

Machine Learning

Prepared by Dr. Yasser Abdelhamid

DS 620
Machine Learning

Lecture 7





OUTLINE

- Decision trees

DECISION TREES

- ❖ A decision tree is a supervised classification algorithm that implements divide and conquer strategy in building a tree structure for the training set.
- ❖ Decision trees vs rule-based Expert systems.
 - Nodes represent features/attributes upon which a decision to be taken
 - Arcs represent different rules.
 - leaf nodes represent output variable classes.

EXAMPLE

❖ If tomorrow the values of Outlook, Temperature, Humidity and Windy were sunny, 74°F, 77% and false respectively, what would the decision be?

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

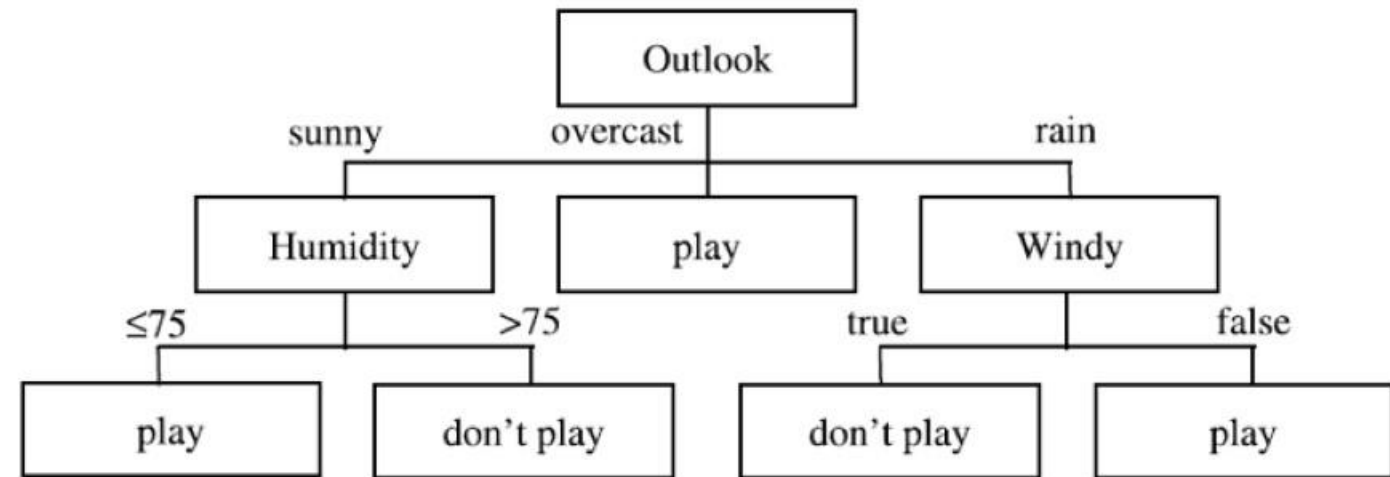
Classes
play, don't play
Outlook
sunny, overcast, rain
Temperature
numerical value
Humidity
numerical value
Windy
true, false

EXAMPLE

❖ If tomorrow the values of Outlook, Temperature, Humidity and Windy were:

- sunny,
- 74°F,
- 77% and
- false respectively,

❖ what would the decision be?



BASIC TERMINOLOGY

- ❖ Objects/Instances, Attributes, and Classes.
- ❖ Splitting on attributes:
 - Testing the value of an attribute and then creating a branch for each of its possible values.
- ❖ Splitting on categorical and continuous attributes.
- ❖ Splitting continues until each branch can be labelled with just one classification.
- ❖ Decision trees as an alternative concise representation of the dataset.
- ❖ Decision trees as a generalization to predict the class of unseen instances.

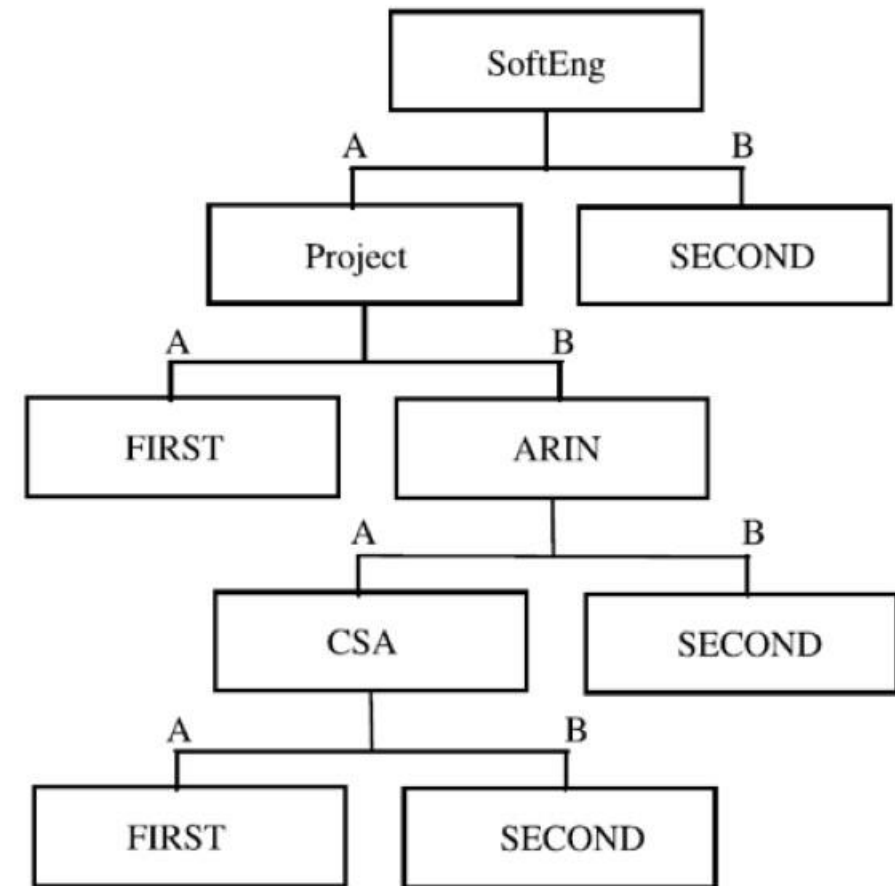
EXAMPLE 2

SoftEng	ARIN	HCI	CSA	Project	Class
A	B	A	B	B	SECOND
A	B	B	B	A	FIRST
A	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	A	B	B	A	FIRST
B	A	A	B	B	SECOND
A	B	B	B	B	SECOND
A	B	B	B	B	SECOND
A	A	A	A	A	FIRST
B	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	B	B	A	B	SECOND
B	B	B	B	A	SECOND
A	A	B	A	B	FIRST
B	B	B	B	A	SECOND
A	A	B	B	B	SECOND
B	B	B	B	B	SECOND
A	A	B	A	A	FIRST
B	B	B	A	A	SECOND
B	B	A	A	B	SECOND
B	B	B	B	A	SECOND
B	A	B	A	B	SECOND
A	B	B	B	A	FIRST
A	B	A	B	B	SECOND
B	A	B	B	B	SECOND
A	B	B	B	B	SECOND

Classes
FIRST, SECOND
SoftEng
A,B
ARIN
A,B
HCI
A,B
CSA
A,B
Project
A,B

EXAMPLE 2

- ❖ Think of the root node as corresponding to the original training set.
- ❖ All other nodes correspond to a subset of the training set.
- ❖ A branch is a path from root node to one of the leaf nodes.
- ❖ There are five leaf nodes and hence five branches.
- ❖ Each branch corresponds to a classification rule.



TYPES OF REASONING

- ❖ Deduction : Making an inference based on accepted facts or premises.
- ❖ if premises are true then actions are true.

All Men Are Mortal

John is a Man

Therefore John is Mortal

TYPES OF REASONING

- ❖ Abduction: making a **probable** conclusion from what you know.
- ❖ Conclusion is consistent with the truth of the premises, but it **may not necessarily be correct**.

All Dogs Chase Cats
Fido Chases Cats
Therefore Fido is a Dog

TYPES OF REASONING

- ❖ Induction: Making an inference based on an observation.
- ❖ A generalization is based on repeated observations.
Example: After many observations of x and y occurring together, learn the rule
if x then y
- ❖ Example: if I see 1,000 dogs with four legs I might reasonably conclude that **“if x is a dog then x has 4 legs”**

TDIDT

(TOP DOWN INDUCTION ON DECISION TREES)

- ❖ Decision trees are generated by repeatedly splitting on the values of attributes.
- ❖ This process is known as **recursive partitioning**.

TDIDT: BASIC ALGORITHM

IF all the instances in the training set belong to the same class
THEN return the value of the class

ELSE (a) Select an attribute A to split on⁺

(b) Sort the instances in the training set into subsets, one for each value of attribute A

(c) Return a tree with one branch for each *non-empty* subset, each branch having a descendant subtree or a class value produced by applying the algorithm recursively

⁺ Never select an attribute twice in the same branch

TDIDT(CONT.)

- ❖ The maximum possible length for a branch is M (Number of attributes), hence the algorithm is **guaranteed to terminate**.
- ❖ No two instances with the same values of all the attributes may belong to different classes (*Adequacy Condition*).
- ❖ **The algorithm does not specify the method of selecting an attribute for split.**
- ❖ Making a good choice of attributes to split on at each stage is crucial to the success of the TDIDT approach.

ATTRIBUTE SELECTION

❖ Is it feasible to look for the best selection of attributes?

Dataset	take first	take last	random					most	least
			1	2	3	4	5		
contact_lenses	42	27	34	38	32	26	35	42	26
lens24	21	9	15	11	15	13	11	21	9
chess	155	56	94	52	107	90	112	155	52
vote	40	79	96	78	116	110	96	116	40
monk1	60	75	82	53	87	89	80	89	53
monk2	142	112	122	127	109	123	121	142	109
monk3	69	69	43	46	62	55	77	77	43

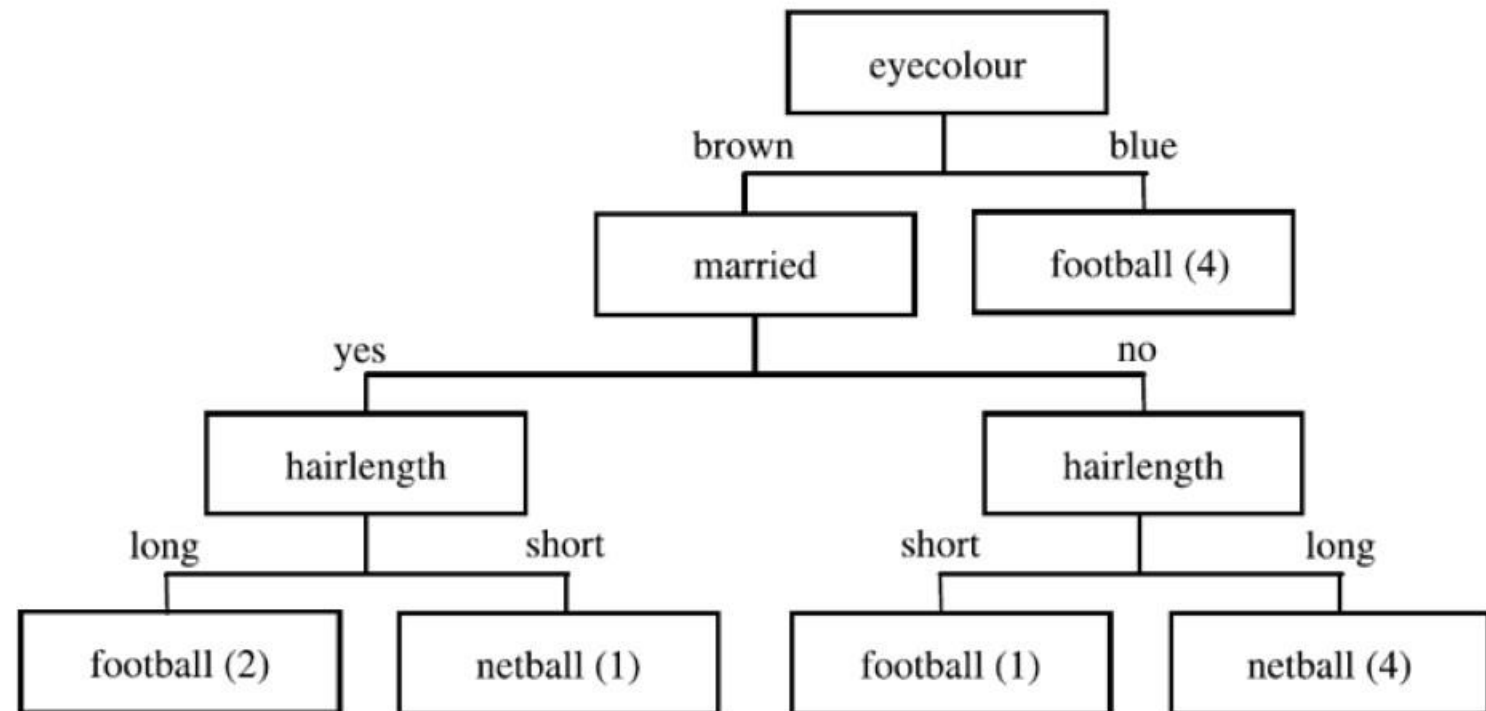
EXAMPLE

❖ Using this dataset can you make a set of generalization rules to decide whether a person will play Football or Netball

eyecolour	married	sex	hairlength	class
brown	yes	male	long	football
blue	yes	male	short	football
brown	yes	male	long	football
brown	no	female	long	netball
brown	no	female	long	netball
blue	no	male	long	football
brown	no	female	long	netball
brown	no	male	short	football
brown	yes	female	short	netball
brown	no	female	long	netball
blue	no	male	long	football
blue	no	male	short	football

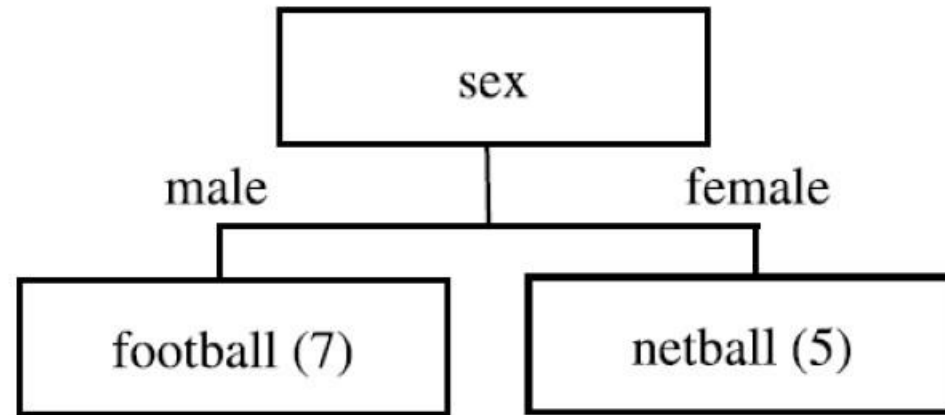
SOLUTION 1

❖ Five branches
(rules)



SOLUTION 2

❖ Only two rules



❖ The only way to know which one gives better results for unseen data is to use them both and compare the results.

SELECTION USING ENTROPY AND INFORMATION GAIN

- ❖ Select the attribute that minimizes the value of **entropy**, thus maximizing the **information gain**.
- ❖ There is **no guarantee** that using entropy will always lead to a small decision tree
- ❖ Experience shows that it generally produces trees with **fewer branches**.
- ❖ Experience also shows that **small trees tend to give more accurate predictions** than large ones, although there is certainly no guarantee of infallibility.

Dataset	excluding entropy		entropy
	most	least	
contact lenses	42	26	<u>16</u>
lens24	21	<u>9</u>	<u>9</u>
chess	155	52	<u>20</u>
vote	116	40	<u>34</u>
monk1	89	53	<u>52</u>
monk2	142	109	<u>95</u>
monk3	77	43	<u>28</u>

EXAMPLE

- ❖ Lenses dataset
- ❖ 24 instances
- ❖ 3 classes

age	Value of attribute			Class
	specRx	astig	tears	
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

classes
 1: hard contact lenses
 2: soft contact lenses
 3: no contact lenses

age
 1: young
 2: pre-presbyopic
 3: presbyopic

specRx
 (spectacle prescription)
 1: myopia
 2: high hypermetropia

astig
 (whether astigmatic)
 1: no
 2: yes

tears
 (tear production rate)
 1: reduced
 2: normal

ENTROPY

- ❖ Entropy is a measure of the 'uncertainty' contained in a training set, due to the presence of more than one possible classification.

ENTROPY

- ❖ Let K = number of classes.
- ❖ p_i for $i = 1$ to K is the number of occurrences of class i divided by the total number of instances, which is the **probability of the class i** .

$$E = - \sum_{i=1}^K p_i \log_2 p_i$$

- ❖ E is the entropy of the training set, it is measured in '**bits**' of information, **where p_i not equal 0**.

ENTROPY

- ❖ The value of $-p_i \log_2 p_i$ is positive **for values of p_i greater than zero and less than 1.**
- ❖ When $p_i = 1$ the value of $-p_i \log_2 p_i$ is zero.
- ❖ E is positive or zero for all training sets.
- ❖ E is **zero** if all the instances belong to the **same class.**
- ❖ E takes its **maximum value** when the instances are **equally distributed** amongst the K possible classes.

ENTROPY OF LENSES DATASET

- ❖ Total instances is 24.
- ❖ 4 instances with classification 1
- ❖ 5 instances with classification 2
- ❖ 15 instances with classification 3
- ❖ $p_1 = 4/24$,
- ❖ $p_2 = 5/24$ and
- ❖ $p_3 = 15/24$.
- ❖ E_{start} is the entropy of the whole dataset regardless to any attributes.
- ❖ $E_{\text{start}} = -(4/24) \log_2(4/24) - (5/24) \log_2(5/24) - (15/24) \log_2(15/24)$
- ❖ $= 0.4308 + 0.4715 + 0.4238 = 1.3261$ bits

USING ENTROPY FOR ATTRIBUTE SELECTION

- ❖ Partition the initial training set into smaller training sets repeatedly using attributes, until the entropy of each of these subsets becomes zero.
- ❖ Splitting on *any* attribute has the property that **the average entropy of the resulting subsets will be less than (or occasionally equal to) to that of the previous training set.**

SPLITTING ON AGE

Value of attribute				Class
age	specRx	astig	tears	
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

Value of attribute				Class
age	specRx	astig	tears	
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1

Age=1

Value of attribute				Class
age	specRx	astig	tears	
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3

Age=2

Value of attribute				Class
age	specRx	astig	tears	
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

Age=3

ENTROPY OF AGE ATTRIBUTE

❖ Training set 1 (age = 1)

- Entropy $E1 = -(2/8) \log_2(2/8) - (2/8) \log_2(2/8) - (4/8) \log_2(4/8)$
- $= 0.5 + 0.5 + 0.5 = 1.5$

❖ Training set 2 (age = 2)

- Entropy $E2 = -(1/8) \log_2(1/8) - (2/8) \log_2(2/8) - (5/8) \log_2(5/8)$
- $= 0.375 + 0.5 + 0.4238 = 1.2988$

❖ Training Set 3 (age = 3)

- Entropy $E3 = -(1/8) \log_2(1/8) - (1/8) \log_2(1/8) - (6/8) \log_2(6/8)$
- $= 0.375 + 0.375 + 0.3113 = 1.0613$

AVERAGE ENTROPY

- ❖ Average entropy of the three training sets produced by splitting on attribute age is
- ❖ $E_{new} = (8/24)E_1 + (8/24)E_2 + (8/24)E_3 = \mathbf{1.2867 \text{ bits}}$

$$\text{Information Gain} = E_{start} - E_{new}$$

- ❖ Information gain from splitting on attribute *age* is
 - $1.3261 - 1.2867 = 0.0394 \text{ bits}$

AVERAGE ENTROPY

- ❖ Notice that when the average entropy becomes smaller, the information gain becomes larger, and vice versa.
- ❖ To conclude, we need to maximize information gain (or minimize attribute average entropy).

MAXIMIZING INFORMATION GAIN

❖ We have four candidate attributes to split on in our dataset, *age*, *specRx*, *astig* and *tears*.

Splitting on *age* attribute generates average entropy of

1.2867, hence Information Gain = $1.3261 - 1.2867 = 0.0394$ bits

Splitting on *specRx* attribute generates average entropy of 1.2866 hence
Information Gain = $1.3261 - 1.2866 = 0.0395$ bits

Splitting on *astig* attribute generates average entropy of

0.9491 hence Information Gain = $1.3261 - 0.9491 = 0.3770$ bits

Splitting on *tears* attribute generates average entropy of 0.7773 hence
Information Gain = $1.3261 - 0.7773 = 0.5488$ bits

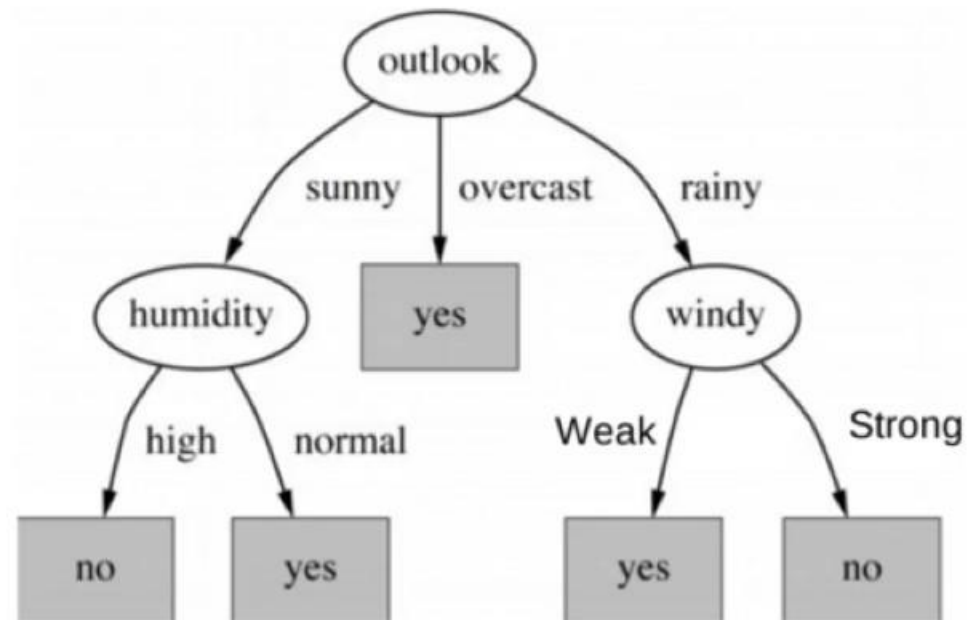
❖ Which attribute shall we select?

EXAMPLE

S. No.	Outlook	Temperature	Humidity	Windy	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

MAKE A DECISION TREE THAT PREDICTS WHETHER
TENNIS WILL BE PLAYED ON THE DAY?

Final decision tree



ALGORITHMS

ALGORITHMS

CART

- ***GINI INDEX***

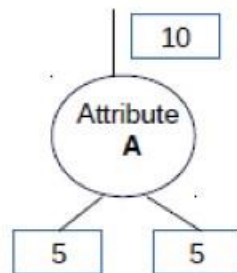
ID₃

- ***ENTROPY FUNCTION***
- ***INFORMATION GAIN***

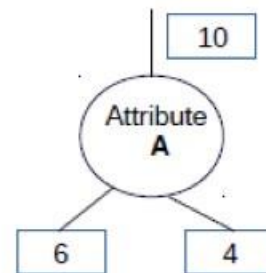
ENTROPY

- ❖ Given a binary (two-class) classification, C , and a set of examples, S , the class distribution at any node can be written as (p_0, p_1) , where $p_1 = 1 - p_0$, and the entropy, of S is the sum of the information:

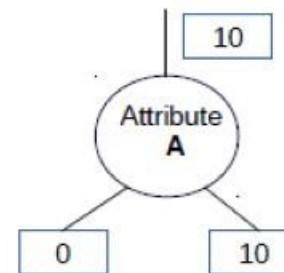
$$\text{Info}(S) = - p_0 \log_2 p_0 - p_1 \log_2 p_1$$



Entropy =1
Highest impurity
Useless classifier



Entropy =0.97
High impurity
Poor classifier



Entropy =0
Impurity= 0
Good classifier

GENERAL FORM

- ❖ In the general case, the target attribute can take on **m** different values (multi-way split) and the entropy of *S* relative to this **m**-wise classification is given by:

$$Inf(S) = - \sum_i^m p_i \log_2(p_i)$$

INFORMATION GAIN

- ❖ Information gain is an attribute selection measure.
- ❖ It measures how much information a given attribute X tells us about the Class Y.
- ❖ According to this measure, we select the attribute with the highest information gain score.

$$Info(Y|X) = \sum_j P(X=v_j) * Info(Y|X=v_j)$$

- ❖ Other notations:

- $Info_X(Y)$.
- $I(X)$.
- $Info(Y | X=v_j)$.

INFORMATION GAIN

❖ Let p_i be the probability that an arbitrary tuple in D (a data set) belongs to class C_i , estimated by $|C_{i,D}|/|D|$

❖ Expected information (entropy) needed to classify a tuple in D :

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

❖ Information needed (after using attribute A to split D into v partitions) to classify D :

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} * Info(D_j)$$

❖ Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

EXAMPLE

Class-labeled training tuples from the *AlIElectronics* customer database.

<i>RID</i>	<i>age</i>	<i>income</i>	<i>student</i>	<i>credit_rating</i>	<i>Class: buys_computer</i>
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

EXAMPLE

$$Info(D) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) = 0.940 \text{ bits.}$$

$$\begin{aligned} Info_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) \\ &\quad + \frac{4}{14} \times \left(-\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \right) \\ &\quad + \frac{5}{14} \times \left(-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) \\ &= 0.694 \text{ bits.} \end{aligned}$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

- ❖ **Gain(age) = 0.246 bits (Highest Information Gain)**
- ❖ Gain(student) = 0.151 bits
- ❖ Gain(credit rating) = 0.048 bits
- ❖ Gain(income) = 0.029 bits

Class-labeled training tuples from the *AlIElectronics* customer database.

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no



SUMMARY

- Decision trees

