
sticky	sunny	low	rested	yes
frosted	foggy	mid	rested	no
fresh	windy	low	rested	yes
fresh	windy	low	rested	yes
fresh	foggy	low	rested	yes
fresh	foggy	low	rested	yes
sticky	sunny	mid	rested	yes
frosted	foggy	low	injured	no