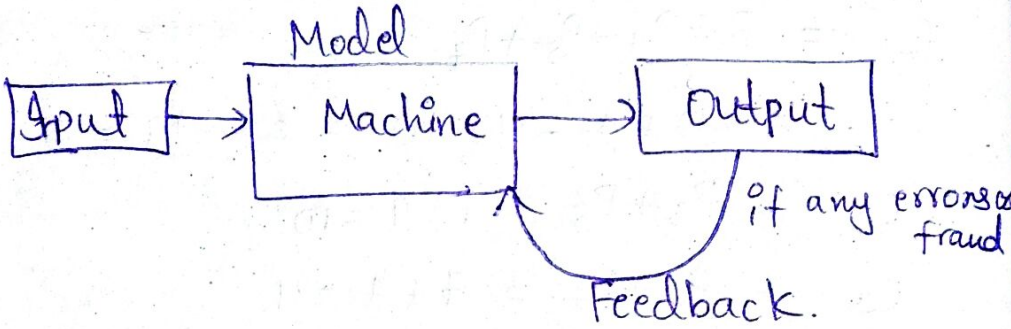
Machine Learning

UNIT-1

(1) Introduction:

. A machine which learns from the human instructions as input and provides output as required information, if any unwanted info is getting then by providing feedback to machine, it gives the required information is known as machine learning.

Model:



(1) Well-posed learning Problems:

. A computer program is said to learn from experience 'E' in content to some task 'T' & some performance measure P, if its performance on T, as was measured by P, upgrades with experience E.

. Any problem can be segregated as well-posed learning problem if it has 3 traits -

i) Task
ii) Performance
iii) Experience.

Certain examples that efficiently defines well-posed learning problems are:

Eg. 1 - A checkers learning problem.

Task - Playing checkers game.

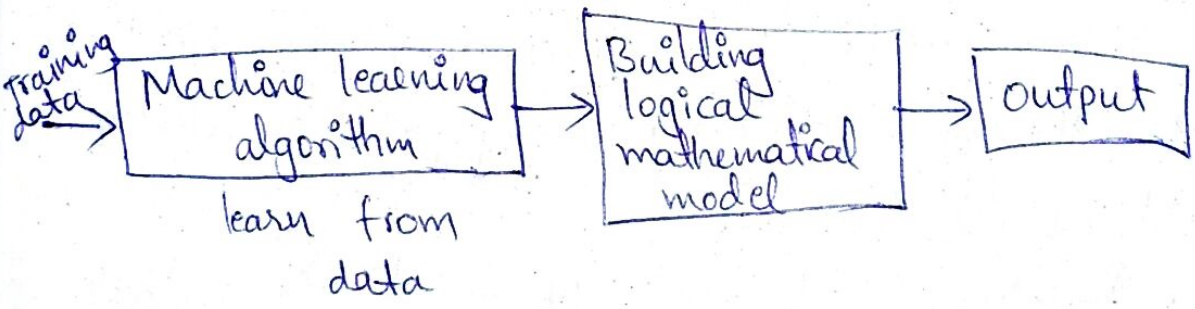
Performance measure - % (percent) of games won against oppaser.

Experience - Playing implementation games against itself

(2) Designing a learning system:

When we fed the training data to machine learning algorithm, this algorithm will produce a mathematical model & with the help of it, the machine will make a prediction & take a decision without being explicitly programmed.

Also, during training data, the more machine will work with it, the more it will get experience & the more it will get experience the more efficient result is produced.



Steps for designing learning system are:

i) Choosing training experience:

It is based on 3 attributes-

a) Training experience need to provide direct/indirect feedback with regarding choices

b) The learner need to control the sequence of training examples.

c) The representation of the distribution of examples over which performance measured.

Step-2: choosing target function:
. It means, according to the algorithm the legal moves will be taken.

step-3: Choosing representation for target function.
. Moves need to be taken choosen with the optimized using specified representation.

step-4: Choosing function approximation algorithm:
. Not only the optimized moves, but also it will approximates which steps need to be choosen provide feedback.

step-5: Final design
After all the steps, the system will be designed.

(3) Perspectives & issues in ML:-

Perspectives:
. ML involves searching a very large space of possible hypothesis to determines one that best fits the observed data & any prior knowledge held by the learner.
Eg: In checkers learner, the hypothesis is located by spaeching through the vast space with available training examples.
. Hypothesis representations are appropriate for learning different kinds of target functions.

Issues:
. What learning algorithms to be used?
. How much training data is sufficient?

. When and how prior knowledge can guide the learning process?
. What is the best strategy for choosing a next training experience?
. How can the learner automatically alter its representation to improve its learning ability?

Concept learning & the general to specific ordering:-

(1) Introduction:

. Learning involves acquiring general concepts from specific training examples.

Eg:- People continually learn general concepts or categories such as "bird", "car", "situations in which I should study more in order to pass the exam," etc.,

. Each such concept can be viewed as describing some subset of objects or events defined over a larger set.

. Alternatively, each concept can be thought of as a boolean-valued function defined over this larger set.

Eg:- (A funct^n defined over all animals whose value is true for birds & false for every other animal)

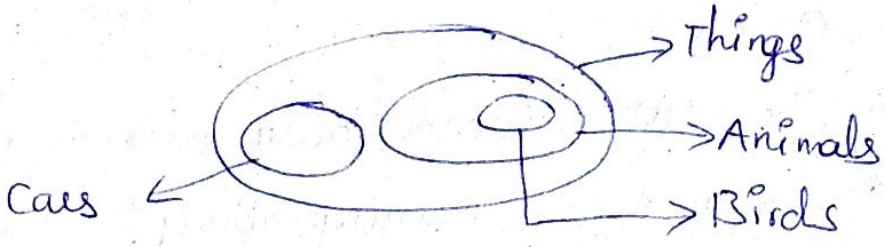
Definition:
Concept-learning:

Inferring a boolean-valued function from training examples of its input and output.

(2) Concept-learning task:-

. A concept is a subset of objects or events defined over a larger set.

Eg:- We refer to the set of everything as the set of things. Animals are a subset of things, & birds are a subset of animals.



Example of a concept learning task:-

. Concept: Good days for watersports (values: yes, no)

. Attributes/features:

sky (values: sunny, cloudy, rainy) etc;

Day Example	sky	AirTemp	Humidity	Wind	water	Fore cast	Enjoy sport
1	sunny	warm	Normal	strong	warm	same	Yes
2	sunny	warm	High	strong	warm	same	Yes
3	Rainy	cold	High	strong	warm	change	No
4	sunny	warm	High	strong	cool	change	Yes

instances

class

AirTemp (values: warm, cold)

Humidity (values: Normal, High)

Wind (values: strong, weak)

Water (warm, cool)

Forecast (values: same, change).

. Eg: of a training point:

<sunny, warm, high, strong, warm, same, yes>

. Chosen Hypothesis representation:

→ "?" means any value is acceptable

→ "0" means no value is acceptable.

Example of a hypothesis: <?, cold, high, ?, ?, ?>

(If the air temperature is cold and humidity high then it is a good day for water sports).
- It is no, so our hypothesis is incorrect.

. The goal of a concept-learning task is to infer the "best" concept-description from the set of all possible hypotheses ("best" means "which best generalizes to all (known or unknown) elements of the instance space").

Notations & terminology:

. Set of items over which the concept is defined is called the set of instances (denoted by X).

. The concept to be learned is called Target concept (denoted by $c: X \rightarrow \{0, 1\}$)

. Set of training examples is set of instances X, along with their target concept value $c(x)$.

. Members of concept (instances for which $c(x) = 1$) are called positive examples.

. Non-members of the concept (instances for which $c(x) = 0$) are called negative examples.

. H represents the set of all possible hypotheses. H is determined by human designer's choice of a hypothesis representation.

. The goal of concept-learning is to find a hypothesis $h: X \rightarrow \{0, 1\}$ such that $h(x) = c(x)$ for all x in X.

③ Concept learning as search :-

. Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation.

. The goal of this search is to find the hypothesis that best fits the training examples.

Eg:- Consider the instances X and hypotheses H in the EnjoySport learning task. The attribute sky has three possible values, & AirTemp, Humidity, Wind, Water, Forecast each have two possible values, the instance space X contains exactly.

. 3.2.2.2.2.2 = 96 distinct instances.

. 5.4.4.4.4.4 = 5120 syntactically distinct hypotheses within H (considering '0' & '?' in addition)

. 1+4*3.3.3.3.3 = 973 semantically distinct hypotheses (cant just one '0' for each attribute since every hypo having one or more '0' symbols is empty).

④ Find-S : Finding a maximally specific hypothesis :-

. Concept learning will be like following two categories:

i) General hypothesis: which it means explaining it with real examples with his/her choices.

G = {'?', '?', ..., '?'}

↳ No. of attributes.

The choices which are taken will be in general i.e with no conditions.

ii) Specific Hypothesis: Here, there will be a specified condition to fulfil their need.

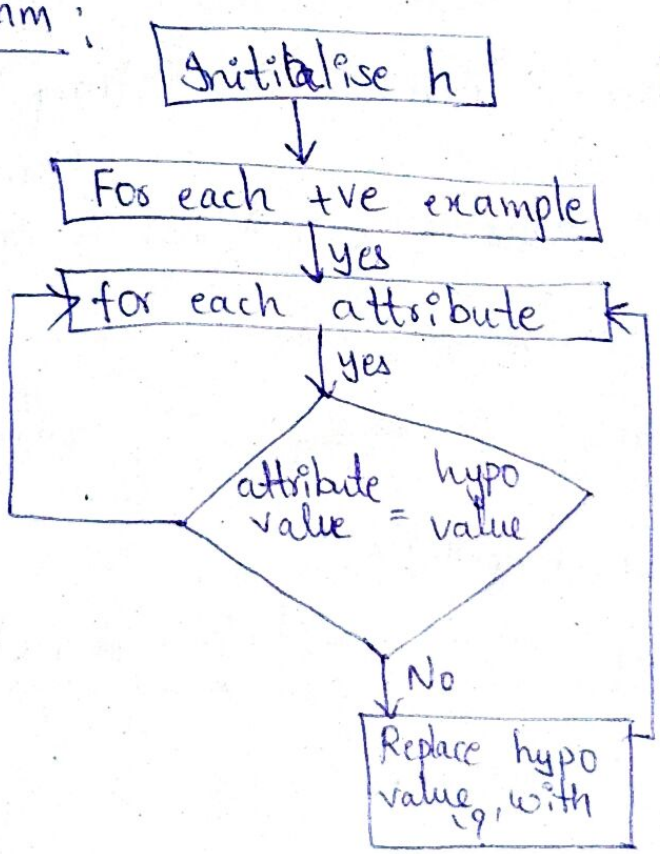
Eg:- If a person wants an apple which is red and sweet, means it has a condition which is specified with color red & taste sweet.

$S = \{ '\emptyset', '\emptyset', \ldots, '\emptyset' \}$
↳ No. of attributes.

Find-S:

- It is most specific hypothesis
- In this, we will consider only +ve examples.

Algorithm:



h to most specific hypothesis.
$h = \{ '\emptyset', '\emptyset', \}$

Dataset for finding a maximally specific hypothesis.

concept: Days on which person enjoys Sport

-table at concept learning task.

step-1: $h_0 = \{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$

$h_0 = \{$'sunny', 'warm', . . , 'warm', 'same'$\}$

step-2:

$h_0 = \{$'sunny', 'warm', 'Normal', 'strong', 'warm', 'same'$\}$

→ compare these first row with second row if is not equal replace '?'

After comparison:

$h_0 = \{$'sunny', 'warm', '?', 'strong', 'warm', 'same'$\}$

3rd instance is -ve example so excluded (No condition)

step-3: $h_0 = \{$'sunny', 'warm', '?', 'strong', 'warm', 'same'$\}$

compare it with 4th instance & if it has nt Equal replace '?'

After comparison:

$h_0 = \{$'sunny', 'warm', '?', 'strong', '?', '?'$\}$

~~step-4~~

∴ This is our final hypothesis.

(5) Version spaces and the candidate elimination algorithm:

Version space:

. It is an intermediate of general & specific hypothesis.

. It returns all (G+specific) general(-ve) the possible hypothesis based on the training dataset (examples). +ve

specific ($S = \{\emptyset, \emptyset, \emptyset, \emptyset\}$)

Version space ↔ ↑ General ($G = \{?, ?, ?, ?\}$)

Candidate elimination algorithm:

- It uses version space.
- Consider both +ve & -ve results.
- We have both specific and general hypothesis.
- For a +ve example; we tend to generalize specific hypothesis.
- For a -ve example: We tend to make general hypothesis more specific.

Algorithm:

step-1: Initialize 'G' & 'S' as most general and specific hypothesis.

step-2: For each example e:

if e is +ve: Make specific hypothesis more general (like as in Find-S)

else (Negative)

- Make general hypothesis more specific.

Dataset:

concept: Days on which person enjoys sport.

↑ 6 attributes

step-1: $S_0 = \{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$ $G_0 = \{?, ?, ?, ?, ?, ?\}$

step-2: 1st row is yes (+ve) so make specific hypothesis more specific

$S_1 = \{'sunny', 'warm', 'Normal', 'strong', 'warm', 'same'\}$

$G_1 = \{?, ?, ?, ?, ?, ?\}$

step-3:

2nd row also yes.

$S_2 = \{'sunny', 'warm', '?', 'strong', 'warm', 'same'\}$

$G_2 = \{?, ?, ?, ?, ?, ?\}$

Step-4: 3rd row -ve so generalize the general hypothesis

$S_3 = \{$'sunny', 'warm', '?', 'strong', 'warm', 'same'$\}$

↑ cold ↑ no need any value ↑ same — it is not equal.

$G_3 = \{$<'sunny' ? ? ? ? ?> <? 'warm', ? ? ? ?> <? ? ? ? ? change same$\}$

↓

Here we are dividing the set becoz there we have rainy but here it is sunny.

Step-5: 4th row is +ve so make specific hypothesis more specific.

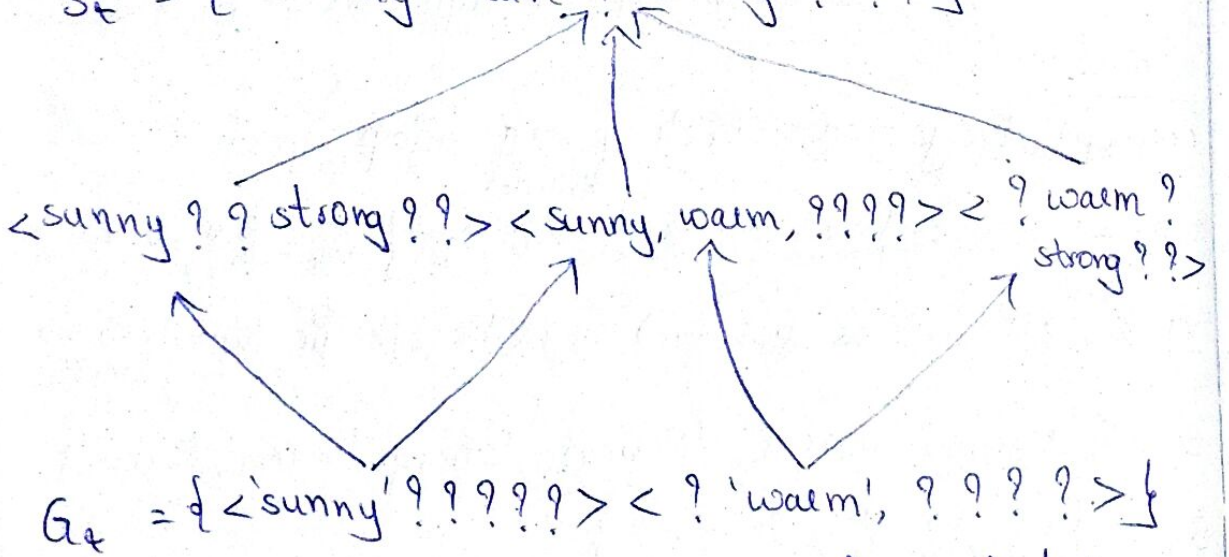
$S_4 = \{$'Sunny', 'warm', '?', 'strong', '?', '?'$\}$

↓ generalize so remove set in G_3

$G_4 = \{$<'sunny' ? ? ? ? ?> <? 'warm', ? ? ? ?>$\}$

Now the version space will be:

$S_4 = \{$<sunny warm ? strong ? ? >$\}$



$G_4 = \{$<'sunny' ? ? ? ? ?> <? 'warm', ? ? ? ? >$\}$

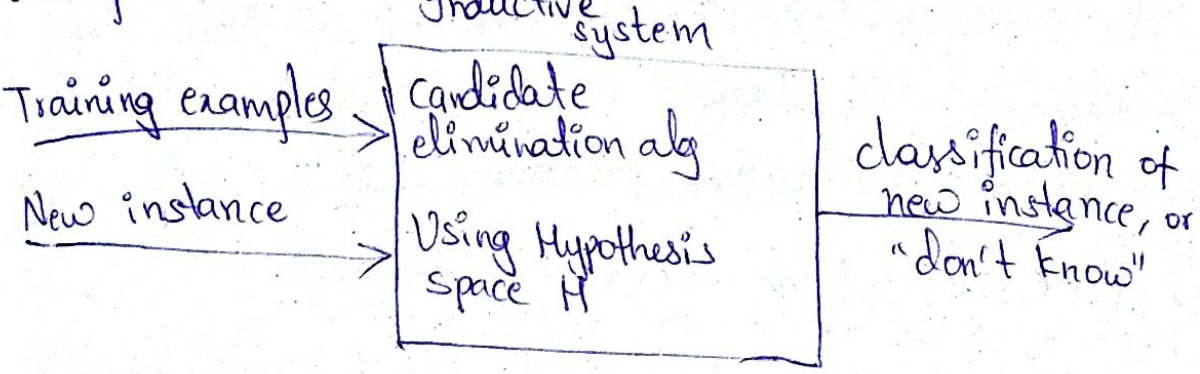
⑥ Remarks on version spaces and candidate elimination:

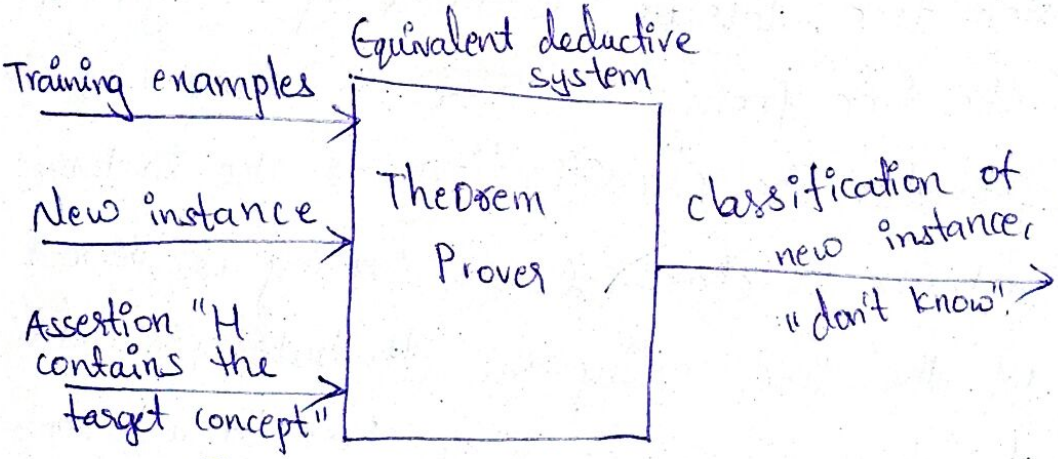
. The taeget concept is exactly learned when the S and G boundary sets converge to a single, identical, hypothesis.

i) there are no errors in the training examples, and
ii) There is some hypothesis in 'H' that correctly describes the target concept.

⑥ Inductive bias:-

. Any preference on the space of all possible hypotheses other than consistency with training examples.





Inductive bias made explicit

→ Decision Tree learning:-

① Introduction:-

• A decision tree is a tree where each node represents a feature (attribute), each link (branch) represents a decision (rule) & each leaf represents

an outcome.

. There are a lot of algorithms in ML which is utilized in our day to day life. One of the important algorithm is decision tree that is used for classification & also a solu^{n} for regression problems.

. It is analyzed using algorithmic approach where a dataset is split into subsets as per conditions.

. The tree is like in the form of if-then-else statements. The deeper is the tree & more are the nodes, the better is the model.

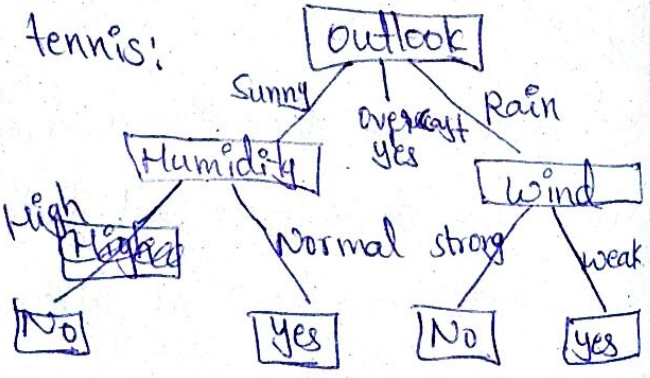
(2) Decision tree representation:

. Decision tree classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance.

. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in figure. This process is then repeated for the subtree rooted at the new node.

Eg:

. A decision for playing tennis:



. The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis & returning the classification associated with the particular leaf (in this case Yes or No)

For eg: The instance:

(outlook = Rain, Temp = Hot, Humidity = high, wind = strong).

. This instance would be sorted down the left-most branch of this decision tree & would be classified as a negative instance.

i.e outlook = Rain (Right) → Humidity wind = strong

= No (Negative instance).

. For +ve instance:

(outlook = sunny ^ Humidity = Normal) V

(outlook = overcast) V

(outlook = Rain ^ wind = weak).

③ Appropriate problems for decision tree learning:

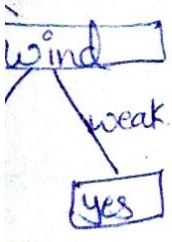
Decision tree learning is generally best suited to problems with the following characteristics:

i) Instances are represented by attribute-value pairs.

Eg: Temperature - value (Hot, cold, mild).

attribute

ii) The target function has discrete o/p values - The decision tree is usually used for boolean classification (e.g: yes or no) kind of example.



iii) Disjunctive descriptions may be required.
iv) The training data may contain errors
v) The training data may contain missing attribute values.

Problems are:

. Many practical problems have been found to fit these characteristics:
  - Equipment classification
  - Medical diagnosis
  - Credit risk analysis
  - Several tasks in natural language processing.

. The problems, in which the task is to classify examples into one of a discrete set of possible categories, are often referred to as classification.

④ The basic decision tree learning algorithm:

. There are two basic algorithms:
  i) CART (classification & Regression tree)
     . GINI Index
  ii) ID_3
     . Entropy function
     . Information gain.

ID_3 algorithm:

. Compute the entropy for data-set entropy(S)
. For every attribute/feature:
→ Calculate entropy for all other values
entropy(A)

→ Take average information entropy for the current attribute

→ Calculate gain for the current attribute.

. Pick the highest gain attribute.

. Repeat until we get the tree we desired.

calculate entropy (Amount of uncertainity (No) in dataset):

$$\text{Entropy} = \frac{-p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

where p = positive
n = negative

calculate Average information:

$$I(\text{Attribute}) = \sum \frac{p_i+n_i}{p+n}\,\text{Entropy}(A)$$

Calculate Information Gain: (Difference in entropy before & after splitting dataset on attribute A).

$$\text{Gain} = \text{Entropy}(S) - I(\text{Attribute})$$

Eg:

S.No	outlook	Temp	Humidity	Windy	Play Tennis
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

Now P(positive) = 9, N(Negative) = 5, Total = 14

. calculate entropy(S):

$$\text{Entropy} = \frac{-p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

$$\text{Entropy}(S) = \frac{-9}{9+5}\log_2\left(\frac{9}{9+5}\right) - \frac{5}{9+5}\log_2\left(\frac{5}{9+5}\right)$$

$$= \frac{-9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

. For each attribute:

→ calculate entropy. let say for outlook - it has sunny, Rainy & overcast values.

outlook	PlayTennis	Outlook	playtennis	Outlook	playtennis
Sunny	No	Rainy	yes	overcast	yes
Sunny	No	Rainy	yes	overcast	yes
Sunny	No	Rainy	~~yes~~ No	overcast	yes
Sunny	yes	Rainy	~~No~~ yes	overcast	yes.
Sunny	yes	Rainy	No		

outlook	P	n	entropy
Sunny	2	3	0.971
Rainy	3	2	0.971
overcast	4	0	0

→ $\frac{-2}{2+3}\left(\log\frac{2}{2+3}\right) - \frac{3}{2+3}\log\left(\frac{3}{2+3}\right)$

→ Calculate Average information entropy.

$$I(\text{outlook}) = \frac{P_{sunny}+n_{sunny}}{p+n}\text{Entropy}(\text{outlook}=\text{sunny}) + \frac{P_{Rainy}+n_{rainy}}{p+n}\text{Entropy}(\text{outlook}=\text{rainy}) + \frac{P_{overcast}+n_{overcast}}{p+n}\text{Entropy}(\text{outlook}=\text{overcast})$$

$$I(\text{outlook}) = \frac{2+3}{9+5}\times 0.971 + \frac{3+2}{9+5}\times 0.971 + \frac{4+0}{9+5}\times 0 = 0.693$$

Total = 14

→ calculate gain: attribute is outlook.

Gain = Entropy(S) − I(Attribute)

Gain = 0.940 − 0.693 = 0.247

. For attribute: Temp

Temp	P	n	entropy
Hot	2	2	1
mild	4	2	0.918
cool	3	1	0.811

→ calculate Average information entropy.

$$I(Temp) = \frac{P_{hot}+n_{hot}}{p+n} E(Temp=Hot) + \frac{P_{mild}+n_{mild}}{p+n} E(Temp=mild) + \frac{P_{cool}+n_{cool}}{p+n} E(Temp=cool)$$

$$= \frac{2+2}{9+5} \times 1 + \frac{4+2}{9+5} \times 0.918 + \frac{3+1}{9+5} \times 0.811 = 091$$

→ calculate gain:

Gain = 0.940 − 0.911 = 0.029

. For attribute Humidity

→ calculate entropy

Humidity	P	n	entropy
High	3	4	0.985
Normal	6	1	0.591

→ calculate Average information entropy.

$$I(Humidit) = \frac{P_{High}+n_{high}}{p+n} E(Humidity=high) + \frac{P_{normal}+n_{normal}}{p+n} E(Humidity=normal)$$

$$= \frac{3+4}{9+5} 0.985 + \frac{6+1}{9+5} 0.591 = 0.78$$

→ calculate gain: = **0.940 − 0.788 = 0.152**

• For attribute : windy

→ calculate entropy:

windy	p	n	entropy
strong	3	3	1
weak	6	2	0.811

→ calculate average information entropy:

$$I(\text{windy}) = \frac{p_{strong} + n_{strong}}{p+n} E(\text{windy} = \text{strong}) + \frac{p_{weak} + n_{weak}}{p+n} E(\text{windy} = \text{weak})$$

$$= \frac{3+3}{9+5}(1) + \frac{6+2}{9+5}(0.811) = 0.892$$

→ Calculate Gain:

$$\text{Gain} = \text{Entropy}(S) - I(\text{windy})$$

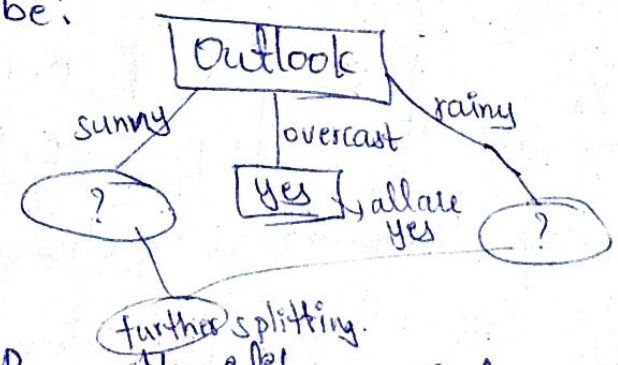
$$\boxed{\text{Gain} = 0.940 - 0.892 = 0.048}$$

• Pick the highest gain attribute:

∴ outlook = 0.247 is the greatest

∴ Root node = outlook

• Tree can be:



outlook	Temp	Humidity	windy	play-tennis
Sunny	Hot	High	weak	No
Sunny	Hot	High	strong	No
Sunny	mild	High	weak	No
Sunny	cool	Normal	weak	yes
Sunny	mild	Normal	strong	yes

$P = 2, \quad n = 3$

$$E(S) = \frac{-2}{2+3}\log_2\left(\frac{2}{2+3}\right) - \frac{3}{2+3}\log_2\left(\frac{3}{2+3}\right) = 0.971$$

. For each attribute;

→ calculate entropy

Humidity	P	n	entropy
High	0	3	0
normal	2	0	0

→ calculate Average information entropy I(humidity) = 0

→ calculate gain Gain = 0.971 − 0 = 0.971

. For windy

→ calculate entropy

→ I(windy) = 0.951

→ Gain = 0.020

windy	P	n	entropy
strong	1	1	1
weak	1	2	0.918

. For temp

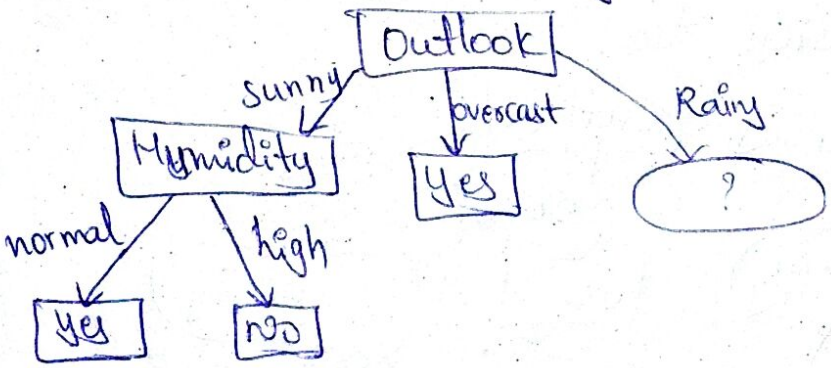
→ calculate entropy

→ I(temp) = 0.4

→ Gain = 0.571

Temp	P	n	entropy
Hot	0	2	0
cool	1	0	0
mild	1	1	1

Now highest is hUmidity i.e = 0.971



For rainy:

Outlook	Temp	Humidity	windy	playtennis
Rainy	Mild	High	weak	yes
Rainy	cool	Normal	weak	yes
Rainy	cool	Normal	strong	No
Rainy	Mild	Normal	weak	yes
Rainy	Mild	High	Strong	No

p = 3 n = 2 total = 5

$$E(S) = \frac{-3}{3+2}\log_2\left(\frac{3}{3+2}\right) - \frac{2}{3+2}\log_2\left(\frac{2}{2+3}\right) = 0.971$$

. For each attribute : let say Humidity.

→ calculate entropy

→ I(Humidity) = 0.951

→ Gain = 0.020

Humidity	p	n	entropy
High	1	1	1
Normal	2	1	0.918

. For windy:

→ calculate entropy

→ I(windy) = 0

→ Gain = 0.971

windy	p	n	entropy
strong	0	2	0
weak	3	0	0

. For temp:

→ calculate entropy

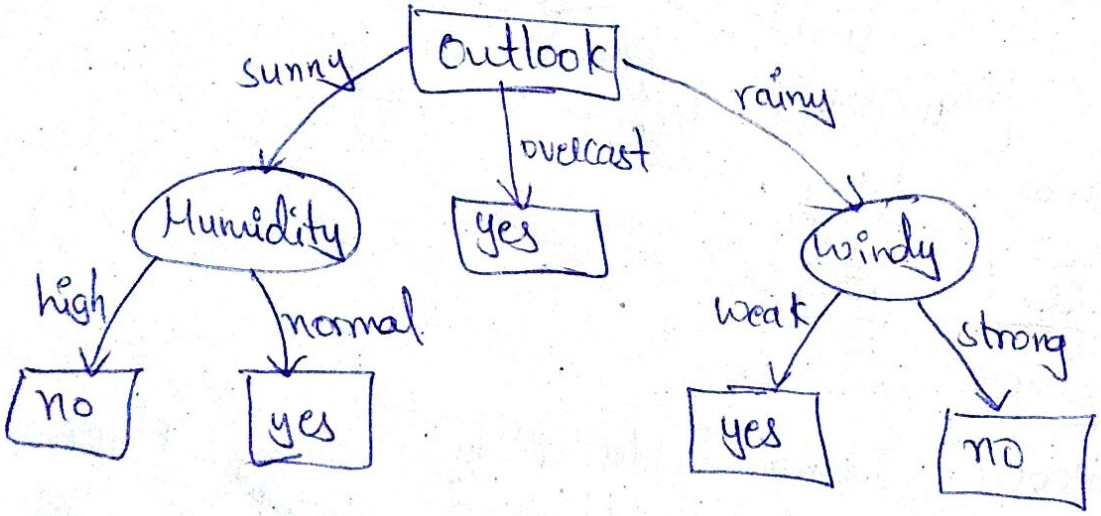
→ I(temp) = 0.951

→ Gain = 0.020

Temp	p	n	entropy
cool	1	1	1
mild	2	1	0.918

. Now the highest is windy i.e = 0.971

∴ Final decision tree:



(5) Hypothesis space search in decision tree learning:

- ID3 can be characterized as searching a space of hypotheses for one that fits the training examples.
- The hypothesis space searched by ID3 is the set of possible decision trees.
- ID3 performs a simple-to-complex, hill-climbing search through this hypothesis space, beginning with the empty tree, then considering progressively more elaborate hypotheses in search of a decision tree that correctly classifies the training data.
- The evaluation function that correctly classifies guides this hill-climbing search is the information gain measure.

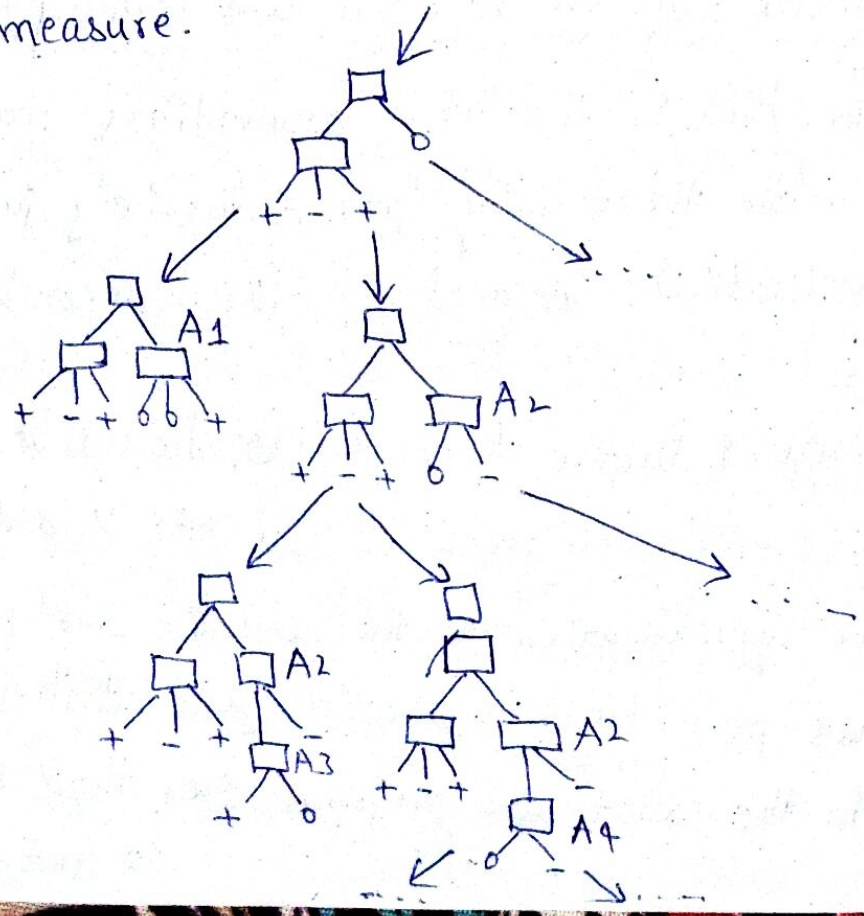


Fig: Hypothesis space search by ID3. ID3 searches through the space of possible decision trees from simplest to increasingly complex, guided by the information gain heuristic.

ID3 capabilities and limitations in search space & search category:

- ID3 hypothesis space of all decision trees is a complete space of finite discrete-valued functions relative to the available attributes.
- ID3 maintains only a single current hypothesis as it searches through the space of decision trees.
- ID3 in its pure form performs no backtracking in its search.

(6) Inductive bias in decision tree learning:

- Inductive bias is the set of assumptions that together with the training data, deductively justify the classifications assigned by the learner to future instances.
- Approximate inductive bias of ID3: shorter trees are preferred over larger trees it uses BFS (Breadth first search)
- A closer approximation to the inductive bias of ID3: Trees that place high information gain attributes close to the root are preferred over those that do not.

ID3 algorithm	Candidate-elimination
i) ID3 searches a completely hypothesis space.	i) Candidate-elimination searches an incomplete hypothesis space.
ii) It searches incompletely through this space, from simple to complex hypotheses, until its termination condition is met.	ii) It searches this space completely, finding every hypothesis consistent with training data.
iii) Its inductive bias is solely a consequence of the ordering of hypotheses by its search strategy.	iii) Its inductive bias is solely a consequence of the expressive power of its hypothesis representation.
iv) It's hypothesis space introduces no additional bias	iv) Its search strategy introduces no additional bias.
v) This form of bias is called a preference bias or search bias.	v) This form of bias is typically called a restriction bias or language bias.
vi) Preference bias is more desirable, becoz it allows the learner to work within a complete hypothesis space that is assured to contain the unknown target function.	vi) It is less desirable, becoz it introduces the possibility of excluding the unknown target function altogether. And it also strictly limits the set of potential hypotheses.

Prefer short hypothesis:

. Occam's razor: is the problem-solving principle that the simplest solu" tends to be the right one. i.e. it should consist the fewest assumptions.

. Occam's razor: "Prefer the simplest hypothesis that fits the data".

. Favour of Occam's razor:

. Fewer short hypotheses than long ones:
  → short hypotheses fits the training data which are less likely to be coincident.
  → Longer hypotheses fits the training data might be coincident.

. Many complex hypotheses that fit the current data but fail to generalize correctly to subsequent data.

(7) Issues in decision tree learning:

i) Avoiding overfitting the data:

. Overfitting happens when the learning algorithm continues to develop hypothesis that reduce training set error at the cost of an increased test set error.

→ Approach to avoid overfitting:

. Pre-pruning that stops growing the tree earlier, before it perfectly classifies the training set.

. Post-pruning allows the tree to perfectly classify the training set, & then prune the tree.

ii) Incorporating continuous-valued attributes:
. The attributes which have continuous values can't have a proper class prediction. Eg: AGE or temperature can have any value, & there is no solu" for it until a range is defined in decision tree itself.

iii) Attributes with many values:
. If attributes have a lot values, then the gain could select any value for further processing. This reduces the accuracy for classification.

iv) Handling attributes with costs:
. The complexity of gain calculation increases if varying cost is associated with every same entry of a tuple to be classified. The solu" to this is replacing the gain calculation.

v) Handling examples with missing attribute values:
. It is possible to have missing value in the training set. To avoid this, most common value among the examples can be selected for the tuple in consideration.

vi) Unable to determine depth of decision tree:
. If the training set does not have an end value i.e the set is given to be continuous, this can lead to an infinite decision tree building.