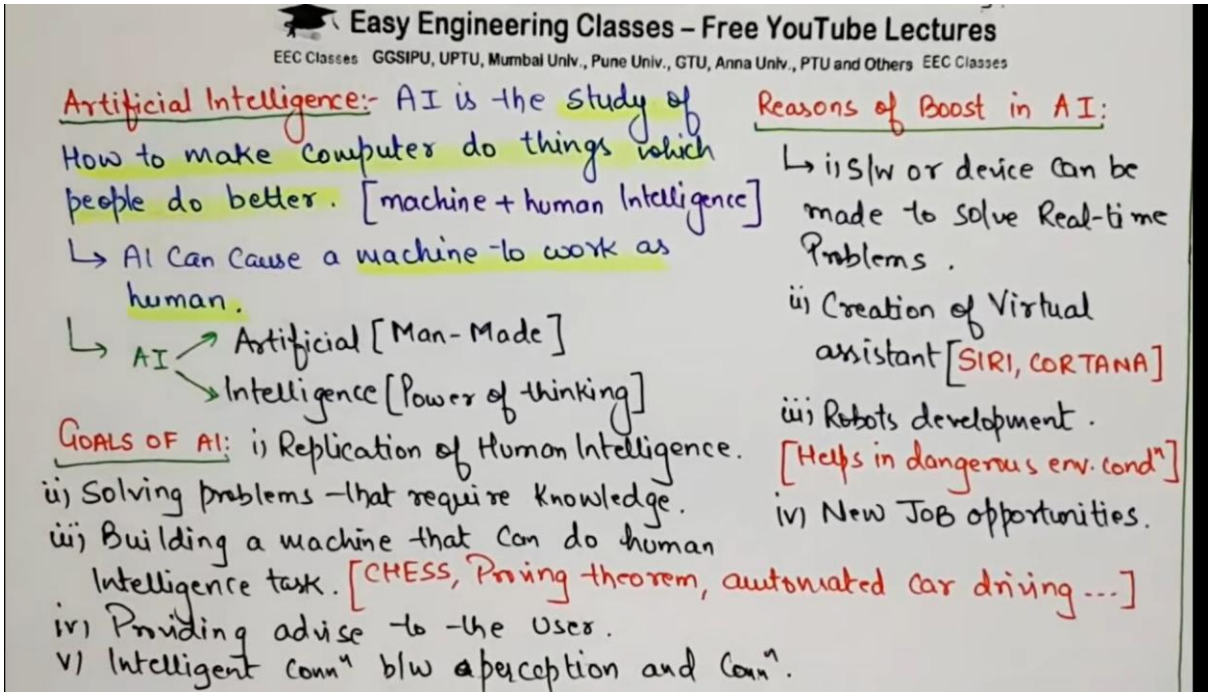
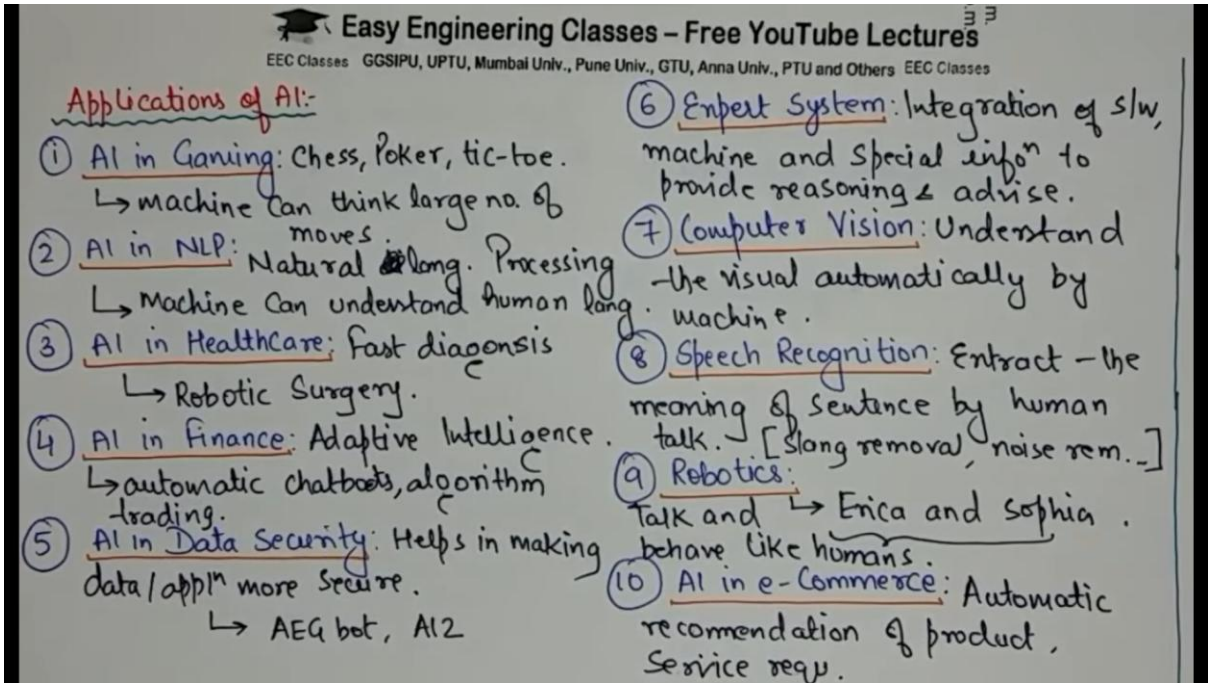
Notes Made by Dare-Marvel

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

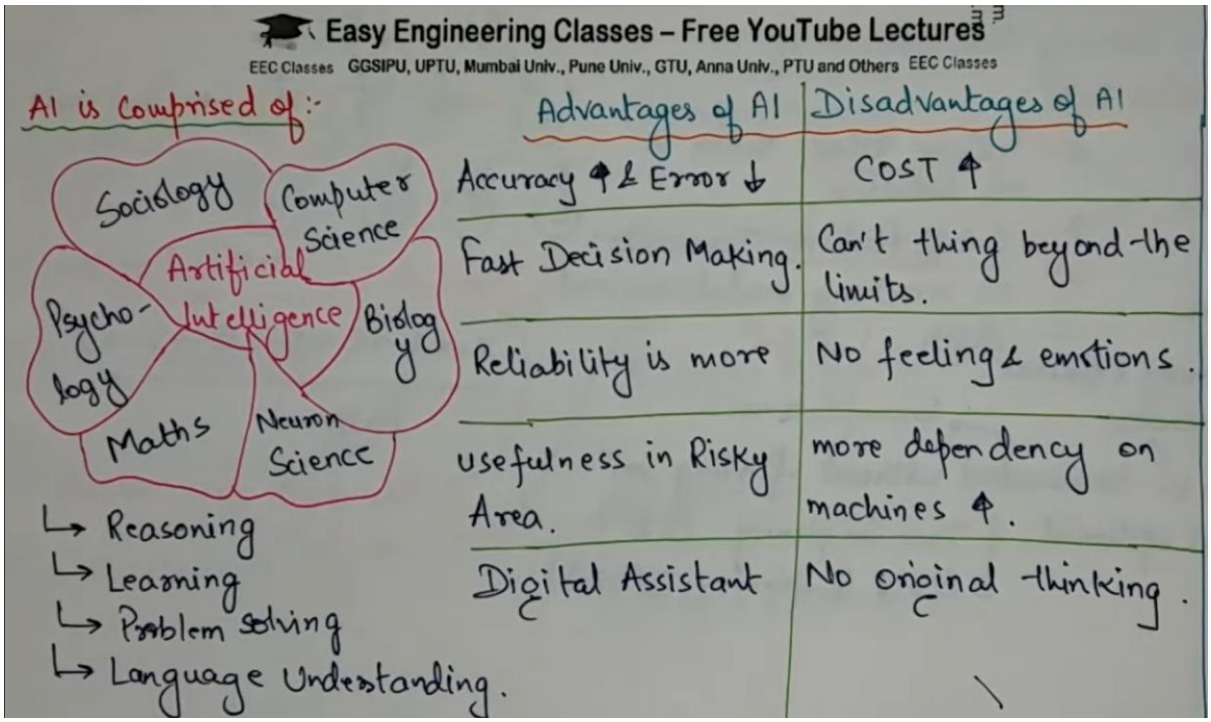
Artificial Intelligence Introduction, Goals, Reasons of Boost



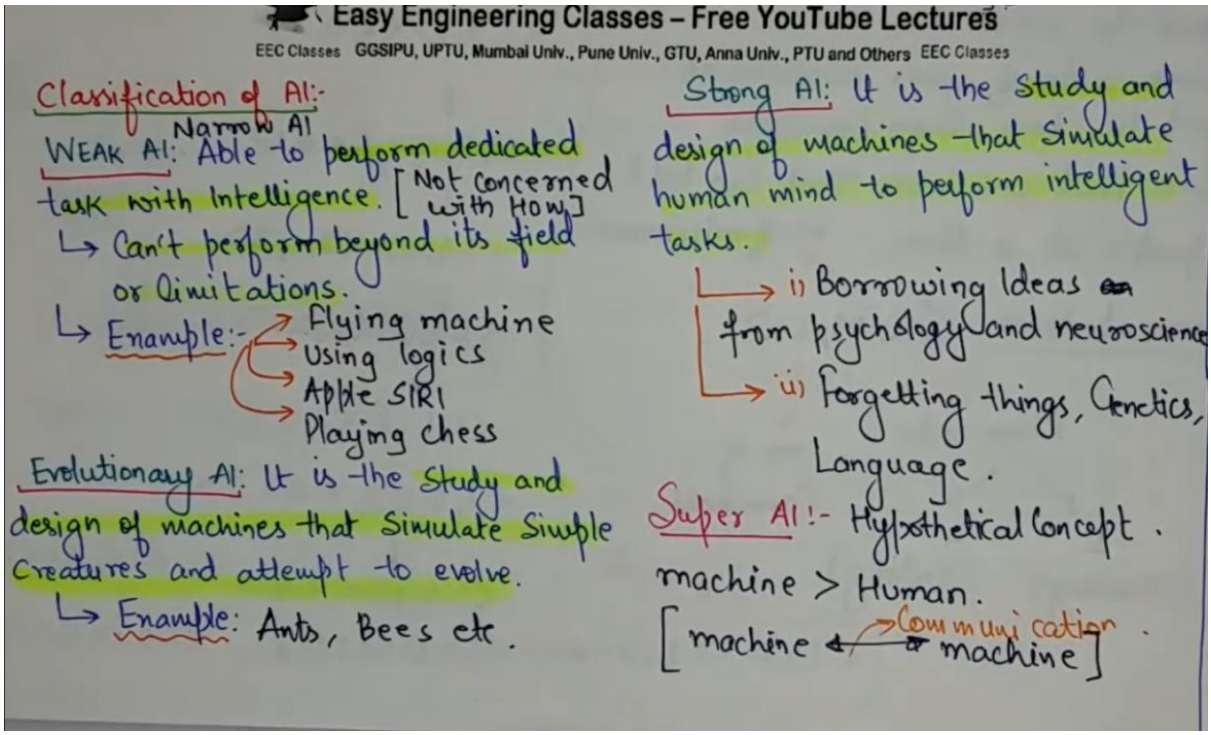
Applications of Artificial Intelligence in various domains



Composition of Artificial Intelligence | Advantages, Disadvantages of Artificial Intelligence



Classification of Artificial Intelligence | Weak, Evolutionary, Strong



Artificial Intelligence Agents | Types of AI Agents

https://www.geeksforgeeks.org/agents-artificial-intelligence/

Types of AI Agents:- [Responsible for any work output obtained from system.

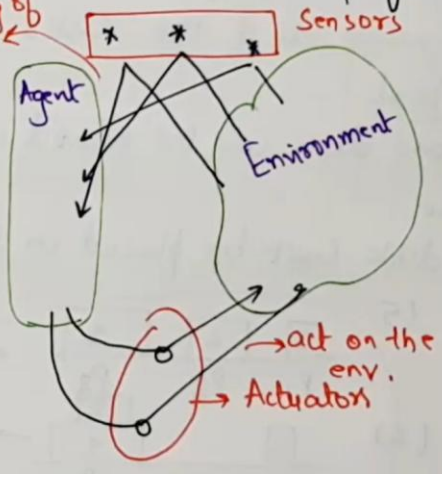
AI is defined as Study of Rational Agents.
- Person
- Firm
- Machine/s/w

→ Make Decisions

Agent → Info^n gathering.
→ action perform.

AI System is Composed of → Agent, Environment

Perceiving of Info^n.

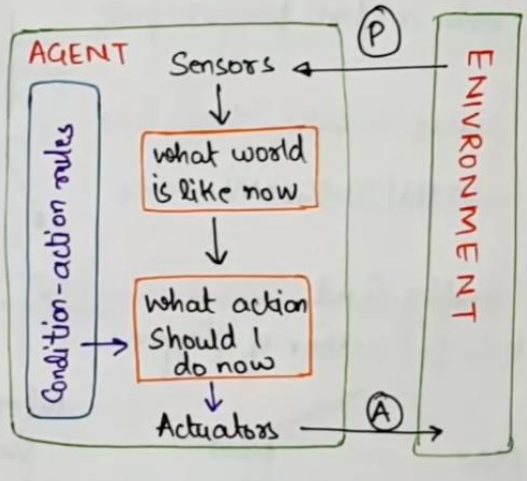


Agent is anything like:-
- i) perceiving its environment through Sensors.
- ii) Acting upon that environment through actuators.

① Simple Reflex Agents:

Works only on Current situation/ perception and ignores the history of previous State. True

→ Condition-Action Rule



Limitations:-
- i) Very limited Intelligence.
- ii) No knowledge about non-perceptual parts of state
- iii) Can go into infinite loop.

(2) Model based Reflex Agent:

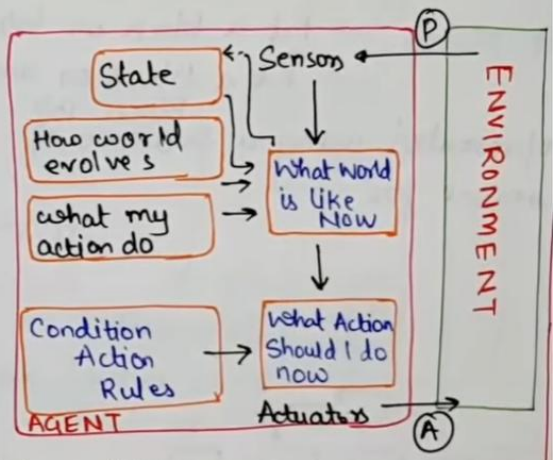
↳ Works by finding the rule whose cond^n matches Current Situation.

↳ Can work in partially observable env., and track situation.

↳ Agent keeps track of internal State which is adjusted by each percept and that depends on percept history.

↳ Model: How things happen in world.

↳ Agent State updation required Info^n:
- (1) How world is evolving
- (2) How agent's action affect the word.



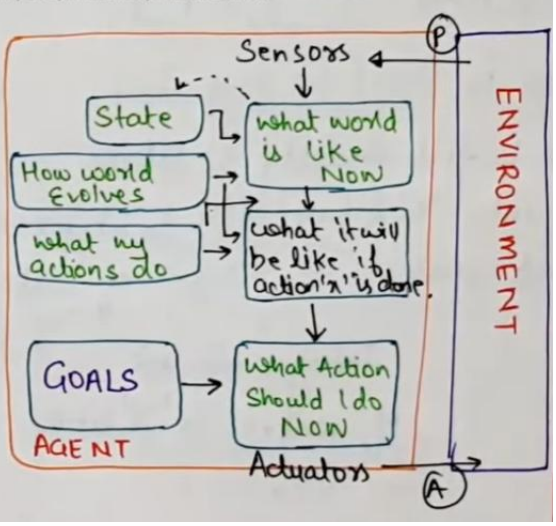
(3) Goal-Based Agents:

↳ Focuses only on reaching the goal set.

↳ Agent takes decision based on how far it is currently from the Goal State.

↳ Every action is taken to minimize distance to Goal State.

↳ More flexible Agent.

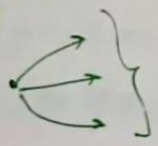


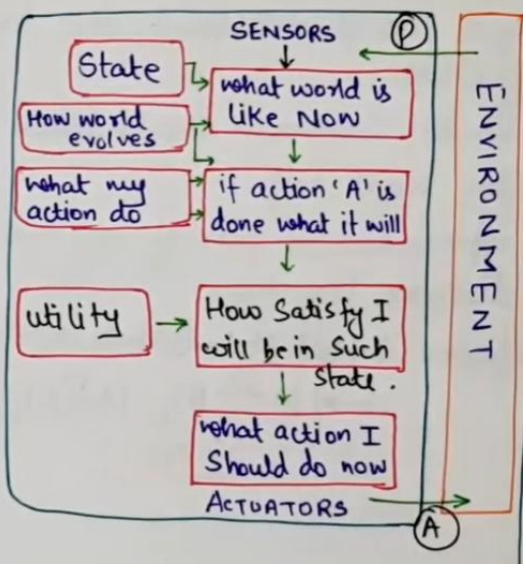
(4) Utility-Based Agents:-

↳ Agents are more Concerned about the utility (Preference) for each state.

↳ Act based not only on goals but also the best way to achieve goal.

↳ Useful when there are multiple possible alternatives and agent has to choose in order to perform best action.

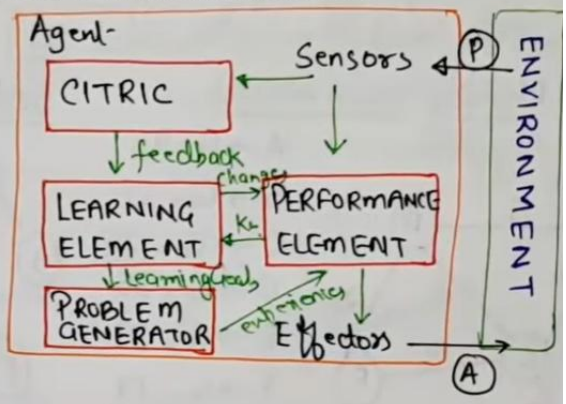




(5) Learning Agents:-

↳ Can Learn from its past experiences.

↳ Starts to ACT with basic knowledge and then able to act by adapting Learning.



Components:-

(i) Learning Element:

(ii) Citric:

(iii) Performance Element:

(iv) Problem Generator:

Components:-

(i) Learning Element: Makes improvement in System by learning from env.

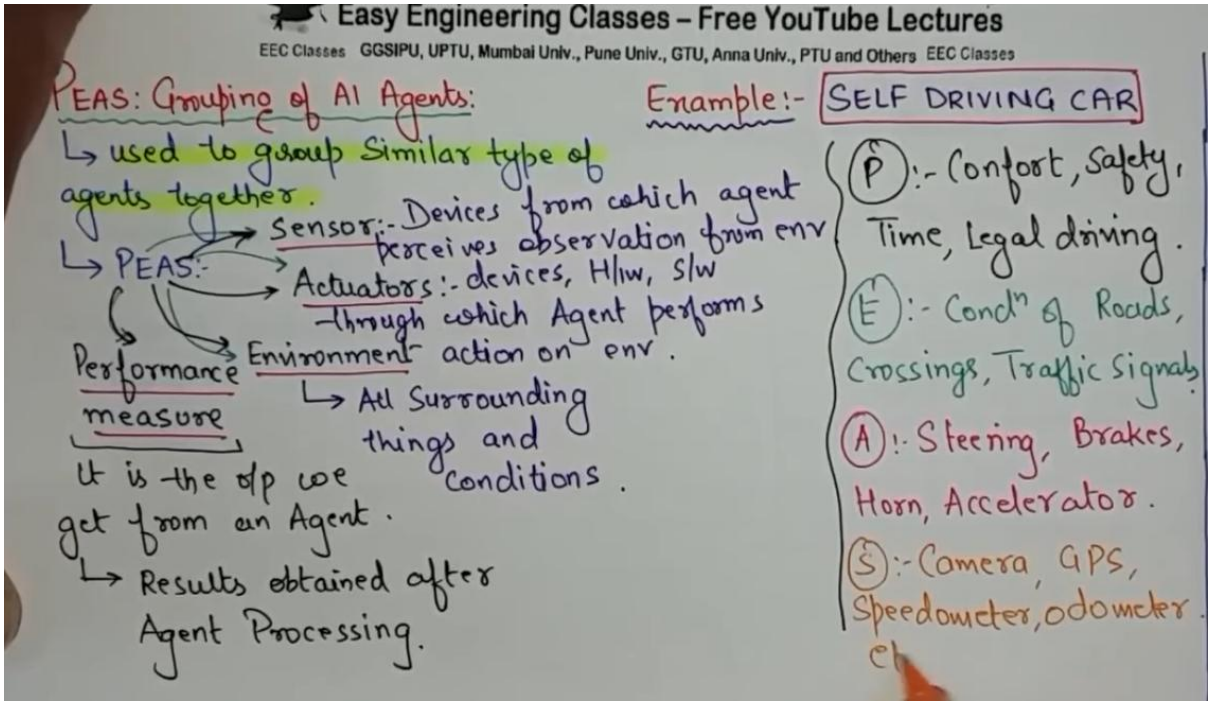
(ii) Citric: Gives feedback about Agent's Performance based on Standard.

(iii) Performance Element: Selects the action to Perform

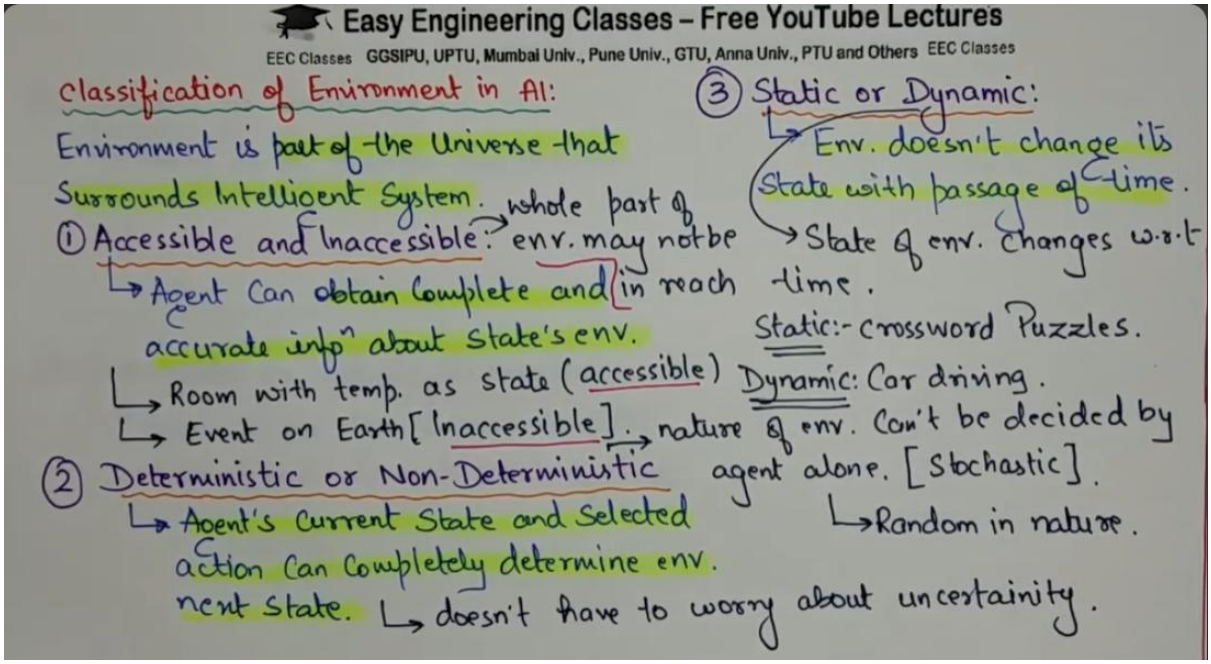
(iv) Problem Generator: Suggests the action.] new info gain.

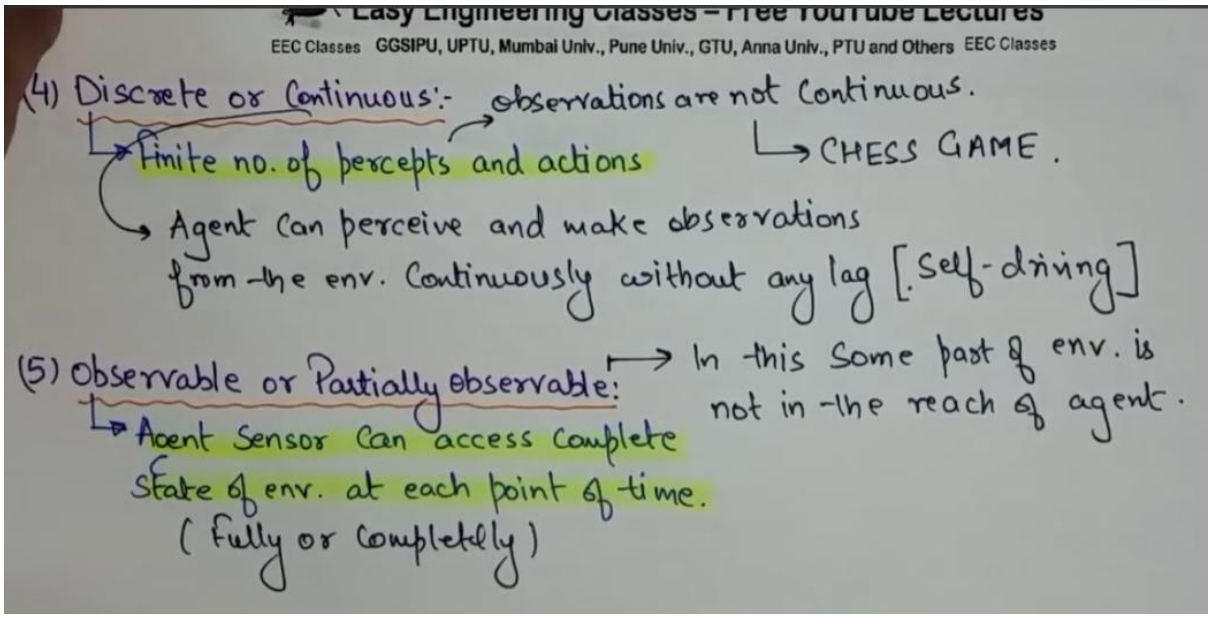
PEAS in Artificial Intelligence | Grouping of AI Agents with Example of Self Driving Car

https://www.geeksforgeeks.org/understanding-peas-in-artificial-intelligence/

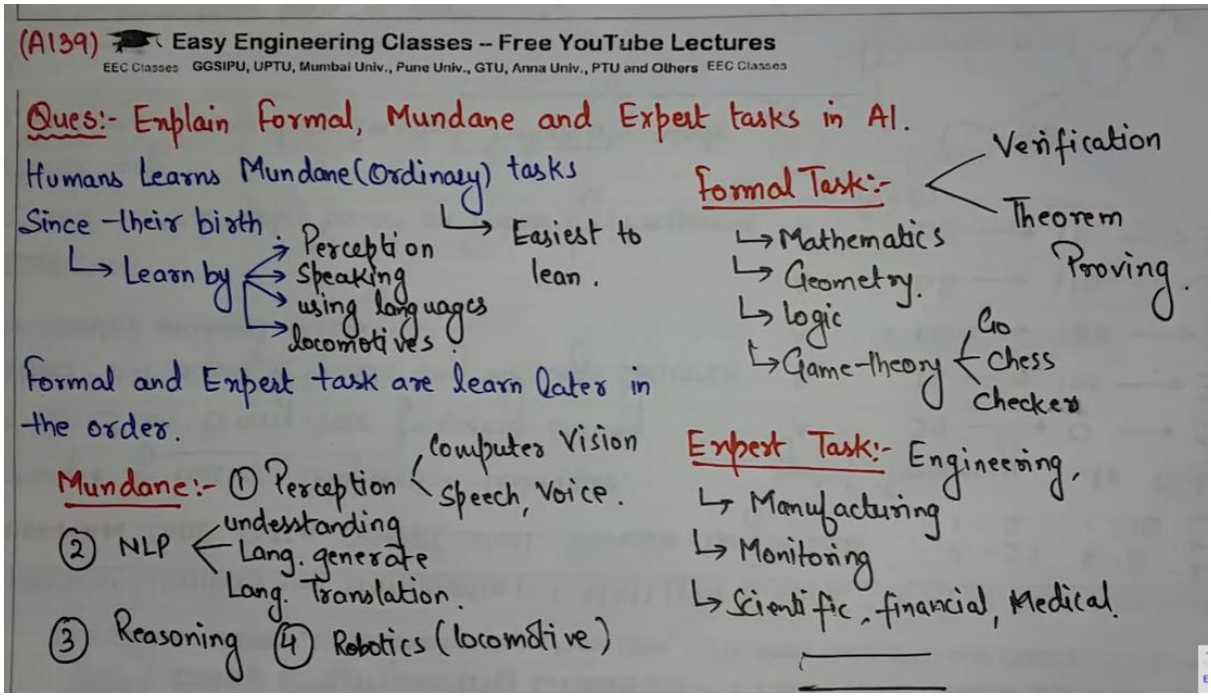


Classification of Environment in Artificial Intelligence

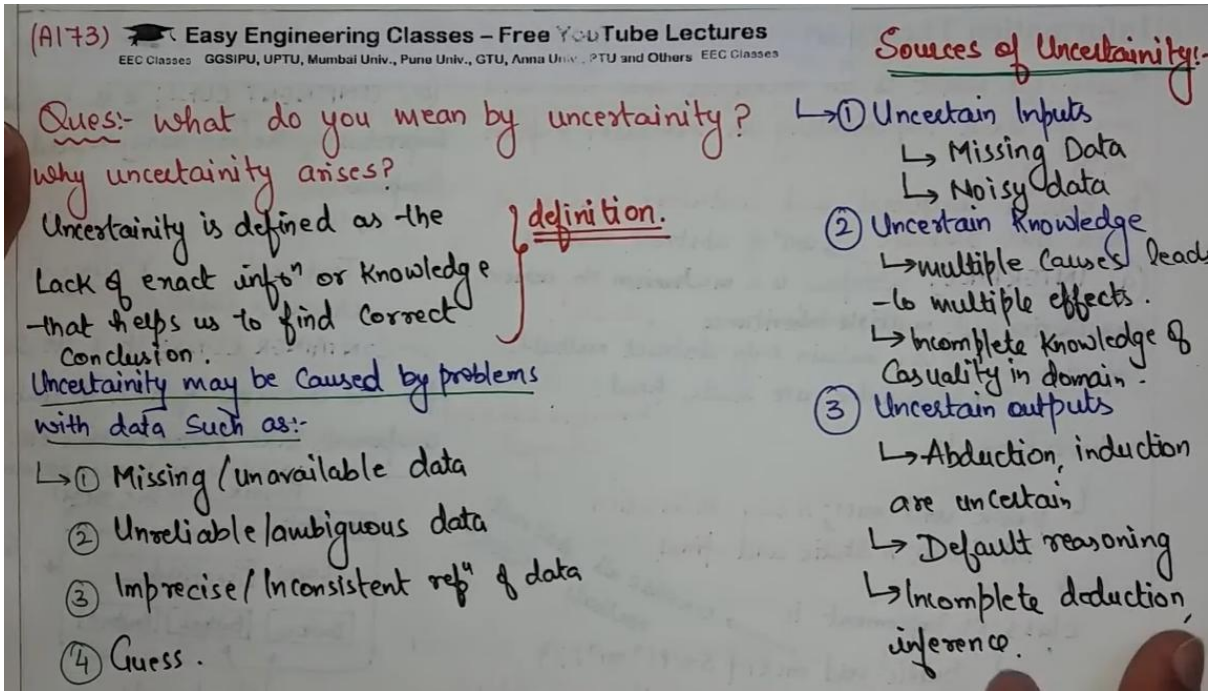


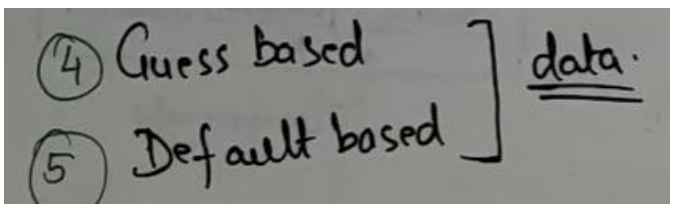


Artificial Intelligence Task | Formal, Expert, Mundane Task

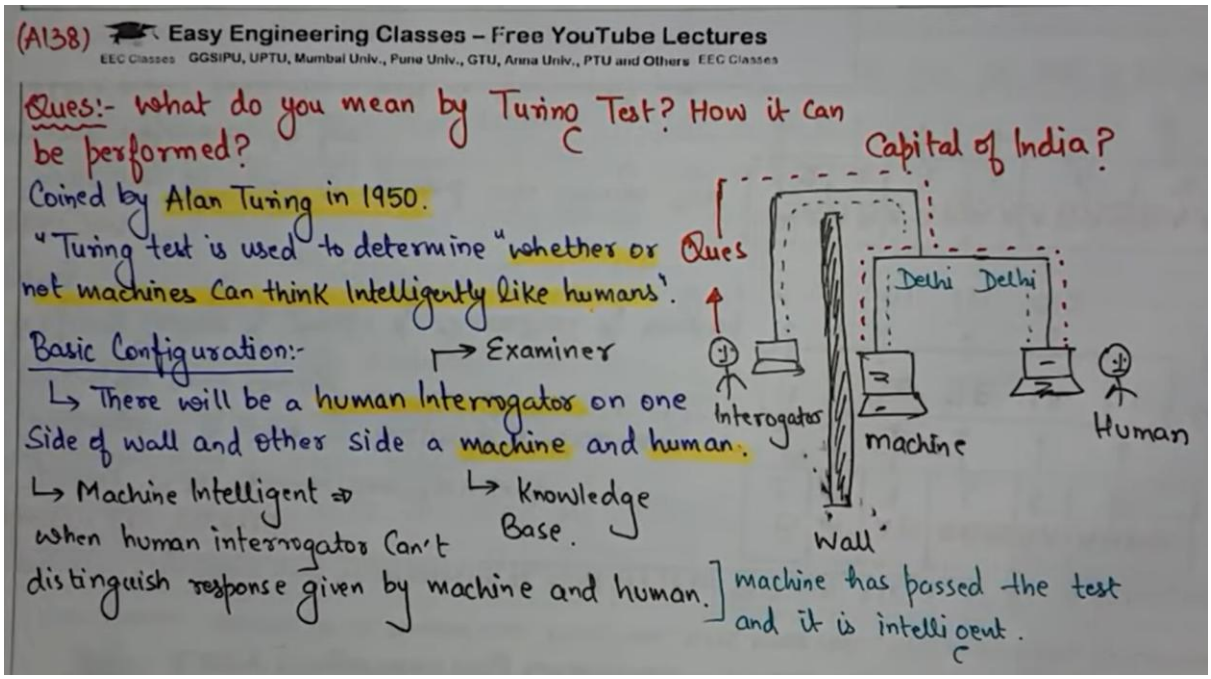


Uncertainty in Artificial Intelligence | Sources of Uncertainty

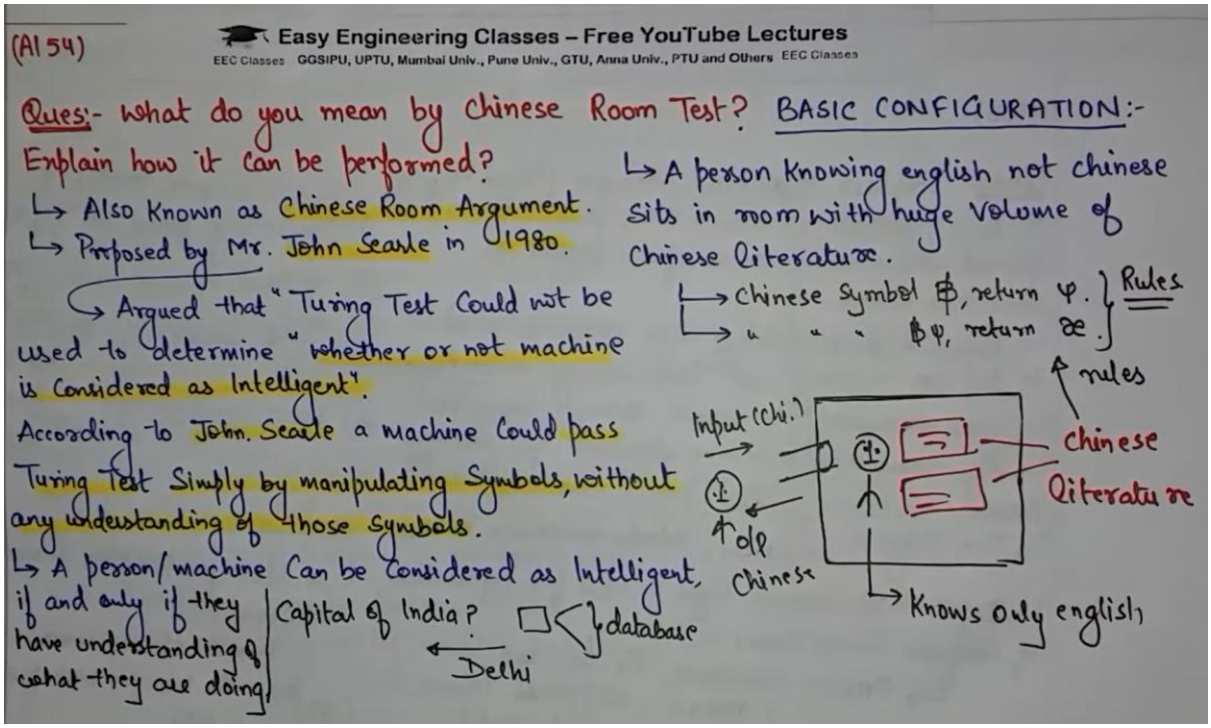




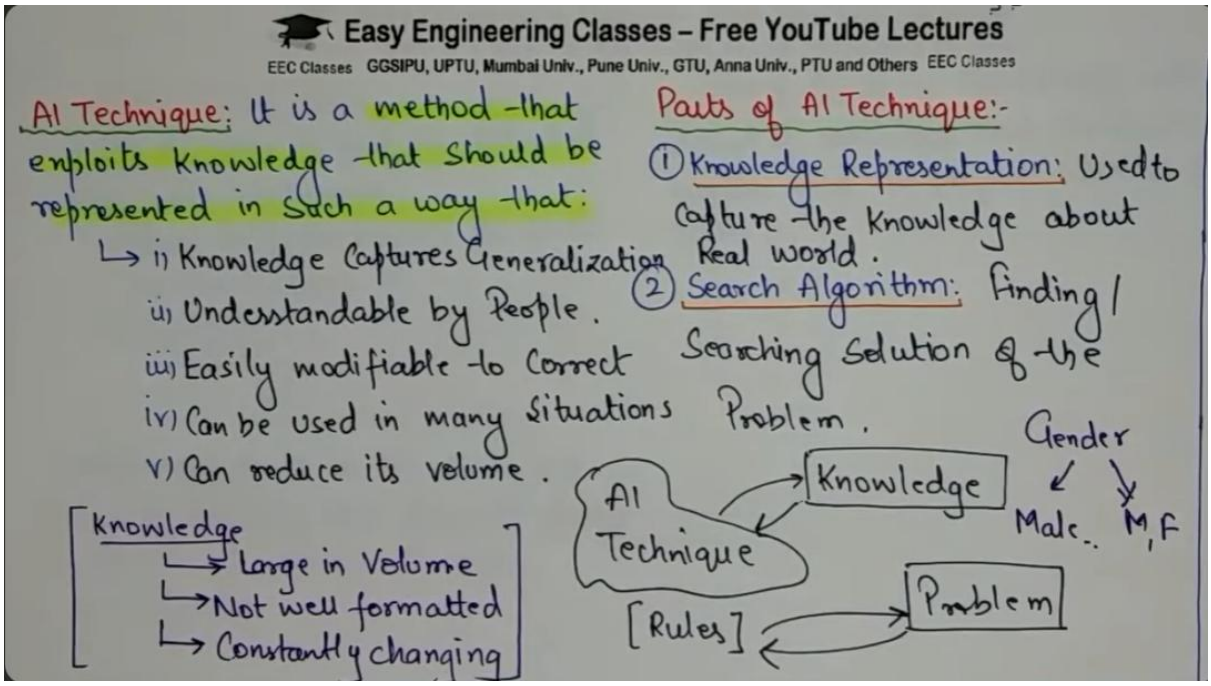
Turing Test in Artificial Intelligence with Configuration and Steps to Perform

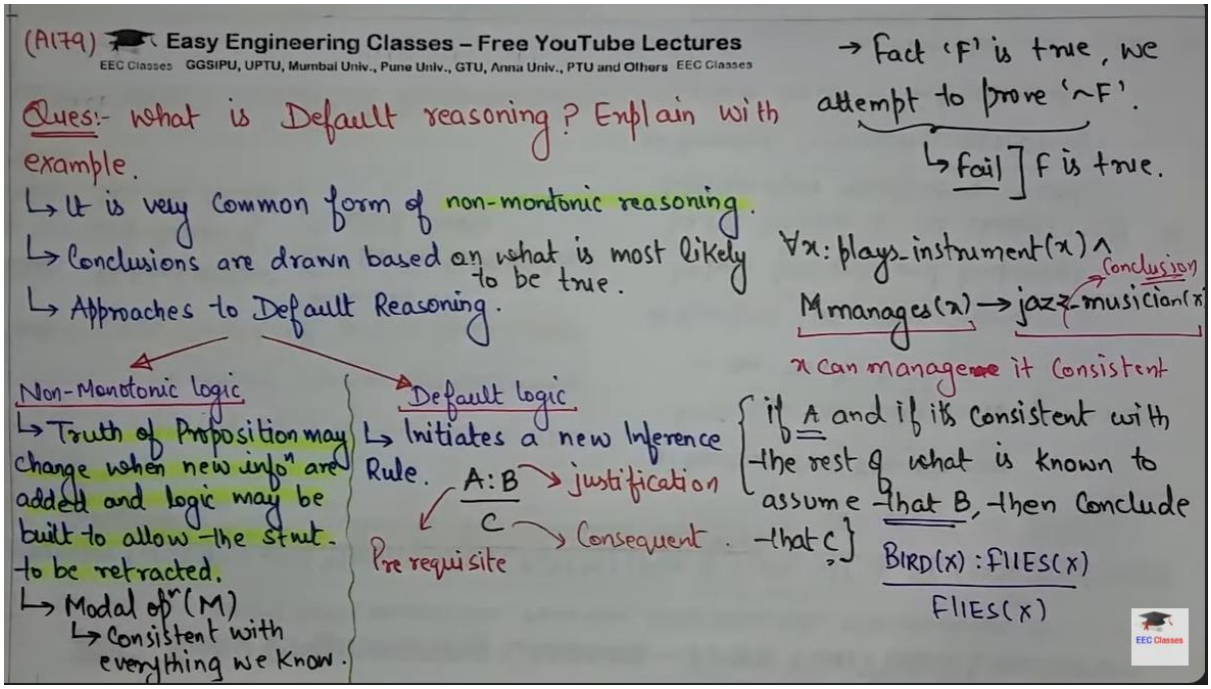


Chinese Room Test in Artificial Intelligence with Configuration & Steps to Perform

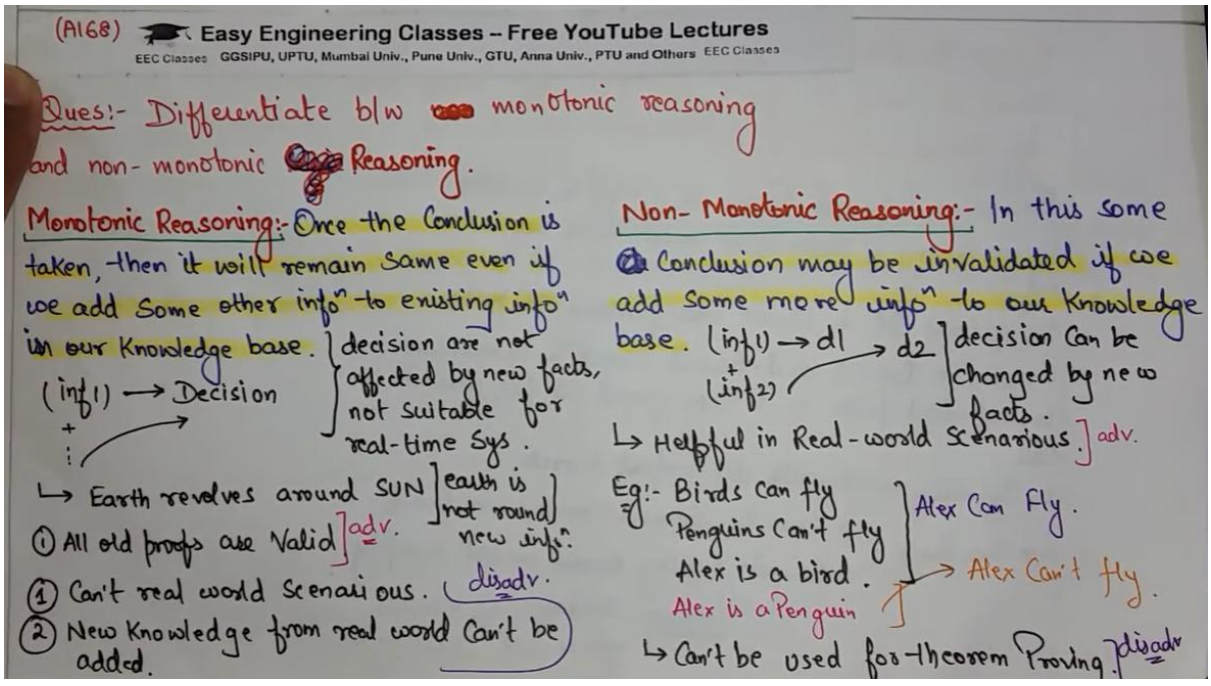


Artificial Intelligence Technique | Knowledge Representation, Search Algorithm

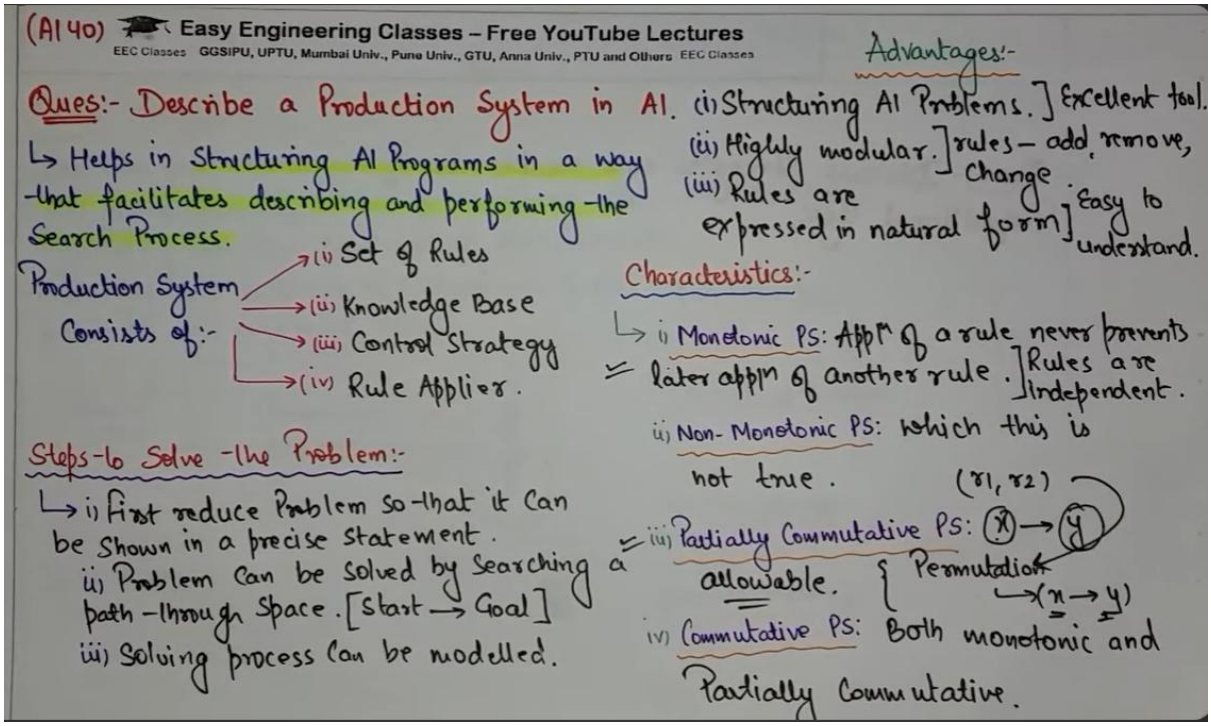




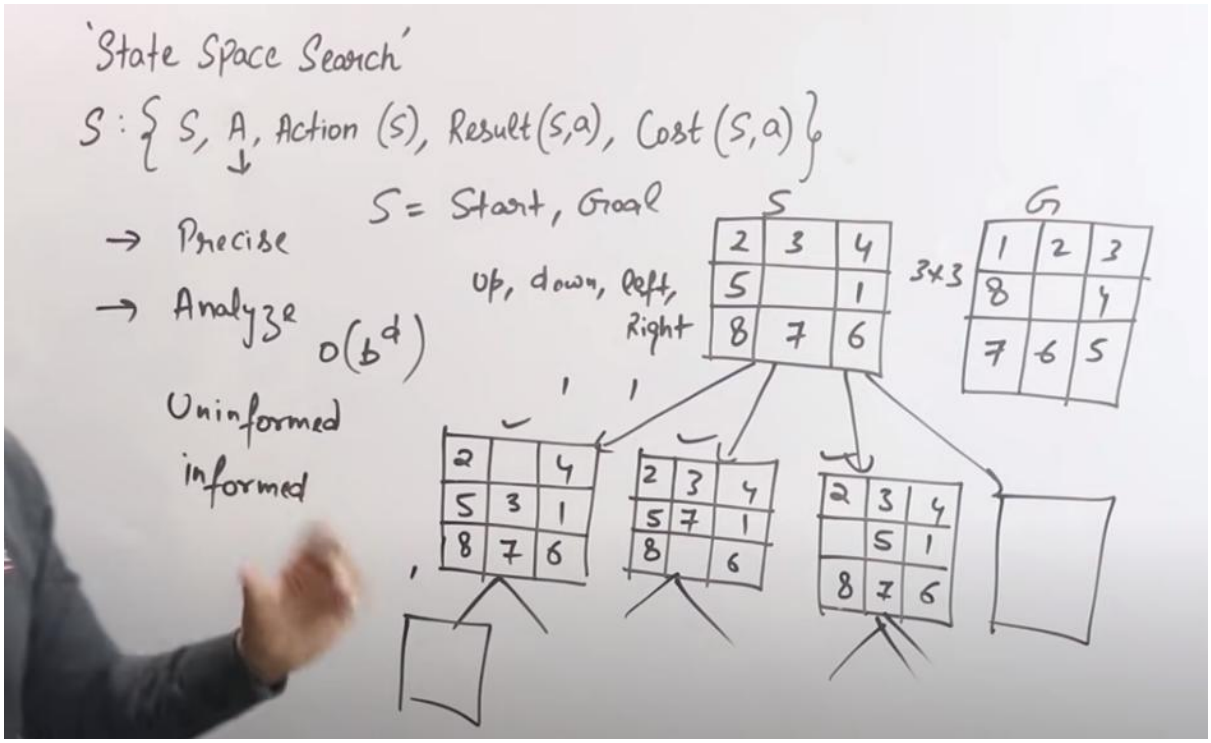
Monotonic vs Non-Monotonic Reasoning in Artificial Intelligence

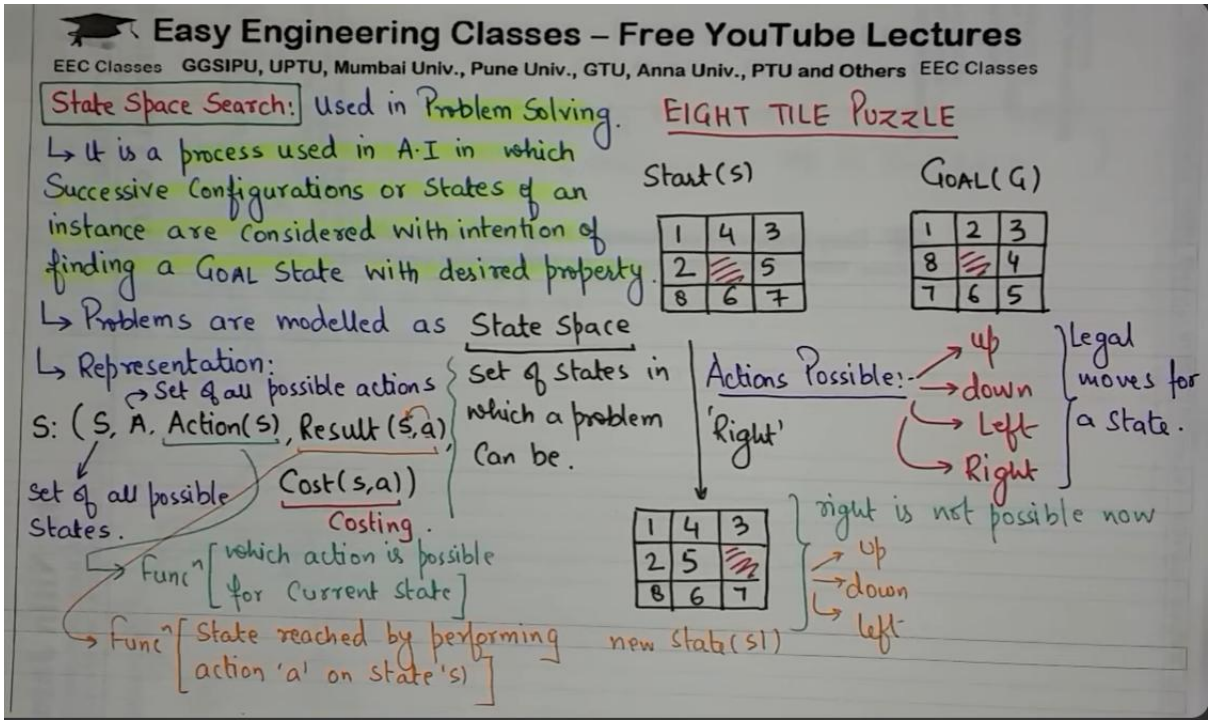


Production System in Artificial Intelligence

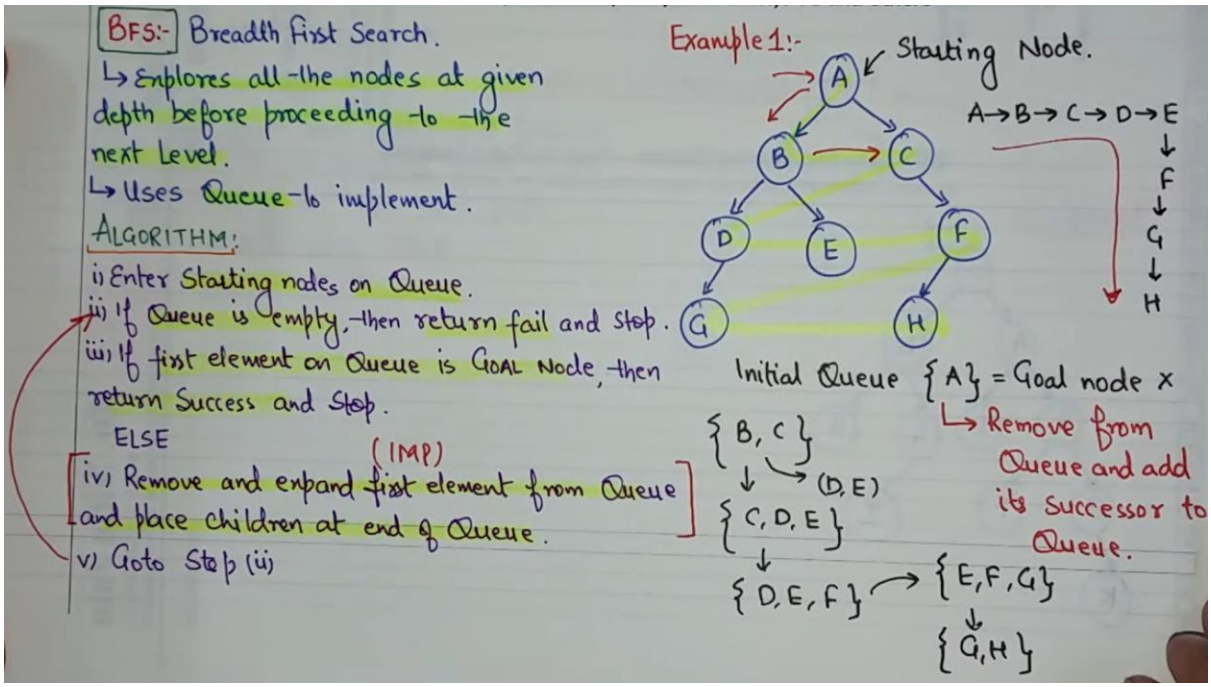


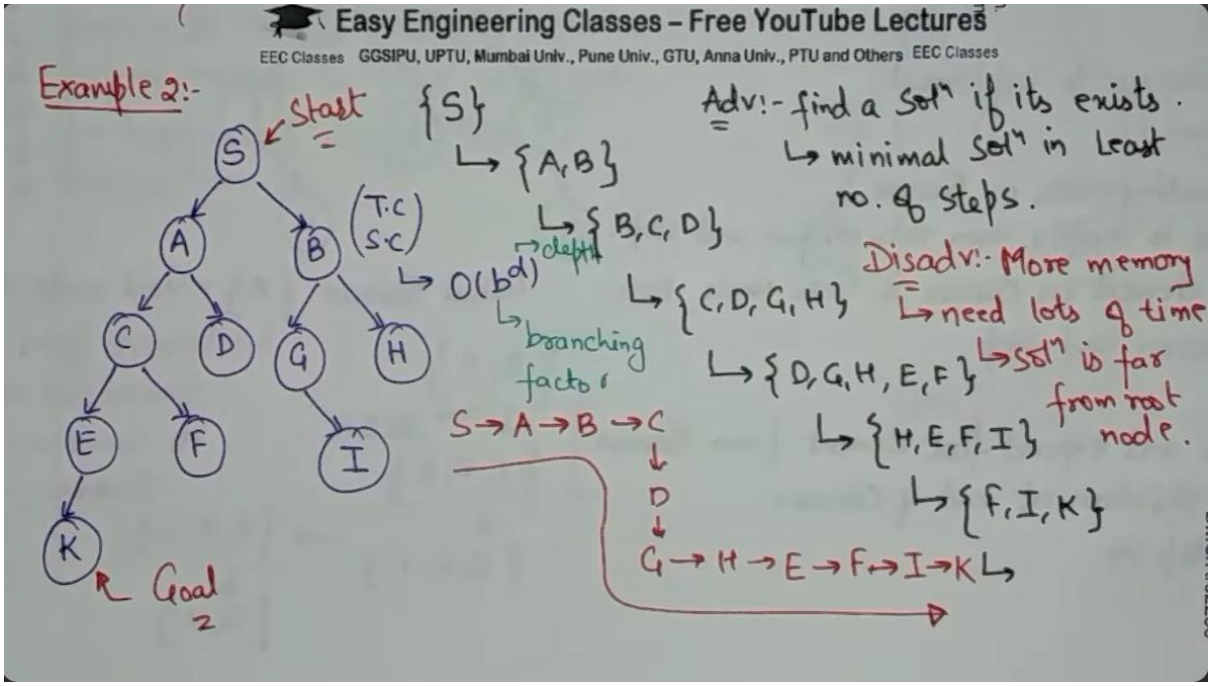
State Space Search



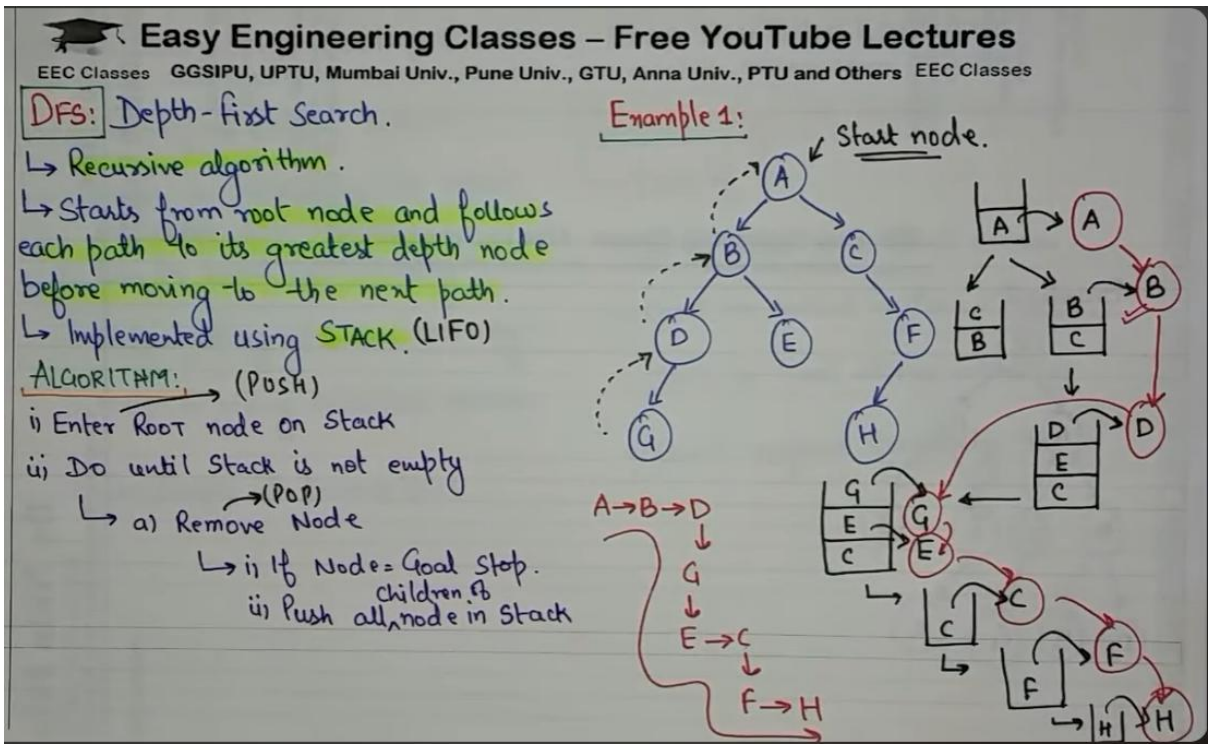


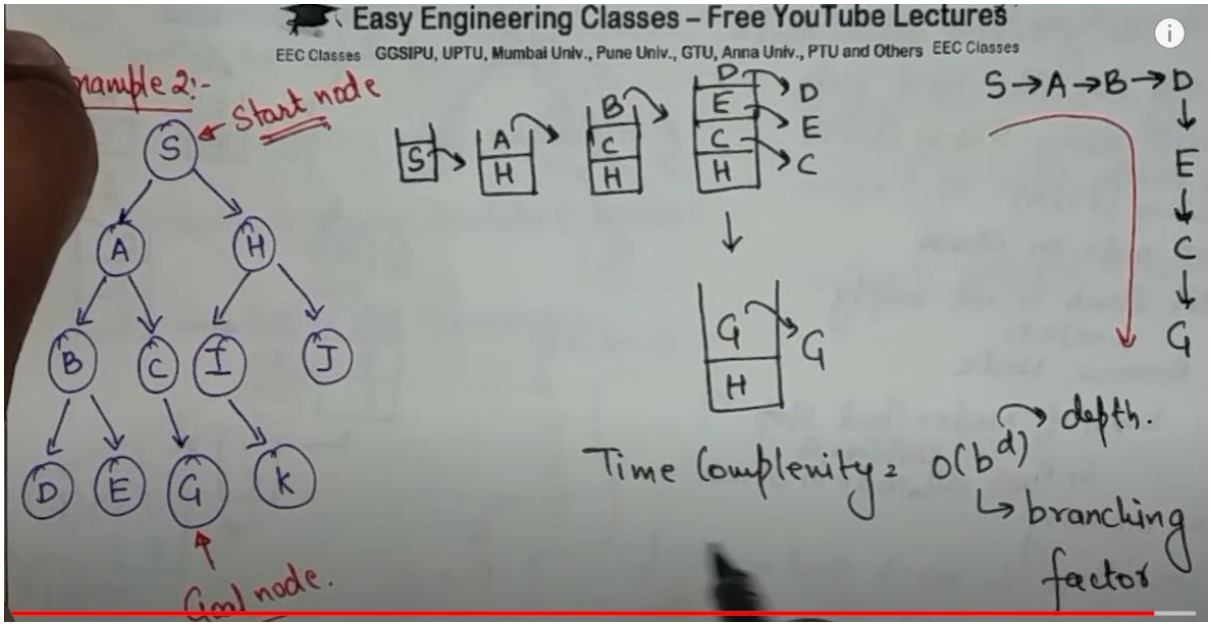
Breadth First Search





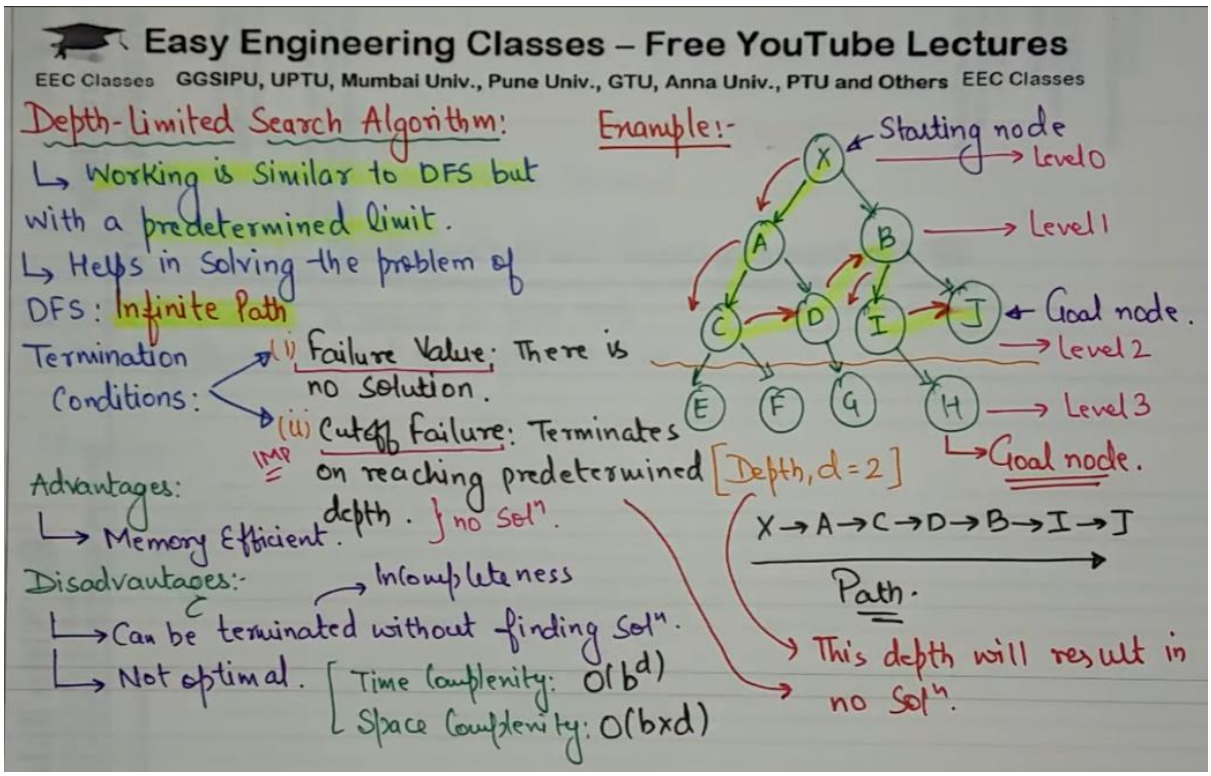
Depth First Search:





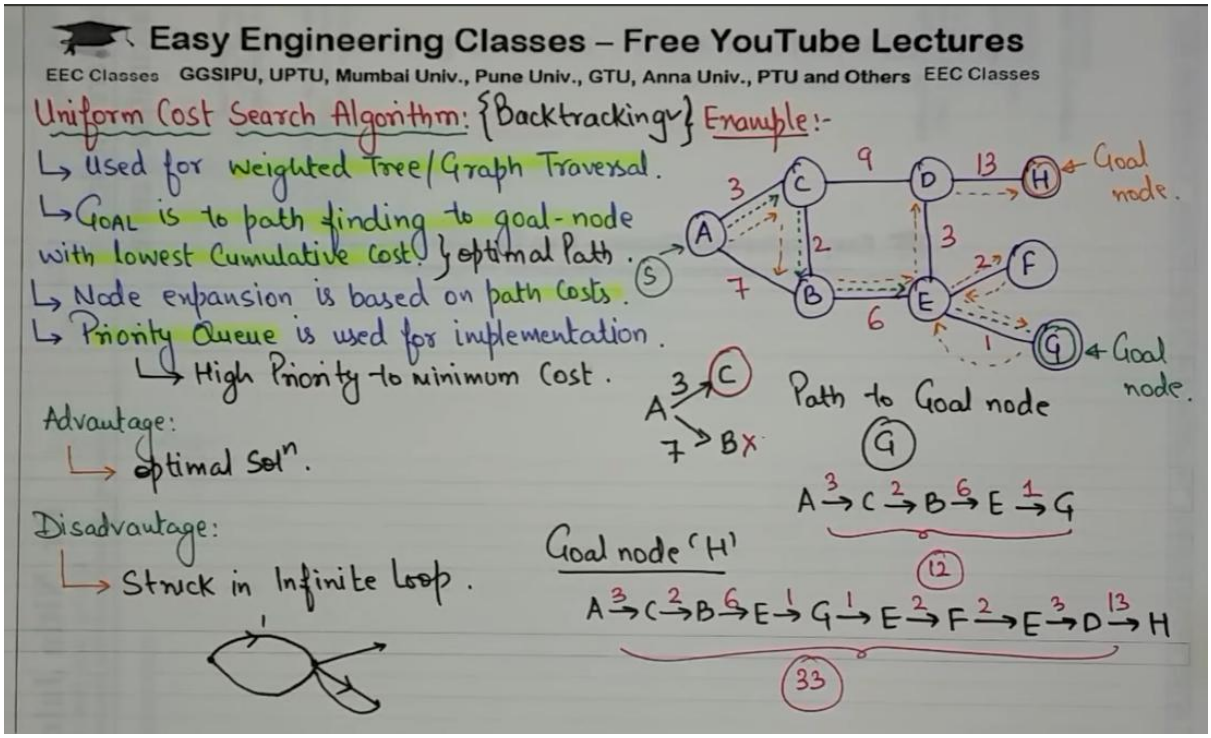
Depth Limited Search

https://iq.opengenus.org/depth-limited-search/



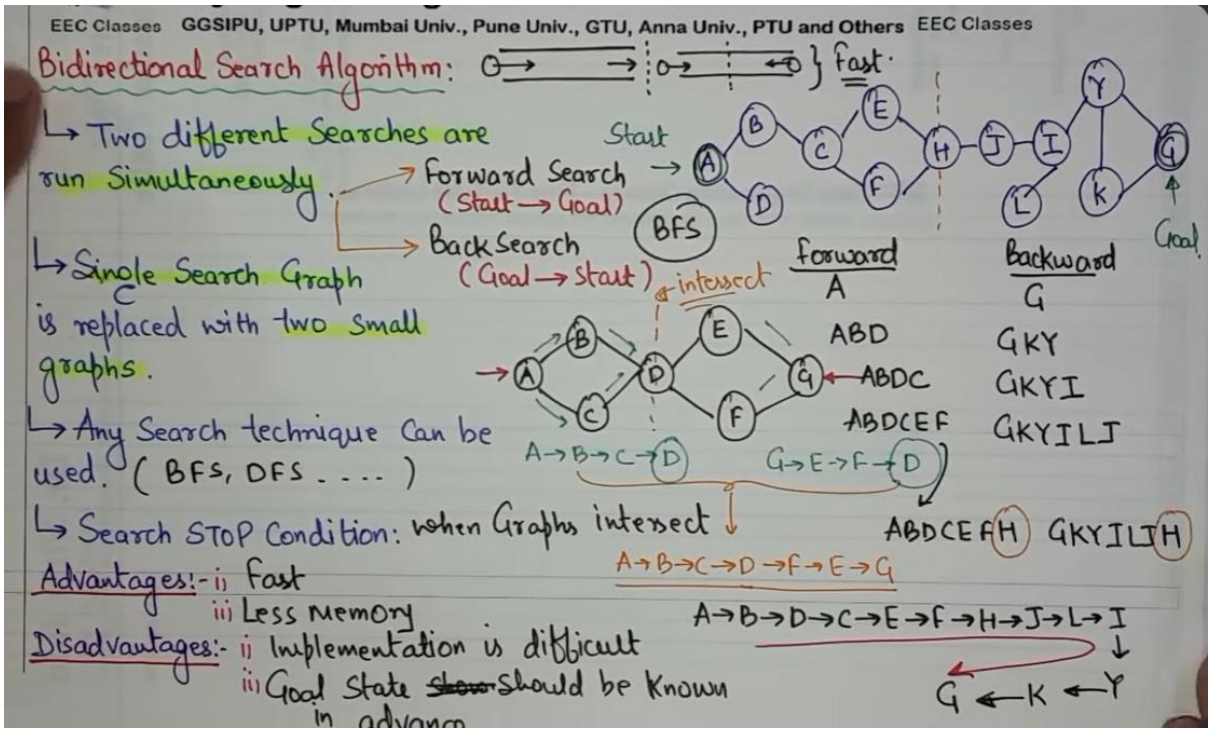
Uniform Cost Search

https://www.educative.io/answers/what-is-uniform-cost-search



Bi-directional search

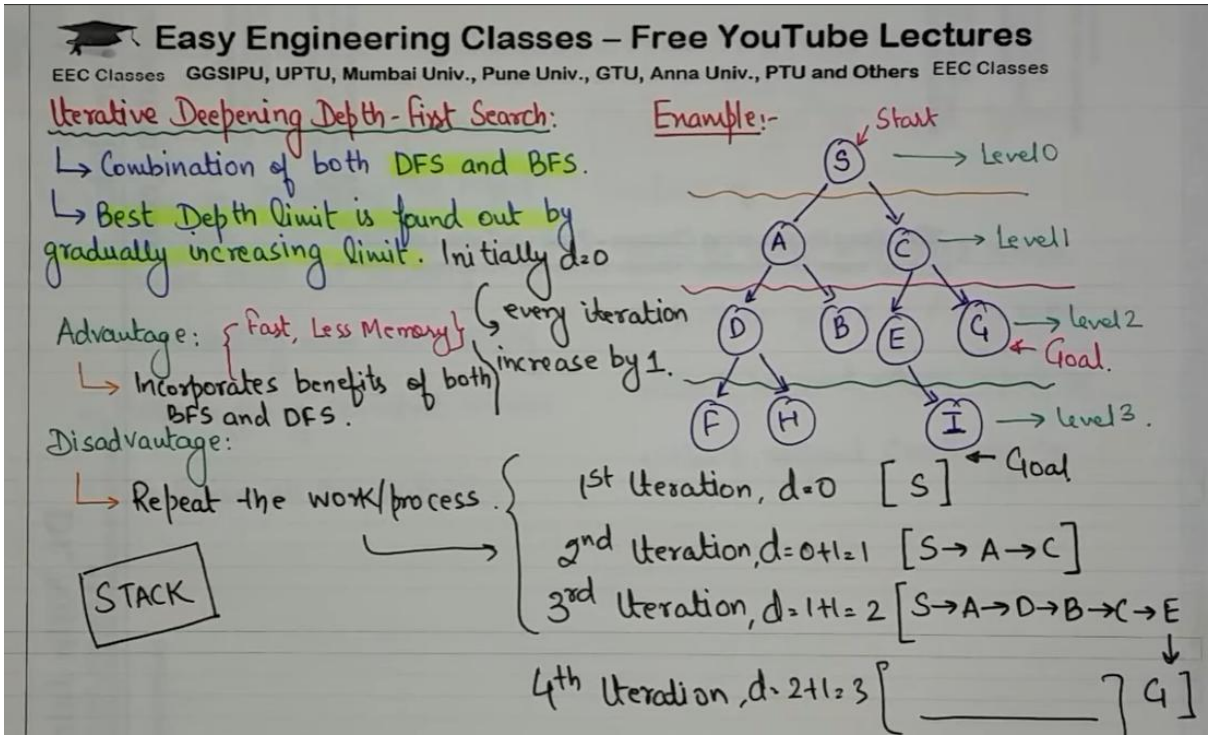
https://iq.opengenus.org/bidirectional-search/



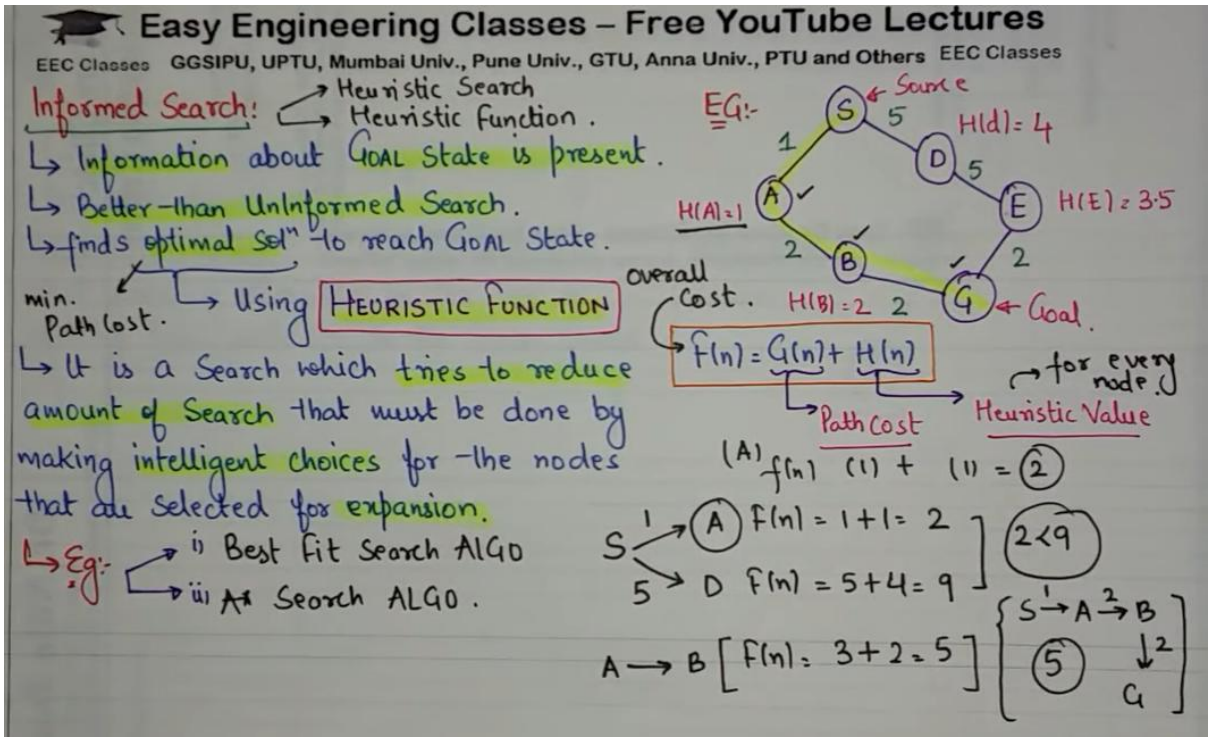
Iterative Deepening Depth First Search

https://iq.opengenus.org/iterative-deepening-search/

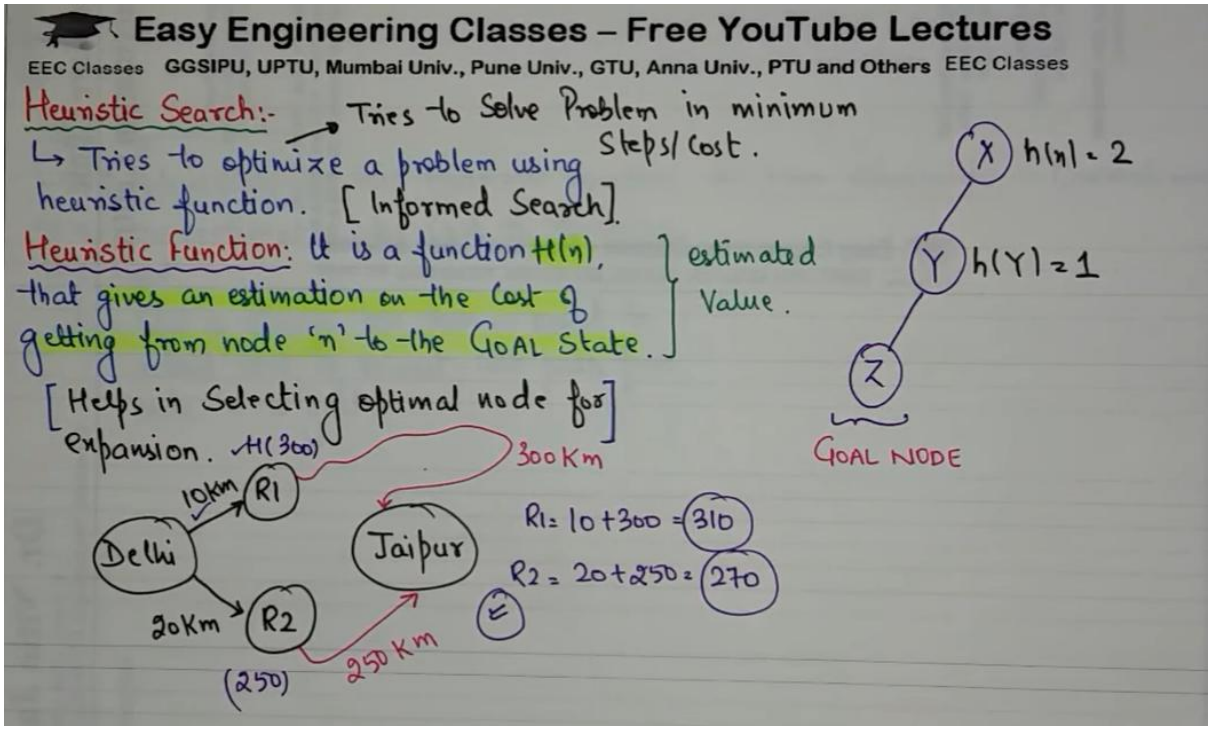
https://www.educative.io/answers/what-is-iterative-deepening-search



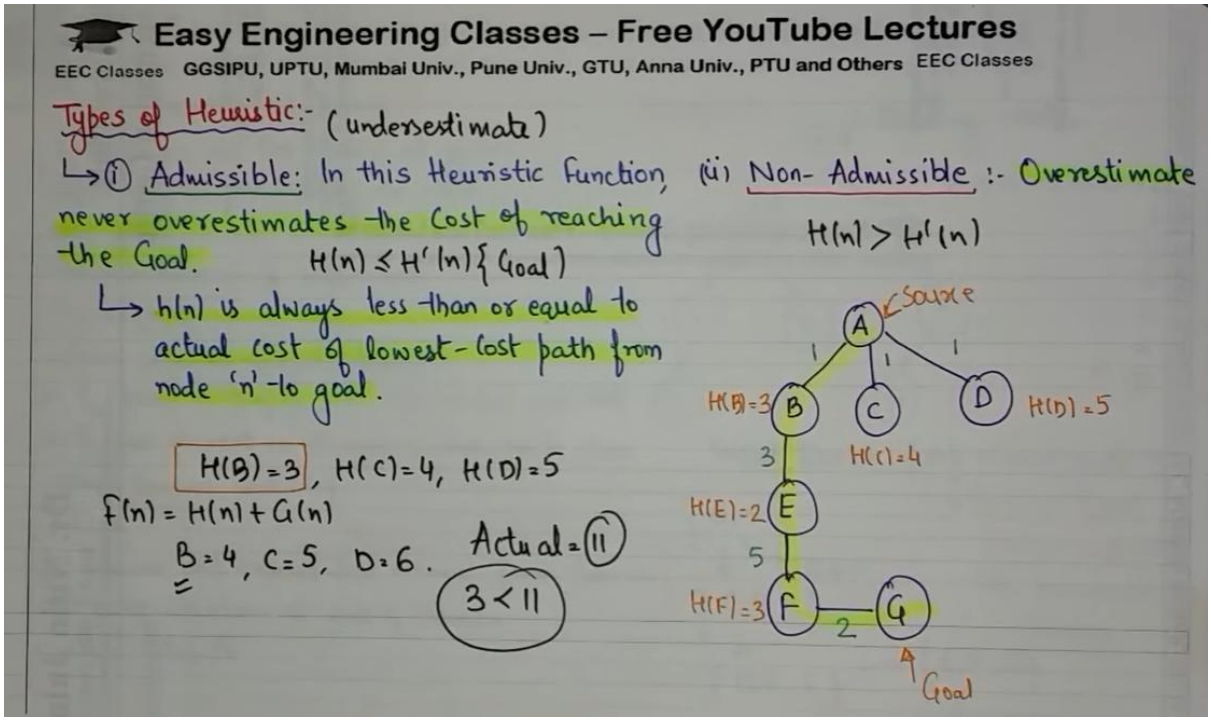
Informed Search:



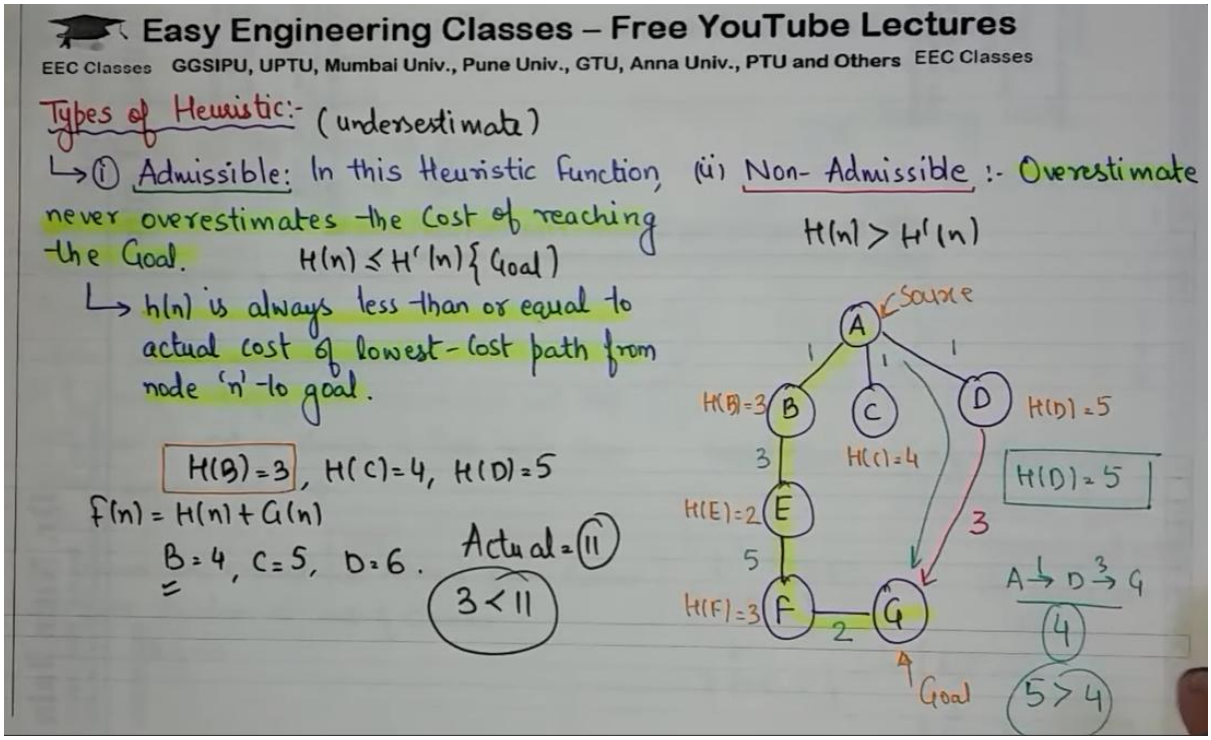
Heuristic Search



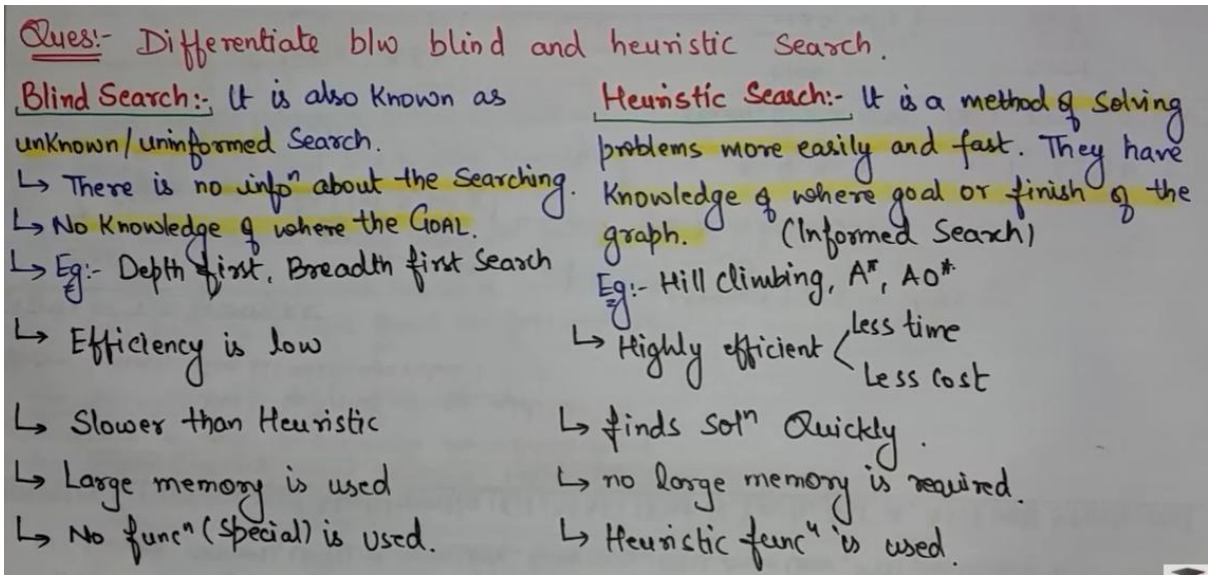
Admissible Heuristic Search



Non-Admissible Heuristic Search

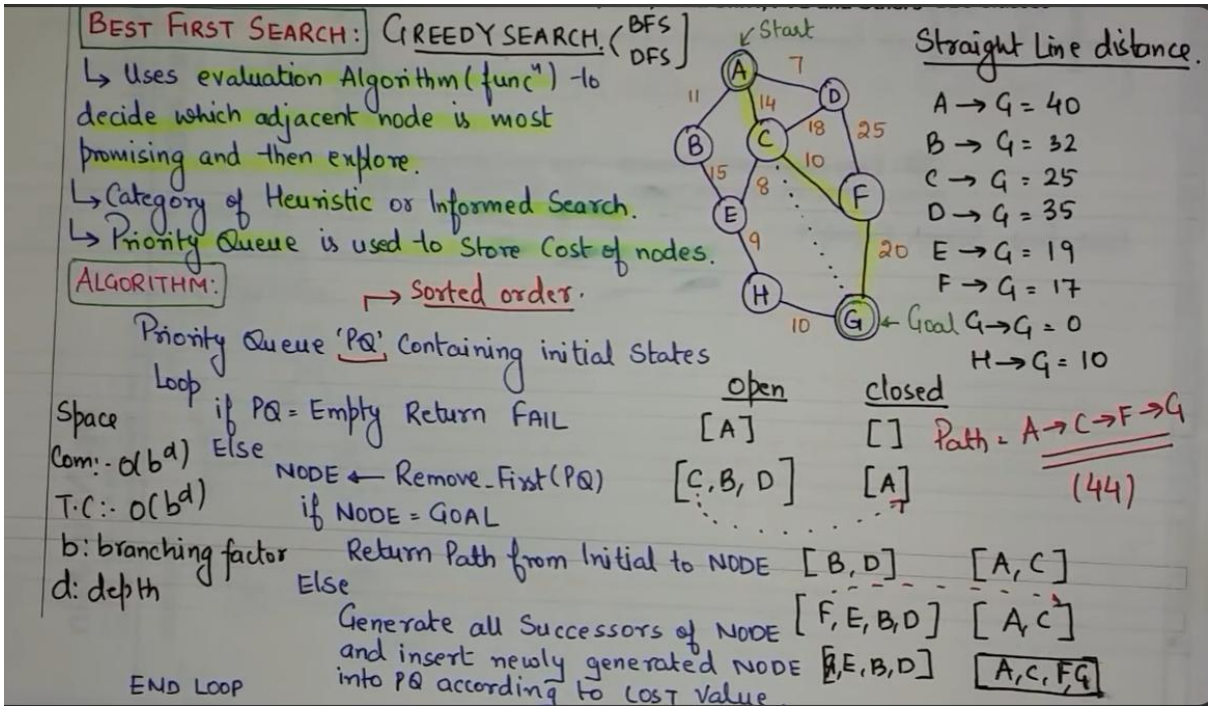


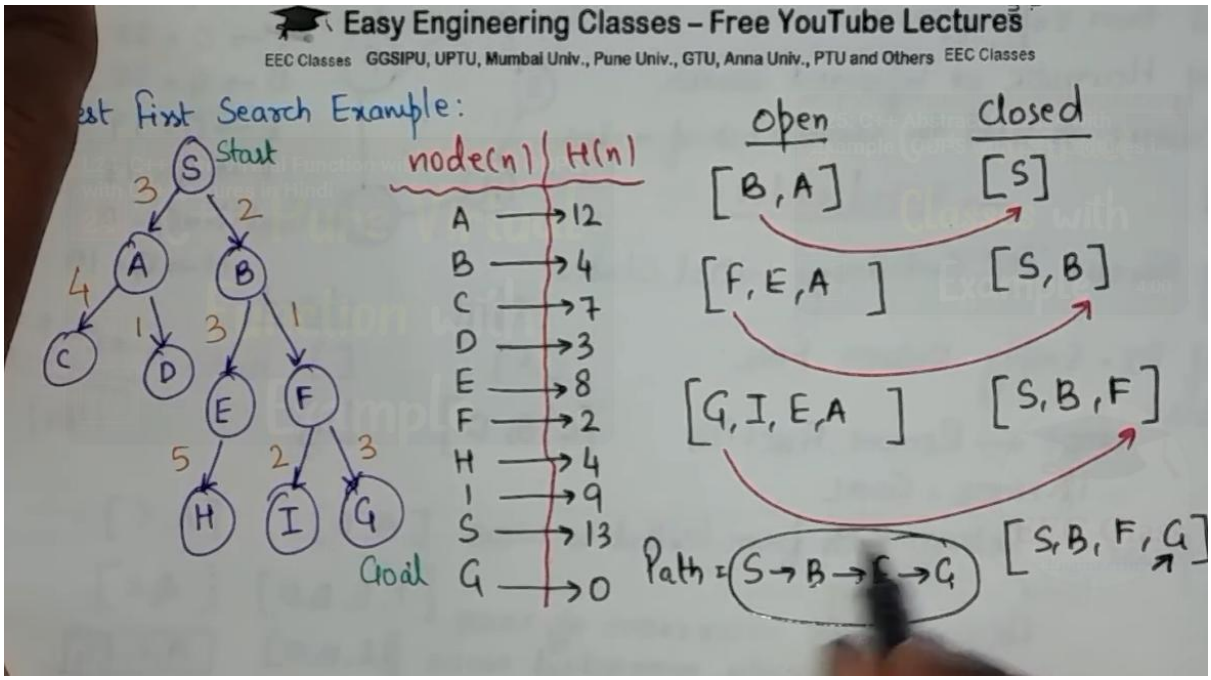
Difference between Blind and Heuristic Search



Best First Search

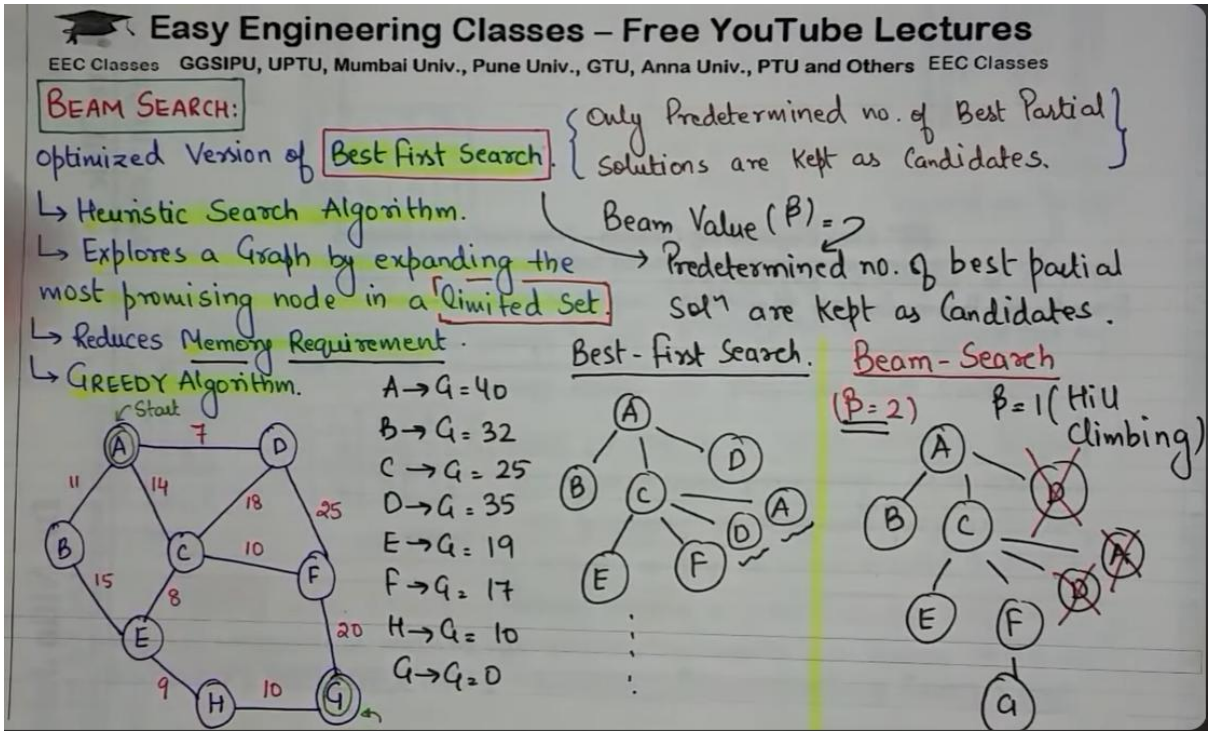
https://www.geeksforgeeks.org/best-first-search-informed-search/





Beam Search

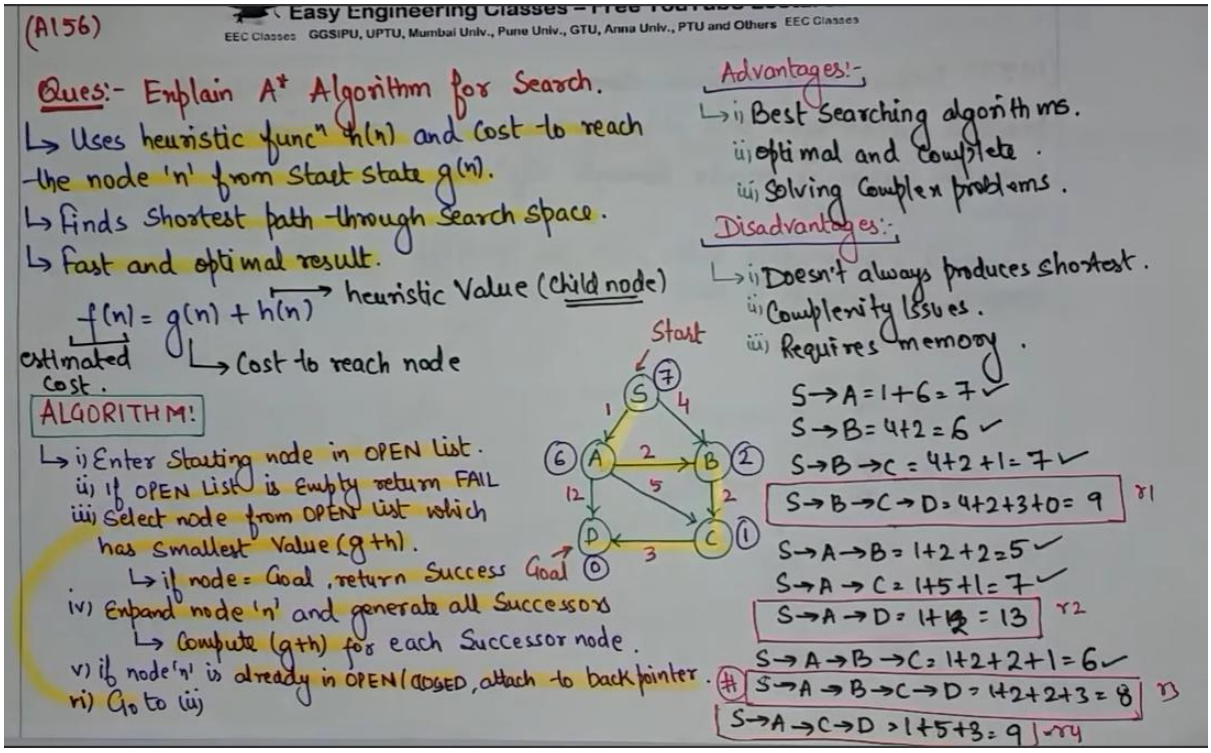
https://www.javatpoint.com/define-beam-search



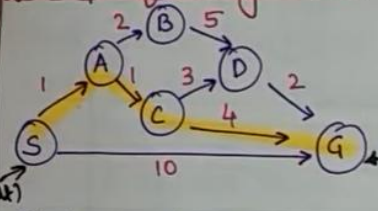
A* Search

https://www.gatevidyalay.com/a-algorithm-a-algorithm-example-in-ai/

https://www.educative.io/answers/what-is-the-a-star-algorithm



Example of A* Algorithm:-



State	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

$S \to A = 1+3 = 4$

$S \to G = 10+0 = 10$

$S \to A \to B = 1+2+4 = 7$ ✓

$S \to A \to C = 1+1+2 = 4$ ✓

$S \to A \to C \to D = 1+1+3+6 = 11$ ✓

$= S \to A \to C \to G = 1+1+4 = 6$.

$S \to A \to B \to D = 1+2+5+6 = 14$ =

$S \to A \to C \to D \to G = 1+1+3+2 = 7$

$S \to A \to B \to D \to G = 1+2+5+2 = 10$

(2) A* Search Algorithm:

→ A* search Alg finds the shortest path through the search space using the heuristic function.

→ It uses h(n), & cost to reach the node n from the start st g(n).

→ This alg expands less search tree and provides optimal resu faster.

→ It is similar to UCS except that it uses g(n) + h(n) ins of g(n)

g(n)

→ This alg expands less search tree and provides optimal result faster.

→ It is similar to UCS except that it uses g(n) + h(n) inste of g(n)

> A* use search heuristic as well as the cost to reach the node. Hence we combine both costs as,

$$f(n) = g(n) + h(n) \quad \{\text{fitness number}\}$$

$f(n)$ = Estimated cost of the cheapest soln

$g(n)$ = Cost to reach node n from start state

$h(n)$ = Cost to reach from node n to goal node.

Algorithm of A* Search

Step1: Place the starting node in OPEN list

Algorithm of A* Search

Step1: Place the starting node in OPEN list

Step2: check if the OPEN list is Empty or not, if the list is Empty then return failure & stops.

Step3: Select the node from the OPEN list which has the smal value of evaluation function (g+h) if node n is goal no then return success & stop, otherwise

Step4: Expand node n & generate all of its successors, & put n in the closed list.

- For each successor 'n', check whether 'n' is already in the OPEN or CLOSED list,
- If not then compute evaluation function for 'n' an place into OPEN list.

Step 5: Else if node 'n' is already in OPEN & CLOSED, then it should be attached to the back pointer which reflects the lowest g(n') value.

Step 6: Return to step 2

the lowest g(n') value.

Step 6: Return to step 2

Advantages:

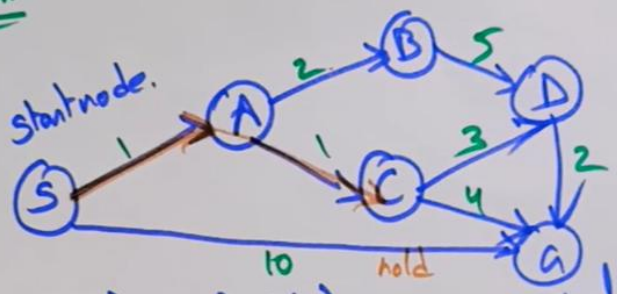
- It is best alg than other search alg
- It is optimal & complete
- It can solve very complex problems

Disadvantages:

- It does not always produce shortest path
- It is not practical for various large-scale problems

A* Search Algorithm

Example:



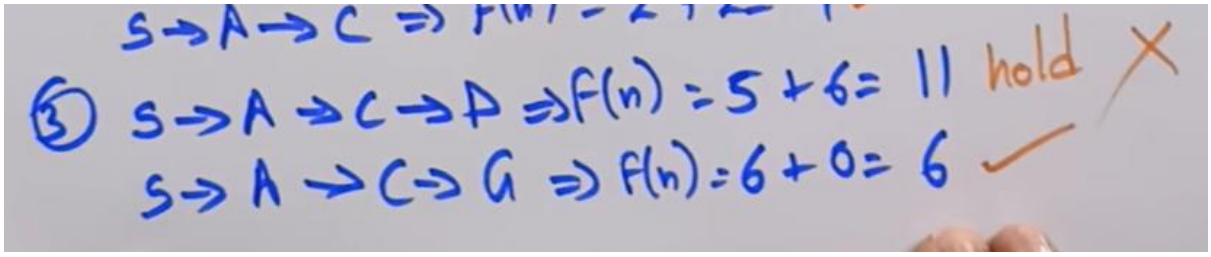
State	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

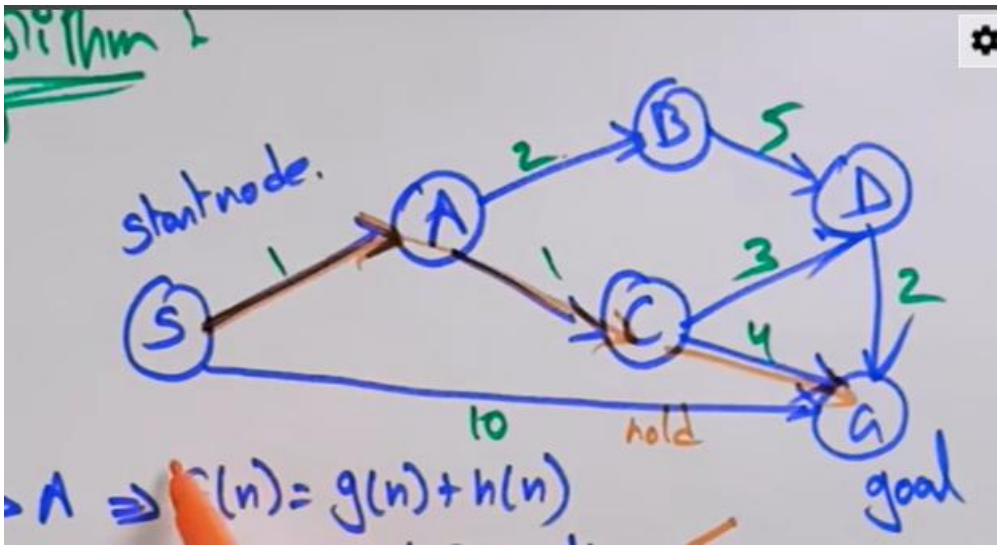
① $S \rightarrow A \Rightarrow f(n) = g(n) + h(n)$
$= 1 + 3 = 4$

$S \rightarrow G = f(n) = 10 + 0 = 10$ hold

② $S \rightarrow A \rightarrow B \Rightarrow f(n) = 3 + 4 = 7$ hold

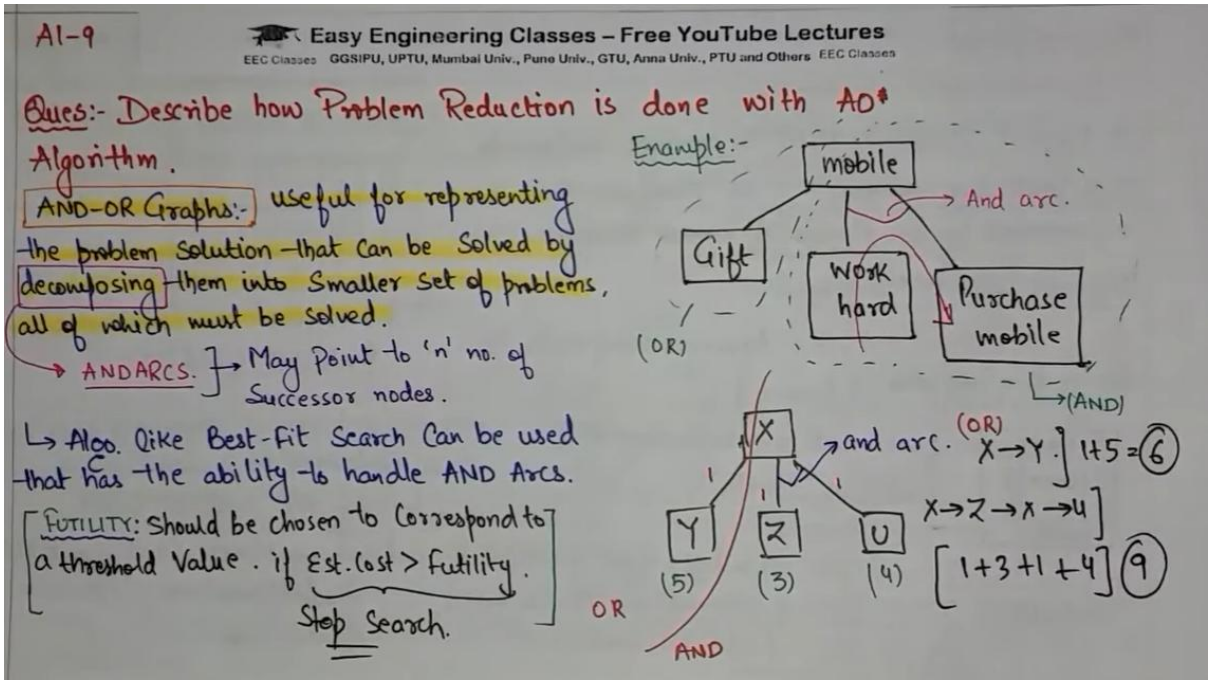
$S \rightarrow A \rightarrow C \Rightarrow f(n) = 2 + 2 = 4$



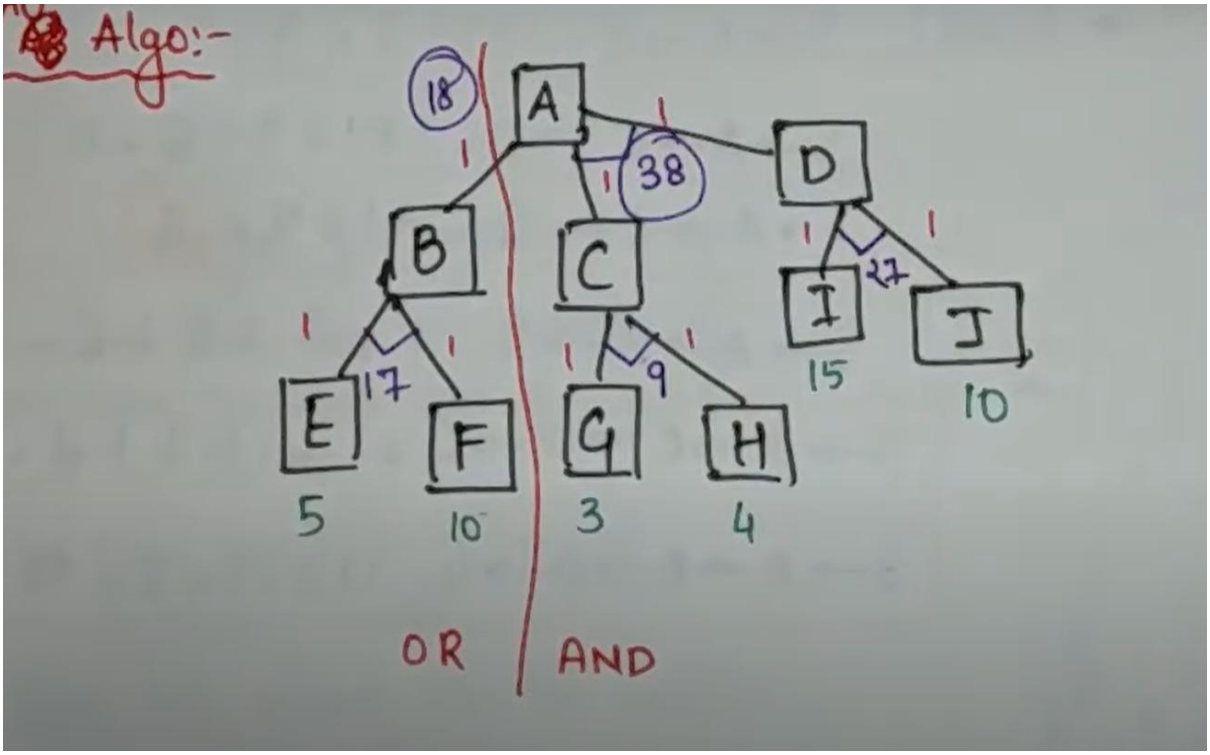


AO* Search

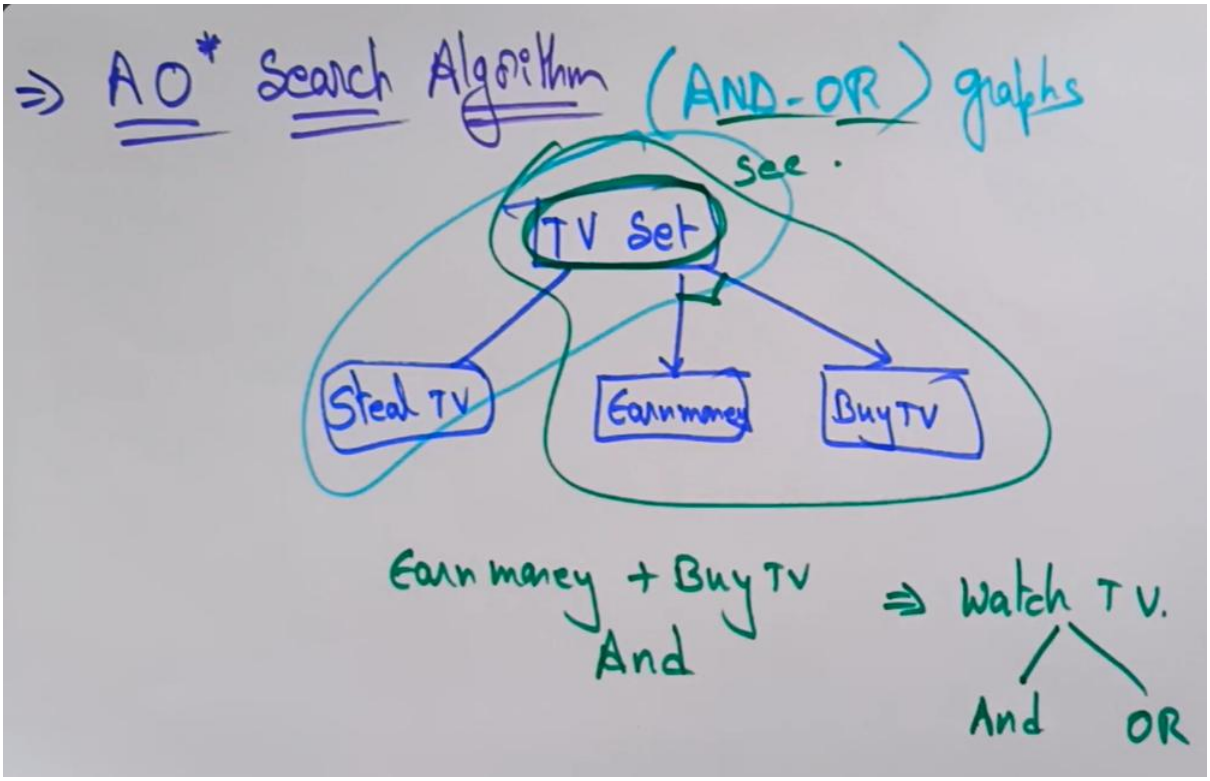
https://iq.opengenus.org/ao-algorithm/



Algo:-



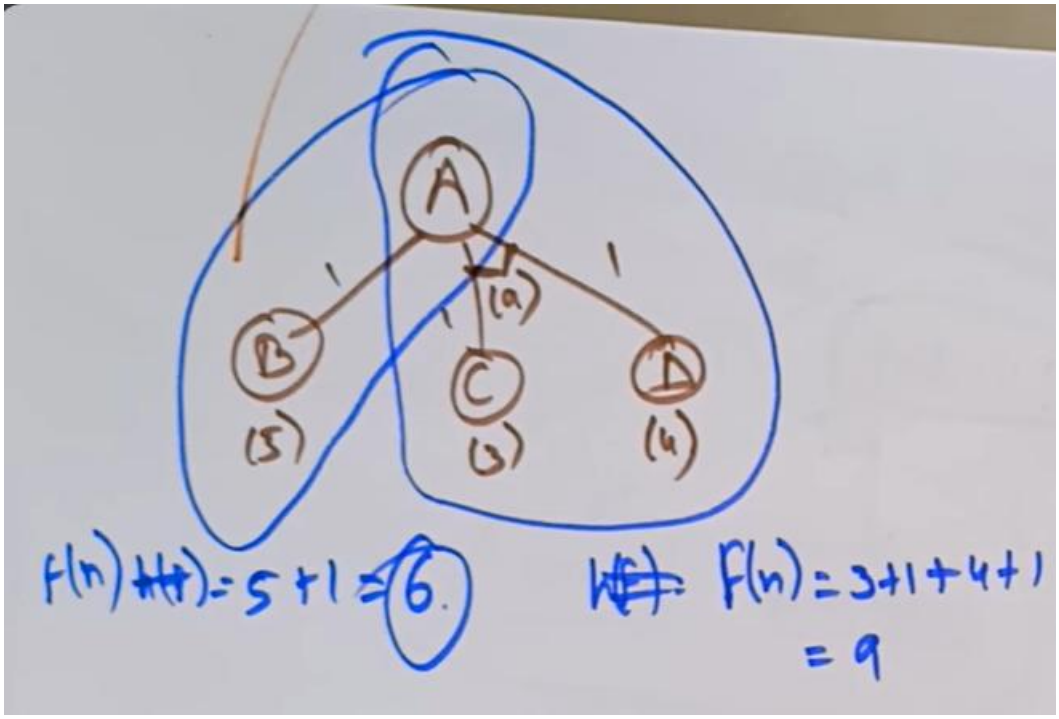
⇒ AO* Search Algorithm (AND-OR) graphs



Earn money + Buy TV ⇒ Watch TV.

And

And OR

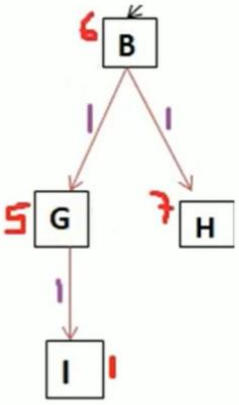


AO* Search Algorithm in Artificial Intelligence

- AO* algorithm is a heuristic search algorithm in AI.
- AO* algorithm uses the concept of AND-OR graphs to decompose any complex problem given into smaller set of problems which are further solved.
- **Working of AO* algorithm:**
- The AO* algorithm works on the formula given below :

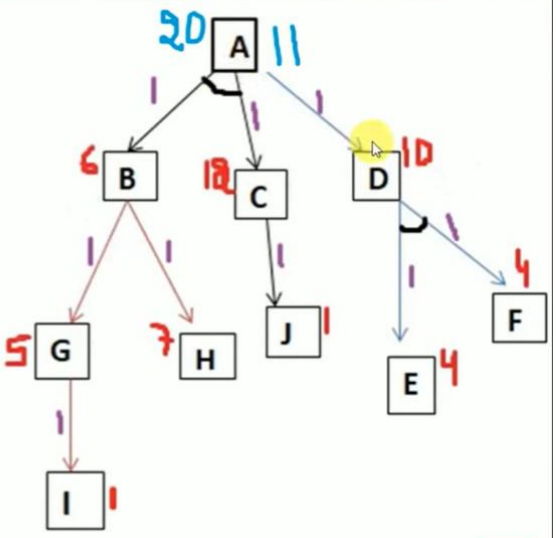
$$f(n) = g(n) + h(n)$$

- where,
- $g(n)$: The actual cost of traversal from initial state to the current state.
- $h(n)$: The estimated cost of traversal from the current state to the goal state
- $f(n)$: The actual cost of traversal from the initial state to the goal state.



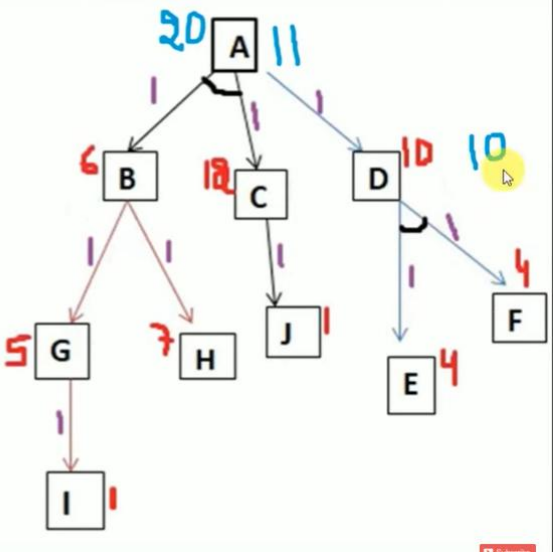
AO* Search Algorithm – Solved Example - Artificial Intelligence

- In the diagram we have two ways from **A to D** or **A to B-C** (because of and condition).
- Calculate cost to select a path
- **F(A-D) = 1+10 = 11**
- **F(A-BC) = 1 + 1 + 6 +12 = 20**
- As we can see **F(A-D)** is less than **F(A-BC)** then the algorithm choose the path **F(A-D).**



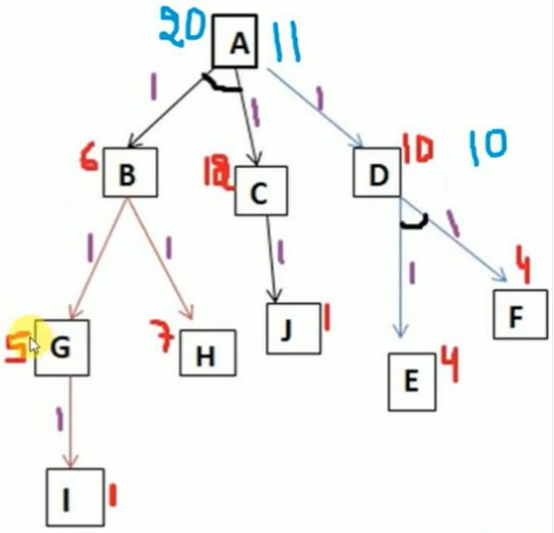
AO* Search Algorithm – Solved Example - Artificial Intelligence

- Form D we have one choice that is **F-E.**
- **F(A-D-FE) = 1+1+ 4 +4 =10**
- Basically **10** is the cost of reaching **FE from D.**
- And **Heuristic value of node D** also denote the cost of reaching **FE from D**.
- So, the new Heuristic value of D is 10.
- And the Cost from A-D remain same that is **11**.



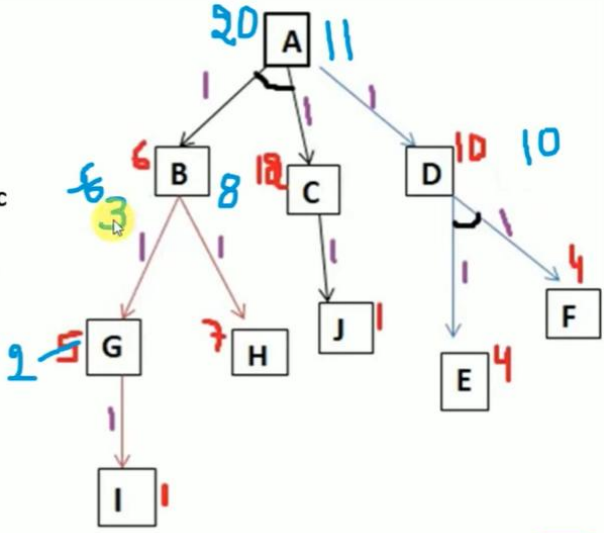
AO* Search Algorithm – Solved Example - Artificial Intelligence

- Let's take a look at F(A-BC)
- Now from B we have two path G and H , let's calculate the cost
- F(B-G)= 5+1 =6 and F(B-H)= 7 + 1 = 8
- So, cost from F(B-H) is more than F(B-G) we will take the path B-G.



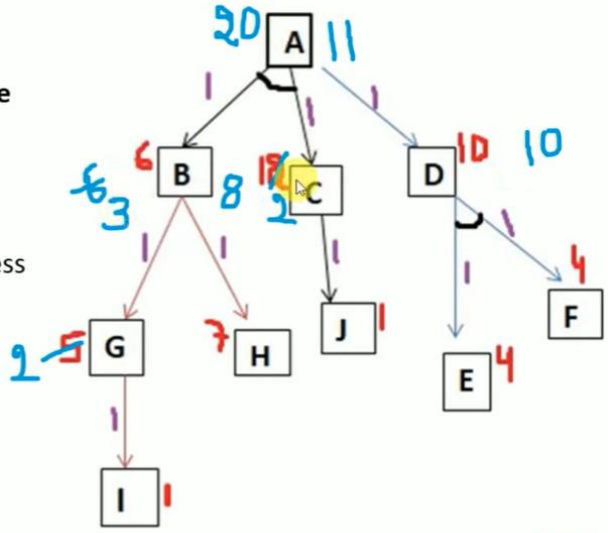
AO* Search Algorithm – Solved Example - Artificial Intelligence

1. The Heuristic value from **G to I is 1 but let's calculate the cost form G to I.**
2. **F(G-I) = 1 +1 = 2.** which is less than **Heuristic value 5**. So, the new **Heuristic value form G to I is 2.**



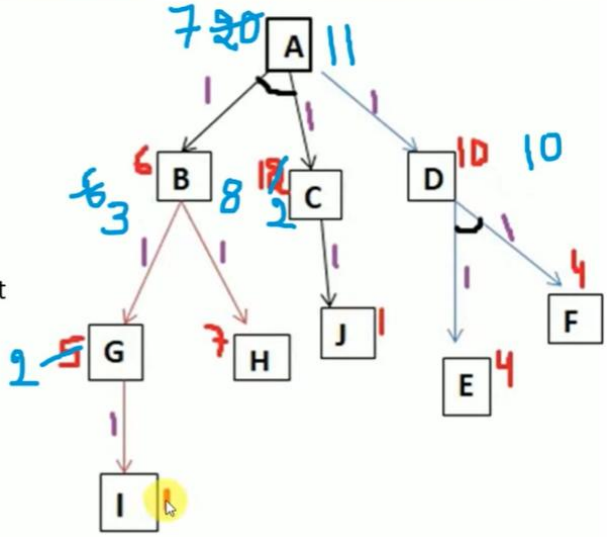
AO* Search Algorithm – Solved Example - Artificial Intelligence

1. **But A is associated with both B and C .**
2. As we can see from the diagram **C only have one choice or one node to explore that is J.** The Heuristic value of C is 12.
3. Cost form C to **J= F(C-J) = 1+1= 2** Which is less than Heuristic value
4. No**w the New Heuristic value of C is 2.**

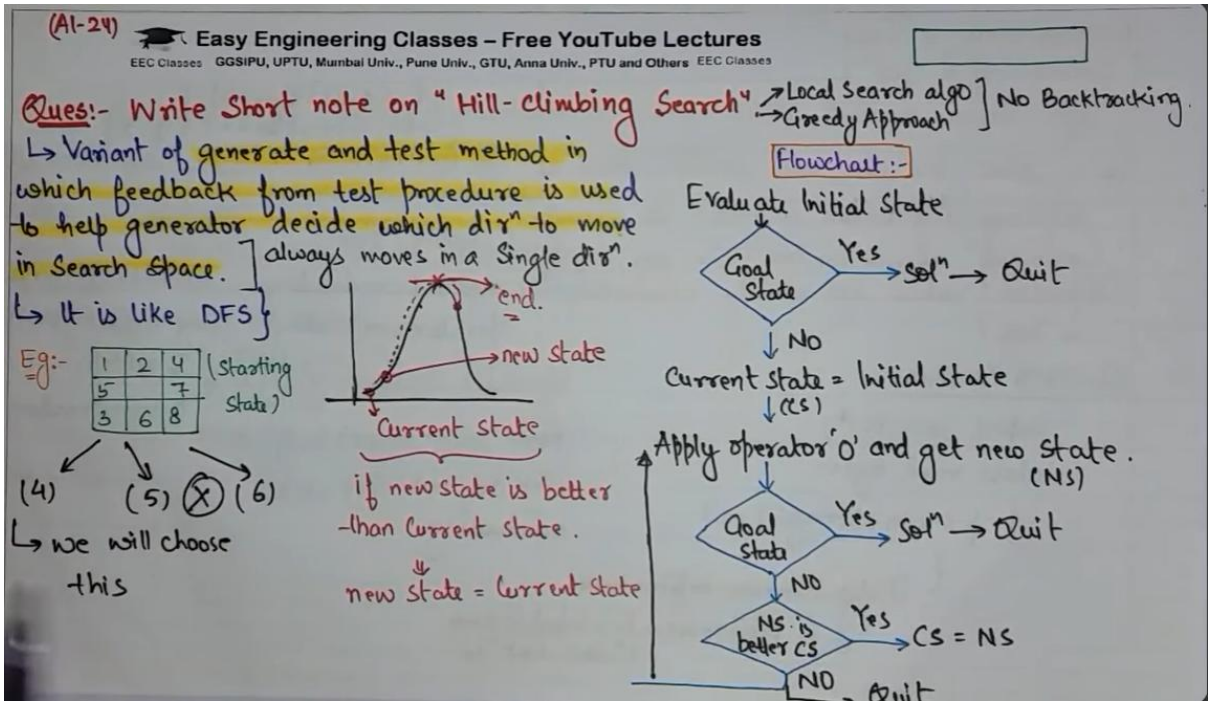


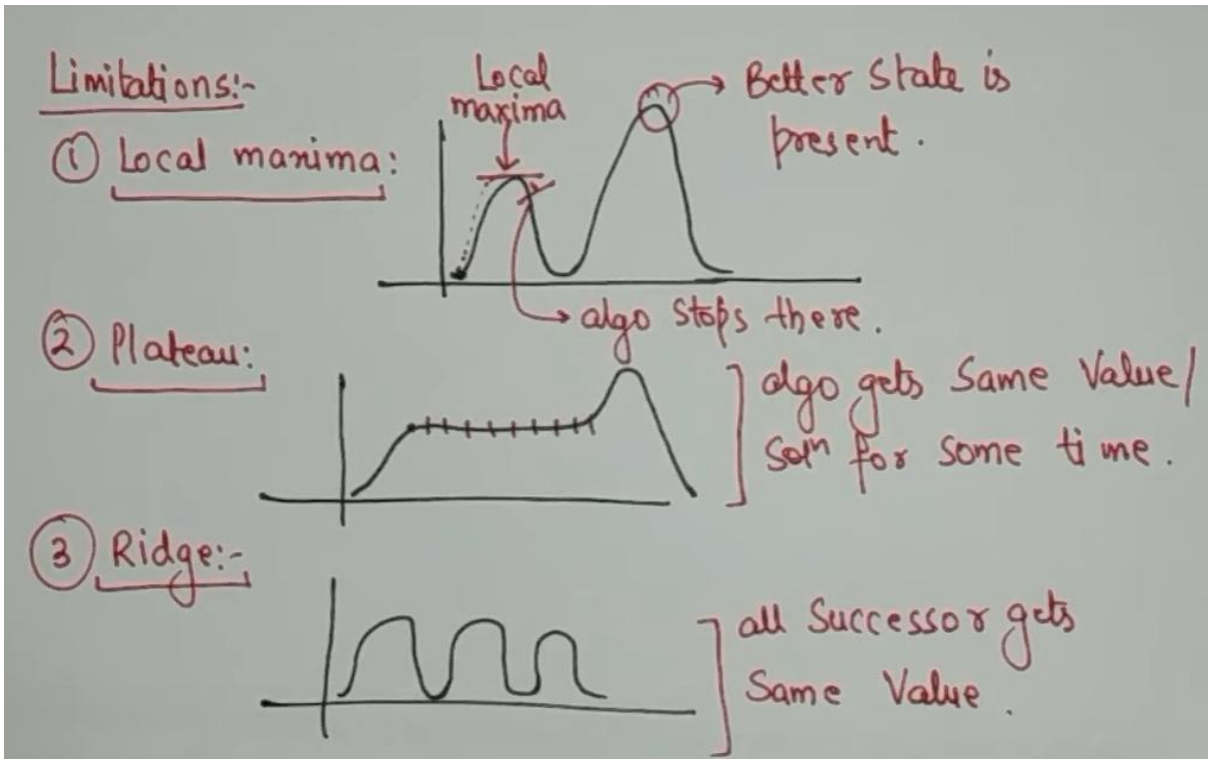
AO* Search Algorithm – Solved Example - Artificial Intelligence

1. **And the New Cost from A- BC that is**
2. **F(A-BC) = 1+1+2+3 = 7**
3. **which is less than F(A-D)=11.**
4. In this case Choosing path A-BC is more cost effective and good than that of A-D.



Hill Climb Search





Adversarial Search

→ multiple agent environment

→ Search where we examine the problem which arise when we try to plan ahead of the world and other agents are planning against us.

→ Single agent → sequence of actions → aim to find solution

→ multiple agent → same search space → searching for solution

→ eg: game playing

→ each agent is opponent of other agent and playing against each other. each agent need to consider action of other agent and effect of that action on their performance.

Search where two or more players with conflicting goals are trying to explore the same search space for solution → adversarial search → known as Games

Games → model as search problems → heuristic evaluation function → solve games

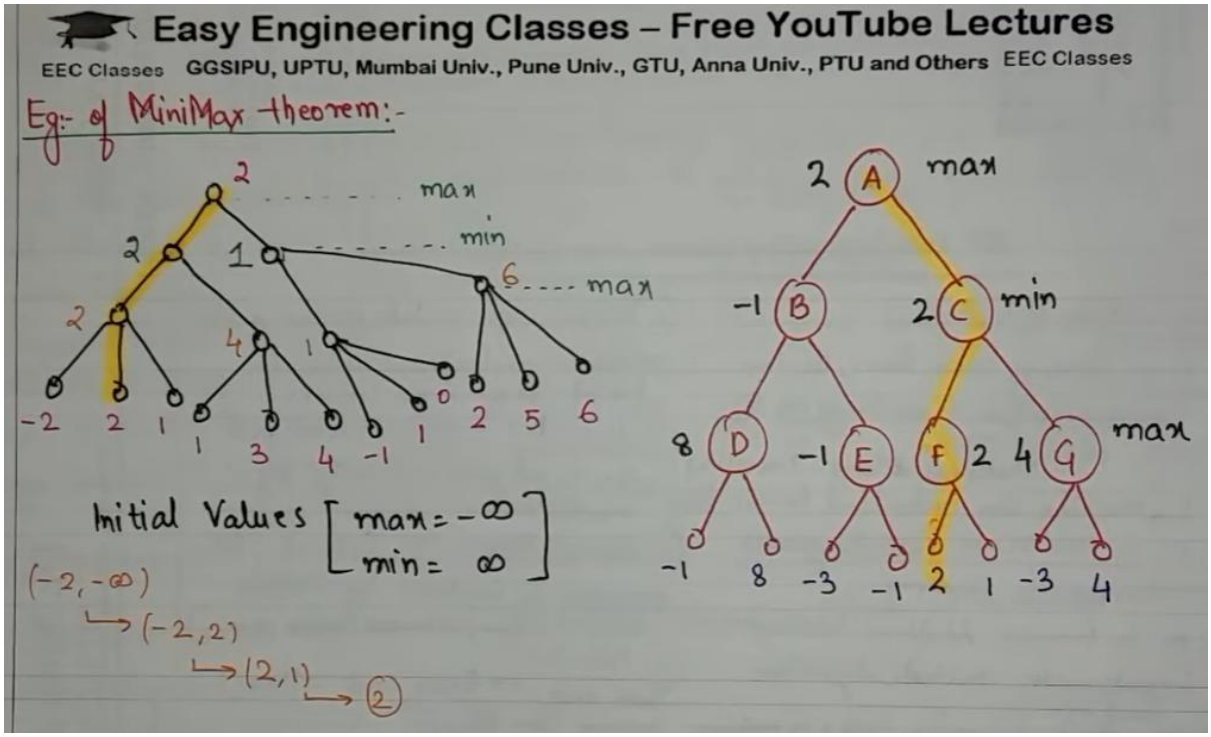
Minimax algorithm

(A155) Easy Engineering Classes – Free YouTube Lectures
EEC Classes GGSIPU, UPTU, Mumbai Univ., Pune Univ., GTU, Anna Univ., PTU and Others EEC Classes

Ques:- Explain Min& Max theorem. [Game Playing]
It is a specialized Search Algo that returns optimal sequence of moves for a player in Zero Sum Game.
↳ Recursive/Backtracking algo which is used in decision making and Game theory. → Two Player.
↳ uses recursion to search through Game Tree.
↳ Algo Computes minimax decision for current State.
↳ Two Players → MAX [Selects maximum Value]
→ MIN [Selects minimum Value]
↳ Depth-first Search Algo is used for exploration of complete Game Tree.
↳ CHESS, checkers, Tic-tac-toe...
[max = −∞ / min = ∞] Worst Values. [Initial].

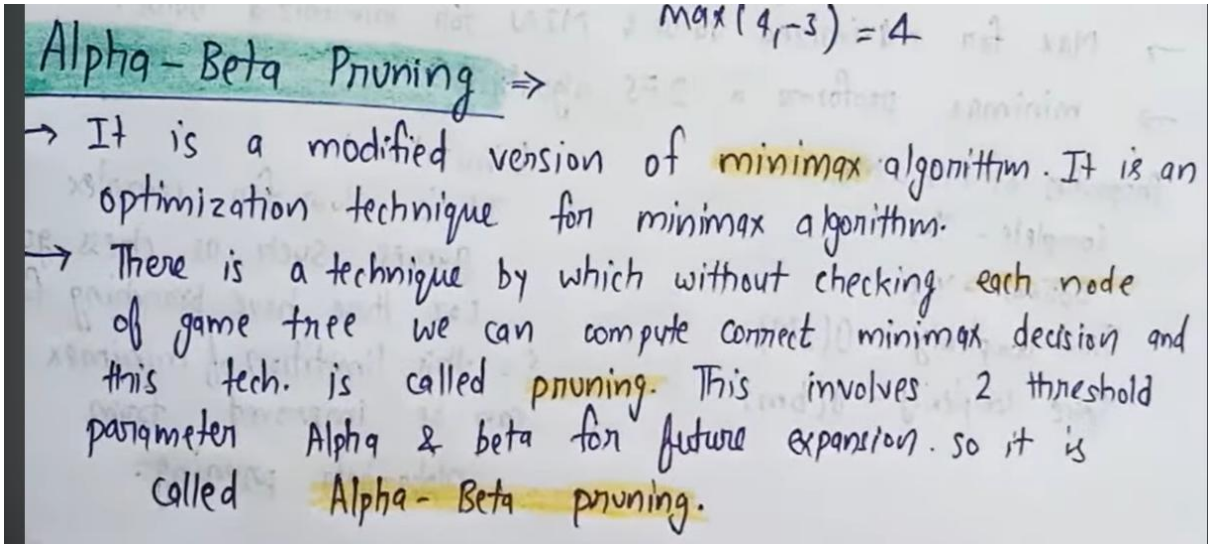
Properties:- → COMPLETE
① Definitely found soln (if exists)
② optimal
③ Time Complexity = $O(b^m)$ (m: maximum depth; b: branching factor of game tree)
④ Space Complexity = $O(b^m)$

Limitations:-
↳ Slow for Complex Games such as chess.
↳ 35 choices/moves.
$(35)^{100}$ d = 100 (for both players)
↳ BIG.



Alpha Beta Pruning

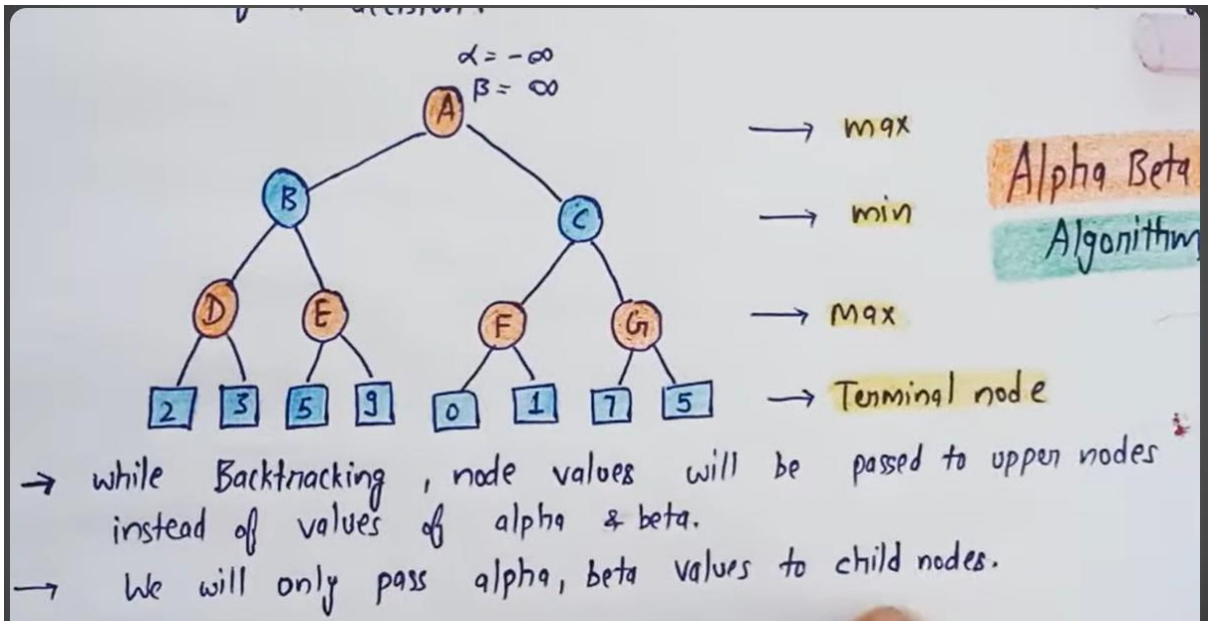
https://www.javatpoint.com/ai-alpha-beta-pruning



Alpha :- The best (highest value) choice we have found so far at any point along the path of maximizer. (Max player)
The initial value of alpha is $-\infty$.

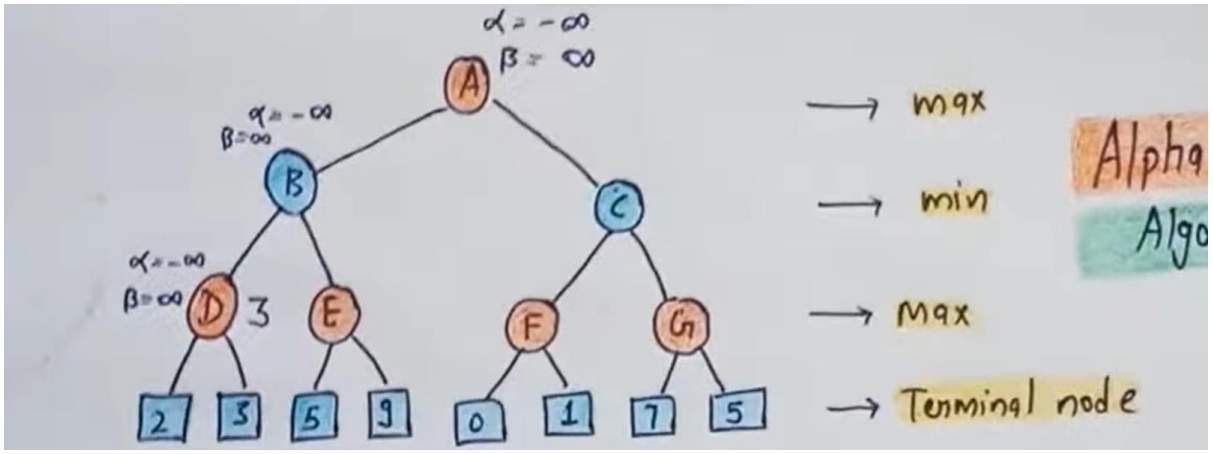
Beta:- The best (lowest value) ——— of minimizer. $+\infty$. (Min

→ It removes all the nodes which are not really affecting the final decision.

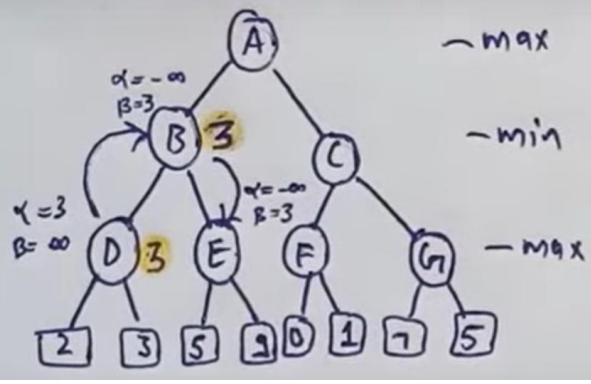


→ while Backtracking, node values will be passed to upper nodes instead of values of alpha & beta.

→ We will only pass alpha, beta values to child nodes.

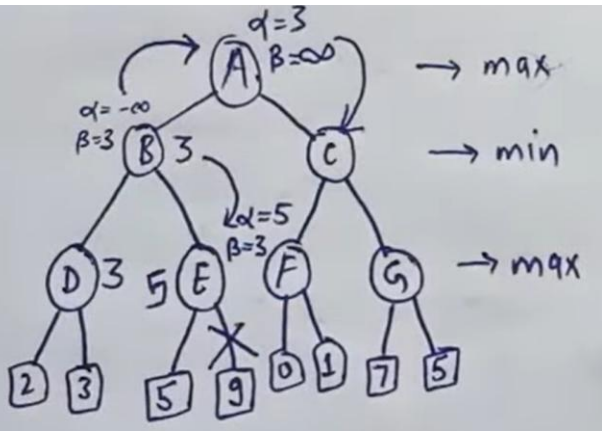


max(2,3) = 3 , so node D value will also 3. , α=3



→ max
→ min
→ max

α=-∞ , β = min(3, ∞) = 3



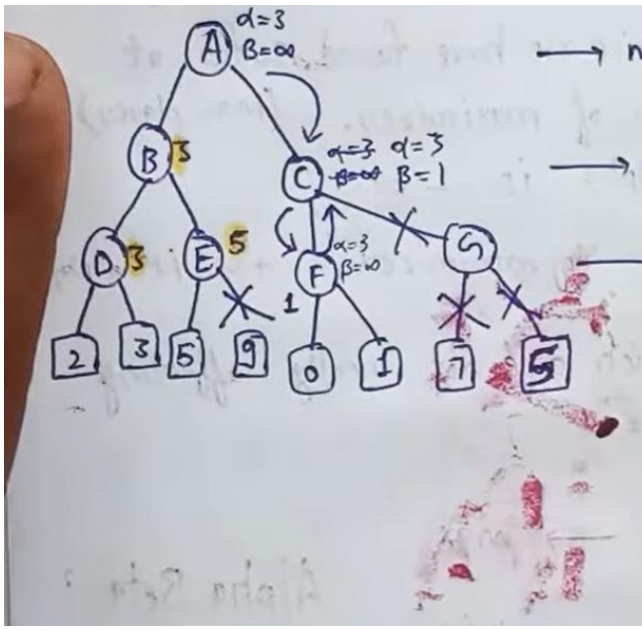
→ max
→ min
→ max

max(-∞, 5) = 5 = node α=5 α>=β

B to A , backtracking

α = max(-∞, 3) = 3

α = 3



→ max
→ min
→ max

C to F

max(0,3) = 3

max(1,3) = 3

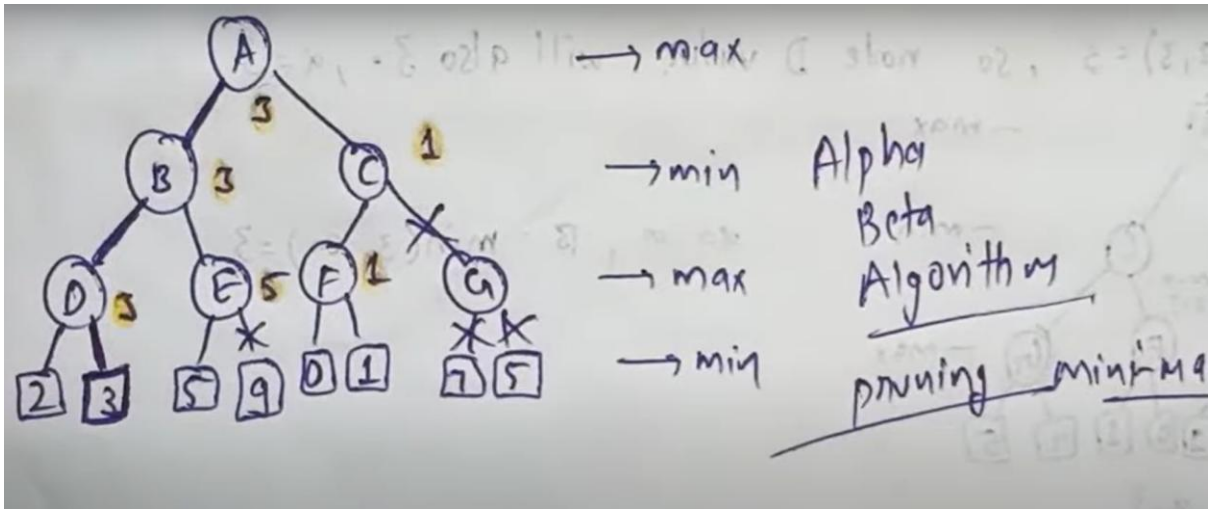
F to C

β = min(1, ∞) = 1

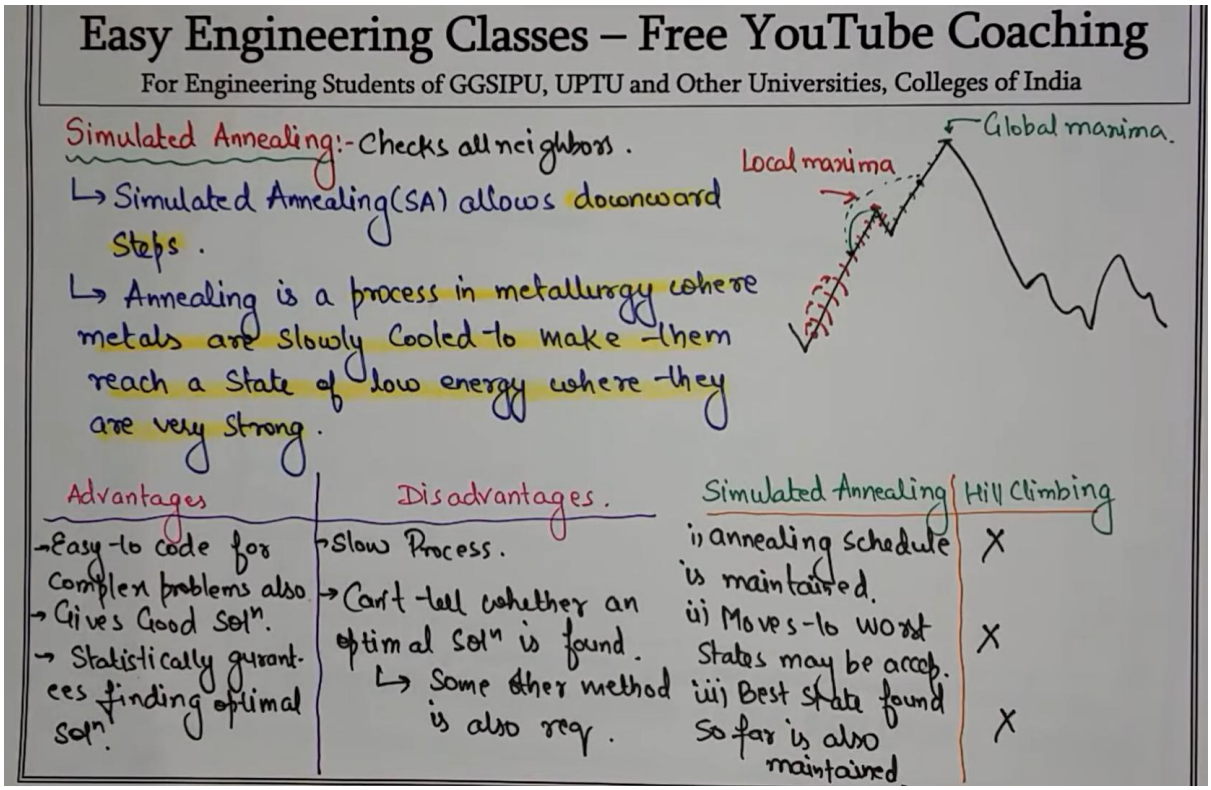
α = 3

β = 1

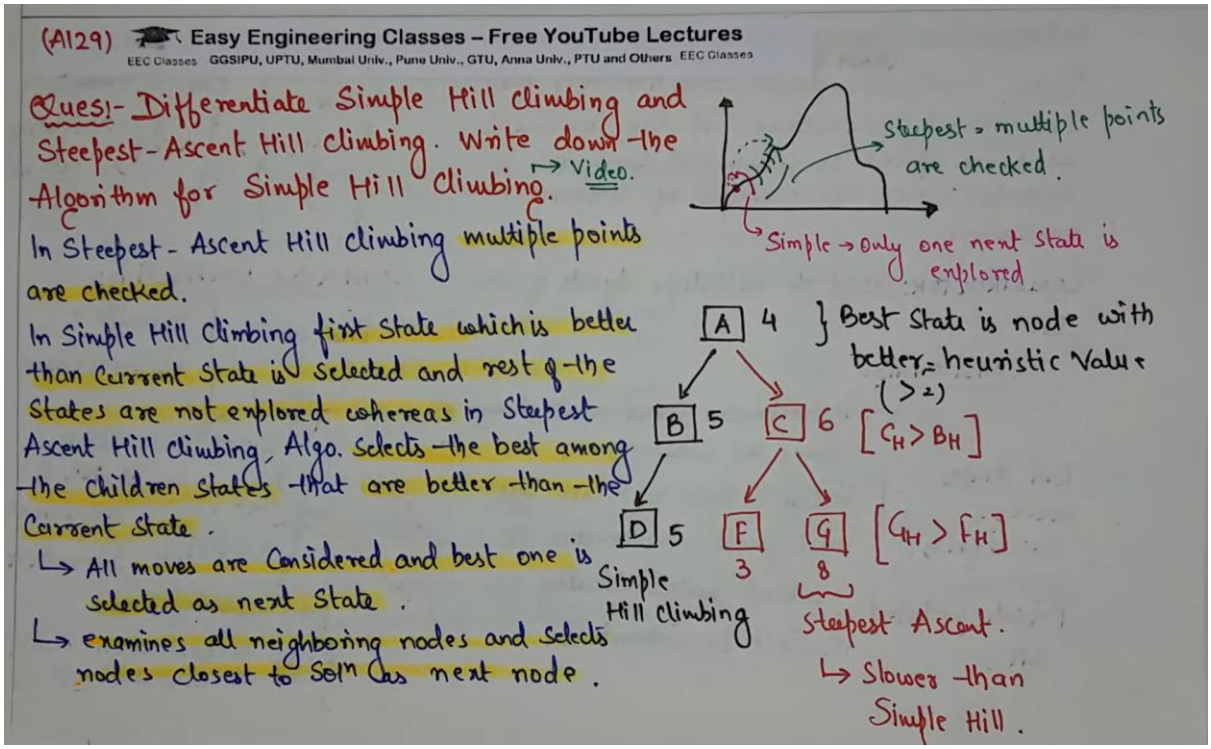
α >= β pruning

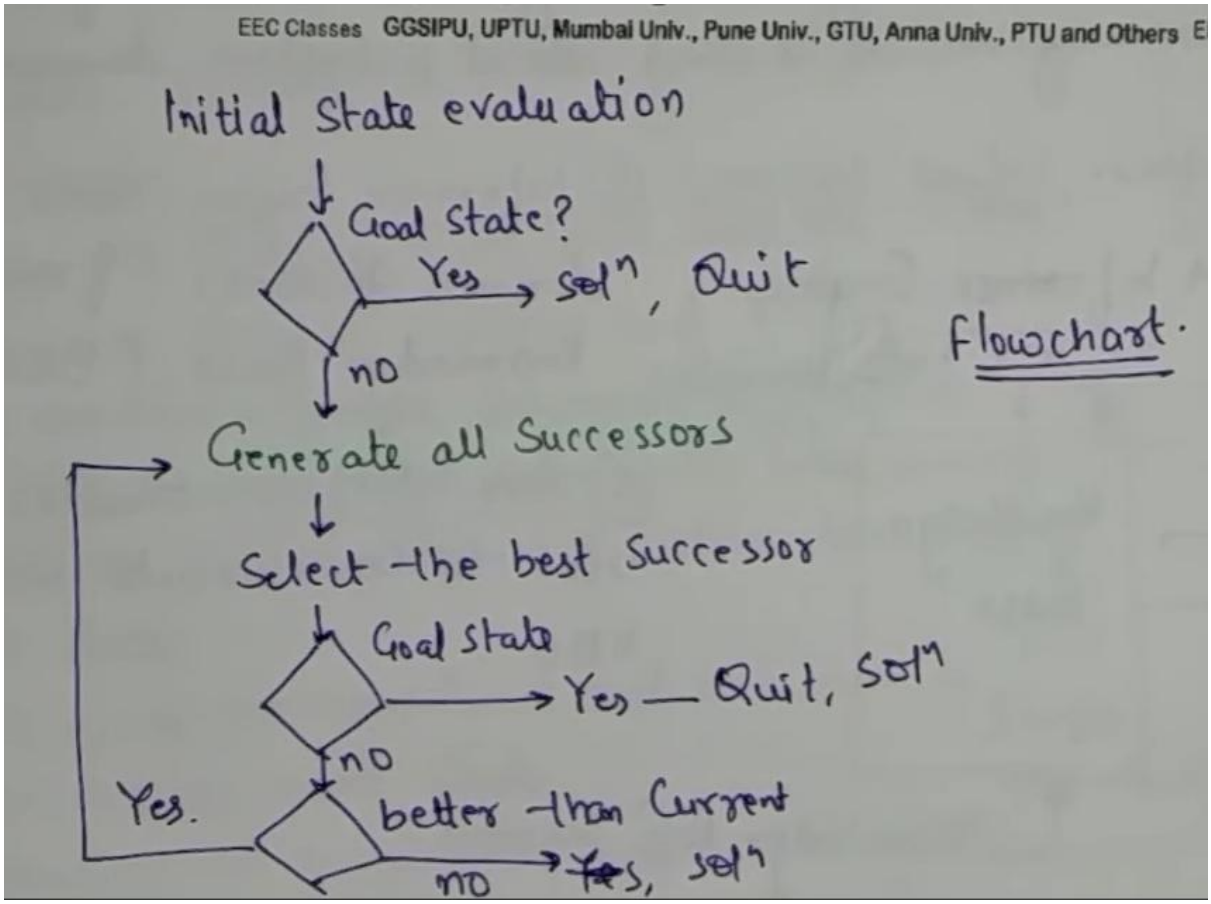


L32: Simulated Annealing in Artificial Intelligence | Difference Hill Climbing & Simulated Annealing

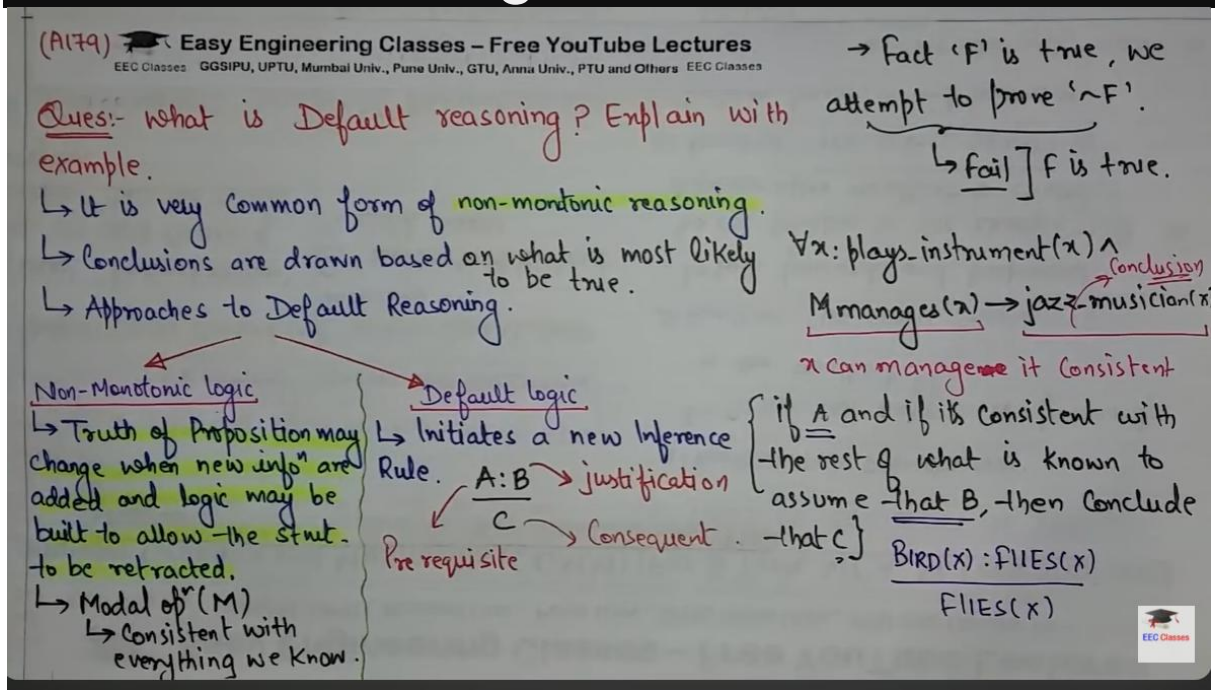


Steepest Ascent Hill Climbing | Artificial Intelligence | Compare with Simple Hill Climbing





Default Reasoning



Propositional Logic

Propositional Calculus:-

↳ It is a system that deals with the method used for manipulation of the symbols according to some rules.

X: It is Hot
Y: It is Humid
Z: It is raining

{All, Some ↳ (PL) x}

ALPHABET SET:

↳ i) Set of Variables or Propositional Symbols P, Q, R
ii) Logical Constants ↗ True (T) ↘ false (F)
iii) Two Parentheses "(" and ")" (X, Y)
iv) Set of logical operators

IMP:- ↳ i)

coord	symbol.	example	
not	¬	¬X	
ii) and	∧	X∧Y	
iii) or	∨	X∨Y	
iv) implies	→	(X→Y)	if X then Y
v) if and only if	↔	(X↔Y)	

(1) if it is humid ('Y') then it is hot. (X)
↳ (Y→X)

(2) if it is hot (X) and humid (Y) then it is not raining (Z).
(X∧Y)→¬Z

v) Set of equivalence relations or laws: (P, Q, R) are Variables.

↳ Commutative Laws: $P \wedge Q \cong Q \wedge P$, $P \vee Q \cong Q \vee P$

↳ Associative Laws: $(P \wedge Q) \wedge R \cong P \wedge (Q \wedge R)$, $(P \vee Q) \vee R \cong P \vee (Q \vee R)$

↳ Double Negation: $\sim(\sim P) \cong P$

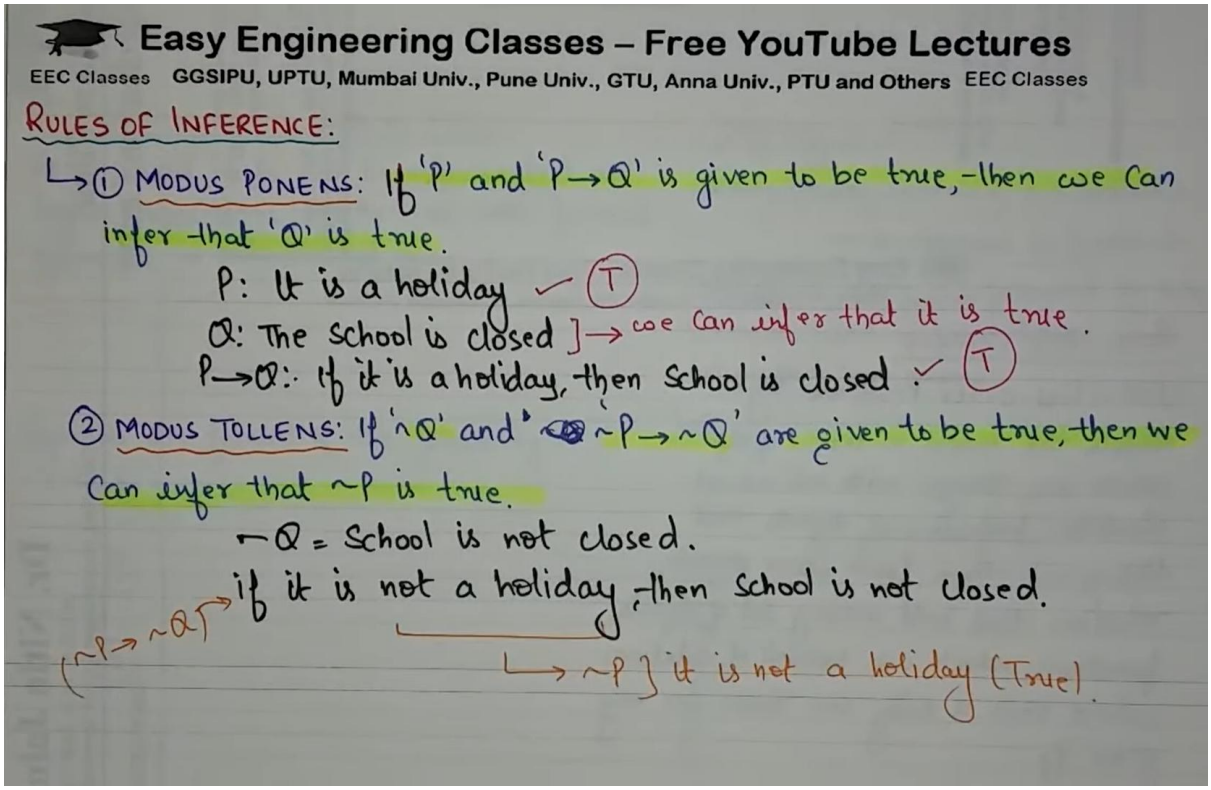
Imp ↳ De-Morgan's Law: $\neg(P \vee Q) = \neg P \wedge \neg Q$, $\neg(P \wedge Q) \cong \neg P \vee \neg Q$

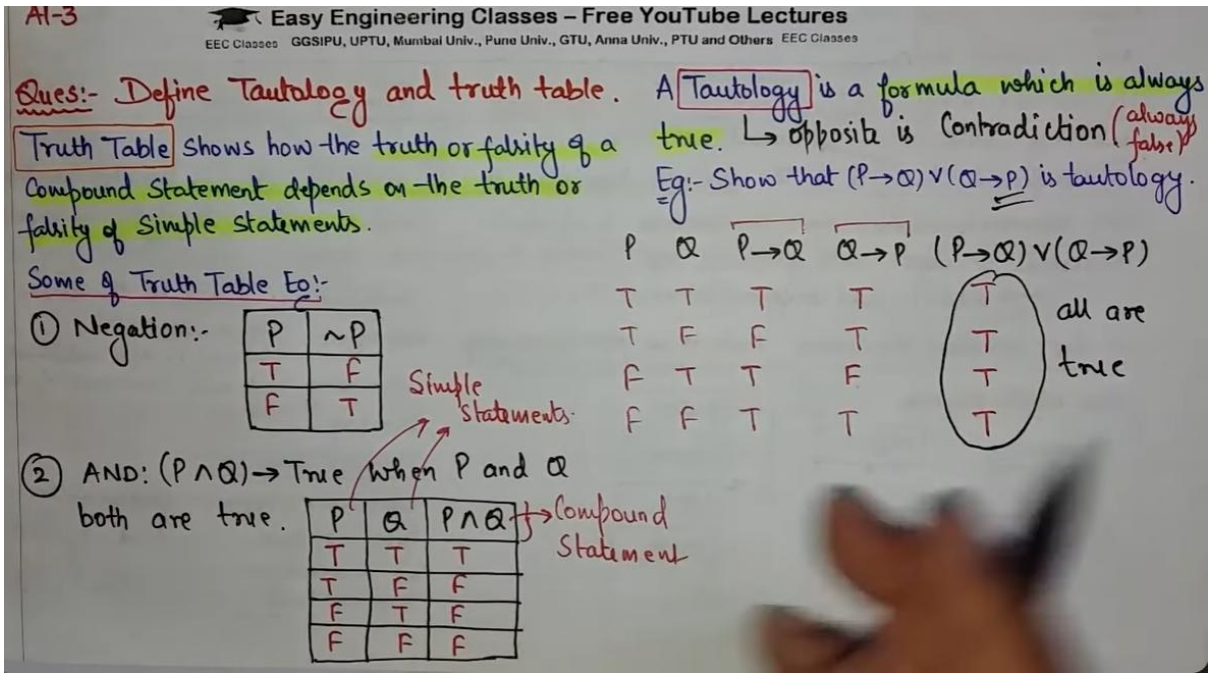
↳ Absorption Law: $P \wedge (P \vee Q) \cong P$, $P \vee (P \wedge Q) \cong P$

↳ Law of Contradiction: $P \wedge \neg P \equiv$ False $\begin{matrix} P=1 \\ \neg P=0 \end{matrix}$ ⓪ $\begin{matrix} P=0 \\ \neg P=1 \end{matrix}$ ⓪

↳ Law of excluded middle: $P \vee \neg P =$ True $\begin{matrix} P=1 \\ \neg P=0 \end{matrix}$ ① $\begin{matrix} P=0 \\ \neg P=1 \end{matrix}$ ①

↳ Law of impotency: $P \wedge P \cong P$ $\left[\begin{matrix} P=1 \to 1 \\ P=0 \to 0 \end{matrix}\right]$





First Order Logic (FOL)

First-Order Logic in Artificial Intelligence

- FOL is another way of knowledge representation in A.I. It is an extension to PL. (Propositional Logic)
- FOL is also know nas Predicate logic. It is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.

FOL does not only assume that the world contains facts like PL but also assumes

- objects: A, B, people, no:, Colors, Wars, pits, Wumpus

FOL does not only assume that the world contains facts like PL but also assumes

1. objects: A, B, people, no:, Colors, Wars, pits, Wumpus, ...

2. Relations: It can be unary relation such as: n-any relation such as: the sister of, brother of, has color.
   - Function: Father of, best friend, end of, ...

- As a natural lang, FOL has two main parts
  a. Syntax
  b. Semantics

Syntax of FOL: Basic elements

Constant : 1, 2, A, Bhanu, Hyderabad, ...

Variables : $x, y, z, a, b, \ldots$

Predicates : Brother, father, $>, \ldots$

Functions : Sqrt, ...

Connectives : $\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$

Equality : $=$

Quantifiers : $\forall, \exists$

Atomic Sentences:

- Atomic Sentences are the most basic sentences of FOL. These Sentences are formed from a predicate symbol followed by parenthesis with ~~Square~~ a sequence of terms.
- We can represent atomic Sentences as

  Predicate (term1, term2, ..., termn).

  Eg: Hari and Raghu are brothers: $\Rightarrow$ Brothers (Hari, Ra

We can represent atomic Sentences as

Atomic Predicate (term1, term2, ..., termn)

Eg:- Hari and Raghu are brothers: ⇒ Brothers (Hari, Raghu)

Tommy is a dog: ⇒ dog (Tommy)

omplex Sentences

- Complex Sentences are made by combining atomic Sentences using Connectives.

irst order logic statements divided into 2 parts:-

Subject: It is the main part of stmt

Predicate: A predicate can be defined as a relation, which binds two atoms together in a stmt.

Consider the stmt:

stmt - "x is an integer."

It consists of two parts,

first part: x is the subject of stmt

second part: "is an integer" is known as predicate

x is an Integer

↓ ↓

Predicate

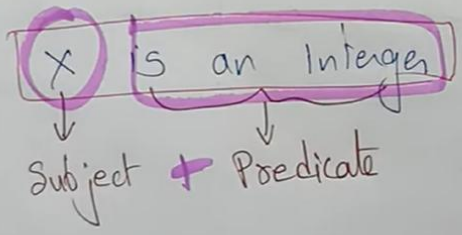
Consider the stmt

stmt → "X is an integer"

It consists of two parts,

first part: x is the subject of stmt

second part: "is an integer" is known as predicate.



Quantifiers in FOL :-

→ A Quantifier is a lang ele which generates quantifica

→ these are the symbols that permit to determine or identify the range and scope of the variable in the logic expression.

There are two types of quantifier :-

a. Universal Quantifier (for all, everyone, everything)

b. Existential Quantifier (for some, atleast one)

Universal Quantifier:

It is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of particular thing.

→ It is represented by a symbol ∀.

(In UQ we use implication "→")

If x is a variable, the ∀x is read as:

- for all x
- for each x

If x is a variable, the ∀x is read as:

- for all x
- for each x
- for every x.

range

Existential Quantifier:

It is a type of quantifier, which express that the stmt within its scope is true for at least one instance of something.

It is denoted by ∃.

{In Existential quantifier we always use AND or conjunction symbol (∧).}

⇒ If x is a variable, then existential quantifier will be ∃(x) is read as:

- There exists a 'x', for some 'x', for atleast on

It is denoted by ∃.

{In Existential quantifier we always use AND or conjunction symbol (∧).}

⇒ If x is a variable, then existential quantifier will

- ∃(x) is read as:
  - There exists a 'x', for some 'x', for atleast on

Eg: {All man drink coffee.} ∀

- Let (x) is variable.
  - x_1 drink coffee ∧ x_2 drink coffee ....
    ∧ x_n drink coffee.

So, we can write this as,

$\forall x \, man(x) \rightarrow drink(x, coffee)$

There are all x where x is a man who drink coffee.

Some Examples of FOL using Quantifier

① All birds fly.

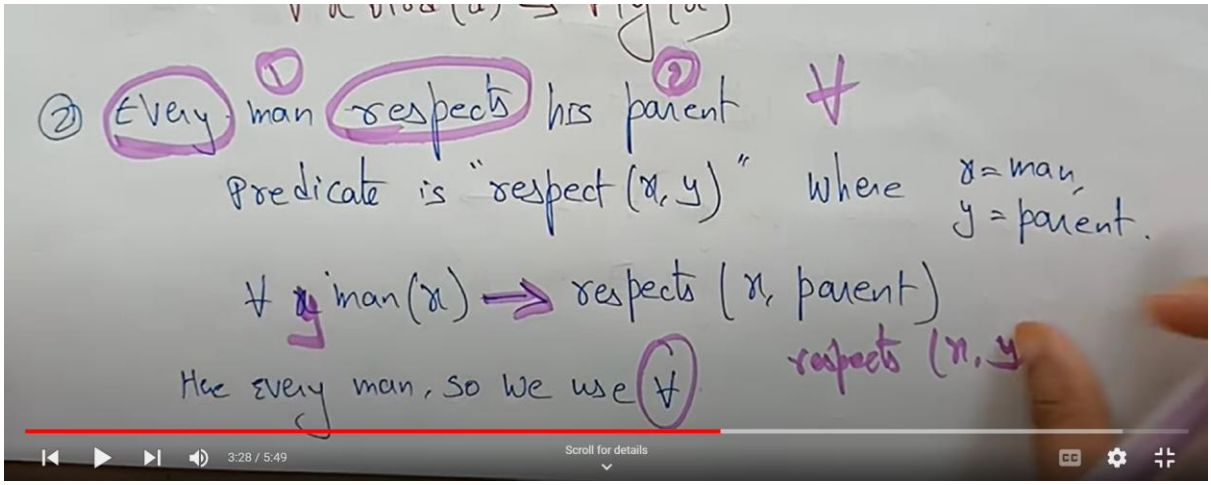
Predicate is "fly(bird)"

$\forall x \; bird(x) \rightarrow fly(x)$

② Every man respects his parent

Predicate is "respect(x, y)" where x = man, y = parent.

$\forall x \; man(x) \rightarrow respects(x, parent)$



x_1 is intelligent $\vee$ x_2 is intelligent ... $\vee$ x_n is intelli

$\exists x : boy(x) \wedge intelligent(x)$

$\Rightarrow$ There are some ~~boy~~ x where x is a boy who is intelligent.

* The main connective for $\forall$ is $\rightarrow$.

The main connective for $\exists$ is $\wedge$.

Eg: Some boys are intelligent. ∃

x_1 is intelligent ∨ x_2 is intelligent .... ∨ x_n is intelligent

$\exists x : boy(x) \wedge intelligent(x)$

→ There are some ~~boy~~ x where x is a boy who is intelligent.

* The main connective for ∀ is '→'.
The main connective for ∃ is '∧'.

Inference in First Order logic:

→ Inference in FOL is used to ~~ded~~ deduce new fact Sentences from existing sentences.

Before understanding FOL inference rule, let's unders some basic terminologies used in FOL.

① Substitution

It is a fundamental operation performed on terms and formulas. It occurs in all inference sys in FOL.

① Substitution

It is a fundamental operation performed on terms and formulas. It occurs in all inference sys in FOL.

F[a/x]

Substitute a constant "a" in place of variable "x".

FOL logic does not only use predicate & terms making atomic sentences but also uses another way, which is equality in FOL.

Eg:- Brother(John) = Smith

Here, the obj refered by Brother(John) is similar to obj refered by Smith. The equality symbol can also be used with negation to represent that two terms are not the same objects.

Eg:- $\neg(x=y)$ which is equivalent to $x \neq y$.

FOL inference rules for quantifier:-

As PL we also have inference rules in FOL.

- → Universal Generalization.
- → Universal Instantiation
- → Existential Instantiation
- → Existential Introduction.

① Universal Generalization:

It is a valid inference rule which states that if premise P(c) is true for any arbitrary element c in

universe of discourse, then we can have a conclusion as $\forall x P(x)$.

- It can be represented as, $\frac{P(c)}{\forall x P(x)}$

Eg:- Let's represent, P(c): "A byte contains 8 bits";
So, for $\forall x P(x)$ "All bytes contain 8 bits;" it will also be true.

② Universal Instantiation:-
It is also called universal elimination. It can be applied multiple times to add new sentences.

be true.

② Universal Instantiation:- = elimination

It is also called universal elimination. It can be applied multiple times to add new sentences.

P(c) by substituting a ground term c (a constant within domain x) from $\forall x P(x)$ for any object in the Universe of discourse.

U.I $\frac{\forall x P(x)}{P(c)}$ U.G = $\frac{P(c)}{\forall x P(x)}$

Eg:- If "every person like ice-cream" $\Rightarrow \forall x P(x)$ so we can infer that, "John likes the ice-cream" $\Rightarrow P(c)$

3) Existential Instantiation :- = elimination

→ It is also called Existential Elimination, which is a valid inference rule in FOL.

→ It can be applied only once to replace the existential sentence

→ This rule states that one can infer P(c) from the formula given in the form $\exists x\, P(x)$ for a new constant symbol c.

$$\frac{\exists x\, P(x)}{P(c)}$$

Existential introduction :-

→ It is also called as existential generalization.

→ This rule states that if there is some element c in the universe of discourse which has a property P, then we can infer that there exists something in the universe which has property P.

$$\frac{P(c)}{\exists x\, P(x)}$$

Eg :- "Pinky got good marks in ma

"Therefore, some one got good

h has property P. c: Pinky

$$\frac{P(c)}{\exists x\, P(x)}$$

Eg :- "

" Pinky got good marks in math

Someone "Therefore, some one got good marks in Maths".

Unification

It is all about making the expressions look identical. So, for the given expressions to make them look identical we need to do substitution.

eg:- $p(x, f(y))$ ① ; $p(a, f(g(z)))$ ②

$x = a$

$y = g(z)$

unification : $[a/x, g(z)/y]$

⇒ In the above eg, substitute x with a, & y with $g(z)$. ← it will be represented as a/x & $g(z)/y$.

⇒ With both the substitutions, the first expression will be

unification : $[a/x, g(z)/y]$

⇒ In the above eg, substitute x with a, & y with $g(z)$. ← it will be represented as a/x & $g(z)/y$.

⇒ With both the substitutions, the first expression will be

identical to the second expression & the substitution set will be $[a/x, g(z)/y]$

Conditions for Unification :-

① Predicate symbol must be same, atoms or expressio with different predicate symbol can never be unified.

② Number of arguments in both expressions must be ide

③ Unification will fail if there are two similar variable present in same expression.

Unification Algorithm

⇒ Algorithm : Unify (L_1, L_2)

Step 1 :- If L_1 or L_2 is a variable or constant, then :

(a) If L_1 & L_2 are identical return NIL.

(b) Else if L_1 is a variable, then if L_1 occurs in L_2 then return FAIL, else return $\{(L_2/L_1)\}$

(c) Else if L_2 is a variable, then if L_2 occurs in L_1 then return FAIL, else return $\{(L_1/L_2)\}$

(d) Else return FAIL.

Step 2 :- If the initial predicate symbol in L_1 & L_2 are not identical, then return FAIL.

Step 3 :- If L_1 & L_2 have a different number of arguments, then return FAIL.

Step 4 :- Set SUBST to NIL

Step 5 :- Loop ⟹ { for i ← 1 to no: of arguments in L_1 :

a) Call unify with the ith argument of L_1 and the ith argument of L_2 putting result in S.

b) If S = FAIL then return FAIL

c) If S is not equal to NIL then

Step 6 :- Return SUBST

Step3 :- If L_1 & L_2 have a different
Then return FAIL

Step4 :- Set SUBST to NIL

Step5 :- Loop ⟹ { for i ← 1 to no: of arguments in L_1:
a) Call unify with the ith argument of L_1 and the ith argument of L_2 putting result in S.
b) If S = FAIL then return FAIL
c) If S is not equal to NIL then
(i) Apply S to the remainder of both L_1 & L_2
(ii) SUBST = APPEND (S, SUBST) }

Step6 :- Return SUBST

Substitute

Implementation of the Algorithm

Step1 :- Initialize the substitution set to be empty.

Step2 :- Recursively unify atomic sentences:
a) check for identical expression match.
b) If one expression is a variable v_i, & other is a term t_i which does not contain variable v_i then:
→ Substitute t_i/v_i in existing substitutions
→ Add t_i/v_i to the substitution setlist.
→ If both the expressions are functions, then function name must be similar, & no: of arguments must be same in both expression.

b) If one expression is a variable v_i, & other is a term t_i which does not contain variable v_i then:
→ Substitute t_i/v_i in existing substitutions
→ Add t_i/v_i to the substitution setlist.
→ If both the expressions are functions, then function name must be similar, & no: of arguments must be same in both expression.

Examples:

Consider $P(x, g(x)) = P(z, y)$

Solutions:

(i) $P(z, y)$: unifies with $[x/z, g(x)/y]$ — $x = z$, $g(x) =$

(ii) $P(z, g(z))$: unifies with $[x/z \text{ or } z/x]$ — $x = z$, $g(x) =$

(iii) $P(\text{Prime}, F(\text{Prime}))$: does not unifies

(y and g does not match)

Resolution in FOLt

→ Resolution is a theorem proving technique that ~~process by building~~ proofs by contradictions.

→ It is used, if there are various stmts are given, & need to prove a conclusion of those stmt.

→ Unification is a key concept in proofs by resolutions.

→ Resolution is a single inference rule which can efficiently operate on conjuctive normal form or clausal form.

clause: Disjunction of literals is called cl

→ It is used, if there are various stmts are given, & need to prove a conclusion of those stmt.

→ Unification is a key concept in proofs by resolutions.

→ Resolution is a single inference rule which can efficiently operate on conjuctive normal form or clausal form. CNF

clause: Disjunction of literals is called clause.

Conjunctive NF: A sentence represented as a conjunction of clauses said to be CNF.

Steps for Resolution

1. Conversion of facts into FOL
2. Convert FOL stmt into CNF
3. Negate the stmt which needs to proove (by contrad
4. Draw resolution graph (unification)

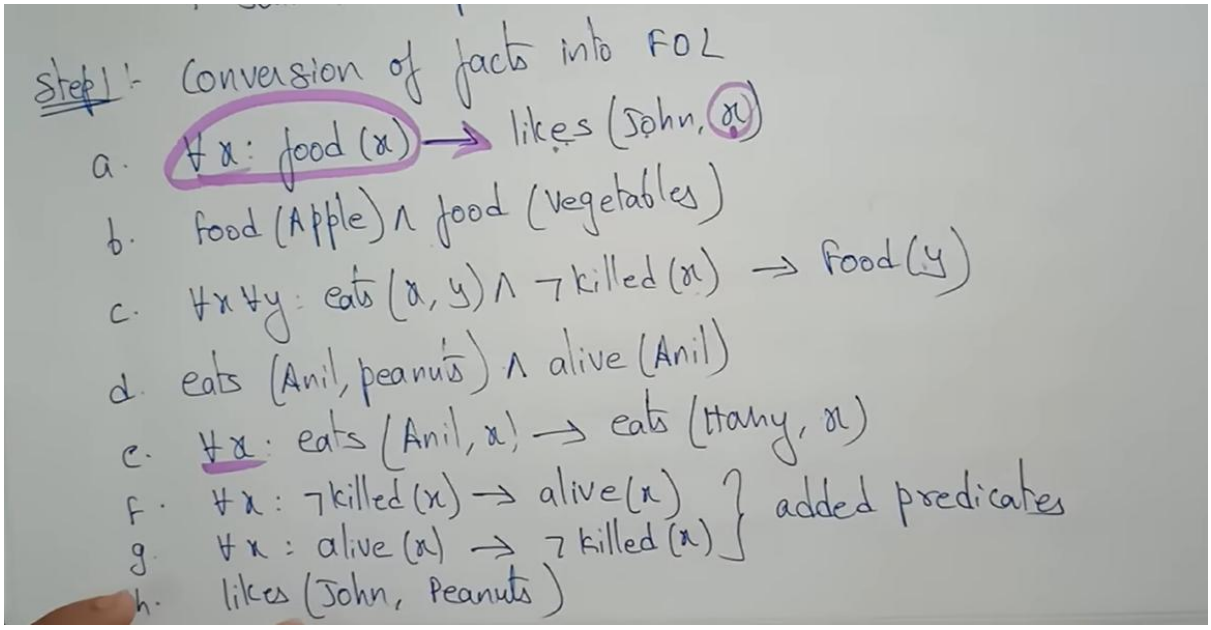
Examples:-

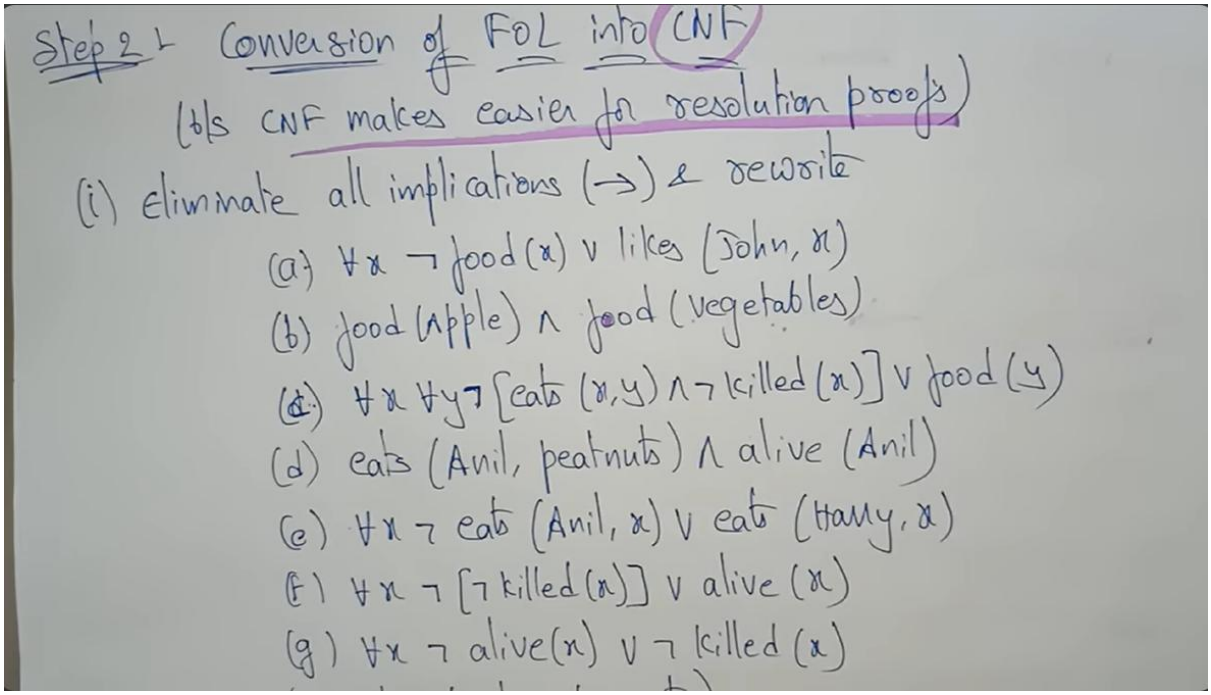
a. John likes all kind of food
b. Apple & vegetable are food
c. Anything anyone eats and not killed is food

Examples:-

a. John likes all kind of food ∀x
b. Apple & vegetable are food
c. Anything anyone eats and not killed is food
d. Anil eats peanuts and still alive

e. Harry eats everything that Anil eats
Proove by resolution that:
f. John likes peanuts.





$A \implies B$

$\equiv \neg A \vee B$ [If you haven't proved this before, you should]

$\equiv A \wedge \neg B$ [DeMorgan's]

$\equiv \neg B \wedge A$ [Propositions commute over conjunction]

$\equiv \neg(\neg B) \vee \neg A$ [DeMorgan's]

$\equiv \neg B \implies \neg A$ [First line in reverse]

(ii) → Move negation (¬) inwards and rewrite

(a) $\forall x \neg food(x) \vee likes(John, x)$

(b) $food(apple) \wedge food(vegetables)$

(c) $\forall x \forall y \neg eats(x, y) \vee killed(x) \vee food(y)$

(d) $eats(Anil, peanuts) \wedge alive(Anil)$

(e) $\forall x \neg eats(Anil, x) \vee eats(Harry, x)$

(f) $\forall x \neg killed(x) \vee alive(x)$

(g) $\forall x \neg alive(x) \vee \neg killed(x)$

(h) $likes(John, peanuts)$

(iii) Rename variables or standardize variables

(a) $\forall x \neg food(x) \vee likes(John, x)$

(b) $food(Apple) \wedge food(vegetables)$

(c) $\forall y \forall z \neg eats(y, z) \vee killed(y) \wedge food(z)$

(d) $eats(Anil, peanuts) \wedge alive(Anil)$

(e) $\forall w \neg eats(Anil, w) \vee eats(Harry, w)$

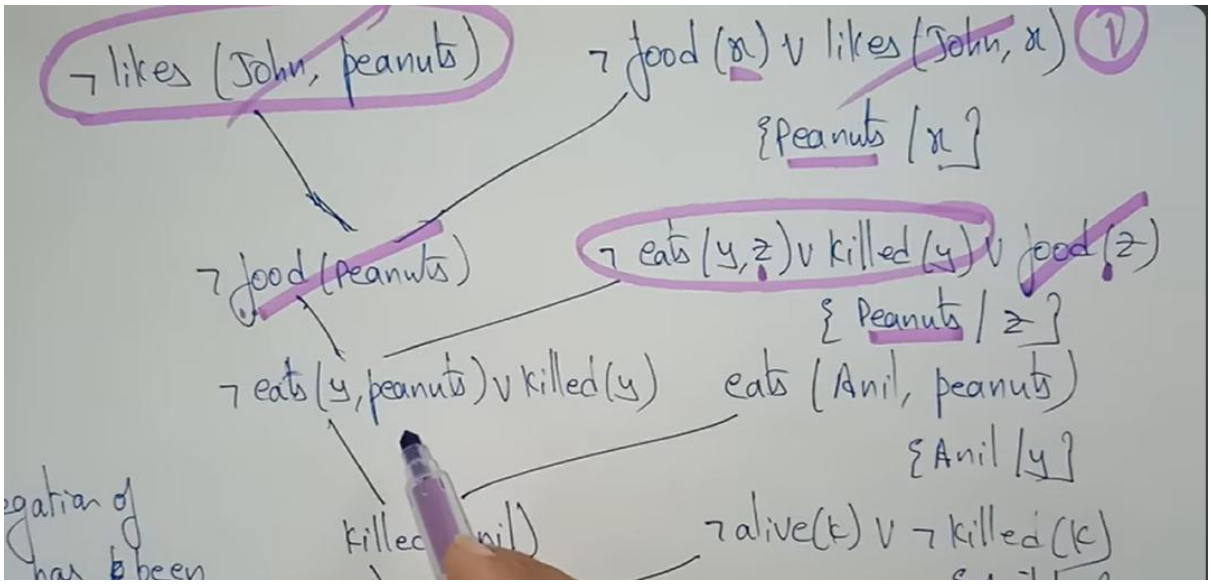
(f) $\forall g \neg killed(g) \vee alive(g)$

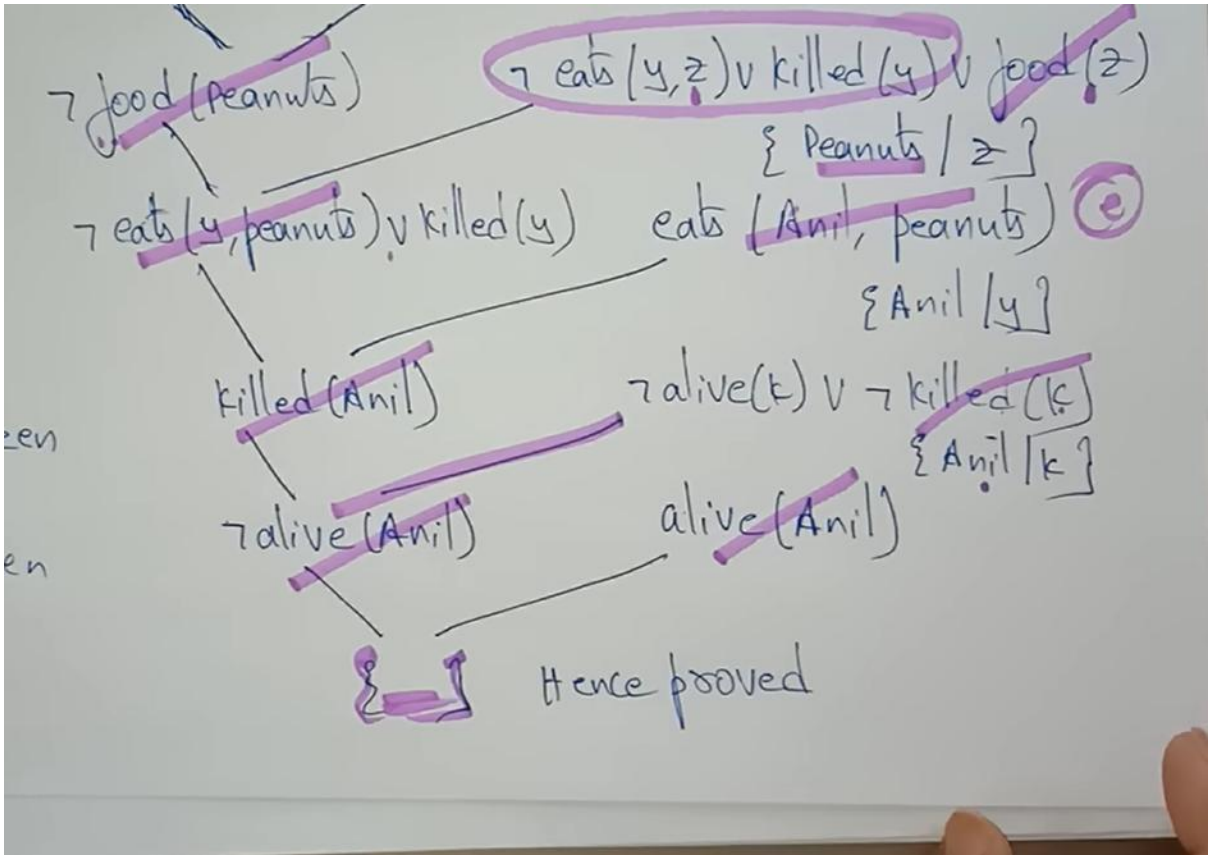
(g) $\forall k \neg alive(k) \vee \neg killed(k)$

(h) $likes(John, peanuts)$

i) Drop universal quantifiers:

(a) ¬ food(x) ∨ likes (John, x)
(b) food (Apple)
(c) food (Vegetables)
(d) ¬ eats (y, z) ∨ killed (y) ∨ food (z)
(e) eats (Anil, peanuts)
(f) alive (Anil)
(g) ¬ eats (Anil, w) ∨ eats (Harry, w)
(h) killed (g) ∨ alive (g)
(i) ¬ alive(k) ∨ ¬ killed (k)
(j) likes (John, peanuts)





Representing Simple Facts in FOPL:

→ models world in terms of objects.

Real-world facts can be represented as logical propositions written as well-formed formulas in propositional logic.

Symbols:-
1. objects
2. Properties
3. Relation.

} Representation.

Symbols are formed of the following:-

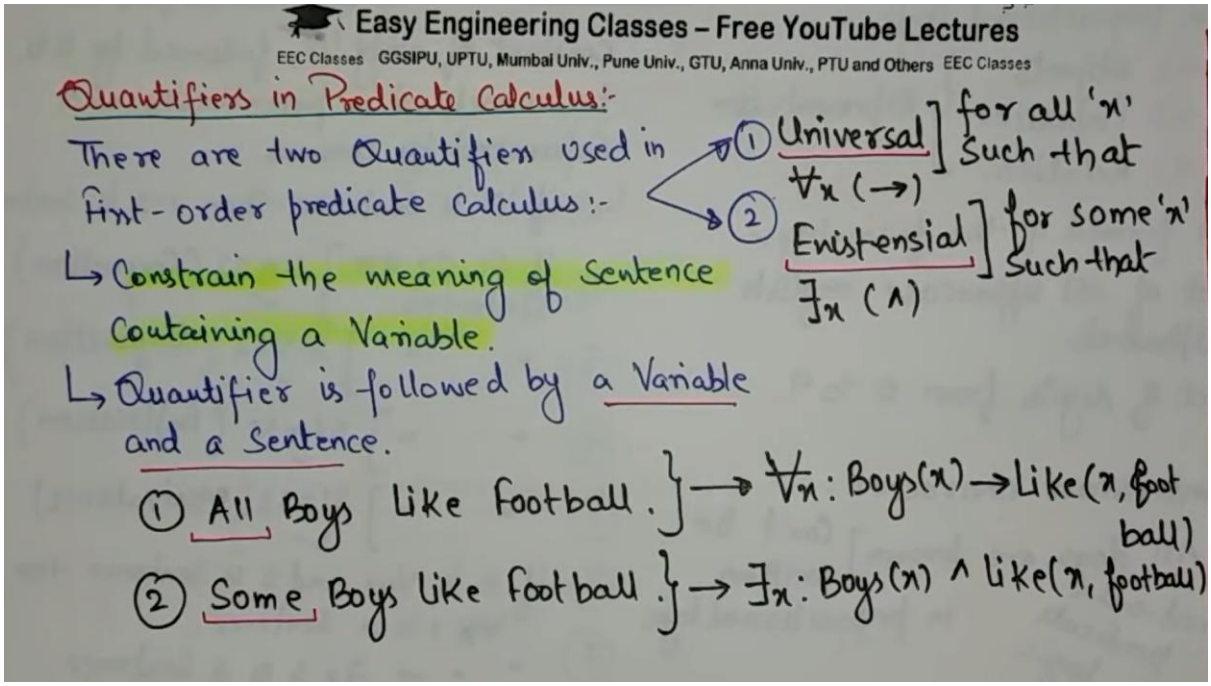
1. Set of all uppercase english Alphabets.
2. Set of digits from 0 to 9.
3. Underscore character.

[All dogs are brown] Can't be written in propositional logic
↳ first-order predicate logic.

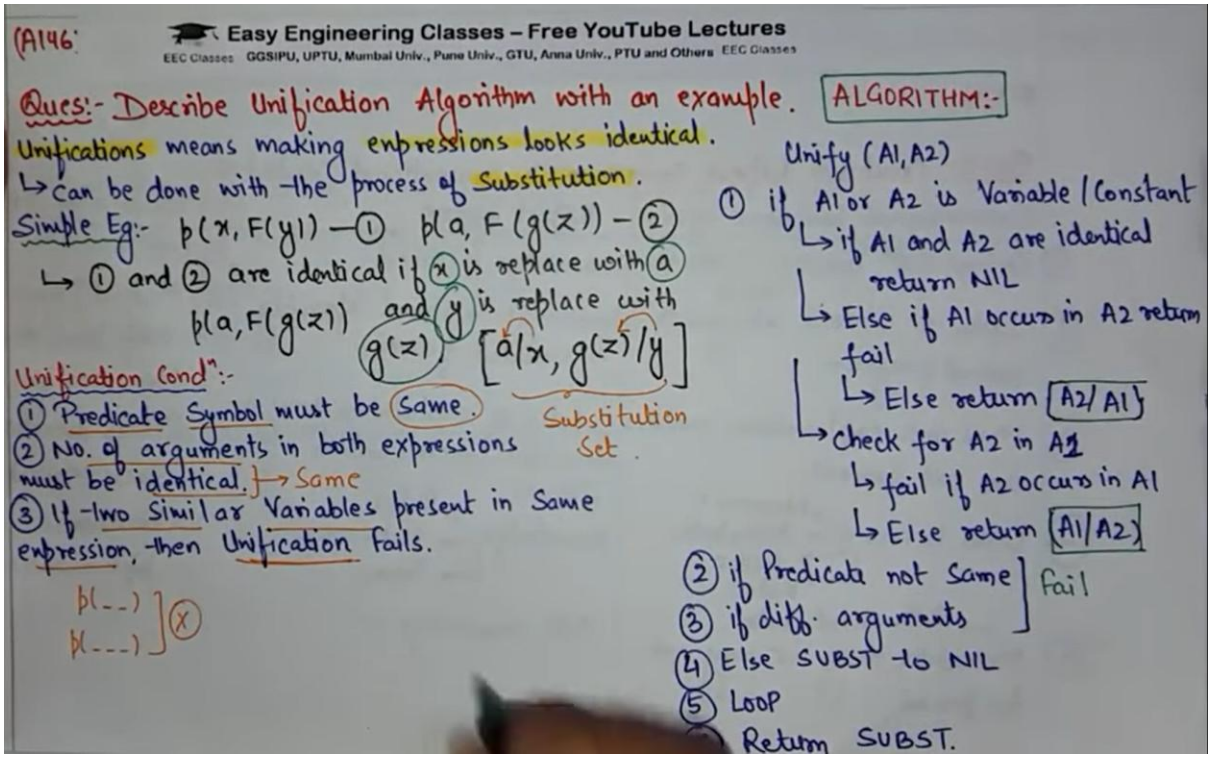
Defining a Sentence :-

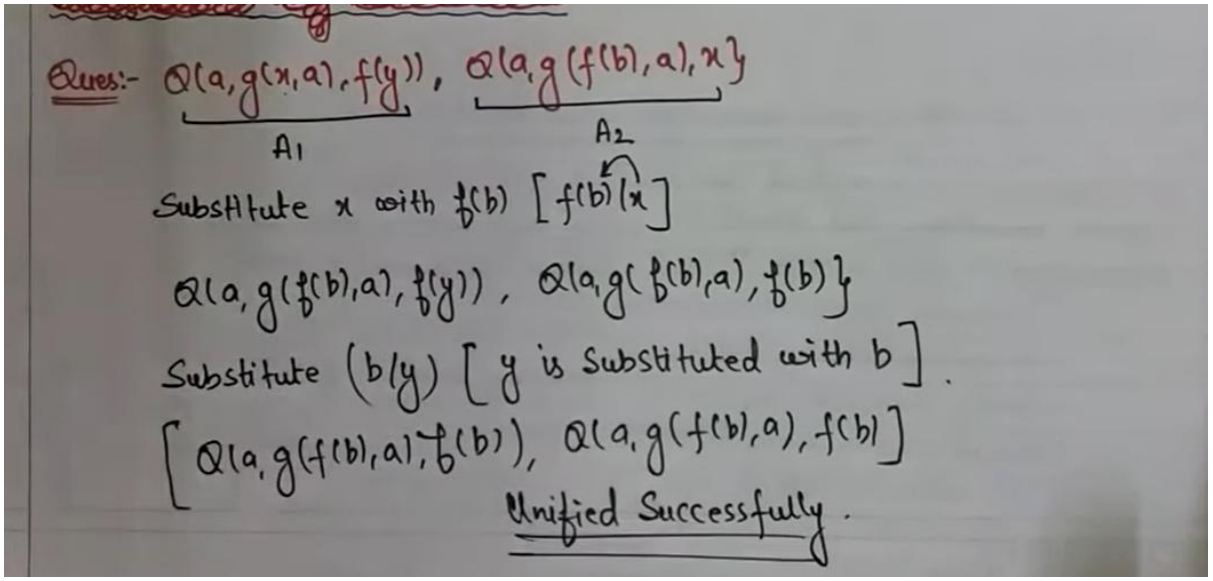
Every atomic sentence is a Sentence.
↳ is defined as predicate constant of arity 'n', followed by $t_1, t_2, \ldots t_n$ terms enclosed in parentheses and separated by Commas.

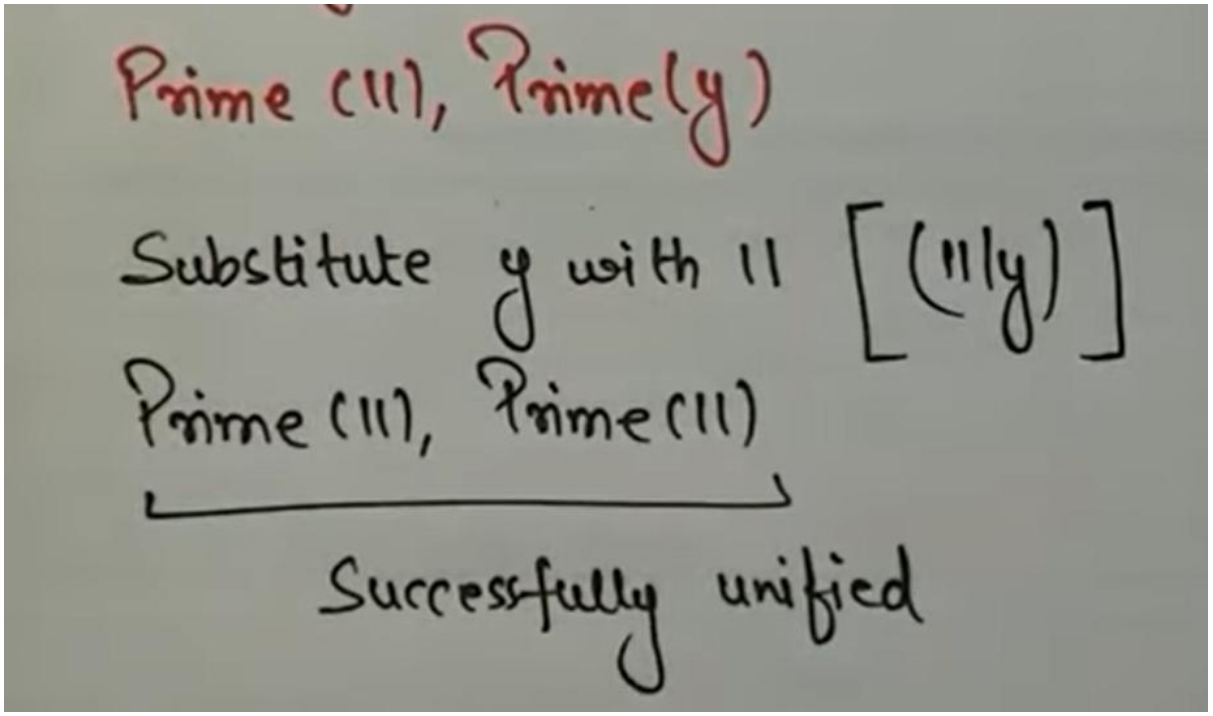
1. if 'S' is Sentence then ¬S is sentence.
2. if S1, S2 are sentences] S1 ∧ S2 (Conjuction)
3. " "] S1 ∨ S2 (disjunction)
4. " "] S1 → S2 (implication)
5. " "] S1 ≡ S2 (equivalence)
6. if x is Var. and s is Sentence the ∀x s is a Sentence.
7. " " → ∃x s is a Sentence.



Unification Algorithm







Forward chaining, Backward chaining

(A166)

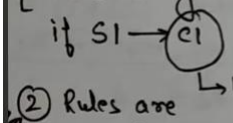
Ques:- Differentiate between Forward and Backward ~~chaining~~ Reasoning.

Forward Reasoning:- It moves forward from start to goal state. Also called as Data Driven Reasoning.

↳ Search Tree has Initial Configurations) at root of tree.

↳ Next Level of Tree is generated by finding all the rules whose left side matches the root node. [IF-THEN RULES]

[Medical Diagonsis]



① Initially System is provided with one or more constraint.

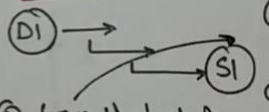
② Rules are Searches for each Constraint.

③ Satisfying rules (R.H.S) becomes L.H.S

Backward Reasoning:- It moves backward from goal to initial state. Also called as Goal Driven Reasoning.

↳ Search Tree has Goal Configuration(s) at the root of the tree.

↳ Next Level of Tree is generated by finding all the rules whose right side matches root nodes.



① Goal State and rules are selected where Goal State resides in THEN part.

② from If part of Selected rules subgoals are made

Properties :-

→ It moves from bottom to up (top)

→ It is a process of making a conclusion based on known facts of data, by starting from the initial state and reach the goal sta

→ Forward chaining approach is also called as data-driven as we rea

to the goal using available data.

→ Forward-chaining approach is commonly used in the Expert System.

Example :-

Rule 1 : If A and C then F
Rule 2 : If A and E then G
Rule 3 : If B then E
Rule 4 : If G then D

Database

A B

knowledge base

A & C → F
A & E → G
B → E
G → D

Problem :- Poove if A and B toue, then D is toue.

AIML

PAGE NO.:

(1) Problem: Prove if A & B true, then D is true

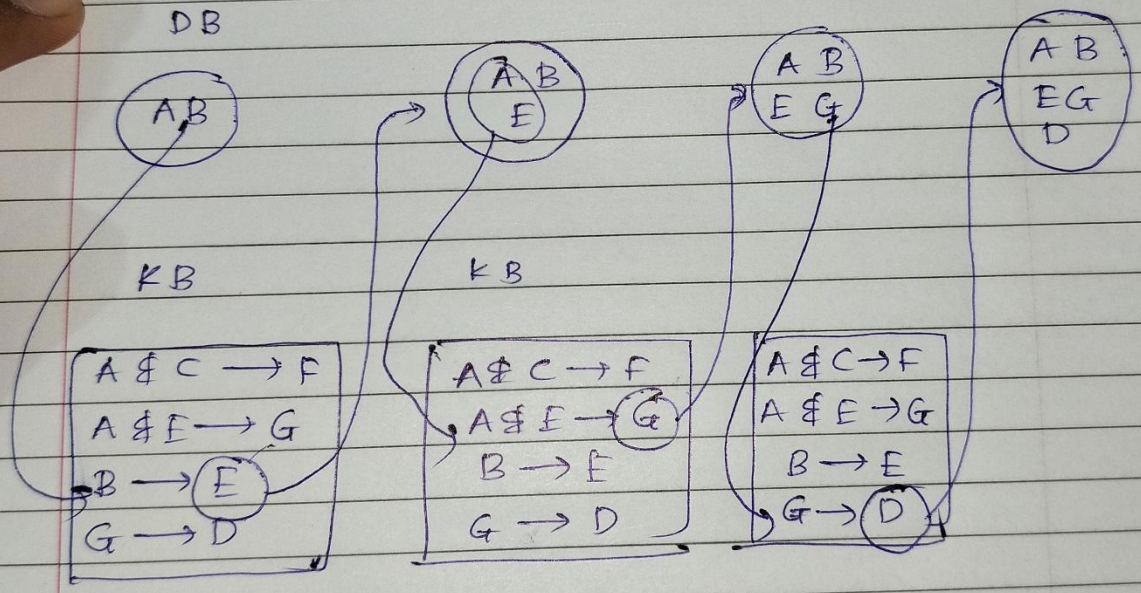
Knowledge base

A & C → F
A & E → G
B → E
G → D

Database

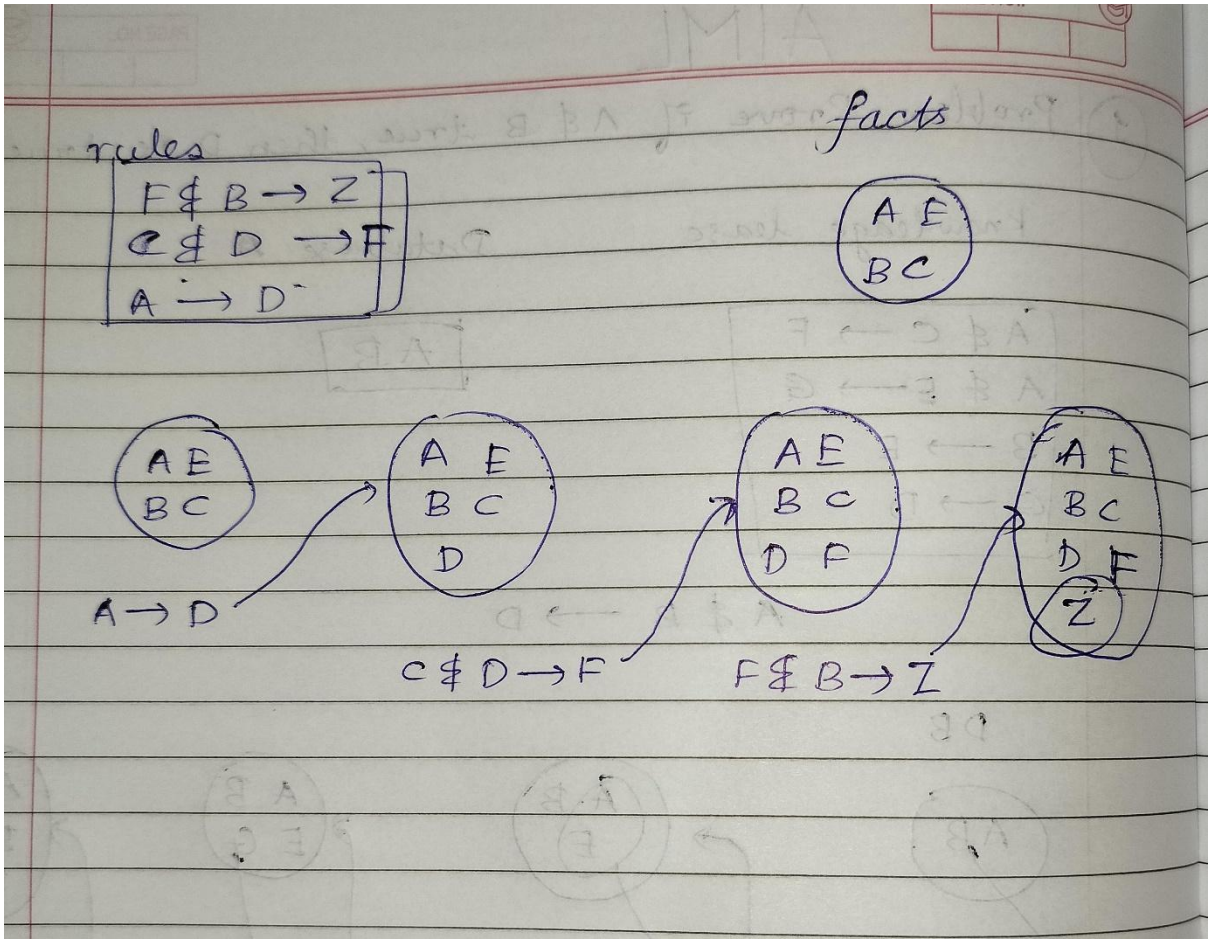
A B

A & B → D



(2) Goal State: Z

termination condition: Stop if Z is derived or no further rules can be applied



Backward Chaining :-

→ A Backward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through rules to find known facts support the goal.

Properties of backward chaining :-

→ It is known as a top-down approach

→ Backward-chaining is based on modus ponens inference rule.

→ In backward chaining, the goal is broken into sub-goal or sub-goals to prove the facts true.

→ It is called a goal-driven approach, as a list of goal decides

Properties of backward chaining :-

→ It is known as a top-down approach

→ Backward-chaining is based on modus ponens inference rule.

→ In backward chaining, the goal is broken into sub-goal or sub-goals to prove the facts true.

→ It is called a goal-driven approach, as a list of goal decides

1:08 / 9:01

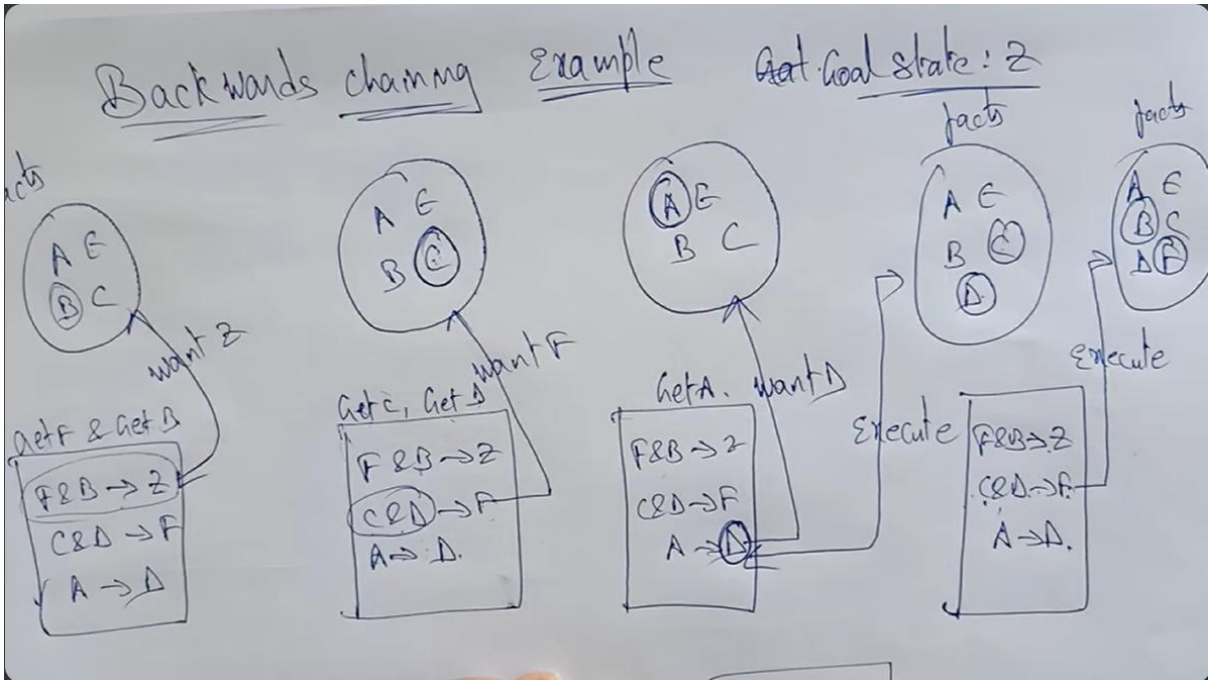
Scroll for details

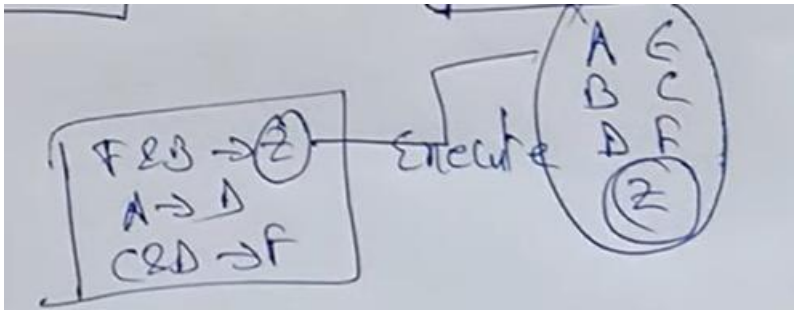
backward chaining example | Artificial intelligence | Lec-40 | Bhanu Priya

Which rules are selected and used

Backward-chaining algorithm is used in game theory, automated theorem proving tools, inference engines, proof assistants, and various AI applications.

The backward-chaining method mostly used a depth-first search strategy for proof.





⇒ Backward Chaining

- If based on the decision the initial state is fetched, then it is called as backward chaining.
- Backward chaining is also called as a decision-driven or goal-driven inference technique.

Example

We have to prove that colonel is a criminal

Given facts

- It is crime for an American to sell weapons to the enemy of America.
- Country Nono is an enemy of America.
- Nono has some Missiles.
- All the missiles were sold to Nono by colonel.
- Missile is a weapon.
- Colonel is American.

Facts to FOL

⇒ American(x) ∧ Weapon(y) ∧ Sell(x, y, z) ∧ enemy(z, America) ⇒ Criminal(x)

⇒ Enemy(Nono, America)

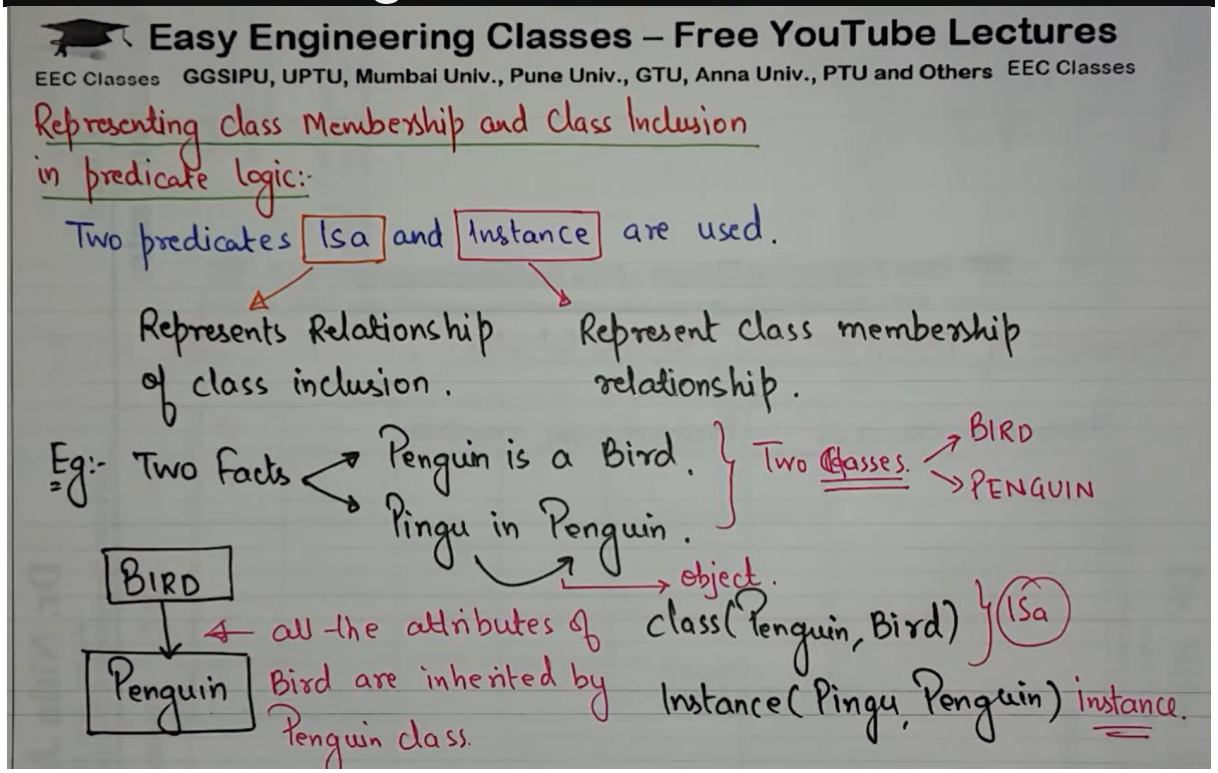
⇒ Owns(Nono, x) Missile(x)

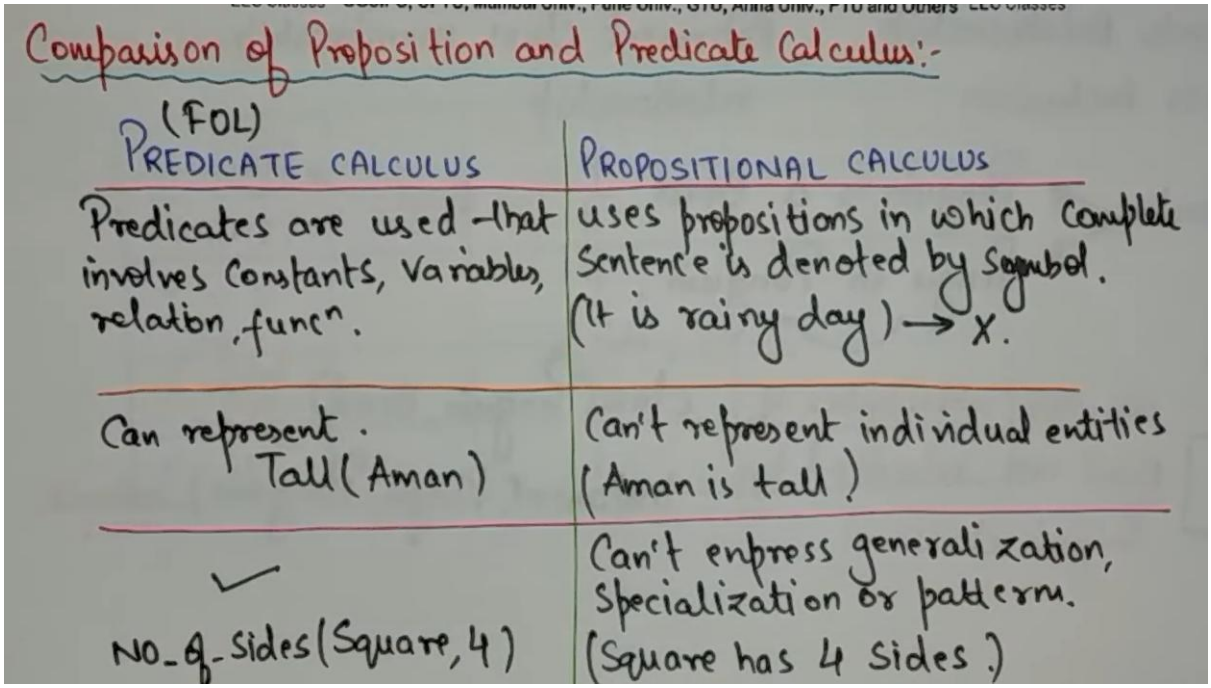
⇒ ∀x Missile(x) ∧ Owns(Nono, x) ⇒ Sell(colonel, x, Nono)

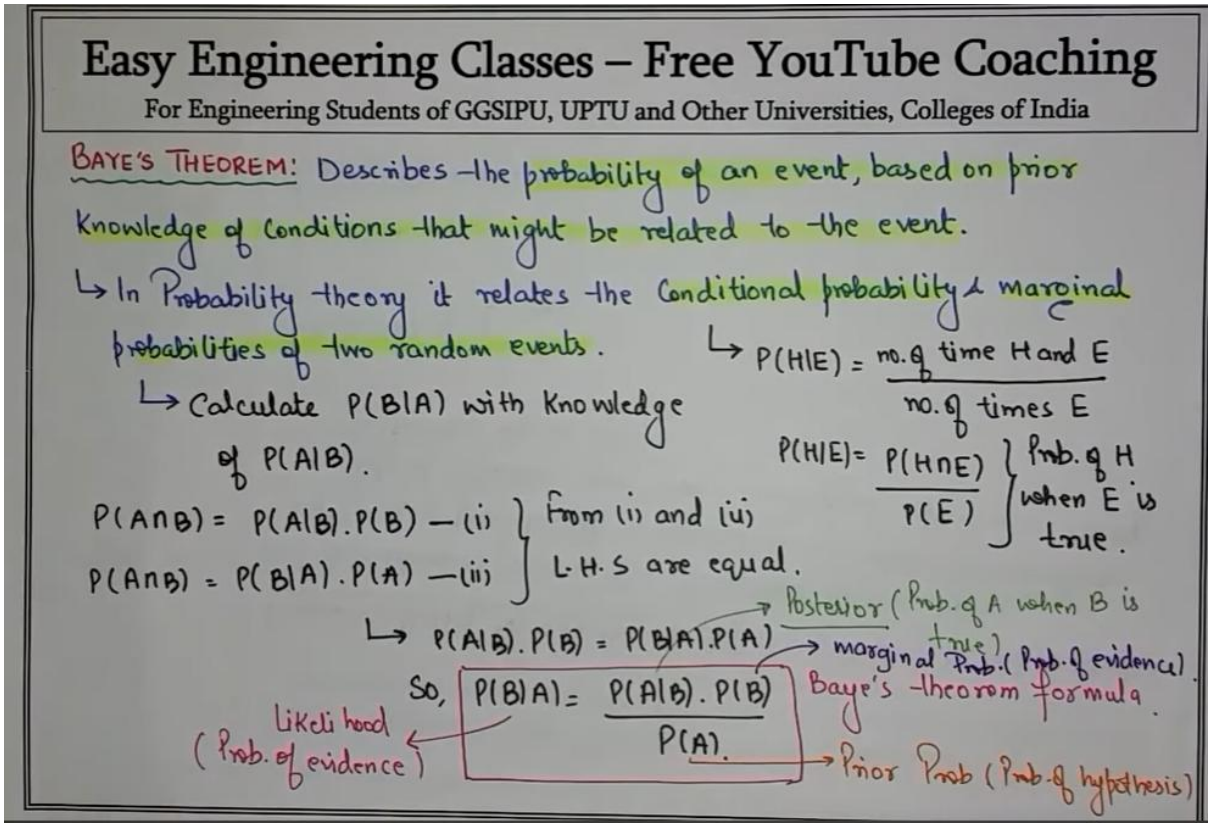
⇒ Missile(x) ⇒ Weapon(x)

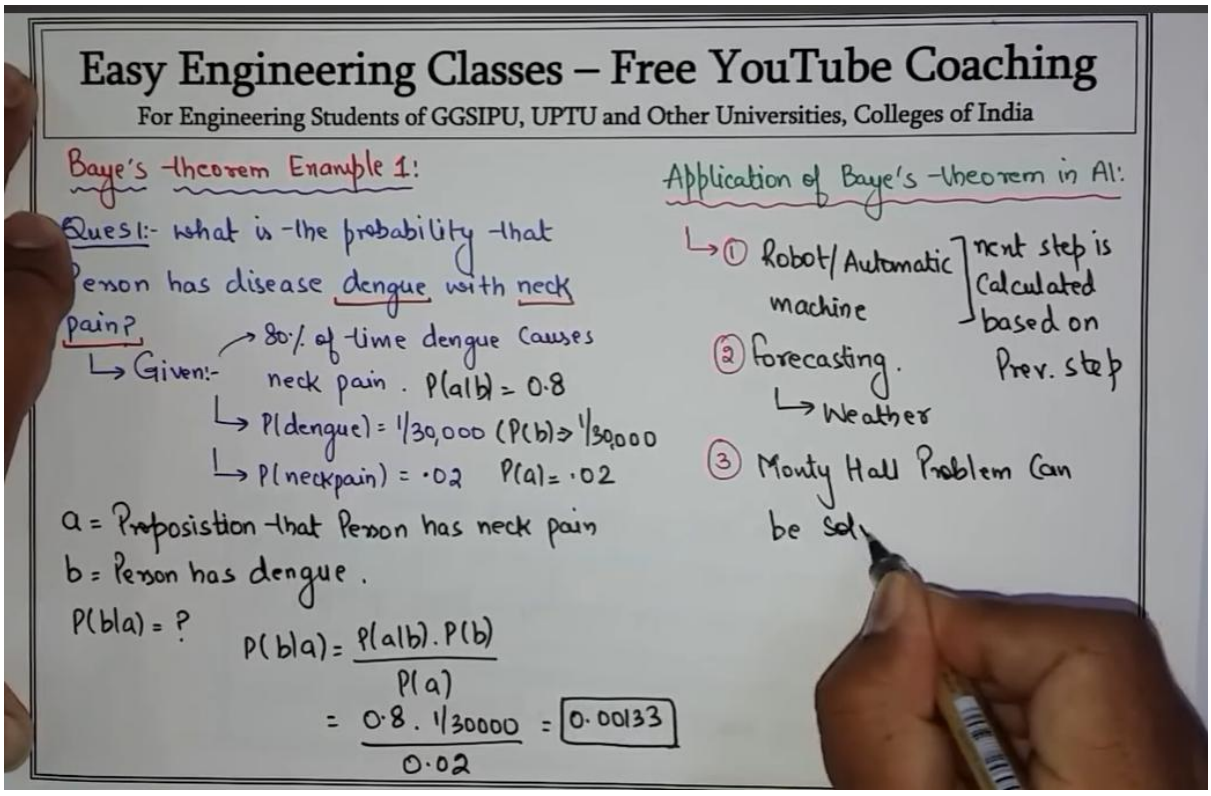
⇒ American(colonel)

Class Membership & Inclusion in Predicate Logic









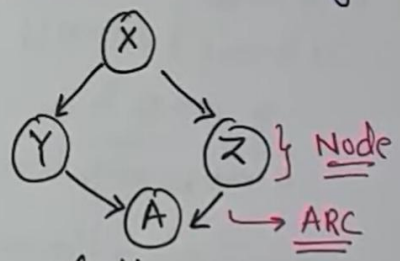
Bayesian Belief Network

Bayesian Belief Nlw in AI: It defines probabilistic independencies and dependencies among the Variables in the Nlw.

↳ "It is a probabilistic graphical model which represents a set of Variables and their conditional dependencies using a directed acyclic graph." (DAG).

↳ Built from Probability distribution.

↳ Consists of ① DAG. ② Table of Conditional Probabilities.



↳ Node: Corresponds to a Random Variable → Continuous / Discrete

↳ Arc / Directed arrows: rep. Casual relationship or Conditional Probabilities among random Variables.

knowledge based agent

logical agents

Agents with some representation of complex knowledge about the world / its environment & uses inference to ~~derive~~ derive new information from the knowledge combined with new i/ps.

Knowledge base :- set of sentences in a formal lang representing facts about the world

Knowledge - based agents

→ Intelligent agents need knowledge about the world to choose good actions/ decisions

→ knowledge = {sentences} in a knowledge representation language (formal lang)

→ A sentence is an assertion about the world.

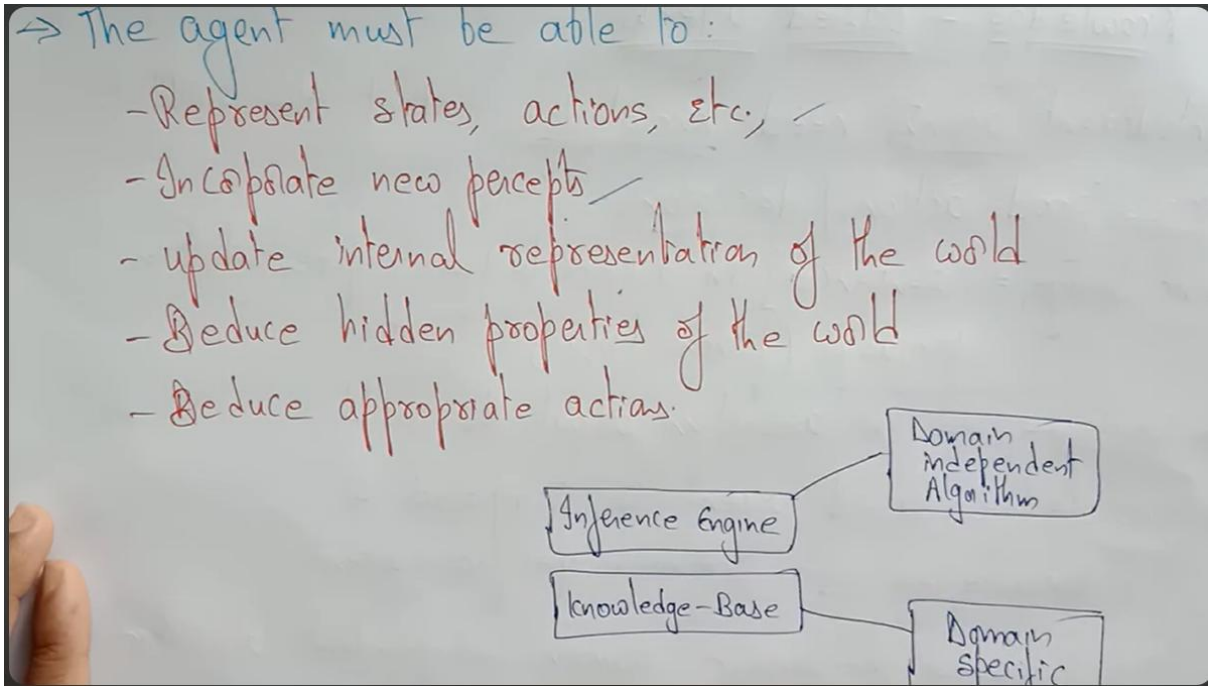
→ A knowledge - based agent is composed of:

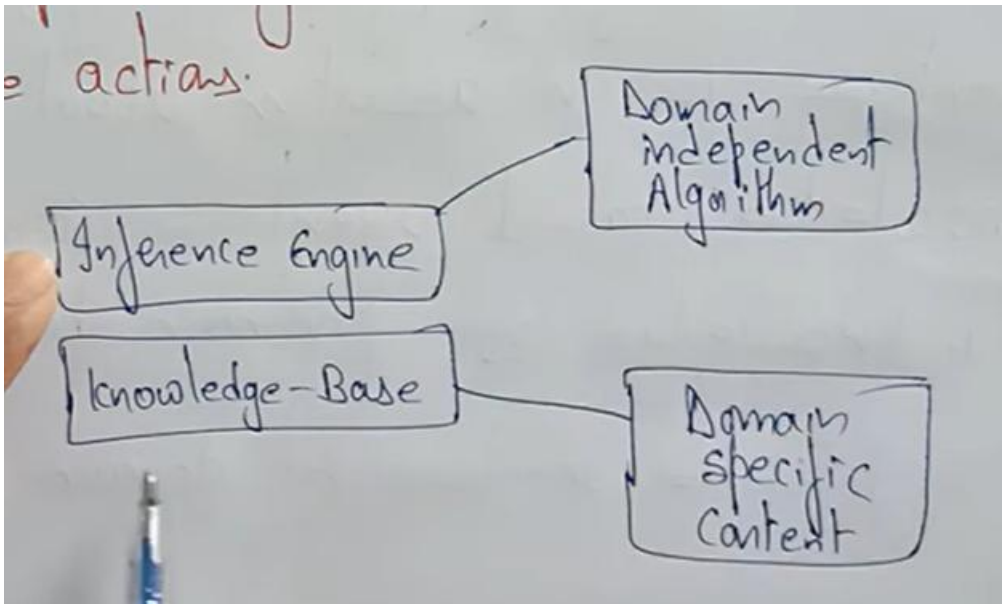
1. knowledge base: domain specific content

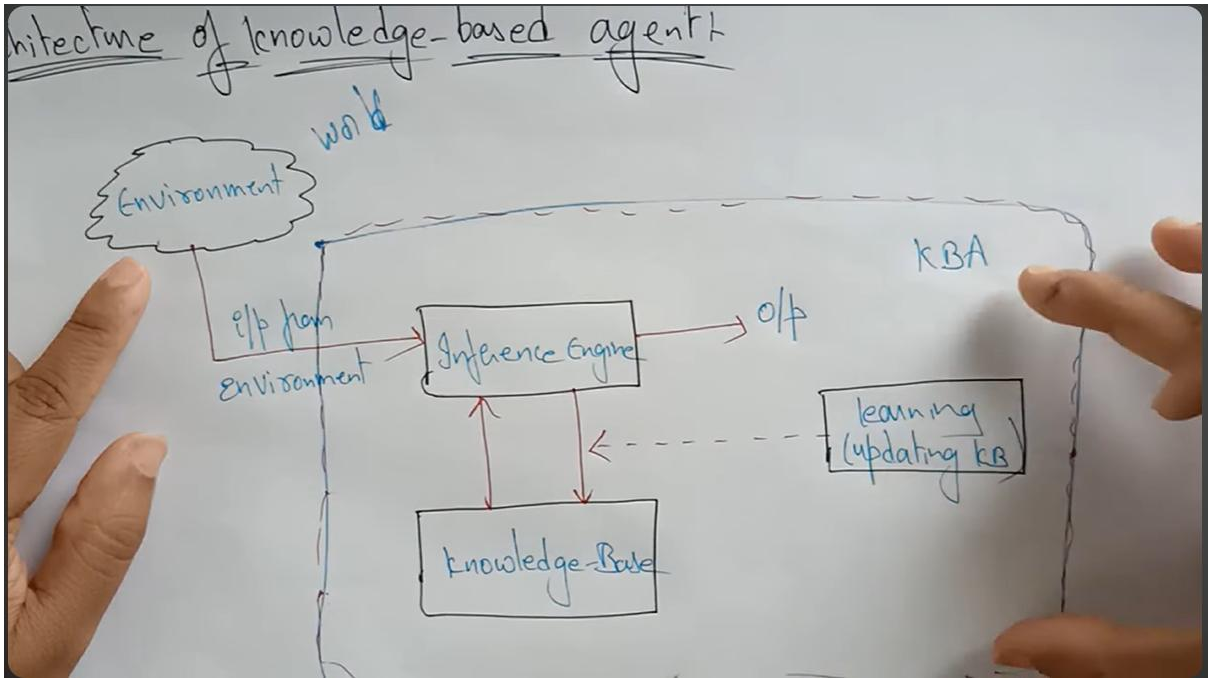
→ A knowledge - based agent is composed of:

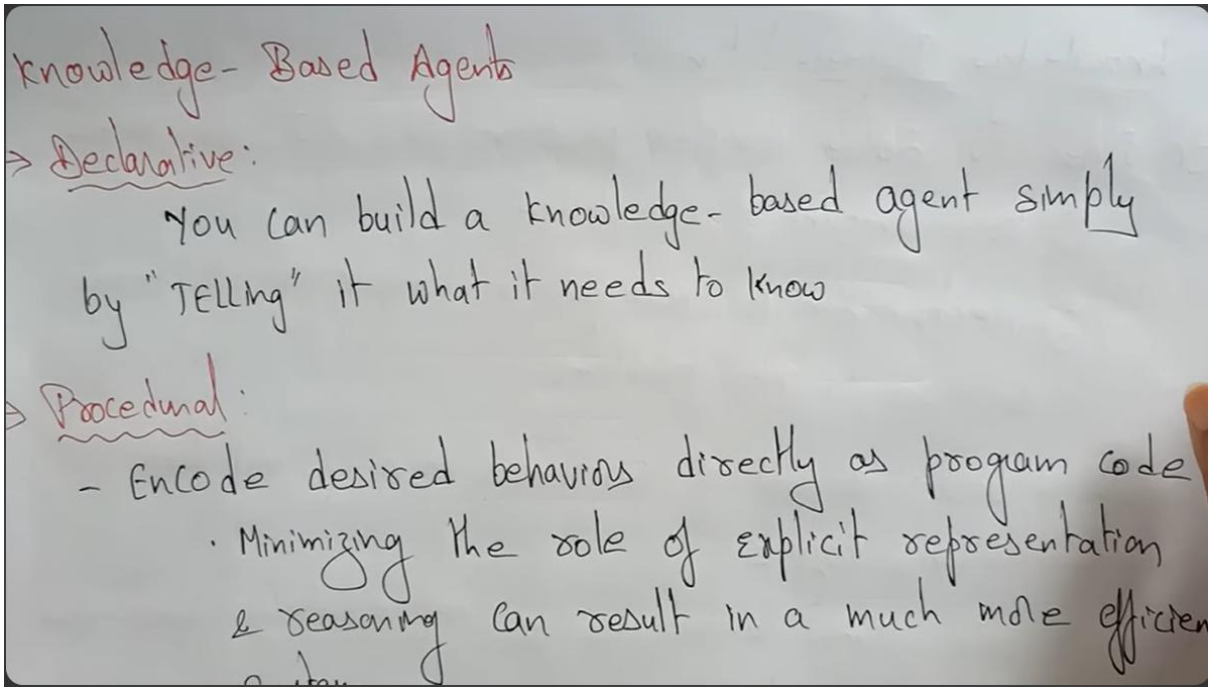
1. knowledge base: domain specific content

2. Inference mechansim: domain- independent algorithm


⇒ The agent must be able to:
- Represent states, actions, etc.,
- Incorporate new percepts
- update internal representation of the world
- Reduce hidden properties of the world
- Reduce appropriate actions.
Inference Engine
Domain independent Algorithm
knowledge-Base
Domain specific


e actions.
Inference Engine
Domain independent Algorithm
knowledge-Base
Domain specific Content





Techniques of knowledge representation

Techniques of knowledge representation

There are mainly four ways of knowledge representation

1. logical representation
2. Semantic n/w representation
3. Frame representation
4. Production rules

Logical representation : (proposition)

- It is a lang with some concrete rules which deals with propositions & has no ambiguity in representation

→ It consists of precisely defined
syntax & semantics
which supports the sound inference. Each sentence can be translated into logics using syntax & Semanti

- Syntax: defines well-formed Sentence in the l
- Semantics: defines the truth of meaning of sentence in a world

Logical Representation

→ Propositional logic
→ First order predicate logic

① Propositional logic: (PL)

→ PL is the simplest logic
→ A proposition is a declarative statement that's either T or

⇒ Propositional logic cannot predicate, it can say either true or false.

⇒ Connectives.

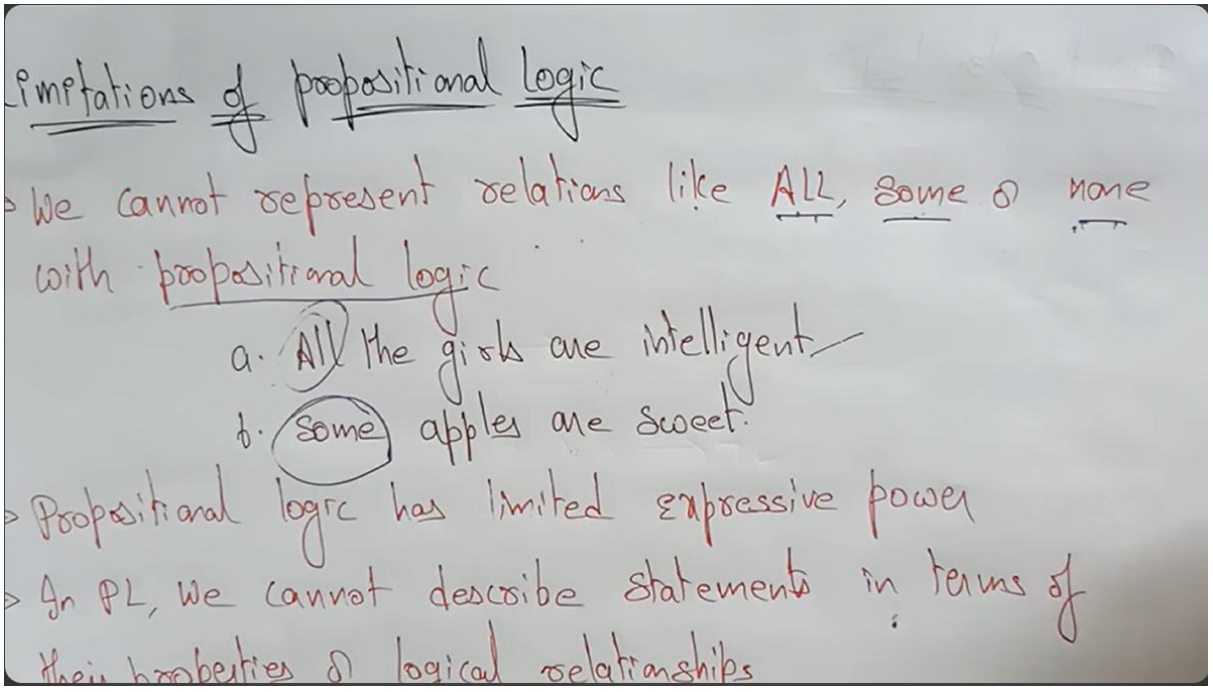
word	Symbol	example
→ Not	¬	¬A
→ and	∧	A∨B
→ OR	∨	A∧B
→ implies	→	A→B
→ if and only if (biconditional stmt)	↔	A↔B

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

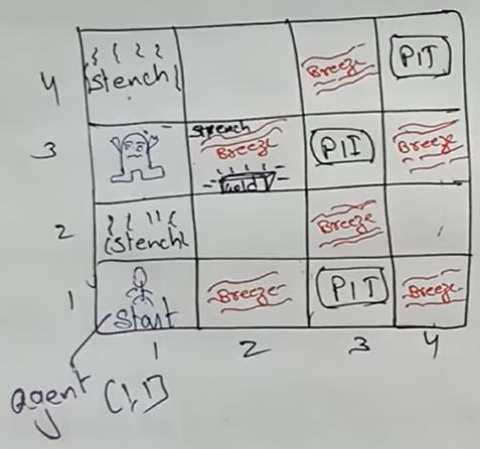
P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

P	Q	$P \leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T



wumpus world problem

mple: Propositional logic (Wumpus World Proving)



① The rooms adjacent to the wumpus are smelly (stench)

② the rooms adjacent to Pits has breeze.

③ There will be glitter in the room if & only if the room has gold.

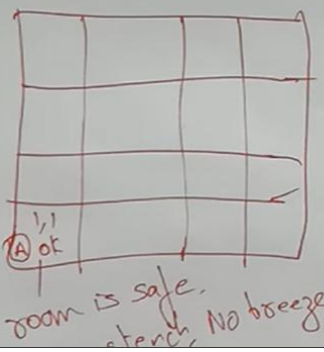
[1,1]

Actuators: left move, right move, grab, realse.

Scensors: stench, breeze, glitter.

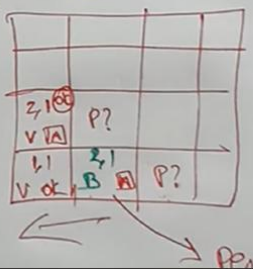
Agent 1st step:

B = breeze G = glitter P = pit V = visited
A = Agent ok = safe S = stench W = wumpus

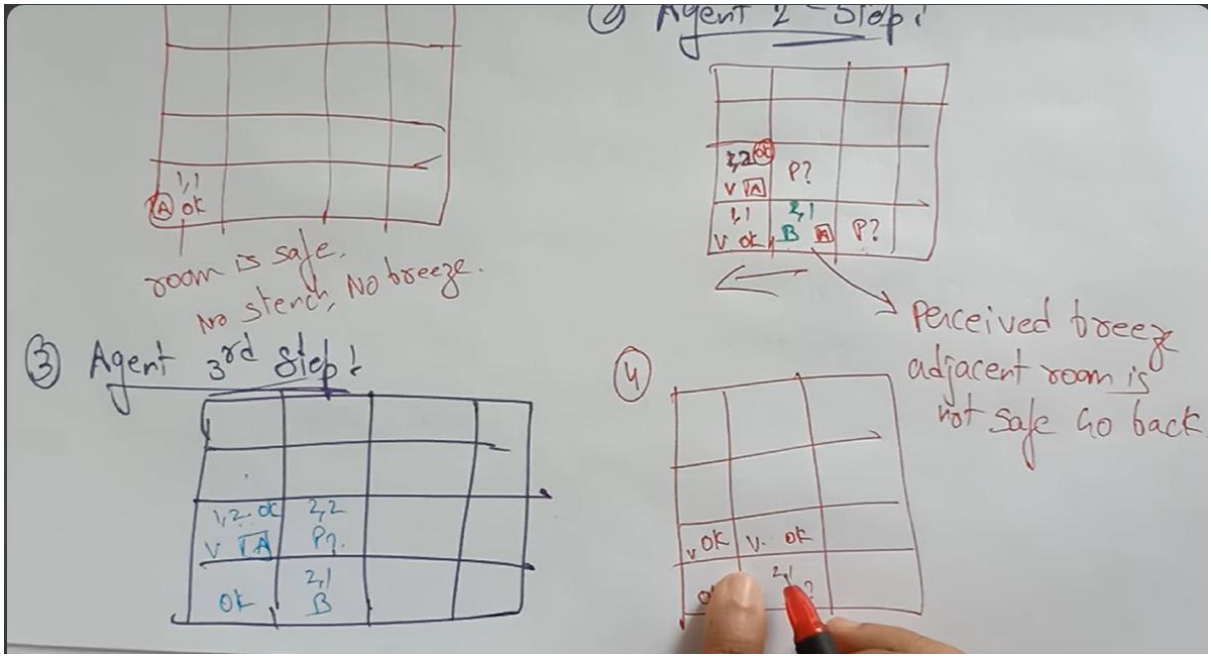


room is safe.
stench, No breeze.

② Agent 2nd step:



Perceived



Atomic Proposition Variables for Wumpus world

1. Let $P_{i,j}$ be true if there is a pit in the room $[i,j]$
2. let $B_{i,j}$ be true if agent perceives breeze in $[i,j]$
3. let $W_{i,j}$ be true if there is Wumpus in square $[i,j]$
4. let $S_{i,j}$ be true if agent perceives stench in $[i,j]$
5. let $V_{i,j}$ be true if that square $[i,j]$ is visited
6. let $G_{i,j}$ be true if there is gold
7. let $OK_{i,j}$ be true if room is safe

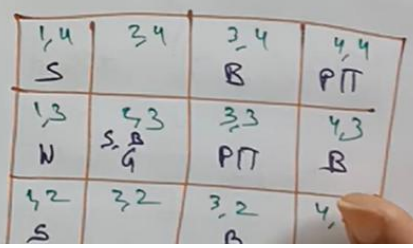
Some propositional rules for wumpus world

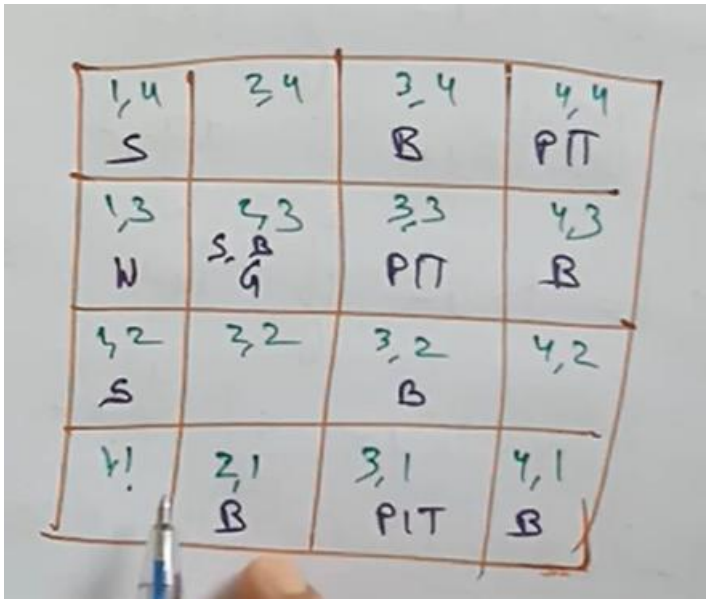
$R_1 \Rightarrow \neg S_{11} \rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$

$R_2 \Rightarrow \neg S_{21} \rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22} \wedge \neg W_{31}$

$R_3 \Rightarrow \neg S_{12} \Rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{22} \wedge \neg W_{13}$

$R_4 \Rightarrow S_{12} \rightarrow W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$



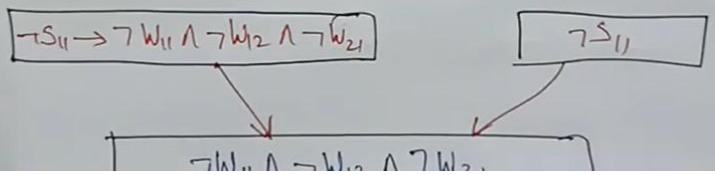


→ We can poove Wumpus is in the room (1,3) using propositional rules

① Apply Modus ponens with $\neg S_{11}$ & R_1:

⇒ We first apply MP rule with R_1 which is

$\neg S_{11} \rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$ & $\neg S_{11}$ which will gives the o/p $\neg W_{11} \wedge W_{12} \wedge W_{12}$.

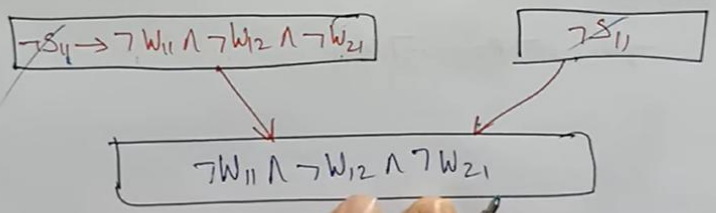


propositional rules

① Apply Modus ponens with $\neg S_{11}$ & R_1:

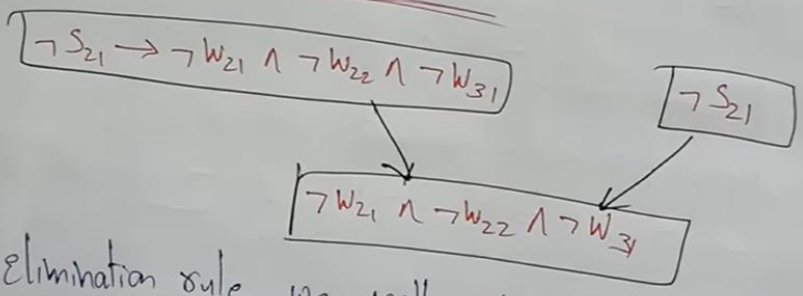
⇒ We first apply MP rule with R_1 which is

$\neg S_{11} \rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$ & $\neg S_{11}$ which will gives the o/p $\neg W_{11} \wedge W_{12} \wedge W_{12}$.



$\neg W_{11}$, $\neg W_{12}$ & $\neg W_{21}$

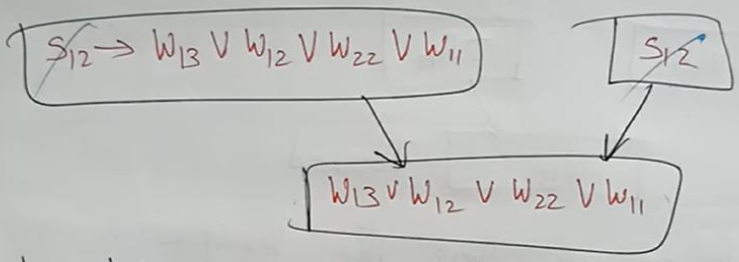
⇒ Apply Modus ponens to $\neg S_{21}$ & R_2:



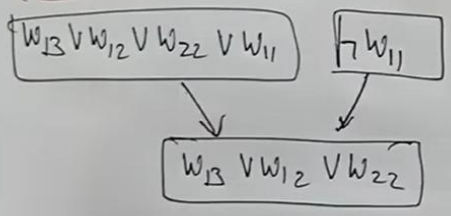
Apply & Elimination rule we will get three statements

$\neg W_{21}$, $\neg W_{22}$, $\neg W_{31}$

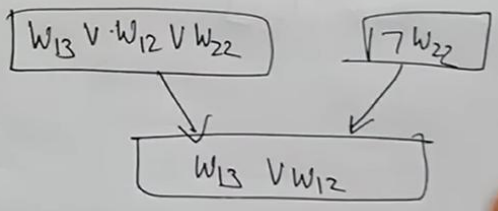
⇒ Apply Modus ponens to S_{12} & R_4:

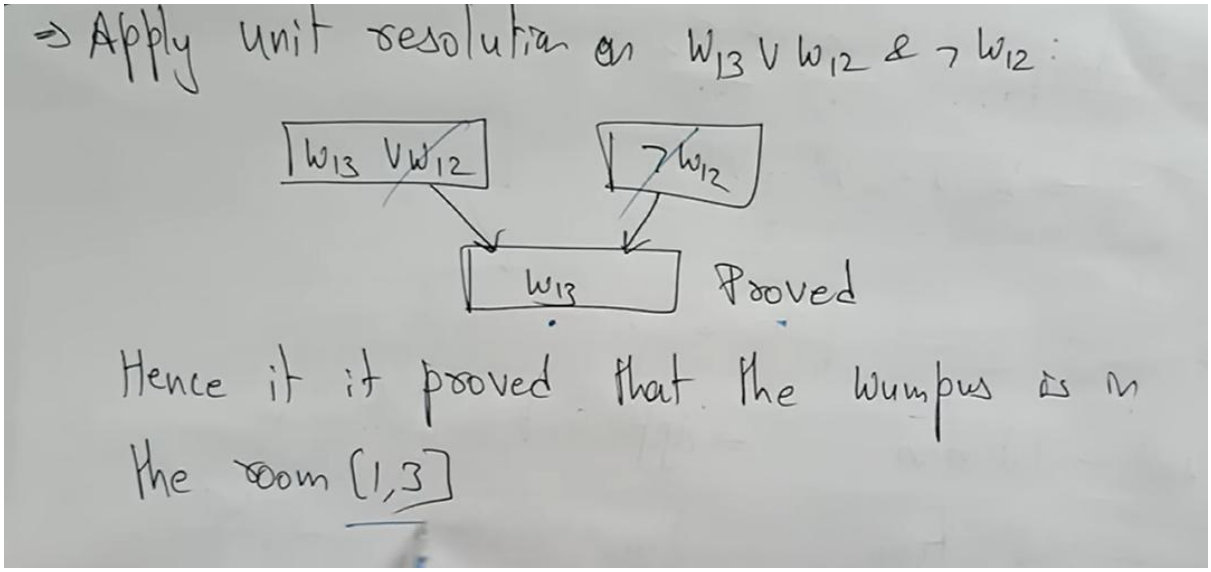


⇒ Apply unit resolution on



⇒ Apply unit resolution on





Steps to Convert the Propositional Logic Statement into CNF Conjuctive Normal Form

Steps to Convert Logic Statement to CNF in AI

- The **conjunctive normal form** also called the **clausal normal form** states that a formula is in CNF if it is a **conjunction of one or more than one clause**, where each **clause is a disjunction of literals**.

$$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B \vee \neg C)$$

Steps to Convert Logic Statement to CNF in AI

1. Eliminate biconditionals and implications:
   - Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   - Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

Steps to Convert Logi

2. Move $\neg$ inwards:

- $\neg(\forall x \ \ p) \equiv \exists x \ \neg p,$
- $\neg(\exists x \ \ p) \equiv \forall x \ \neg p,$
- $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta,$
- $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta,$
- $\neg\neg\alpha \equiv \alpha.$

Steps to Convert Logic Statement to CNF in AI

3. Standardize variables apart by renaming them: each quantifier should use a different variable.

4. Skolemize: each existential variable is replaced by a *Skolem constant* or *Skolem function* of the enclosing universally quantified variables.

   - For instance, $\exists x \, Rich(x)$ becomes $Rich(G1)$ where $G1$ is a new Skolem constant.
   - "Everyone has a heart" $\forall x \ Person(x) \Rightarrow \exists y \ \ Heart(y) \wedge Has(x, y)$ becomes $\forall x \ \ Person(x) \Rightarrow Heart(H(x)) \wedge Has(x, H(x))$, where H is a new symbol (Skolem function).

Steps to Convert Logic Statement to CNF in AI

5. Drop universal quantifiers
   - For instance, $\forall x \; Person(x)$ becomes $Person(x)$.

6. Distribute $\wedge$ over $\vee$:
   - $(\alpha \wedge \beta) \vee \gamma \equiv (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$.

$$A \rightarrow (B \leftrightarrow C)$$

$$= \neg A \vee (B \leftrightarrow C)$$

$$= \neg A \vee ((\neg B \vee C) \wedge (B \vee \neg C))$$

$$= (\neg A \vee (\neg B \vee C)) \wedge (\neg A \vee (B \vee \neg C))$$

$A \rightarrow B \equiv \neg A \vee B$

$B \leftrightarrow C \equiv (B \rightarrow C) \wedge (C \rightarrow B)$

Simple Linear Regression

* Equation of Regression line

$$y' = ax + b$$

$$a = \frac{n\Sigma xy - \Sigma x \Sigma y}{n\Sigma x^2 - (\Sigma x)^2} =$$

$$b = \frac{1}{n}(\Sigma y - a\Sigma x) =$$

=>Expendeture of business (in thousands) for every year is shown in table below:

X (Year)	1	2	3	4	5
Y (Expendeture)	12	19	29	37	45

- Find the expenditure of company in 6th year using line of a model.

Sol:

Sr. No	X	Y
1	1	12
2	2	19
3	3	29
4	4	37
5	5	45

=>Expendeture of business (in thousands) for every year is shown in table below:

X (Year)	1	2	3	4	5
Y (Expendeture)	12	19	29	37	45

- Find the expenditure of company in 6th year using line of a model.

Sol:

Sr. No	X	Y	XY	X^2
1	1	12	12	1
2	2	19	38	4
3	3	29	87	9
4	4	37	148	16
5	5	45	225	25
	$\Sigma X = 15$	$\Sigma Y = 142$	$\Sigma XY = 510$	$\Sigma X^2 = 55$

* Equation of Regression line

$$Y' = aX + b$$

$$a = \frac{n\Sigma xy - \Sigma x \Sigma y}{n\Sigma x^2 - (\Sigma x)^2} = \frac{5 \times 510 - 15 \times 142}{5 \times 55 - (15)^2} = 8.4$$

$$b = \frac{1}{n}(\Sigma y - a\Sigma x) = \frac{1}{5}(142 - 8.4 \times 15) = 3.2$$

Equation of line:

$$Y' = 8.4X + 3.2$$

* for the sixth month

$$Y' = 8.4 \times 6 + 3.2$$

$$= 53.6 \text{ Thousands}$$

Linear Regr

Q: Consider the following & obtain the Multiple Linear Equation

Sr. No	Y	X_1	X_2	$(X_1)^2$	$(X_2)^2$	X_1Y	X_2Y	X_1X_2
1	-37	3	8	9	64	-11.1	-29.6	24
2	3.5	4	5	16	25	14	17.5	20
3	2.5	5	7	25	49	12.5	17.5	35
4	11.5	6	3	36	9	69	34.5	18
5	5.7	2	1	4	1	11.4	5.7	2
	$\Sigma Y = 19.5$	$\Sigma X_1 = 20$	$\Sigma X_2 = 24$	$\Sigma(X_1)^2 = 90$	$\Sigma(X_2)^2 = 148$	$\Sigma X_1Y = 95.8$	$\Sigma X_2Y = 45.6$	$\Sigma X_1X_2 = 99$

1. $\Sigma x_1^2 = \Sigma(X_1)^2 - \frac{(\Sigma X_1)^2}{n}$

$\Sigma x_1 \quad \Sigma x_2 \quad \Sigma x_1 y \quad \Sigma x_2 y \quad \Sigma x_1$

$= 90 \quad = 148 \quad = 95.8 \quad = 45.6 \quad = 9$

1. $\Sigma x_1^2 = \Sigma (x_1)^2 - \frac{(\Sigma x_1)^2}{N} = 90 - \frac{(20)^2}{5} = 90 - 80 = 10$

2. $\Sigma x_2^2 = \Sigma (x_2)^2 - \frac{(\Sigma x_2)^2}{N} = 148 - \frac{(24)^2}{5} = 32.8$

3. $\Sigma x_1 y = \Sigma x_1 y - \frac{(\Sigma x_1)(\Sigma y)}{N} = 95.8 - \frac{20 \times 19.5}{5} = 17.8$

4. $\Sigma x_2 y = \Sigma x_2 y - \frac{(\Sigma x_2)(\Sigma y)}{N}$

$= 45.8 - \frac{24 \times 19.5}{5}$

$= 45.6 - 93.6$

$= -48$

5. $\Sigma x_1 x_2 = \Sigma x_1 x_2 - \frac{(\Sigma x_1)(\Sigma x_2)}{N}$

$= 99 - \frac{20 \times 24}{5}$

$= 99 - 96 = 3$

$\theta_0 = \bar{Y} - \theta_1 \bar{x_1} - \theta_2 \bar{x_2}$

$= \frac{19.5}{5} - 2.28 \times \frac{20}{5} - (-1.67) \times \frac{24}{5}$

$= 3.9 - 9.12 - (-8.016)$

$= 2.796$

$\theta_1 = \frac{(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$

$= \frac{32.8 \times 17.8 - (3) \times (-48)}{10 \times 32.8 - (3)^2}$

$= 2.28$

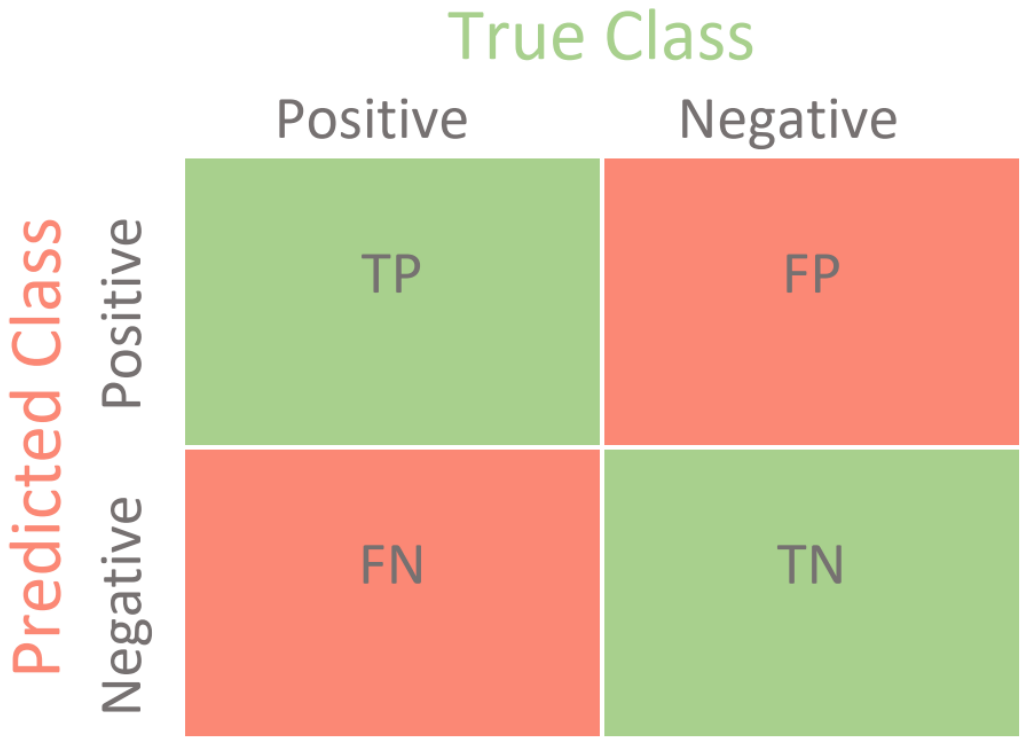
$\theta_2 = \frac{(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$

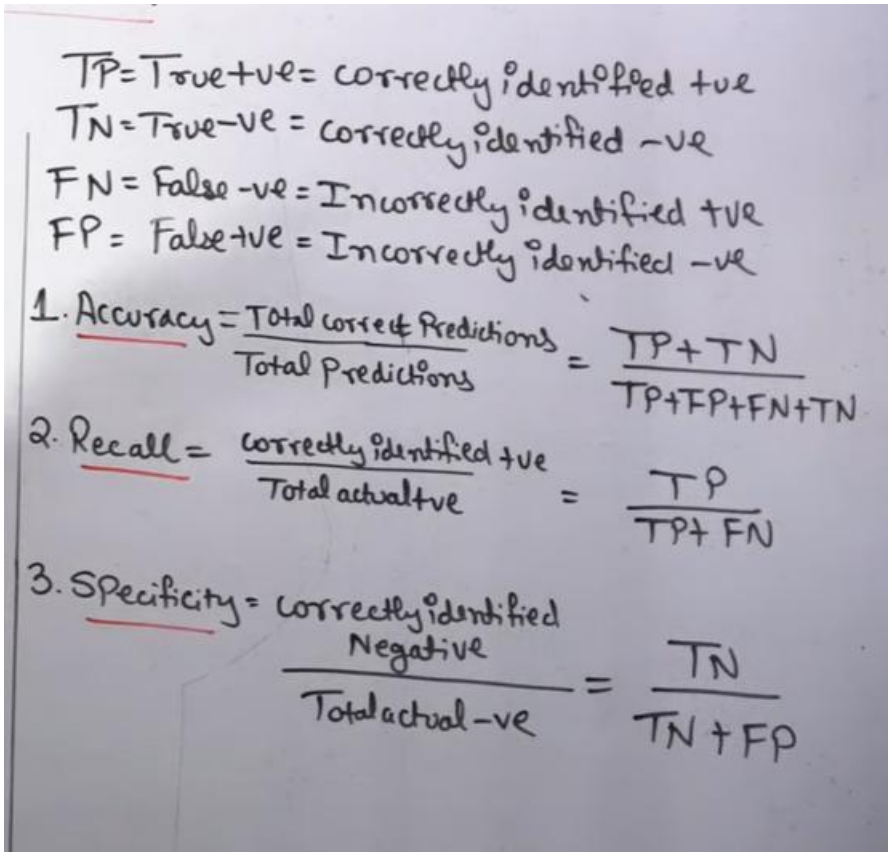
$= \frac{(10)(-48) - (3)(17.8)}{(10)(32.8) - (3)^2}$

$= -1.67$

$Y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

$= 2.796 + 2.28 x_1 - 1.67 x_2$


True Class
Positive
Negative
Predicted Class
Positive
Negative
TP
FP
FN
TN


TP = True +ve = correctly identified +ve
TN = True -ve = correctly identified -ve
FN = False -ve = Incorrectly identified +ve
FP = False +ve = Incorrectly identified -ve
1. Accuracy = Total correct Predictions / Total Predictions = TP+TN / TP+FP+FN+TN
2. Recall = correctly identified +ve / Total actual +ve = TP / TP+FN
3. Specificity = correctly identified Negative / Total actual -ve = TN / TN+FP

$$\text{Precision} = \frac{\text{correctly identified tves}}{\text{Total tves Predicted}} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{\text{Correctly identified tves}}{\text{Total actual tves}} = \frac{TP}{TP+FN}$$

F1 Score

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Kappa Statistics

Kap

Q. Suppose you are analyzing data for 50 people who are applying for a grant. Each proposal of grant was red by two readers where each of them either said yes or no to the proposal. Suppose the disagreement count data were as follows where A & B are readers

		B	
		Yes	No
A	Yes	20	5
	No	10	15

Calculate Cohen's Kappa coefficient.

Step1: Calculate Relative agreement between Raters

$$P_o = \frac{\text{Both said yes + Both said No}}{\text{Total}}$$

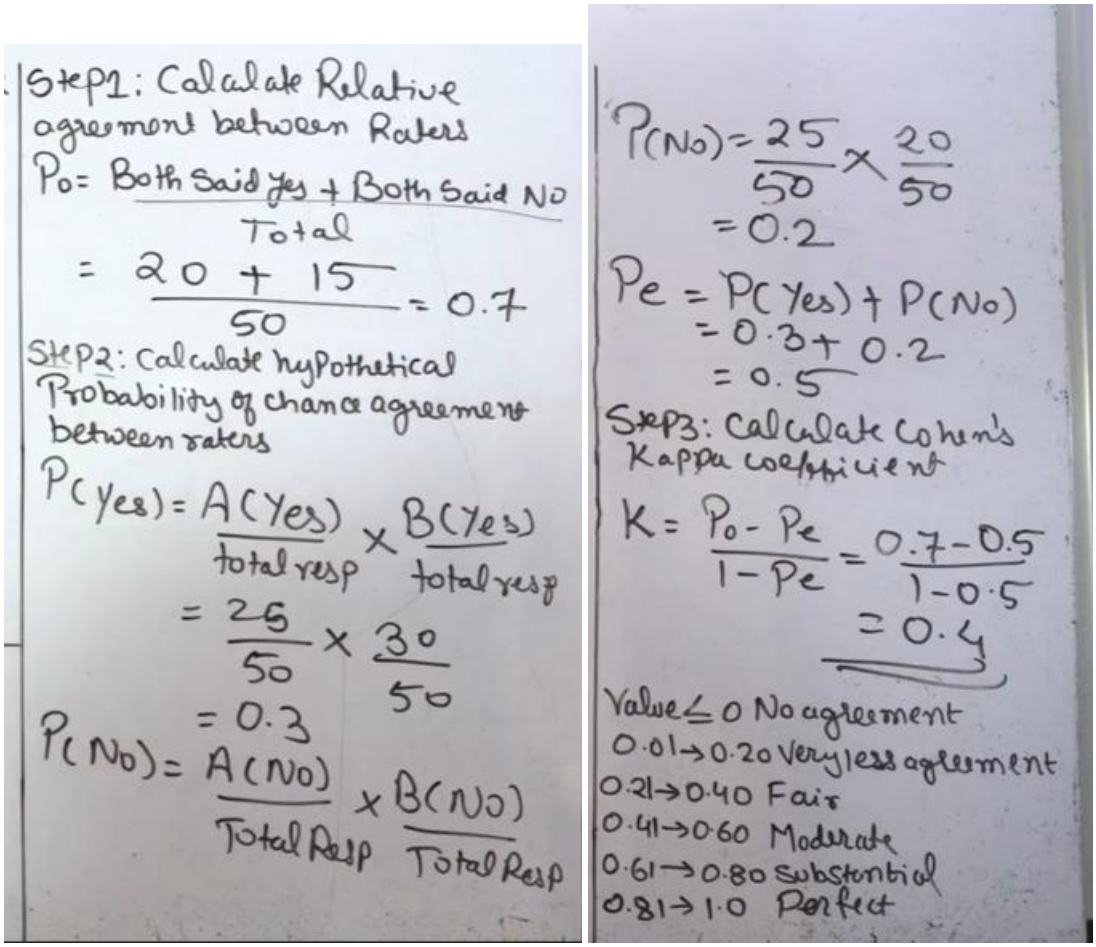
Step2: Calculate hypothetical Probability of chance agreement between raters

P(Yes) & P(No)

Step3: Calculate Kappa Statistics

$$K = \frac{P_o - P_e}{1 - P_e}$$

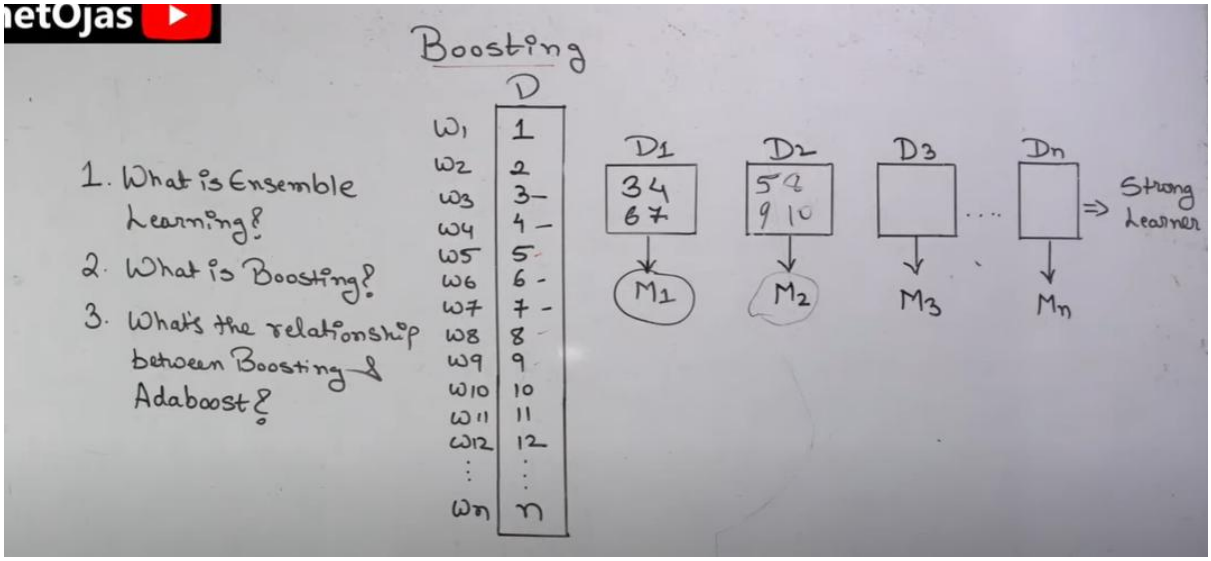
Value ≤ 0 No agreement
0.01 → 0.20 Very less agreement
0.21 → 0.40 Fair
0.41 → 0.60 Moderate
0.61 → 0.80 Substantial
0.81 → 1.0 Perfect

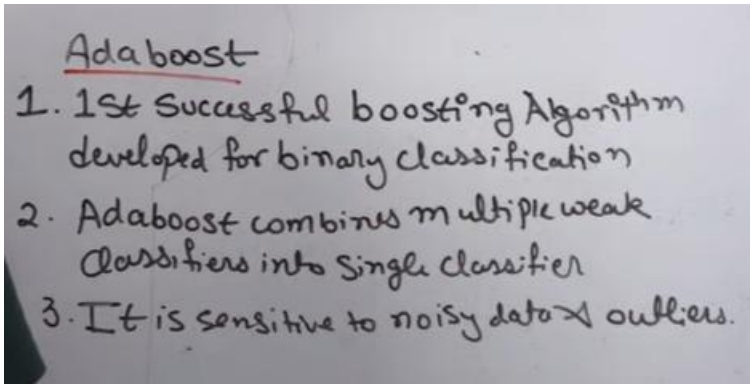


Ensemble Learning

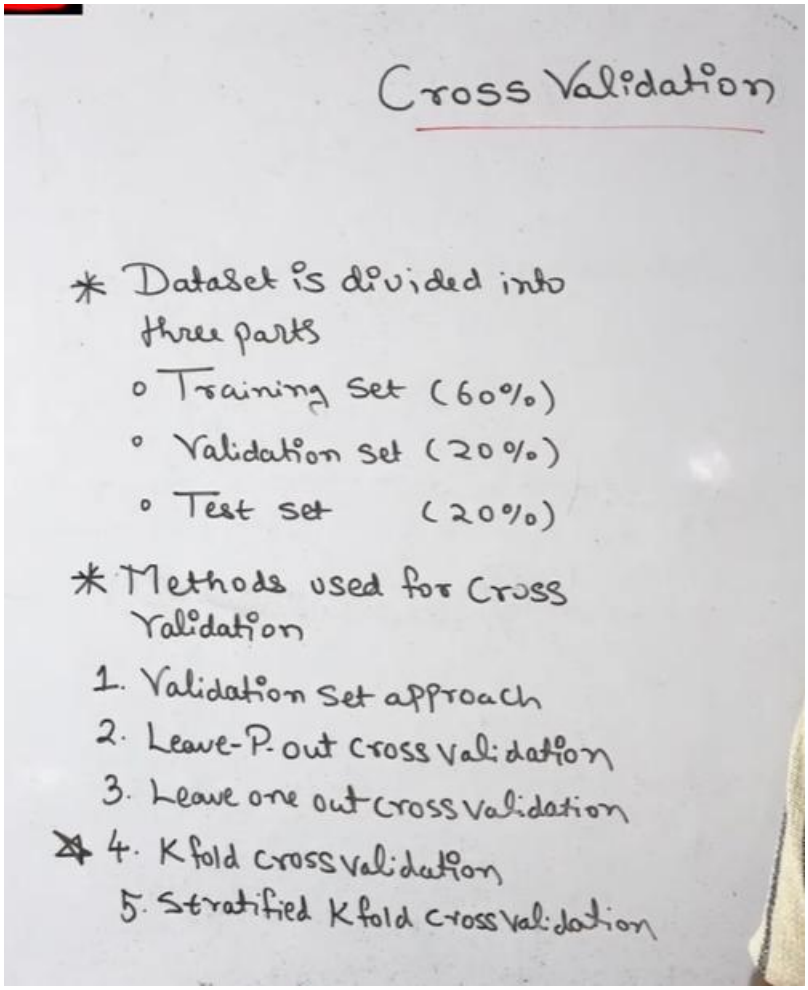
https://www.geeksforgeeks.org/ensemble-classifier-data-mining/

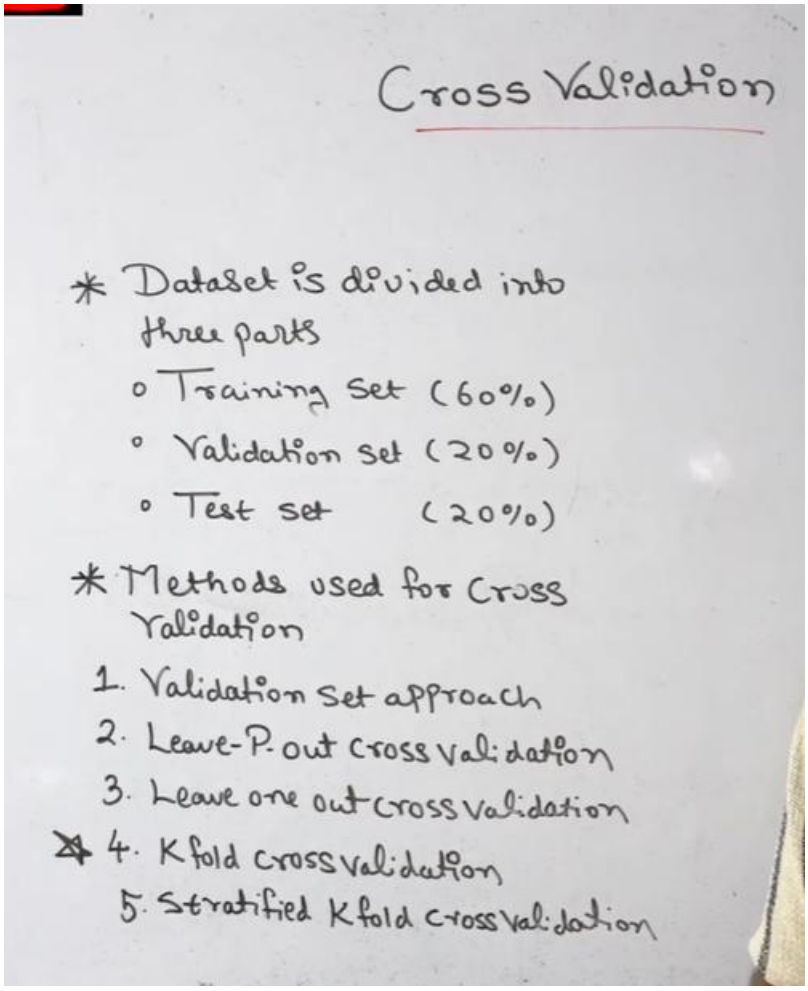
Boosting

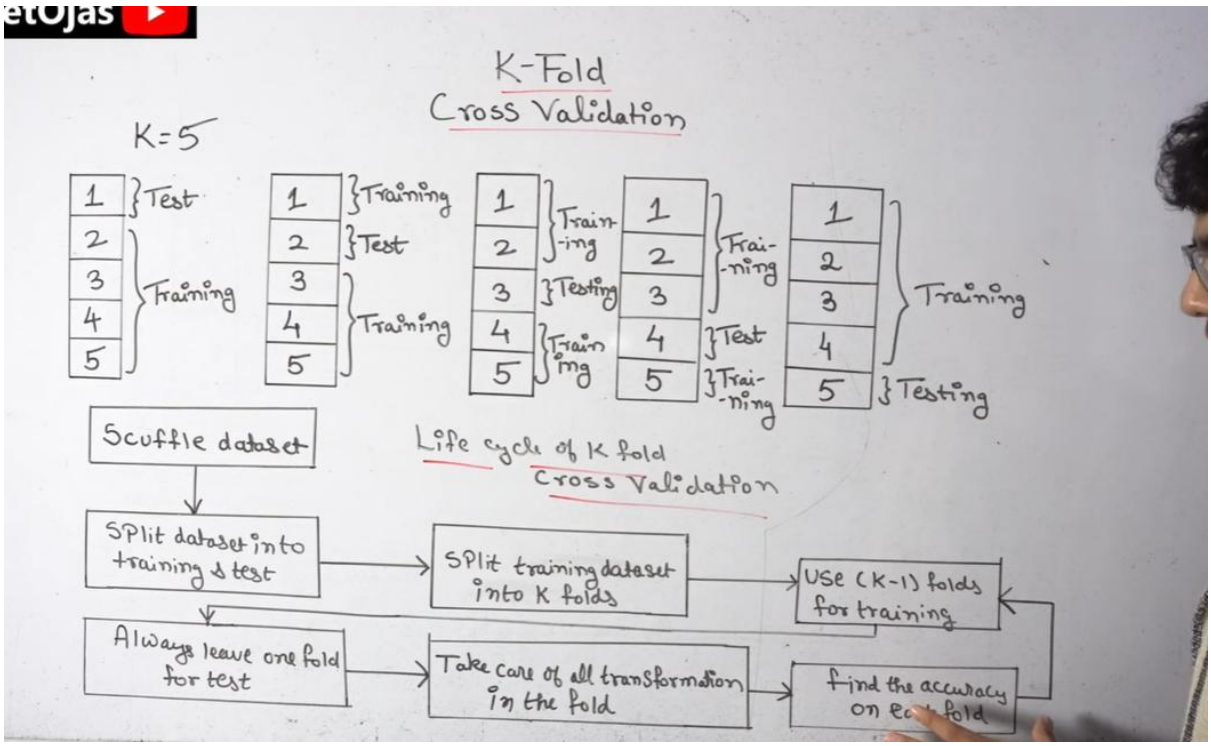


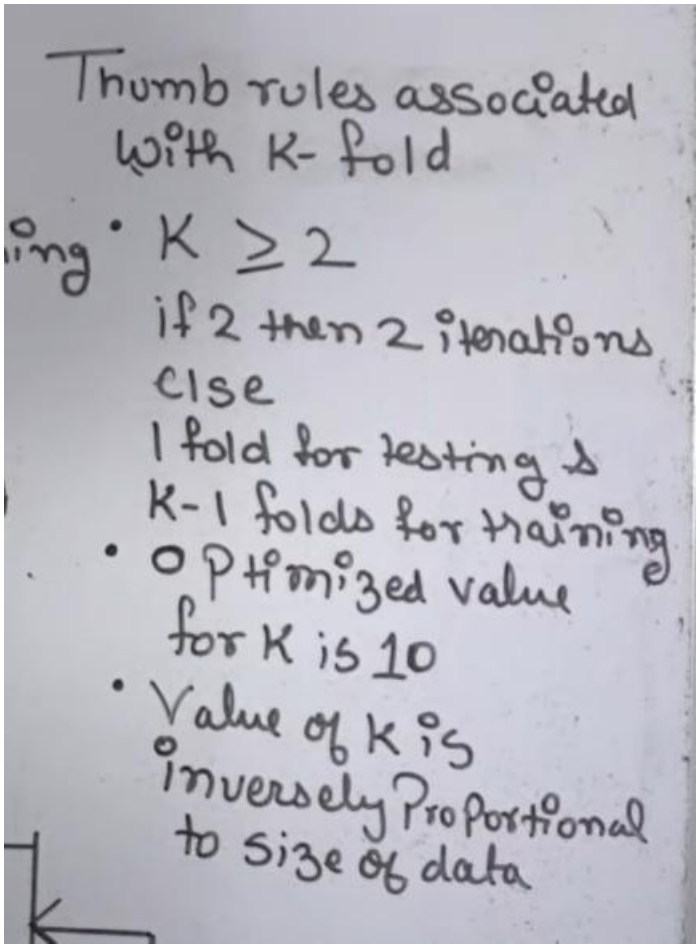


Cross Validation

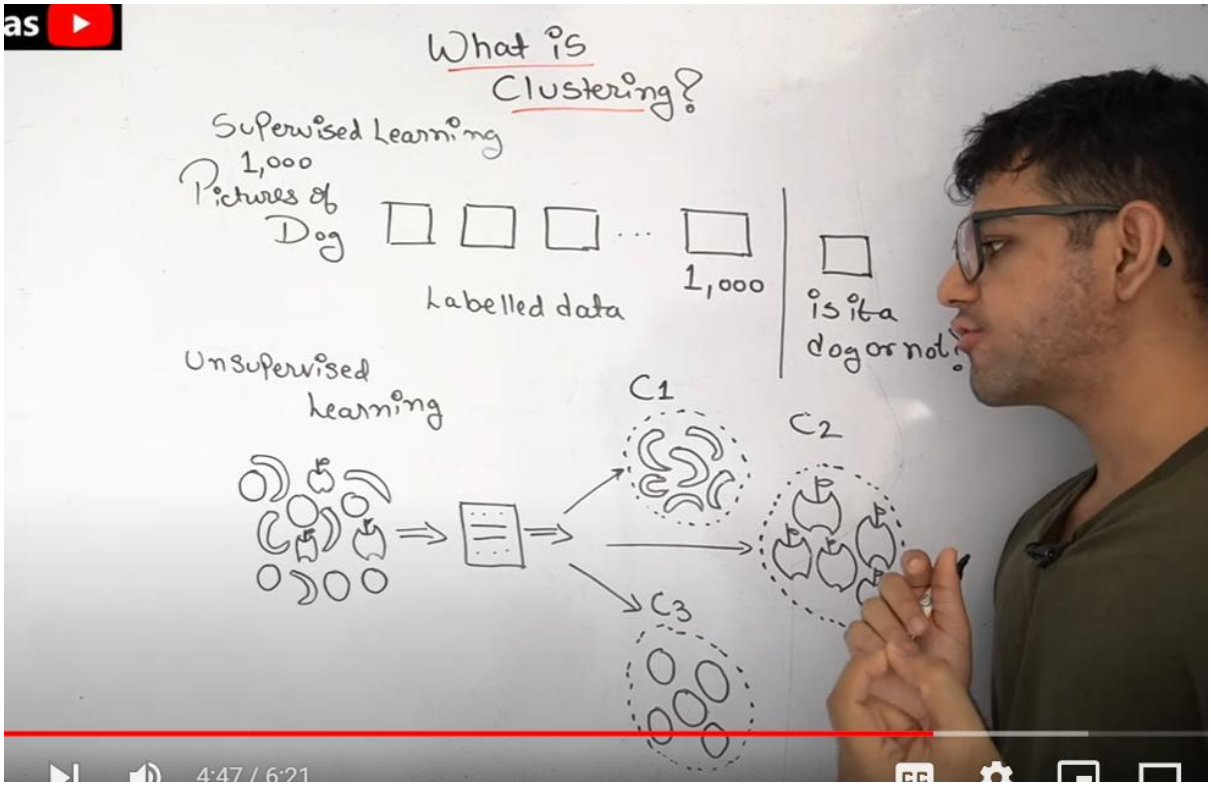



Cross Validation
* Dataset is divided into three parts
o Training Set (60%)
o Validation set (20%)
o Test set (20%)
* Methods used for Cross Validation
1. Validation Set approach
2. Leave-P-out cross validation
3. Leave one out cross validation
4. K fold cross validation
5. Stratified K fold cross validation


K-Fold
Cross Validation
K=5
Test
Training
Training
Test
Training
Testing
Testing
Training
Shuffle dataset
Split dataset into training & test
Always leave one fold for test
Split training dataset into K folds
Take care of all transformation in the fold
Use (K-1) folds for training
Life cycle of K fold Cross Validation

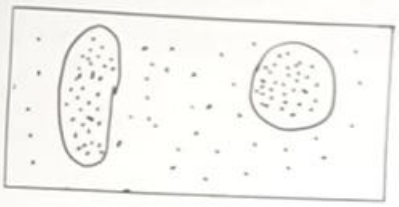


Introduction to Clustering


What is Clustering?
Supervised Learning
1,000
Pictures of Dog
1,000
Labelled data
is it a dog or not?
Unsupervised learning
C1
C2
C3
4:47 / 6:21

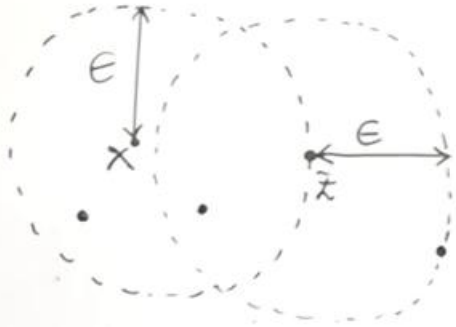
DBScan

PlanetOjas



What is ϵ?

What is Minpoint?



Untitled (1/2) - Sarble Ink

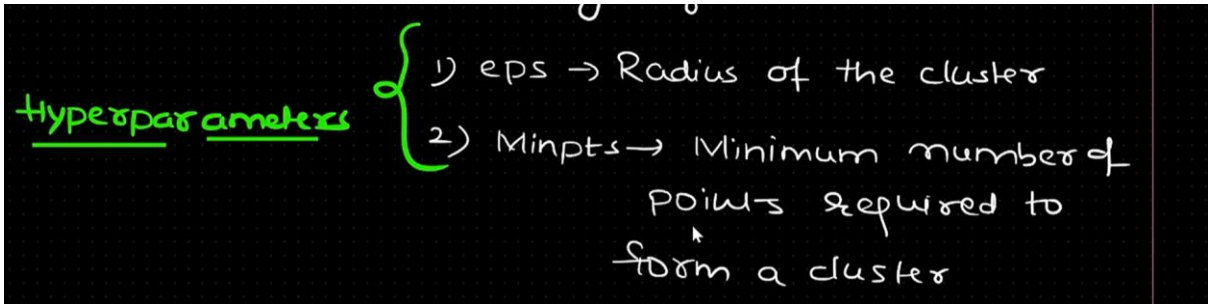
(1/2)

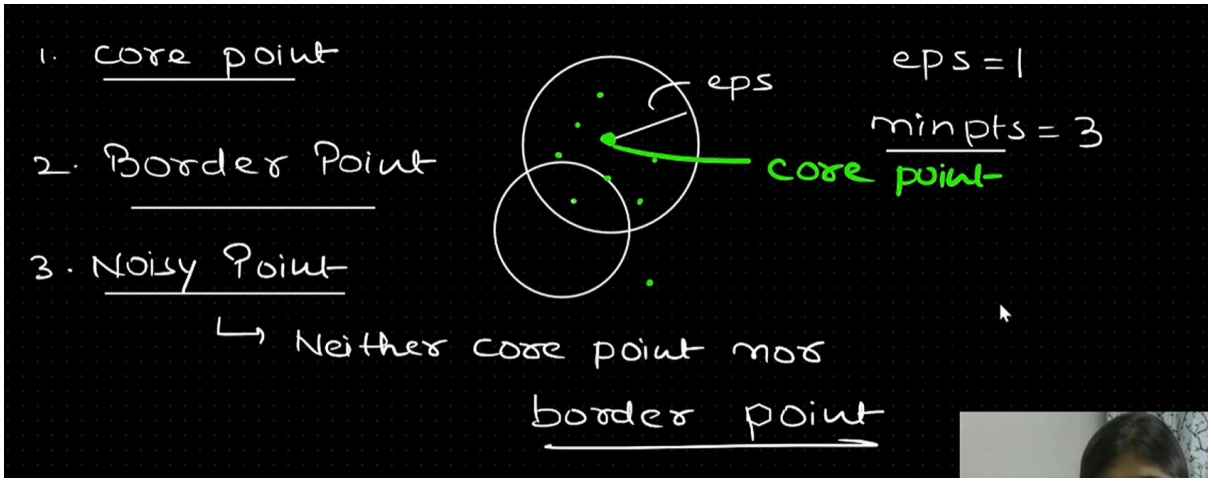
DBSCAN → Density Based Spatial Clustering of Applications with Noise

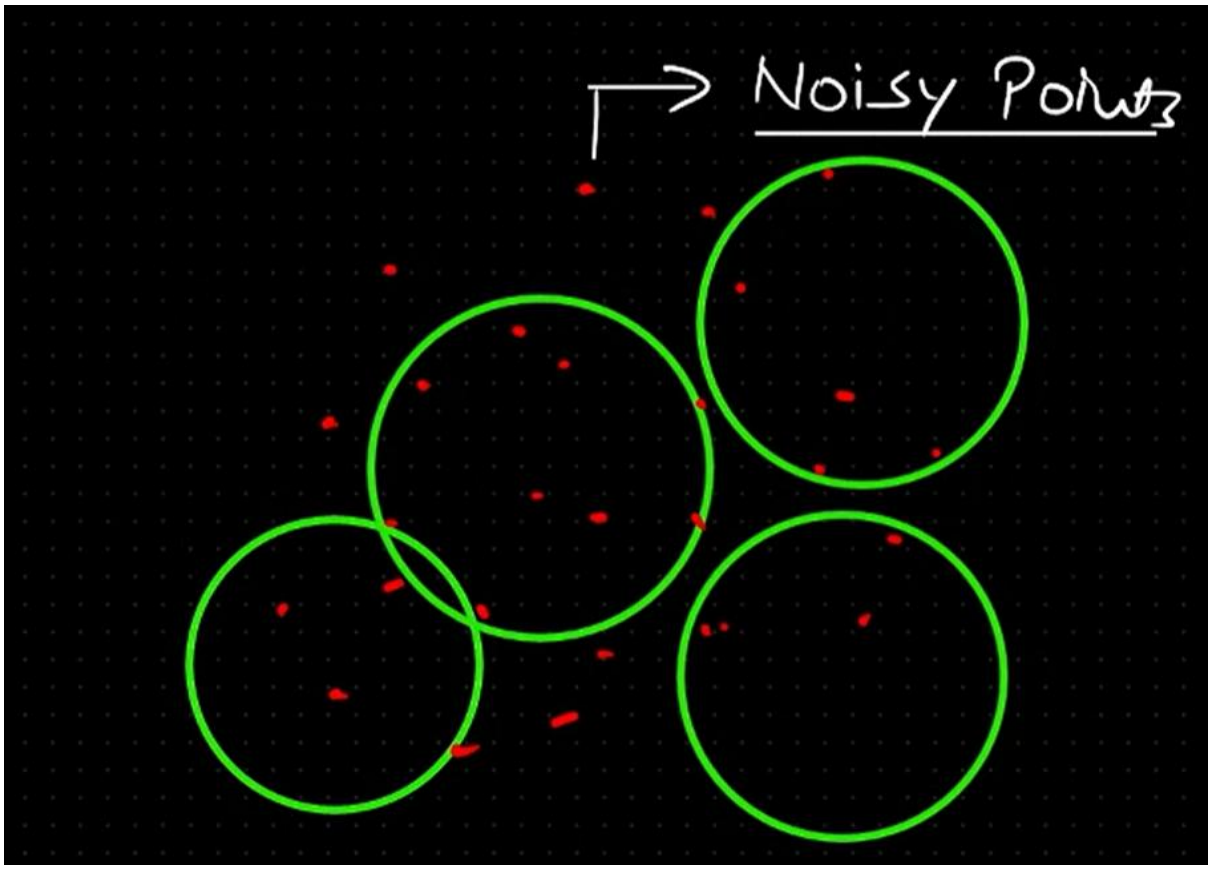
1) clustering algorithm
   ↳ Unsupervised Machine Learning

2) Noisy Points/Outliers

3) Most Powerful clustering algo







Principal Component Analysis Algorithm

Principal Component Analysis – Algorithm

Step 1. Data

- We consider a dataset having n features or variables denoted by $X_1; X_2; \ldots ; X_n$.
- Let there be N examples.
- Let the values of the i^{th} feature X_i be $X_{i1}; X_{i2}; \ldots ; X_{iN}$

Features	Example 1	Example 2	⋯	Example N
X_1	X_{11}	X_{12}	⋯	X_{1N}
X_2	X_{21}	X_{22}	⋯	X_{2N}
⋮				
X_i	X_{i1}	X_{i2}	⋯	X_{iN}
⋮				
X_n	X_{n1}	X_{n2}	⋯	X_{nN}

Principal Component Analysis – Algorithm

Step 2. Compute the means of the variables

Features	Example 1	Example 2	⋯	Example N
X_1	X_{11}	X_{12}	⋯	X_{1N}
X_2	X_{21}	X_{22}	⋯	X_{2N}
⋮				
X_i	X_{i1}	X_{i2}	⋯	X_{iN}
⋮				
X_n	X_{n1}	X_{n2}	⋯	X_{nN}

$$\bar{X}_i = \frac{1}{N}(X_{i1} + X_{i2} + \cdots + X_{iN})$$

Principal Component Analysis – Algorithm

Step 3. Calculate the covariance matrix

Features	Example 1	Example 2	⋯	Example N
X_1	X_{11}	X_{12}	⋯	X_{1N}
X_2	X_2	X_{22}	⋯	X_{2N}
⋮				
X_i	X_{i1}	X_{i2}	⋯	X_{iN}
⋮				
X_n	X_{n1}	X_{n2}	⋯	X_{nN}

$$\mathrm{Cov}(X_i, X_j) = \frac{1}{N-1}\sum_{k=1}^{N}(X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j)$$

$$S = \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \cdots & \mathrm{Cov}(X_2, X_n) \\ \vdots & & & \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \cdots & \mathrm{Cov}(X_n, X_n) \end{bmatrix}$$

Principal Component Analysis – Algorithm

Step 4. Calculate the eigenvalues and eigenvectors of the covariance matrix

i. **Set up the equation:** This is a polynomial equation of degree **n** in. It has **n** real roots and these roots are the eigenvalues of **S**

$$\det(S - \lambda I) = 0$$

ii. **If $\lambda = \lambda'$ is an eigenvalue**, then the corresponding eigenvector is a vector

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

such that $(S - \lambda' I)U = 0$

Principal Component Analysis – Algorithm

Step 4. Calculate the eigenvalues and eigenvectors of the covariance matrix

iii. **We now normalize the eigenvectors**. Given any vector X we normalize it by dividing X by its length. The length (or, the norm) of the vector

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

is defined as

$$\|X\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

We compute the n normalised eigenvectors $e_1, e_2, \ldots, e_n$ by

$$e_i = \frac{1}{\|U_i\|} U_i, \quad i = 1, 2, \ldots, n.$$

Principal Component Analysis – Algorithm

Step 5. Derive new data set

- Order the eigenvalues from highest to lowest.
- The unit eigenvector corresponding to the largest eigenvalue is the first principal component.

N=10 P=5

i) Let the eigenvalues in descending order be $\lambda_1 \geq \lambda_2 \geq .. \geq \lambda_n$ and let the corresponding unit eigenvectors be $e_1, e_2, \ldots, e_n$.

ii) Choose a positive integer p such that $1 \leq p \leq n$.

iii) Choose the eigenvectors corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ and form the following $p \times n$ matrix (we write the eigenvectors as row vectors):

$$F = \begin{bmatrix} e_1^T \\ e_2^T \\ \vdots \\ e_p^T \end{bmatrix}$$

Principal Component Analysis – Algorithm

Step 5. Derive new data set

iv) We form the following $n \times N$ matrix:

$$X = \begin{bmatrix} X_{11} - \bar{X}_1 & X_{12} - \bar{X}_1 & \cdots & X_{1N} - \bar{X}_1 \\ X_{21} - \bar{X}_2 & X_{22} - \bar{X}_2 & \cdots & X_{2N} - \bar{X}_2 \\ \vdots & & & \\ X_{n1} - \bar{X}_n & X_{n2} - \bar{X}_n & \cdots & X_{nN} - \bar{X}_n \end{bmatrix}$$

v) Next compute the matrix:

$$X_{\text{new}} = FX.$$

Note that this is a p × N matrix. This gives us a dataset of N samples having p features.

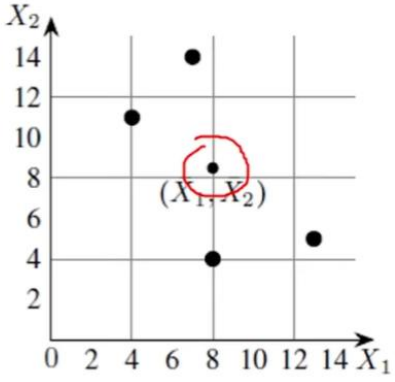
- Given the data in Table, reduce the dimension from 2 to 1 using the Principal Component Analysis (PCA) algorithm.

Feature	Example 1	Example 2	Example 3	Example 4
X_1	4	8	13	7
X_2	11	4	5	14

Step 1: Calculate Mean

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\bar{X}_1 = \frac{1}{4}(4 + 8 + 13 + 7) = 8,$$
$$\bar{X}_2 = \frac{1}{4}(11 + 4 + 5 + 14) = 8.5.$$



Step 2: Calculation of the covariance matrix.

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} \text{Cov}(X_1, X_1) & \text{Cov}(X_1, X_2) \\ \text{Cov}(X_2, X_1) & \text{Cov}(X_2, X_2) \end{bmatrix}$$

$$\text{Cov}(X_1, X_1) = \frac{1}{N-1} \sum_{k=1}^{N} (X_{1k} - \bar{X}_1)(X_{1k} - \bar{X}_1)$$

$$= \tfrac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right)$$

$$= 14$$

Step 2: Calculation of the covariance matrix.

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} \text{Cov}(X_1, X_1) & \text{Cov}(X_1, X_2) \\ \text{Cov}(X_2, X_1) & \text{Cov}(X_2, X_2) \end{bmatrix}$$

$$\text{Cov}(X_1, X_2) = \frac{1}{N-1} \sum_{k=1}^{N} (X_{1k} - \bar{X}_1)(X_{2k} - \bar{X}_2)$$

$$= \tfrac{1}{3}((4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)$$

$$= -11$$

Step 2: Calculation of the covariance matrix.

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} \text{Cov}(X_1, X_1) & \text{Cov}(X_1, X_2) \\ \text{Cov}(X_2, X_1) & \text{Cov}(X_2, X_2) \end{bmatrix}$$

$$\text{Cov}(X_2, X_1) = \text{Cov}(X_1, X_2)$$

$$= -11$$

$$\text{Cov}(X_2, X_2) = \frac{1}{N-1} \sum_{k=1}^{N} (X_{2k} - \bar{X}_2)(X_{2k} - \bar{X}_2)$$

$$= \tfrac{1}{3}\left((11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\right)$$

$$= 23$$

$$S = \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) \end{bmatrix}$$

$$= \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Step 3: Eigenvalues of the covariance matrix

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

The characteristic equation of the covariance matrix is,

$$0 = \det(S - \lambda I)$$

$$= \begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix}$$

$$= (14-\lambda)(23-\lambda) - (-11)\times(-11)$$

$$= \lambda^2 - 37\lambda + 201$$

$$\lambda = \frac{1}{2}(37 \pm \sqrt{565})$$

$$= 30.3849,\ 6.6151$$

$$= \lambda_1,\ \lambda_2 \quad \text{(say)}$$

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Step 4: Computation of the eigenvectors

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = (S - \lambda I)\, U$$

$$= \begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$= \begin{bmatrix} (14-\lambda)u_1 - 11u_2 \\ -11u_1 + (23-\lambda)u_2 \end{bmatrix}$$

$$(14-\lambda)u_1 - 11u_2 = 0$$

$$-11u_1 + (23-\lambda)u_2 = 0$$

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda} = t$$

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

$$\lambda_1 = 30.3849$$

$$\lambda_2 = 6.6151$$

Step 4: Computation of the eigenvectors

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

$$\lambda_1 = 30.3849$$

$$\lambda_2 = 6.6151$$

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda} = t$$

$$u_1 = 11t, \quad u_2 = (14-\lambda)t$$

$$U_1 = \begin{bmatrix} 11 \\ 14-\lambda \end{bmatrix}.$$

Step 4: Computation of the eigenvectors

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

$$\lambda_1 = 30.3849$$

$$\lambda_2 = 6.6151$$

$$U_1 = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix}.$$

- To find a unit eigenvector, we compute the length of U_1 which is given by,

$$\|U_1\| = \sqrt{11^2 + (14-\lambda_1)^2}$$

$$= \sqrt{11^2 + (14-30.3849)^2}$$

$$= 19.7348$$

$$e_1 = \begin{bmatrix} 11/\|U_1\| \\ (14-\lambda_1)/\|U_1\| \end{bmatrix}$$

$$= \begin{bmatrix} 11/19.7348 \\ (14-30.3849)/19.7348 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

Step 5: Computation of first principal components

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

$$\lambda_1 = 30.3849$$

$$\lambda_2 = 6.6151$$

$$e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

$$e_1^T \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix}$$

$$e_1^T \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} X_{11} - \bar{X}_1 \\ X_{21} - \bar{X}_2 \end{bmatrix}$$

$$= 0.5574(X_{11} - \bar{X}_1) - 0.8303(X_{21} - \bar{X}_2)$$

$$= 0.5574(4-8) - 0.8303(11-8,5)$$

$$= -4.30535$$

Step 5: Computation of first principal components

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

Feature	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14
First Principle Components	-4.3052	3.7361	5.6928	-5.1238

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

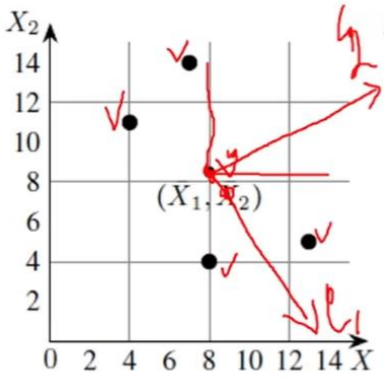
$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

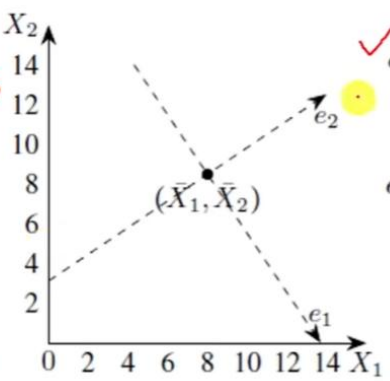
$$\lambda_1 = 30.3849$$

$$\lambda_2 = 6.6151$$

Step 6: Geometrical meaning of first principal components

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14





$$e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

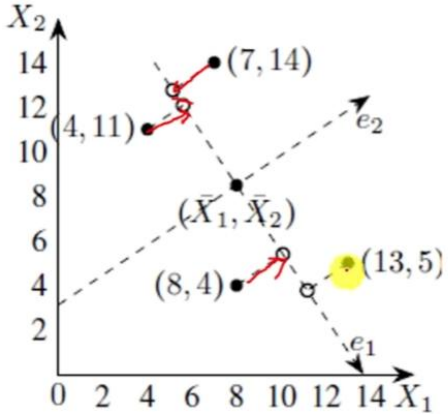
$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

$$\lambda_1 = 30.3849$$

$$\lambda_2 = 6.6151$$

Step 6: Geometrical meaning of first principal components

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14



$$e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

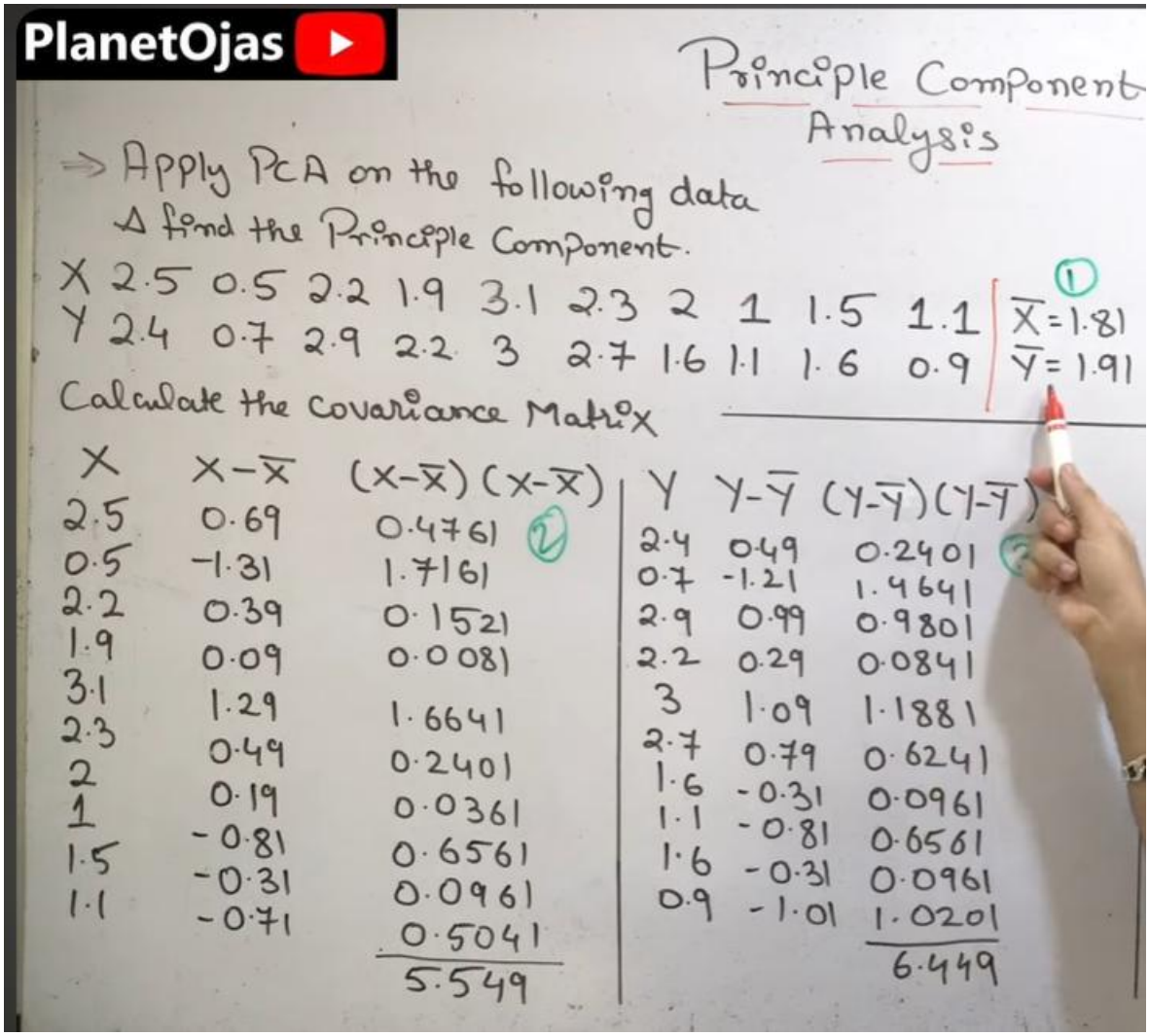
$$\overline{X_1} = 8$$

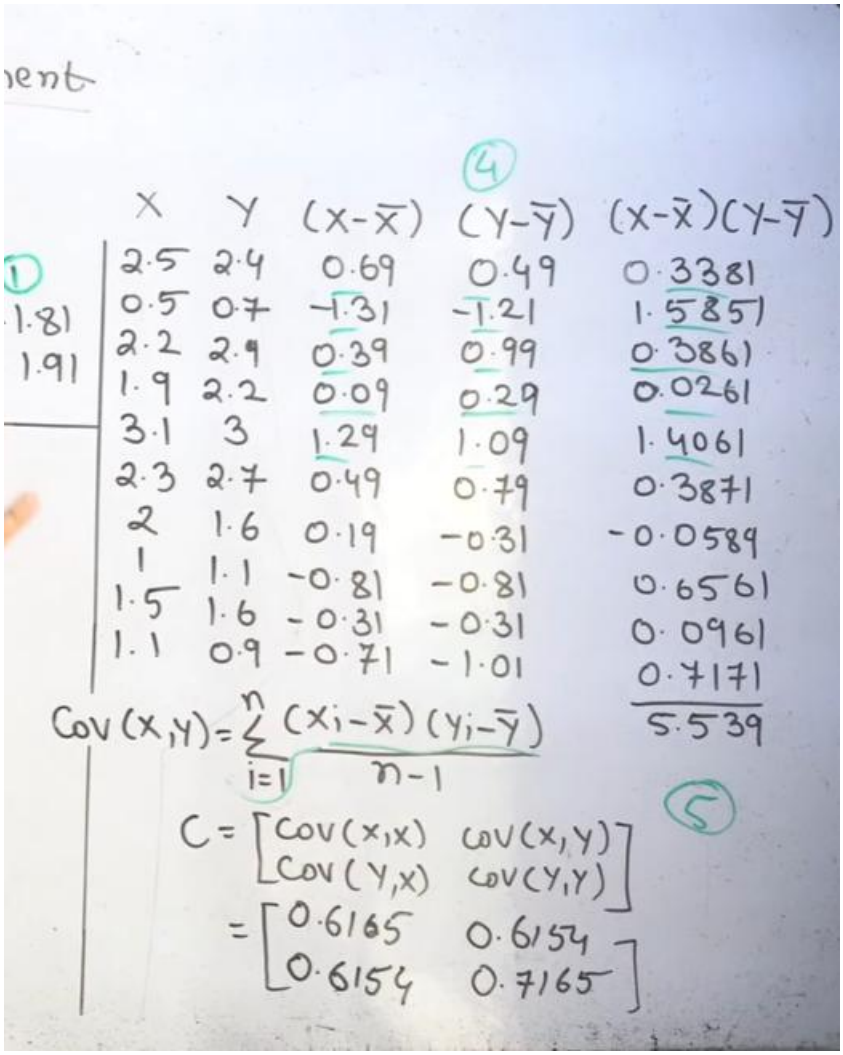
$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

$$\lambda_1 = 30.3849$$

$$\lambda_2 = 6.6151$$


PlanetOjas
Principle Component Analysis
⇒ Apply PCA on the following data & find the Principle Component.
X 2.5 0.5 2.2 1.9 3.1 2.3 2 1 1.5 1.1
Y 2.4 0.7 2.9 2.2 3 2.7 1.6 1.1 1.6 0.9
① $\bar{X} = 1.81$
$\bar{Y} = 1.91$
Calculate the Covariance Matrix
② ③
X | X−X̄ | (X−X̄)(X−X̄)
2.5 | 0.69 | 0.4761
0.5 | −1.31 | 1.7161
2.2 | 0.39 | 0.1521
1.9 | 0.09 | 0.0081
3.1 | 1.29 | 1.6641
2.3 | 0.49 | 0.2401
2 | 0.19 | 0.0361
1 | −0.81 | 0.6561
1.5 | −0.31 | 0.0961
1.1 | −0.71 | 0.5041
5.549
Y | Y−Ȳ | (Y−Ȳ)(Y−Ȳ)
2.4 | 0.49 | 0.2401
0.7 | −1.21 | 1.4641
2.9 | 0.99 | 0.9801
2.2 | 0.29 | 0.0841
3 | 1.09 | 1.1881
2.7 | 0.79 | 0.6241
1.6 | −0.31 | 0.0961
1.1 | −0.81 | 0.6561
1.6 | −0.31 | 0.0961
0.9 | −1.01 | 1.0201
6.449



ent

(1)

1.81
1.91

(4)

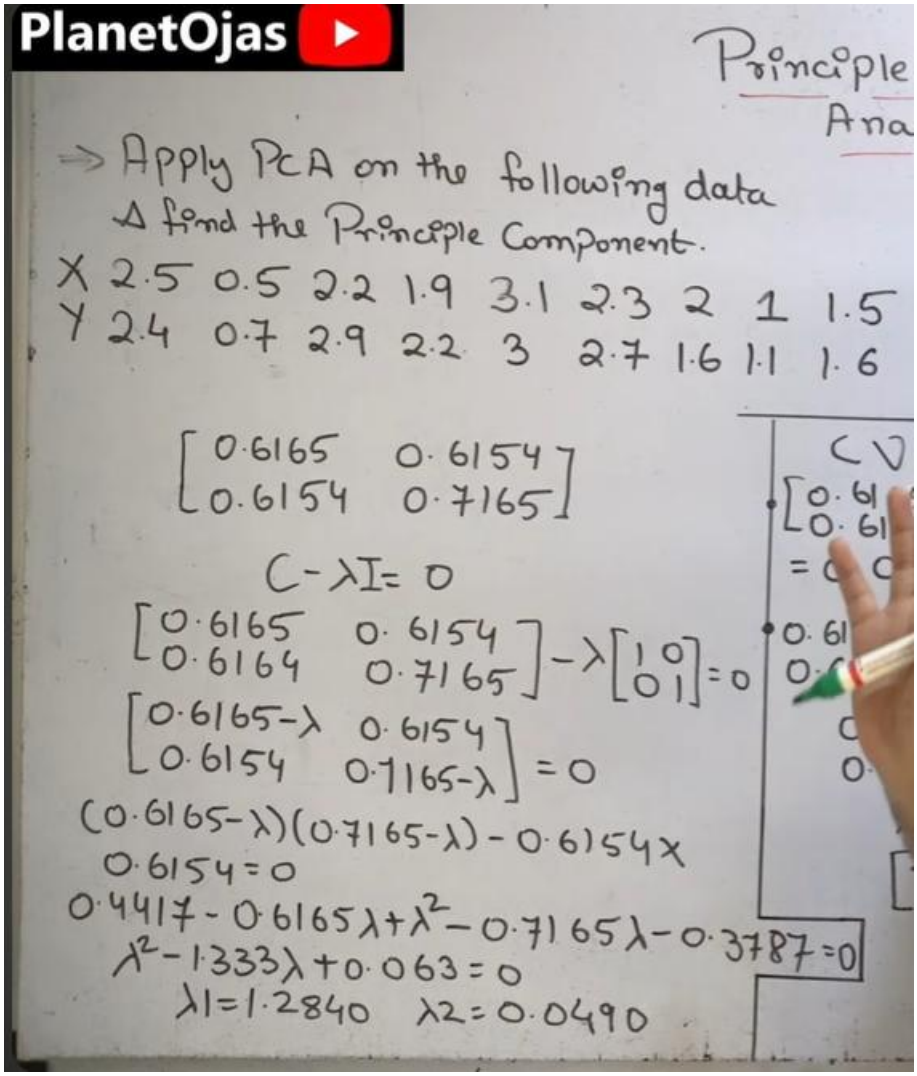
X	Y	$(X-\bar{X})$	$(Y-\bar{Y})$	$(X-\bar{X})(Y-\bar{Y})$
2.5	2.4	0.69	0.49	0.3381
0.5	0.7	-1.31	-1.21	1.5851
2.2	2.9	0.39	0.99	0.3861
1.9	2.2	0.09	0.29	0.0261
3.1	3	1.29	1.09	1.4061
2.3	2.7	0.49	0.79	0.3871
2	1.6	0.19	-0.31	-0.0589
1	1.1	-0.81	-0.81	0.6561
1.5	1.6	-0.31	-0.31	0.0961
1.1	0.9	-0.71	-1.01	0.7171
				5.539

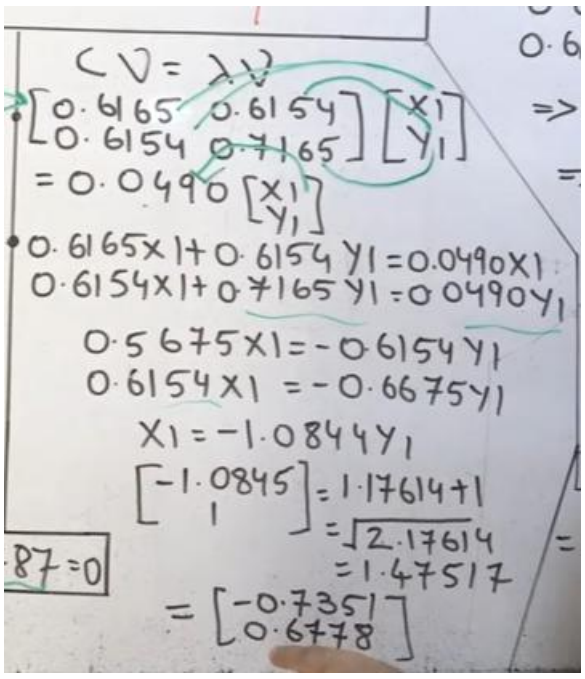
$$Cov(X,Y) = \frac{\sum_{i=1}^{n} (x_i-\bar{x})(y_i-\bar{y})}{n-1}$$

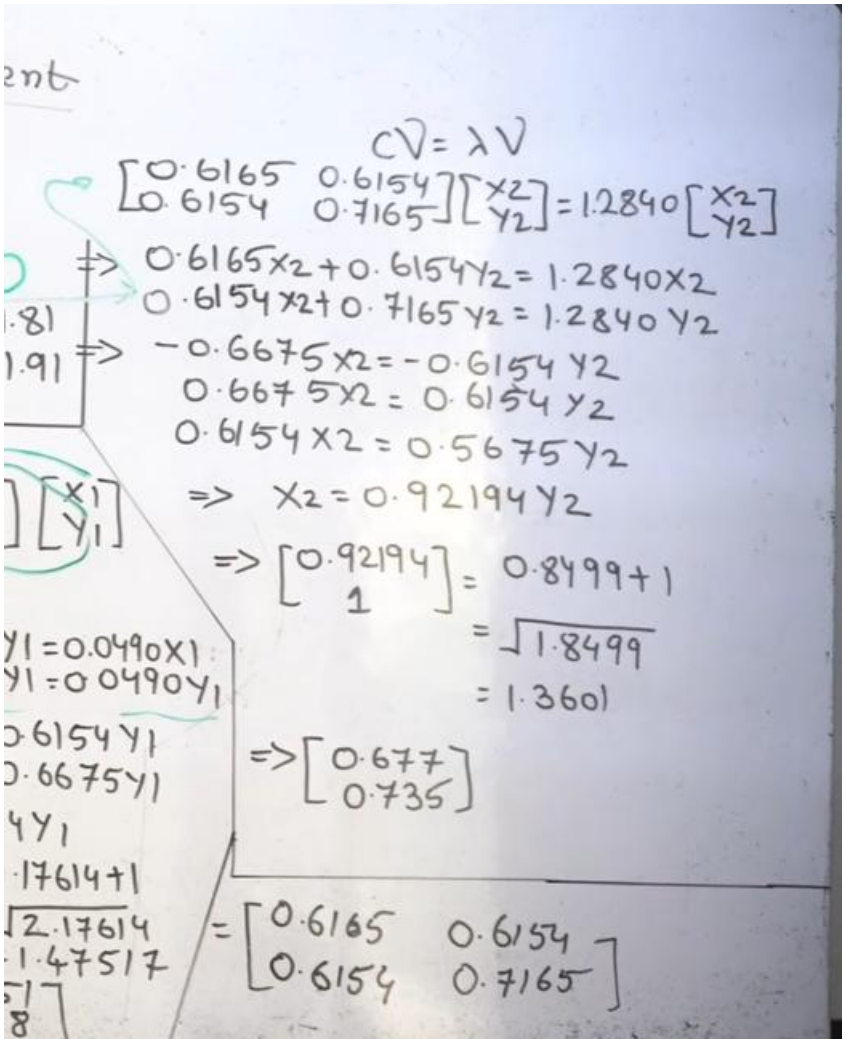
(5)

$$C = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}$$

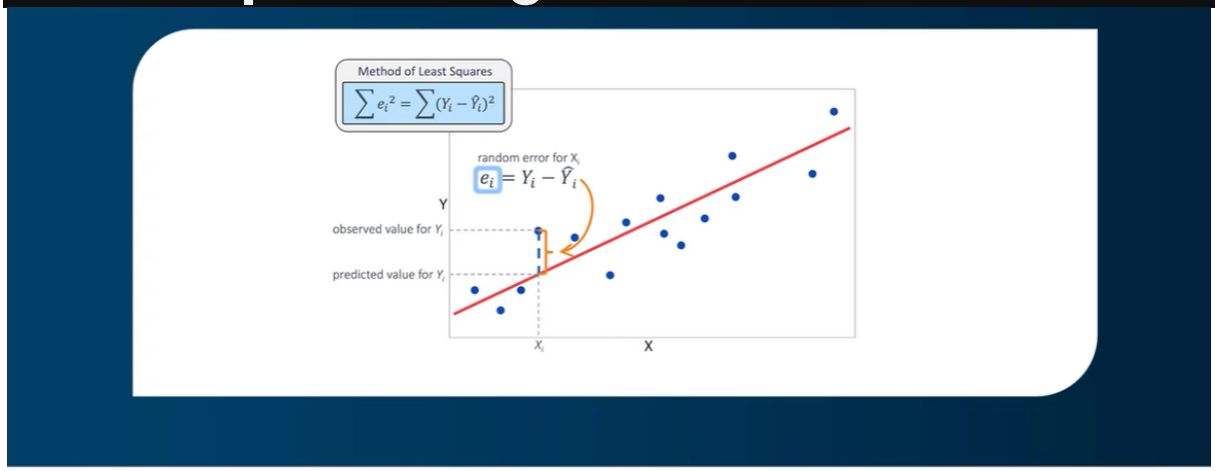
$$= \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix}$$


PlanetOjas
Principle
Ana
→ Apply PCA on the following data
& find the Principle Component.
X 2.5 0.5 2.2 1.9 3.1 2.3 2 1 1.5
Y 2.4 0.7 2.9 2.2 3 2.7 1.6 1.1 1.6
[0.6165 0.6154]
[0.6154 0.7165]
C - λI = 0
[0.6165 0.6154] - λ[1 0] = 0
[0.6164 0.7165] [0 1]
[0.6165-λ 0.6154]
[0.6154 0.7165-λ] = 0
(0.6165-λ)(0.7165-λ) - 0.6154x
0.6154 = 0
0.4417 - 0.6165λ + λ² - 0.7165λ - 0.3787 = 0
λ² - 1.333λ + 0.063 = 0
λ1 = 1.2840 λ2 = 0.0490
CV


CV = λV
[0.6165 0.6154] [X1]
[0.6154 0.7165] [Y1]
= 0.0490 [X1]
[Y1]
0.6165X1 + 0.6154Y1 = 0.0490X1
0.6154X1 + 0.7165Y1 = 0.0490Y1
0.5675X1 = -0.6154Y1
0.6154X1 = -0.6675Y1
X1 = -1.0844Y1
[-1.0845]
[1]
= 1.17614+1
= 2.17614
= 1.47517
= [-0.7351]
[0.6778]
=>
87 = 0



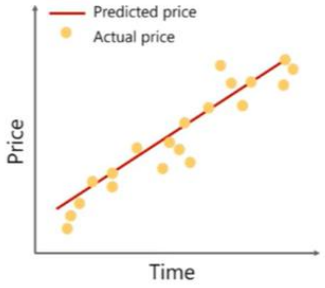
Least Squares Regression



What is the Least Squares Regression Method?

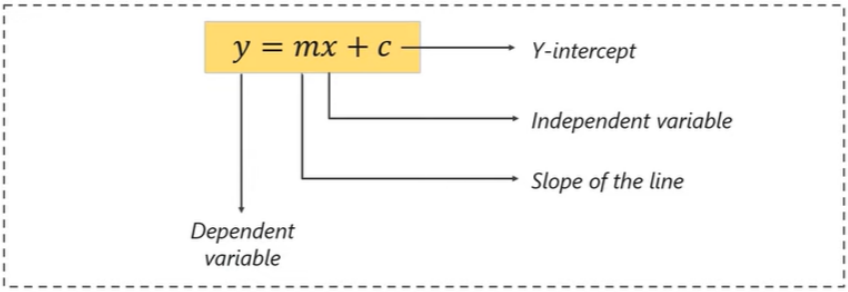
The least-squares regression method is a technique commonly used in Regression Analysis. It is a mathematical method used to find the best fit line that represents the relationship between an independent and dependent variable in such a way that the error is minimized.

The least-squares method is one of the most effective ways used to draw the line of best fit. It is based on the idea that the square of the errors obtained must be minimized to the most possible extent and hence the name least squares method.



What is the Line Of Best Fit?

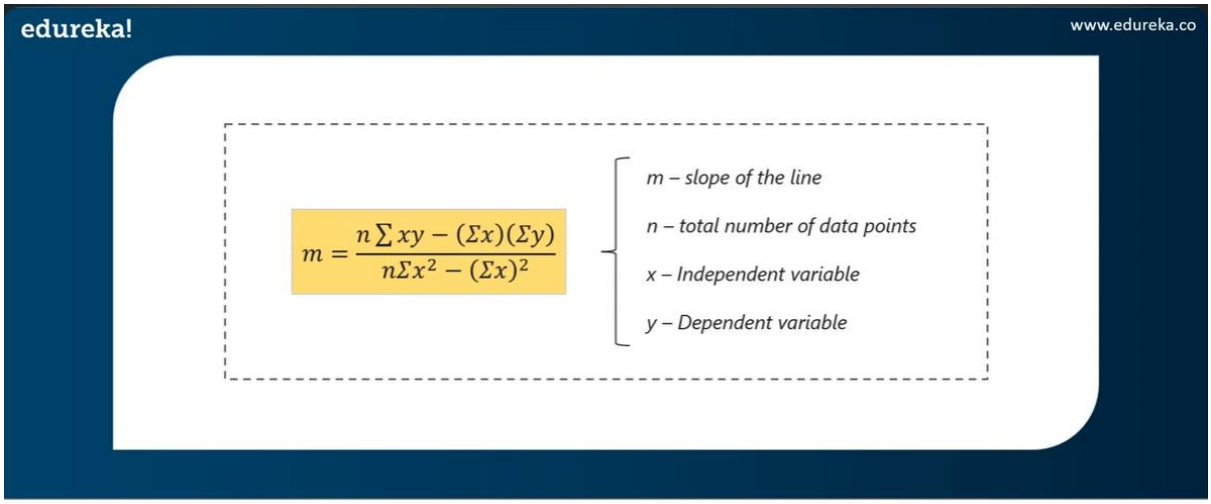
The Line of best fit line is drawn across a scatter plot of data points in order to represent a relationship between those data points.



A simple equation that represents a straight line along 2-Dimensional data, i.e. x-axis and y-axis.

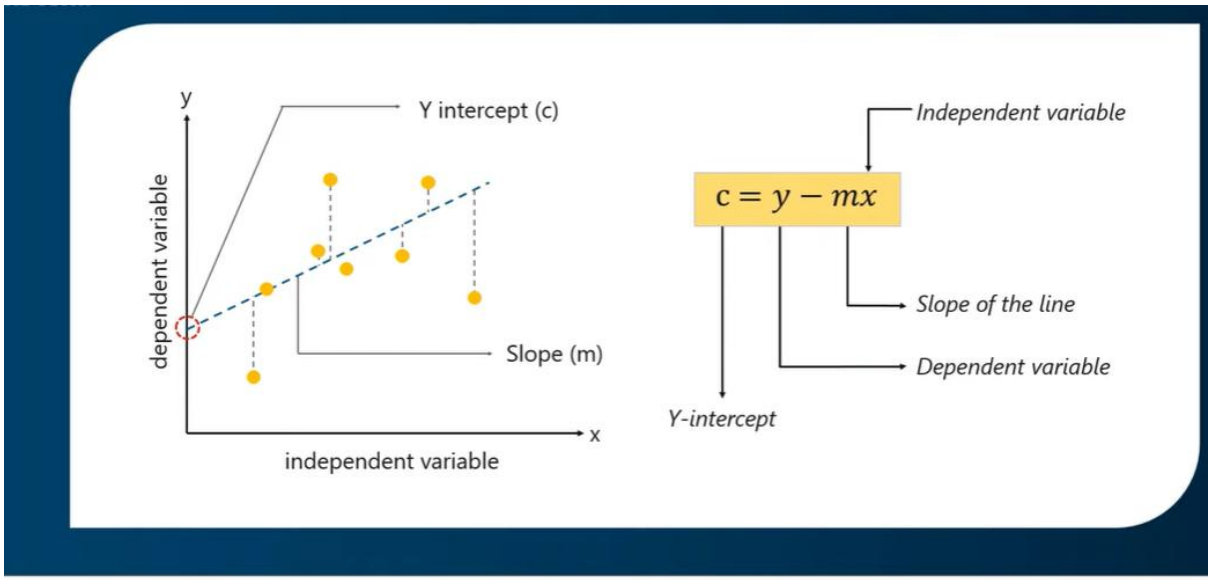
Steps to calculate the Line of Best Fit

To start constructing the line that best depicts the relationship between variables in the data, we first need to get our basics right.



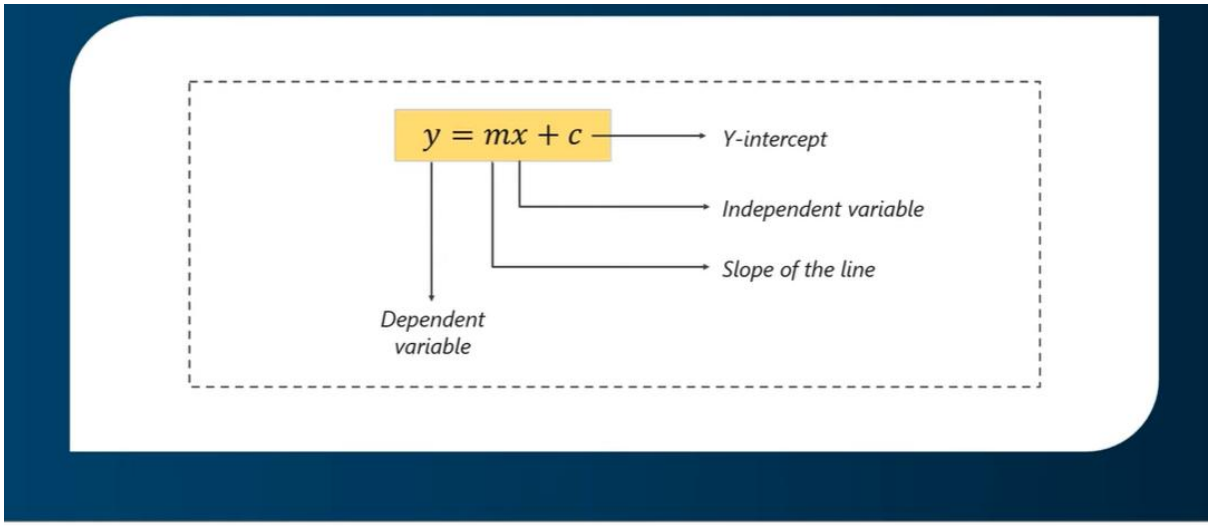
Step 1: Calculate the slope 'm' of the line

The slope of a line characterizes the direction of a line. To find the slope, you divide the difference of the y-coordinates of 2 points on a line by the difference of the x-coordinates of those same 2 points



Step 2: Compute the y-intercept

The y-intercept of a line is the value of y at the point where the line crosses the y axis.



Step 3: Substitute the values in the final equation

A simple equation that represents a straight line along 2-Dimensional data, i.e. x-axis and y-axis.

Tom who is the owner of a retail shop, found the price of different T-shirts vs the number of T-shirts sold at his shop over a period of one week.

Price of T-shirts in dollars (x)	# of T-shirts sold (y)
2	4
3	5
5	7
7	10
9	15

Step 1: Calculate the slope 'm'

$$m = \frac{n\sum xy - (\Sigma x)(\Sigma y)}{n\Sigma x^2 - (\Sigma x)^2} = 1.518 \text{ approx}$$

Step 2: Compute the y-intercept value

$$c = y - mx = 0.305$$

Step 3: Substitute the values in the final equation

$$y = 1.518x + 0.305$$

Tom who is the owner of a retail shop, found the price of different T-shirts vs the number of T-shirts sold at his shop over a period of one week.

Price of T-shirts in dollars (x)	# of T-shirts sold (y)	Y=mx+c	error
2	4	3.3	-0.67
3	5	4.9	-0.14
5	7	7.9	0.89
7	10	10.9	0.93
9	15	13.9	-1.03

Step 1: Calculate the slope 'm'

$$m = \frac{n\sum xy - (\Sigma x)(\Sigma y)}{n\Sigma x^2 - (\Sigma x)^2} = 1.518 \text{ approx}$$

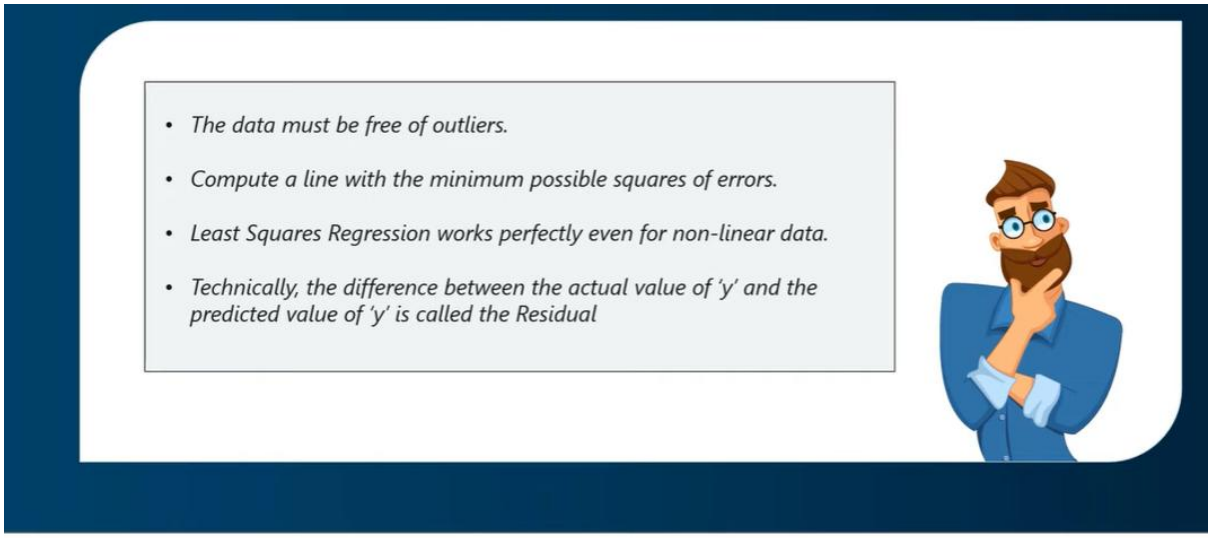
Step 2: Compute the y-intercept value

$$c = y - mx = 0.305$$

Step 3: Substitute the values in the final equation

Consider x = $8,

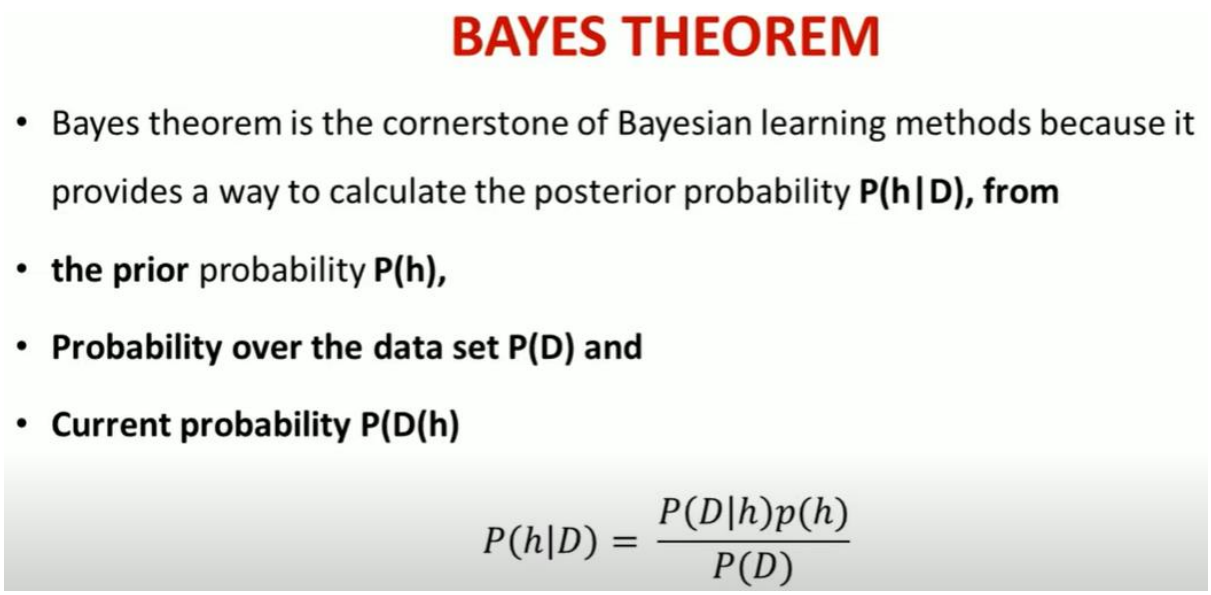
y = 1.518 x 8 + 0.305 = 12.45 = 13 T-shirts



Points To Remember

A few things to keep in mind before implementing the least squares regression method

Naive Bayes Theorem | Maximum A Posteriori Hypothesis | MAP Brute Force Algorithm



Maximum A Posteriori (MAP) Hypothesis

- The learner considers some set of candidate hypotheses H and is interested in finding the most probable hypothesis **h ϵ H** given the observed data **D (or at least one of the maximally probable if there** are several).
- Any such maximally probable hypothesis is called a **maximum a posteriori (MAP) hypothesis.**
- **We can determine the MAP hypotheses by using** Bayes theorem to calculate the posterior probability of each candidate hypothesis.

Maximum A Posteriori (MAP) Hypothesis

- More precisely, we will say that h_{MAP} **is a MAP hypothesis provided**

$$
\begin{aligned}
h_{MAP} &\equiv \underset{h \in H}{\operatorname{argmax}} P(h|D) \\
&= \underset{h \in H}{\operatorname{argmax}} \frac{P(D|h)P(h)}{P(D)} \\
&= \underset{h \in H}{\operatorname{argmax}} P(D|h)P(h)
\end{aligned}
$$

Brute-Force Bayes Concept Learning

BRUTE-FORCE MAP LEARNING algorithm

1. For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} P(h|D)$$

Brute-Force Bayes Concept Learning

BRUTE-FORCE MAP LEARNING algorithm

- This algorithm may require significant computation, because it applies Bayes theorem to each hypothesis in H to calculate $P(h|D)$.
  - While this is impractical for large hypothesis spaces,
  - The algorithm is still of interest because it provides a standard against which we may judge the performance of other concept learning algorithms.

Brute-Force Bayes Concept Learning

BRUTE-FORCE MAP LEARNING algorithm

- Brute Force MAP learning algorithm must specify values for P(h) and P(D|h).
- P(h) and P(D|h) can be chosen to be consistent with the assumptions:
  1. The training data D is noise free.
  2. The target concept c is contained in the hypothesis space H
  3. We have no a priori reason to believe that any hypothesis is more probable than any other.

Brute-Force Bayes Concept Learning

- Given these assumptions, what values should we specify for P(h)?
- Given no prior knowledge that one hypothesis is more likely than another, it is reasonable to assign the same prior probability to every hypothesis h in H.
- Furthermore, because we assume the target concept is contained in H we should require that these prior probabilities sum to 1.
- Together these constraints imply that we should choose

$$P(h) = \frac{1}{|H|} \quad \text{for all } h \text{ in } H$$

Brute-Force Bayes Concept Learning

- What choice shall we make for P(D|h)?
- P(D|h) is the probability of observing the target values D = <d1 . . .dm> for the fixed set of instances <X1 . . . Xm>.
- Since we assume noise-free training data, the probability of observing classification di given h is just 1 if di = h(xi) and 0 if di != h(xi).
- Therefore,

$$P(D|h) = \begin{cases} 1 \text{ if } d_i = h(x_i) \text{ for all } d_i \text{ in } D \\ 0 \text{ otherwise} \end{cases}$$

Brute-Force Bayes Concept Learning

- **Let us consider the** first step of this algorithm, which uses Bayes theorem to compute the posterior probability **P(h|D) of each hypothesis h given the observed training data D.**

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- First consider the case where h is inconsistent with the training data D.
- We know that P(D|h) to be 0 when h is inconsistent with D, we have,

$$P(h|D) = \frac{0 \cdot P(h)}{P(D)} = 0 \text{ if } h \text{ is inconsistent with } D$$

Brute-Force Bayes Concept Learning

Now consider the case where **h is consistent** with D.

We know that P(Dlh) to be 1 when h is consistent with D, we have

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$$P(h|D) = \frac{1 \cdot \frac{1}{|H|}}{P(D)}$$

$$= \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}}$$

$$= \frac{1}{|VS_{H,D}|} \text{ if } h \text{ is consistent with } D$$

Brute-Force Bayes Concept Learning

- To summarize, Bayes theorem implies that the posterior probability P(h|D) under our assumed P(h) and P(D|h) is,

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}$$

Solved example

Does patient have cancer or not?

- Suppose we now observe a new patient for whom the lab test returns a **positive** result.
- Should we diagnose the patient as having cancer or not?

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$$P(cancer|+) = P(+|cancer) * P(cancer) = 0.98 * 0.008 = 0.0078$$

$$P(\neg cancer|+) = P(+|\neg cancer) * P(\neg cancer) = 0.03 * 0.992 = 0.0298$$

Does patient have cancer or not?

- Suppose we now observe a new patient for whom the lab test returns a **negative** result.
- Should we diagnose the patient as having cancer or not?

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$$P(cancer|-) = P(-|cancer) * P(cancer) = 0.02 * 0.008 = 0.00016$$

$$P(\neg cancer|-) = P(-|\neg cancer) * P(\neg cancer) = 0.97 * 0.992 = 0.96224$$

$$h_{MAP} = \neg cancer$$

Generalized Naive Bayes rule to find Probability that Defective Bulb

Bayes Theorem - Defective Bulb by Factory A?

- Three factories A, B, C of an electric bulb manufacturing company produce respectively 35%, 35% and 30% of the total output.
- Approximately 1.5%, 1% and 2% of the bulbs produced by these factories are known to be defective.
- If a randomly selected bulb manufactured by the company was found to be defective, what is the probability that the bulb was manufactures in factory A?

- Let A, B, C denote the events that a randomly selected bulb was manufactured in factory A, B, C respectively.
- Let D denote the event that a bulb is defective.
- We have the following data:

$$P(A) = 0.35,\ P(B) = 0.35,\ P(C) = 0.30$$

$$P(D|A) = 0.015,\ P(D|B) = 0.010,\ P(D|C) = 0.020$$

Bayes Theorem - Defective Bulb by Factory A?

- Generalization of Bayes Theorem

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}.$$

- Let the sample space be divided into disjoint events $B_1, B_2, \ldots, B_n$ and A be any event.
- Then we have

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}$$

- We are required to find P(A|D).
- By the generalization of the Bayes' theorem, we have:

$$\begin{aligned} P(A|D) &= \frac{P(D|A)P(A)}{P(D|A)P(A) + P(D|B)P(B) + P(D|C)P(C)} \\ &= \frac{0.015 \times 0.35}{0.015 \times 0.35 + 0.010 \times 0.35 + 0.020 \times 0.30} \\ &= 0.356. \end{aligned}$$

NAIVE BAYES CLASSIFIER – Example -1

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$\langle Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong \rangle$

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

NAIVE BAYES CLASSIFIER
Example - 1

$$P(PlayTennis = yes) = 9/14 = .64$$

$$P(PlayTennis = no) = 5/14 = .36$$

Outlook	Y	N
sunny	2/9	3/5
overcast	4/9	0
rain	3/9	2/5
Tempreature		
hot	2/9	2/5
mild	4/9	2/5
cool	3/9	1/5

Humidity	Y	N
high	3/9	4/5
normal	6/9	1/5
Windy		
Strong	3/9	3/5
Weak	6/9	2/5

$$\langle Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong \rangle$$

$$v_{NB} = \underset{v_j \in \{yes,no\}}{\text{argmax}} \; P(v_j) \prod_i P(a_i|v_j)$$

$$= \underset{v_j \in \{yes,no\}}{\text{argmax}} \; P(v_j) \quad P(Outlook = sunny|v_j) P(Temperature = cool|v_j)$$

$$P(Humidity = high|v_j) P(Wind = strong|v_j)$$

$$v_{NB}(yes) = P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$$

$$v_{NB}(no) = P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) = .0206$$

$$v_{NB}(yes) = \frac{v_{NB}(yes)}{v_{NB}(yes)+v_{NB}(no)} = 0.205 \qquad v_{NB}(no) = \frac{v_{NB}(no)}{v_{NB}(yes)+v_{NB}(no)} = 0.795$$

NAIVE BAYES CLASSIFIER Example – 2

- Estimate conditional probabilities of each attributes {color, legs, height, smelly} for the species classes: {M, H} using the data given in the table.
- Using these probabilities estimate the probability values for the new instance – (Color=Green, legs=2, Height=Tall, and Smelly=No).

No	Color	Legs	Height	Smelly	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	No	M
3	Green	3	Short	Yes	M
4	White	3	Short	Yes	M
5	Green	2	Short	No	H
6	White	2	Tall	No	H
7	White	2	Tall	No	H
8	White	2	Short	Yes	H

No	Color	Legs	Height	Smelly	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	No	M
3	Green	3	Short	Yes	M
4	White	3	Short	Yes	M
5	Green	2	Short	No	H
6	White	2	Tall	No	H
7	White	2	Tall	No	H
8	White	2	Short	Yes	H

New Instance

(Color=Green, legs=2, Height=Tall, and Smelly=No)

NAIVE BAYES CLASSIFIER EXAMPLE - 2

$$P(M) = \frac{4}{8} = 0.5 \quad P(H) = \frac{4}{8} = 0.5$$

Color	M	H
White	2/4	3/4
Green	2/4	1/4

Legs	M	H
2	1/4	4/4
3	3/4	0/4

Height	M	H
Tall	3/4	2/4
Short	1/4	2/4

Smelly	M	H
Yes	3/4	1/4
No	1/4	3/4

NAIVE BAYES CLASSIFIER - EXAMPLE - 2

$P(M) = \frac{4}{8} = 0.5 \quad P(H) = \frac{4}{8} = 0.5$

Color	M	H
White	2/4	3/4
Green	2/4	1/4

Legs	M	H
2	1/4	4/4
3	3/4	0/4

Height	M	H
Tall	3/4	2/4
Short	1/4	2/4

Smelly	M	H
Yes	3/4	1/4
No	1/4	3/4

$$p(M|New\ Instance) = p(M) * p(Color = Green|M) * p(Legs = 2|M) * p(Height = tall|M) * p(Smelly = no\ |M)$$

$$p(M|New\ Instance) = 0.5 * \frac{2}{4} * \frac{1}{4} * \frac{3}{4} * \frac{1}{4} = 0.0117$$

$$p(H|New\ Instance) = p(H) * p(Color = Green|H) * p(Legs = 2|H) * p(Height = tall|H) * p(Smelly = no\ |H)$$

$$p(H|New\ Instance) = 0.5 * \frac{1}{4} * \frac{4}{4} * \frac{2}{4} * \frac{3}{4} = 0.047$$

$p(H|New\ Instance) > p(M|New\ Instance)$

Hence the new instance belongs to Speices H

NAIVE BAYES CLASSIFIER – Example – 3

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

NAIVE BAYES CLASSIFIER
EXAMPLE - 3

$$p(Yes) = \frac{5}{10} = 0.5$$

$$p(No) = \frac{5}{10} = 0.5$$

Color	Yes	No
Red	3/5	2/5
Yellow	2/5	3/5

Type	Yes	No
Sports	4/5	2/5
SUV	1/5	3/5

Origin	Yes	No
Domestic	2/5	3/5
Imported	3/5	2/5

$$New\ Instance = (Red, SUV, Domestic)$$

$$P(Yes|New\ Instance) = p(Yes) * P(Color = Red|Yes) * P(Type = SUV|Yes) * P(Origin = Domestic|Yes)$$

$$P(Yes|New\ Instance) = \frac{5}{10} * \frac{3}{5} * \frac{1}{5} * \frac{2}{5} = \frac{3}{125} = 0.024$$

$$P(No|New\ Instance) = p(No) * P(Color = Red|No) * P(Type = SUV|No) * P(Origin = Domestic|No)$$

$$P(No|New\ Instance) = \frac{5}{10} * \frac{2}{5} * \frac{3}{5} * \frac{3}{5} = \frac{9}{125} = 0.072$$

$$P(No|New\ Instance) > P(Yes|New\ Instance)$$

NAIVE BAYES CLASSIFIER EXAMPLE – 4

Consider a football game between two rival teams, say team A and team B. Suppose team A wins 65% of the time and team B wins the remaining matches. Among the games won by team A, only 35% of them comes from playing at team B's football field. On the other hand, 75% of the victories for team B are obtained while playing at home.

1. If team B is to host the next match between the two teams, what is the probability that it will emerge as the winner?
2. If team B is to host the next match between the two teams, who will emerge as the winner?

Solution:

Y – Winning football match
X – Hosting football match

Probability that team A wins is $P(Y_A) = 0.65$.

Probability that team B wins is $P(Y_B) = 1 - P(Y_A) = 0.35$

Probability that team B hosted the match it had won is $P(X_B|Y_B) = 0.75$.

Probability that team B hosted the match won by team A is $P(X_B|Y_A) = 0.35$.

1. If team B is to host the next match between the two teams, what is the probability that it will emerge as the winner?

Solution:

$$P(Y_B|X_B) = \frac{P(X_B|Y_B)\times P(Y_B)}{P(X_B)}$$

$$= \frac{P(X_B|Y_B)\times P(Y_B)}{P(X_B|Y_A)P(Y_A) + P(X_B|Y_B)P(Y_B)}$$

$$= \frac{0.75 \times 0.35}{(0.35 \times 0.65 + 0.75 \times 0.35)}$$

$$= 0.5357$$

Probability that team A wins is $P(Y_A) = 0.65$.
Probability that team B wins is $P(Y_B) = 1 - P(Y_A) = 0.35$
Probability that team B hosted the match it had won is $P(X_B|Y_B) = 0.75$.
Probability that team B hosted the match won by team A is $P(X_B|Y_A) = 0.35$.

2. If team B is to host the next match between the two teams, who will emerge as the winner?

Solution:

$$P(Y_A|X_B) = \frac{P(X_B|Y_A)\times P(Y_A)}{P(X_B)}$$

$$= \frac{P(X_B|Y_A)\times P(Y_A)}{P(X_B|Y_A)P(Y_A) + P(X_B|Y_B)P(Y_B)}$$

$$= \frac{0.35 \times 0.65}{(0.35 \times 0.65 + 0.75 \times 0.35)}$$

$$= 0.4642$$

Probability that team A wins is $P(Y_A) = 0.65$.
Probability that team B wins is $P(Y_B) = 1 - P(Y_A) = 0.35$
Probability that team B hosted the match it had won is $P(X_B|Y_B) = 0.75$.
Probability that team B hosted the match won by team A is $P(X_B|Y_A) = 0.35$.

$$P(Y_B|X_B) = 0.5357$$

Naïve Bayes Classifier – Solved Example 5

Record	A	B	C	Class
1	0	0	0	+
2	0	0	1	−
3	0	1	1	−
4	0	1	1	−
5	0	0	1	+
6	1	0	1	+
7	1	0	1	−
8	1	0	1	−
9	1	1	1	+
10	1	0	1	+

How to Compute

- **Conditional Probabilities**
- **Predict the Label for the new instance**

Consider the dataset given in the table and

1. Estimate the conditional probabilities of P(A|+), P(B|+), P(C|+), P(A|-), P(B|-) and P(C|-)
2. Use the conditional probability estimates and predict the class label for the test sample P(A=0, B=1, C=0)

Record	A	B	C	Class
1	0	0	0	+
2	0	0	1	−
3	0	1	1	−
4	0	1	1	−
5	0	0	1	+
6	1	0	1	+
7	1	0	1	−
8	1	0	1	−
9	1	1	1	+
10	1	0	1	+

A	0	1
+	2/5	3/5
-	3/5	2/5

B	0	1
+	4/5	1/5
-	3/5	2/5

$P(A=0|+)=\frac{2}{5}$

C	0	1
+	1/5	4/5
-	0/5	5/5

Record	A	B	C	Class
1	0	0	0	+
2	0	0	1	−
3	0	1	1	−
4	0	1	1	−
5	0	0	1	+
6	1	0	1	+
7	1	0	1	−
8	1	0	1	−
9	1	1	1	+
10	1	0	1	+

A	0	1
+	2/5	3/5
-	3/5	2/5

B	0	1
+	4/5	1/5
-	3/5	2/5

C	0	1
+	1/5	4/5
-	0/5	5/5

$New \Rightarrow P(A=0, B=1, C=0)$

$$P(+|New) = \frac{P(+) * P(A=0|+) * P(B=1|+) * P(C=0|+)}{P(A=0, B=1, C=0)}$$

$$= 0.5 * \frac{2}{5} * \frac{1}{5} * \frac{1}{5} = 0.008$$

$$P(-|New) = \frac{P(-) * P(A=0|-) * P(B=1|-) * P(C=0|-)}{P(A=0, B=1, C=0)}$$

$$= 0.5 * \frac{3}{5} * \frac{2}{5} * \frac{0}{5} = 0.0$$

$P(+|New) > P(-|New)$

Bayesian Belief Network

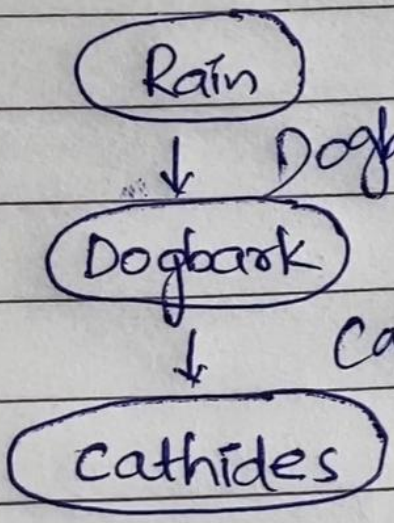
* BAYESIAN BELIEF NETWORKS:

- 2 important concepts.

(1) Directed Acyclic Graph (DAG)

(2) conditional probability Table (CPT)

* Directed acylic Graph:



* conditional Probability table:

	R	~R
B	9/48.	18/48.
~B	3/48	18/48

(B=T & R=T) = 9/48 = 0.19

(B=T & R=F) = 18/48 = 0.375

(B=F & R=T) = 3/48 = 0.06

(B=F & R=F) = 18/48 = 0.375.

* BAYESIAN BELIEF NETWORKS:

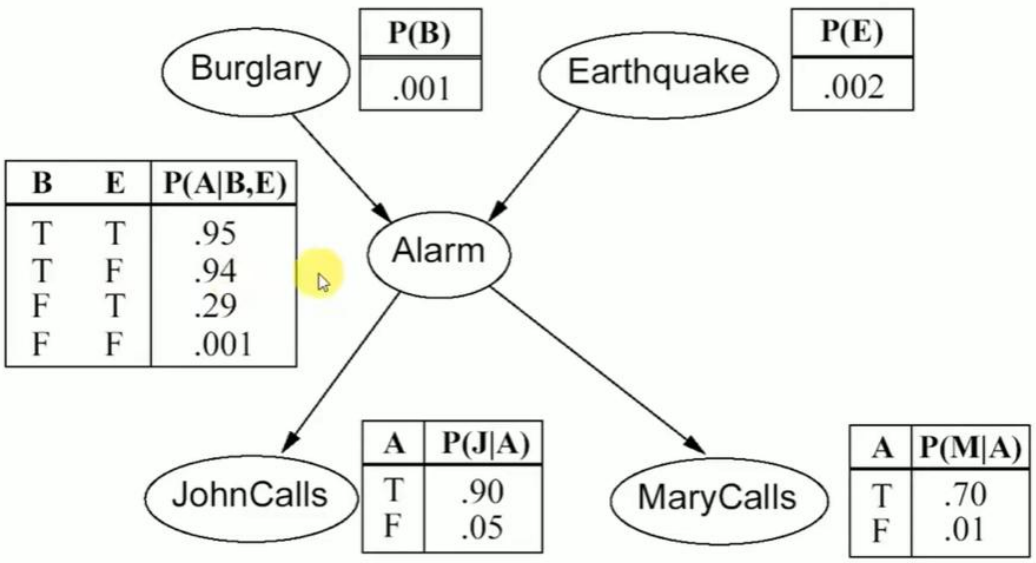
Bayesian belief n/w is a probabilistic graphical model (PGM) that represents conditional dependencies between random variables through DAG.

→ also suitable for representing probabilistic relation between multiple events (more than 2 events)

BAYESIAN BELIEF NETWORKS – EXAMPLE – 1

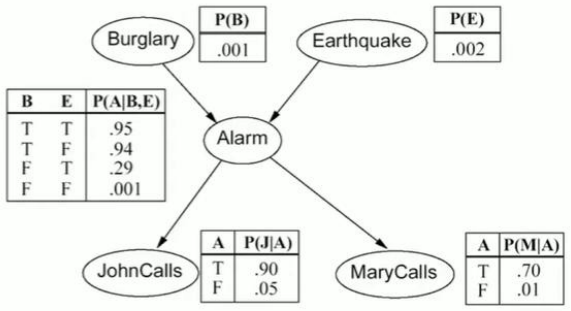
- You have a new burglar alarm installed at home.
- It is fairly reliable at detecting burglary, but also sometimes responds to minor earthquakes.
- You have two neighbors, John and Merry , who promised to call you at work when they hear the alarm.
- John always calls when he hears the alarm, but sometimes confuses telephone ringing with the alarm and calls too.
- Merry likes loud music and sometimes misses the alarm.
- Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.

BAYESIAN BELIEF NETWORKS – EXAMPLE – 1



BAYESIAN BELIEF NETWORKS – EXAMPLE – 1

1. What is the probability that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both John and Merry call?



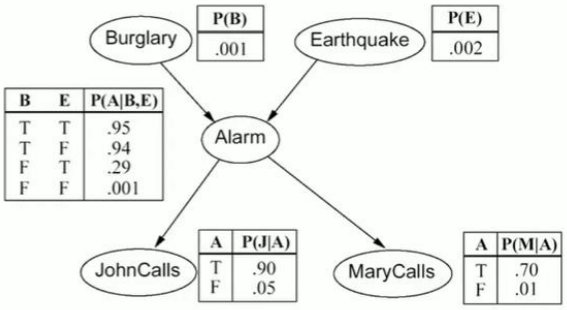
Solution:

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j \mid a)\, P(m \mid a)\, P(a \mid \neg b, \neg e)\, P(\neg b)\, P(\neg e)$$

$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$$

$$= 0.00062$$

2. What is the probability that John call?



Solution:

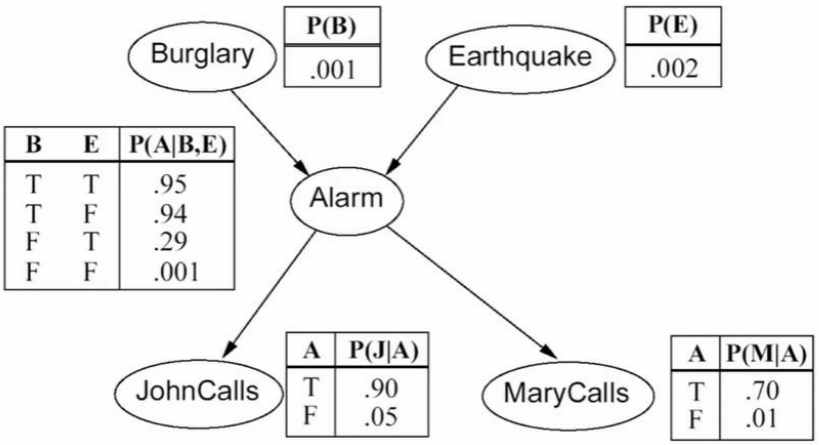
$$P(j) = P(j \mid a)\, P(a) + P(j \mid \neg a)\, P(\neg a)$$

$$= P(j|a)\{P(a|b,e)*P(b,e)+P(a|\neg b,e)*P(\neg b,e)+P(a|b,\neg e)*P(b,\neg e)+P(a|\neg b,\neg e)*P(\neg b,\neg e)\}$$

$$+ P(j|\neg a)\{P(\neg a|b,e)*P(b,e)+P(\neg a|\neg b,e)*P(\neg b,e)+P(\neg a|b,\neg e)*P(b,\neg e)+P(\neg a|\neg b,\neg e)*P(\neg b,\neg e)\}$$

$$= 0.90 * 0.00252 + 0.05 * 0.9974 = 0.0521$$

BAYESIAN BELIEF NETWORKS – EXAMPLE – 2



3. What is the probability that there is a burglary given that John and Merry calls?

BAYESIAN BELIEF NETWORKS – EXAMPLE – 2

- Suppose, we are given for the evidence variables $E_1,\ldots,E_m$, their values $e_1,\ldots,e_m$, and we want to predict whether the query variable X has the value x or not.
- For this we compute and compare the following:

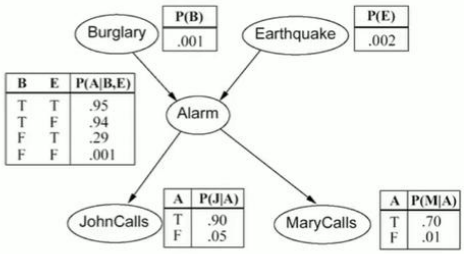
$$P(x \mid e_1,\ldots,e_m) = \frac{P(x,e_1,\ldots,e_m)}{P(e_1,\ldots,e_m)} = \alpha P(x,e_1,\ldots,e_m)$$

$$P(\neg x \mid e_1,\ldots,e_m) = \frac{P(\neg x,e_1,\ldots,e_m)}{P(e_1,\ldots,e_m)} = \alpha P(\neg x,e_1,\ldots,e_m)$$

$$\propto = \frac{1}{(P(x,e_1,\ldots em) + P(\neg x,e_1,\ldots em))}$$

BAYESIAN BELIEF NETWORKS – EXAMPLE – 2

3. What is the probability that there is a burglary given that John and Merry calls?



$$P(b \mid j,m) = \alpha\, P(b) \sum_a P(j|a)\, P(m|a) \sum_e P(a|b,e) P(e)$$

$$= \alpha\, P(b) \sum_a P(j|a)\, P(m|a)\, \{P(a|b,e)P(e) + P(a|b,\neg e)P(\neg e)\}$$

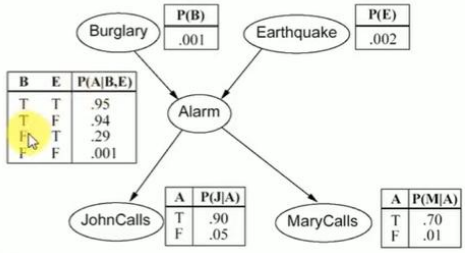
$$= \alpha\, P(b)\, [\, P(j|a)P(m|a)\, \{P(a|b,e)P(e) + P(a|b,\neg e)P(\neg e)\}$$

$$+ P(j|\neg a)P(m|\neg a)\, \{P(\neg a|b,e)P(e) + P(\neg a|b,\neg e)P(\neg e)\}\,]$$

$$= \alpha * .001*(.9*.7*(.95*.002 + .94*.998) + .05*.01*(.05*.002 + .71*.998))$$

$$= \alpha * .00059$$

3. What is the probability that there is a burglary given that John and Merry calls?



$$P(\neg b \mid j,m) = \alpha\, P(\neg b) \sum_a P(j|a)P(m|a) \sum_e P(a|\neg b,e)P(e)$$

$$= \alpha\, P(\neg b) \sum_a P(j|a)P(m|a) \{ P(a|\neg b,e)P(e) + P(a|\neg b,\neg e)P(\neg e)\}$$

$$= \alpha\, P(\neg b) [\, P(j|a)P(m|a) \{ P(a|\neg b,e)P(e) + P(a|\neg b,\neg e)P(\neg e) \}$$

$$+ P(j|\neg a)P(m|\neg a) \{ P(\neg a|\neg b,e)P(e) + P(\neg a|\neg b,\neg e)P(\neg e) \}]$$

$$= \alpha * .999*(.9*.7*(.29*.002 + .001*.998) + .05*.01*(.71*.002 + .999*.998))$$

$$= \alpha * .0015$$

BAYESIAN BELIEF NETWORKS – EXAMPLE – 2

3. What is the probability that there is a burglary given that John and Merry calls?

$$\alpha = \frac{1}{(P(b,j,m) + P(\neg b,j,m))}$$

$$\alpha = \frac{1}{(.00059 + .0015)}$$

$$= 478.5$$

$$P(b \mid j, m) = \propto * P(b, j, m)$$

$$= 478.5 * .00059$$

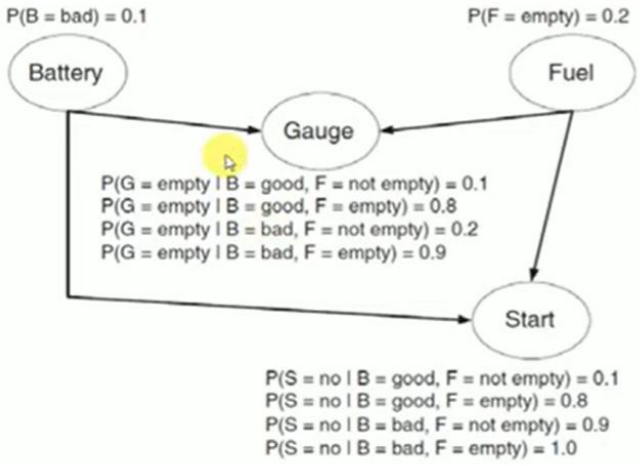
$$= 0.28$$

$$P(\neg b \mid j,m) = \propto * P(\neg b, j, m)$$

$$= 478.5 * .0015$$

$$= 0.72$$

Bayesian Belief Network – Solved Example 3



1. P(B=Good, F=Empty, G=Empty, S=Yes)
2. P(B=Bad, F=Empty, G=Not Empty, S=No)
3. Given the battery is bad, Compute the probability that the car will start

1. **P(B=Good, F=Empty, G=Empty, S=Yes)**

$$
\begin{aligned}
& P(B = good, F = empty, G = empty, S = yes) \\
= \; & P(B = good) \times P(F = empty) \times P(G = empty | B = good, F = empty) \\
& \times P(S = yes | B = good, F = empty) \\
= \; & 0.9 \times 0.2 \times 0.8 \times 0.2 = 0.0288.
\end{aligned}
$$

2. **P(B=Bad, F=Empty, G=Not Empty, S=No)**

$$
\begin{aligned}
& P(B = bad, F = empty, G = not\ empty, S = no) \\
= \; & P(B = bad) \times P(F = empty) \times P(G = not\ empty | B = bad, F = empty) \\
& \times P(S = no | B = bad, F = empty) \\
= \; & 0.1 \times 0.2 \times 0.1 \times 1.0 = 0.002.
\end{aligned}
$$

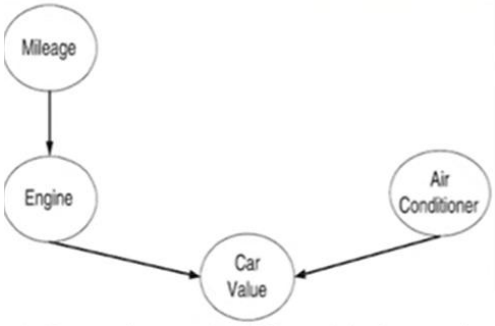
3. **Given the battery is bad, Compute the probability that the car will start.**

$$
\begin{aligned}
& p(S = yes \mid B = bad) \\
&= p(S = yes, B = bad) \Big/ p(B = bad) \\
&= \sum_{\alpha} p(S = yes, B = bad, F = \alpha) \Big/ p(B = bad) \\
&= \sum_{\alpha} p(S = yes | B = bad, F = \alpha) \times p(B = bad, F = \alpha) \Big/ p(B = bad) \\
&= \sum_{\alpha} p(S = yes | B = bad, F = \alpha) \times p(B = bad) \times p(F = \alpha) \Big/ p(B = bad) \\
&= \sum_{\alpha} p(S = yes \mid B = bad, F = \alpha) \times p(F = \alpha) \\
&= P(S = Yes | B = bad, F = Empty) * P(F = Empty) + \\
&\quad P(S = Yes | B = bad, F = Non - Empty) * P(F = No - Empty) \\
&= (0 * 0.2) + (0.1 * 0.8) \\
&= 0.08
\end{aligned}
$$

3. **Given the battery is bad, Compute the probability that the car will start.**

$$
\begin{aligned}
&p(S = yes \mid B = bad) \\
&= p(S = yes, B = bad) \,/\, p(B = bad) \\
&= \sum_{\alpha} p(S = yes, B = bad, F = \alpha) \,/\, p(B = bad) \\
&= \sum_{\alpha} p(S = yes \mid B = bad, F = \alpha) \times p(B = bad, F = \alpha) \,/\, p(B = bad) \\
&= \sum_{\alpha} p(S = yes \mid B = bad, F = \alpha) \times p(B = bad) \times p(F = \alpha) \,/\, p(B = bad) \\
&= \sum_{\alpha} p(S = yes \mid B = bad, F = \alpha) \times p(F = \alpha) \\
&= P(S = Yes \mid B = bad, F = Empty) * P(F = Empty) + \\
&\quad P(S = Yes \mid B = bad, F = Non - Empty) * P(F = No - Empty) \\
&= (0 * 0.2) + (0.1 * 0.8) \\
&= 0.08
\end{aligned}
$$

Bayesian Belief Network - Solved Example – 4

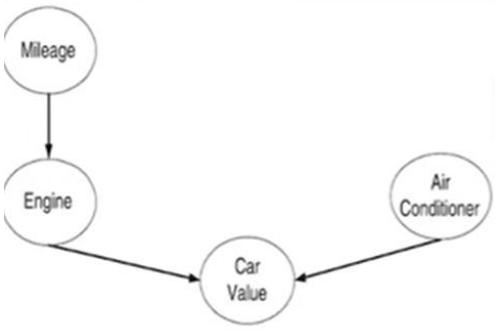


Mileage	Engine	Air Conditioner	No of Records with Car Value=Hi	No of Records with Car Value=Lo
Hi	Good	Working	3	4
Hi	Good	Broken	1	2
Hi	Bad	Working	1	5
Hi	Bad	Broken	0	4
Lo	Good	Working	9	0
Lo	Good	Broken	5	1
Lo	Bad	Working	1	2
Lo	Bad	Broken	0	2

1. Draw the probability table for each node in the network

- $P(Mileage = Hi) = \frac{20}{40} = 0.5$
- $P(Air\ Cond = Working) = \frac{25}{40} = 0.625$

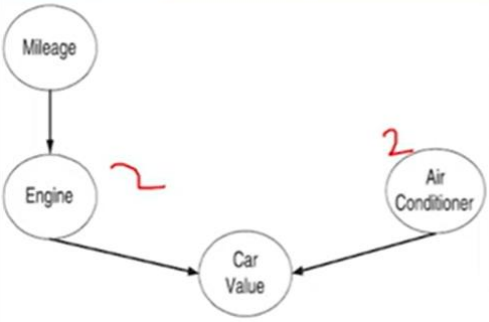
Bayesian Belief Network - Solved Example – 4



Mileage	Engine	Air Conditioner	No of Records with Car Value=Hi	No of Records with Car Value=Lo
Hi	Good	Working	3	4
Hi	Good	Broken	1	2
Hi	Bad	Working	1	5
Hi	Bad	Broken	0	4
Lo	Good	Working	9	0
Lo	Good	Broken	5	1
Lo	Bad	Working	1	2
Lo	Bad	Broken	0	2

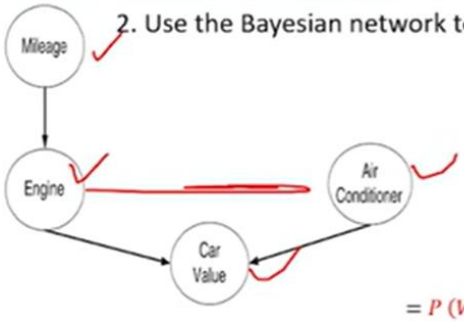
1. Draw the probability table for each node in the network

- $P(Engine = Good \mid Mileage = Hi) = \frac{10}{20} = 0.5$
- $P(Engine = Good \mid Mileage = Lo) = \frac{15}{20} = 0.75$



Mileage	Engine	Air Conditioner	No of Records with Car Value=Hi	No of Records with Car Value=Lo
Hi	Good	Working	3	4
Hi	Good	Broken	1	2
Hi	Bad	Working	1	5
Hi	Bad	Broken	0	4
Lo	Good	Working	9	0
Lo	Good	Broken	5	1
Lo	Bad	Working	1	2
Lo	Bad	Broken	0	2

- $P(Value = High|Engine = Good, Air\ Cond = Working) = \frac{12}{16} = 0.750$
- $P(Value = High|Engine = Good, Air\ Cond = Broken) = \frac{6}{9} = 0.667$
- $P(Value = High|Engine = Bad, Air\ Cond = Working) = \frac{2}{9} = 0.222$
- $P(Value = High|Engine = Bad, Air\ Cond = Broken) = \frac{0}{6} = 0$



2. Use the Bayesian network to compute $P(Engine = Bad, Air\ Conditioner = Broken)$

$$P(Engine = Bad, Air\ Cond = Broken)$$
$$= \sum_{\alpha\beta} P(Engine = Bad, Air\ Cond = Broken, Mileage = \alpha, Value = \beta)$$
$$= \sum_{\alpha\beta} P(Value = \beta|Engine = Bad, Air\ Cond = Broken) \times P(Engine = Bad|Mileage = \alpha) \times P(Mileage = \alpha)\, P(Air\ Cond = Broken)$$
$$= P(Value = High|Engine = Bad, Air\ Cond = Broken) \{P(Engine = Bad|Mileage = hi) \times P(Mileage = hi)\, P(Air\ Cond = Broken) + P(Engine = Bad|Mileage = Lo) \times P(Mileage = Lo)\, P(Air\ Cond = Broken)\} +$$
$$P(Value = Low|Engine = Bad, Air\ Cond = Broken) \{P(Engine = Bad|Mileage = hi) \times P(Mileage = hi)\, P(Air\ Cond = Broken) + P(Engine = Bad|Mileage = Lo) \times P(Mileage = Lo)\, P(Air\ Cond = Broken)\}$$
$$= 0.1453$$

https://www.datacamp.com/blog/curse-of-dimensionality-machine-learning

Association Rule Mining

Association Rule Mining

- Association rule mining is a popular, unsupervised learning technique, used in business to help identify shopping patterns.
- It is also known as market basket analysis.
- It helps find interesting relationships (affinities) between variables (items or events).
- Thus, it can help cross-sell related items and increase the size of a sale.

Association Rule Mining

- All data used in this technique is categorical.
- There is no dependent variable.
- It uses machine-learning algorithms.
- This technique accepts as input the raw point-of-sale transaction data.
- The output produced is the description of the most frequent affinities among items.
- An example of an association rule would be, “a Customer who bought a laptop computer and virus protection software also bought an extended service plan 70 percent of the time.”
- Another example, “A customer who bought bread and milk also bought butter 80% of the time”

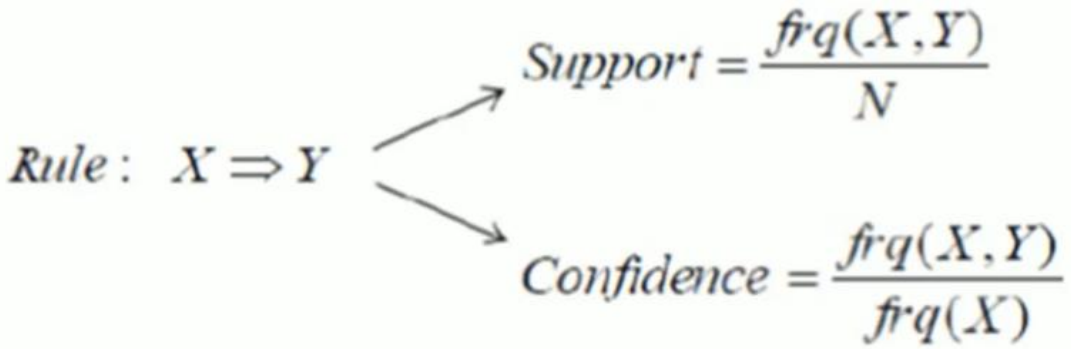
Business Applications of Association Rules

- In business environments a pattern or knowledge can be used for many purposes.
- In sales and marketing, it is used for e-commerce site design, online advertising optimization, product pricing, and sales/promotion configurations.
- This analysis can suggest not to put one item on sale at a time, and instead to create a bundle of products promoted as a package to sell other non-selling items.
- In retail environments, it can be used for store design.
- Strongly associated items can be kept close tougher for customer convenience.
- Or they could be placed far from each other so that the customer has to walk the aisles and by doing so is potentially exposed to other items.

Representing Association Rules

- A generic rule is represented between a set X and Y: **X ⇒ Y [S%, C%]**
- **X, Y**: products and/or services
- **X:** Left-hand-side (LHS or Antecedent)
- **Y:** Right-hand-side (RHS or Consequent)
- **S:** Support: how often **X** and **Y** go together in the total transaction set
- **C:** Confidence: how often **Y** goes together with **X**
- There are 100 transactions.
- Out of 100 product X was bought 80 times and product X and Y were bought 60 times.
- Then the association rule is,
- X ⇒ Y [60%, 75%]

Representing Association Rules

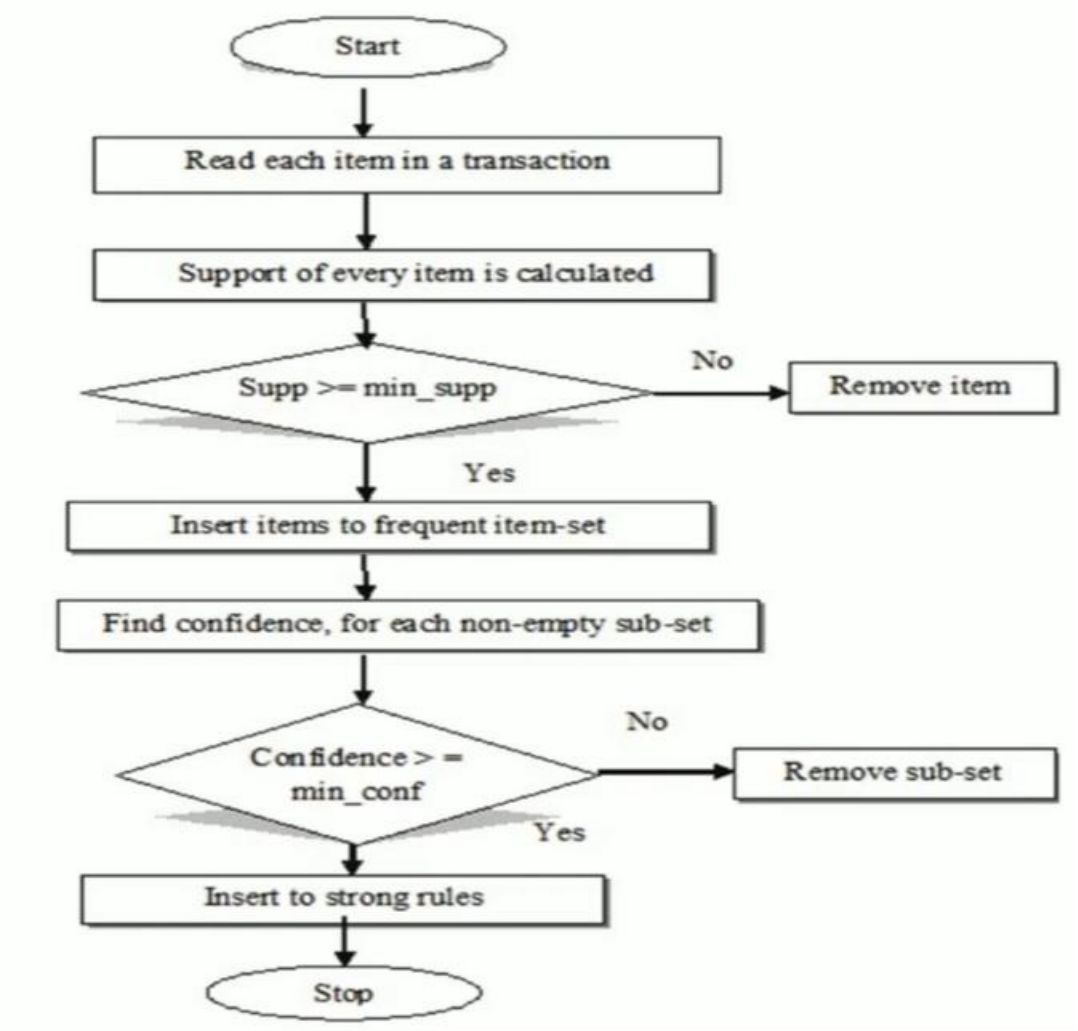


Algorithms for Association Rule

- Not all association rules are interesting and useful, only those that are strong rules and also those that occur frequently.
- In association rule mining, the goal is to find all rules that satisfy the user-specified ***minimum support*** and ***minimum confidence***.
- The resulting sets of rules are all the same irrespective of the algorithm used, that is, given a transaction data set T, a minimum support and a minimum confidence, the set of association rules existing in T is *uniquely determined*.
- The most popular algorithms are Apriori, Eclat, and FP-growth, along with various derivatives and hybrids of the three.

Apriori Algorithm

- This is the most popular algorithm used for association rule mining.
- A frequent itemset is an itemset whose support is greater than or equal to minimum support threshold.
- The Apriori property is a downward closure property, which means that any subsets of a frequent itemset are also frequent itemsets.
- Thus, if (A,B,C,D) is a frequent itemset, then any subset such as (A,B,C) or (B,D) are also frequent itemsets.
- This uses a bottom-up approach; and the size of frequent subsets is gradually increased, from one-item subsets to two-item subsets, then three-item subsets, and so on.
- Groups of candidates at each level are tested against the data for minimum support.



Apriori algorithm

Association Rules Exercise

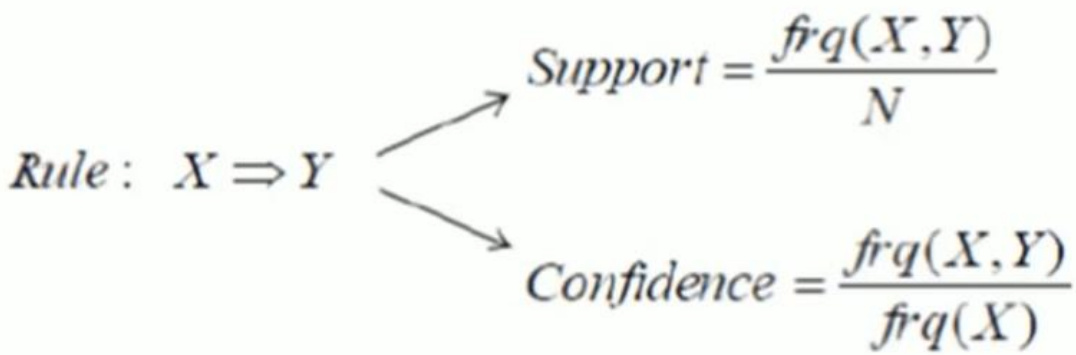
Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

Association Rules Exercise

- Here are a dozen sales transactions.
- The objective is to use this transaction data to find affinities between products, that is, which products sell together often.
- The support level will be set at 33 percent; the confidence level will be set at 50 percent.

Association Rules Exercise



N is total number of transactions

Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

1-item Sets	Frequency
Milk	9
Bread	10
Butter	10
Egg	3
Ketchup	3
Cookies	5

Frequent 1-item Sets	Frequency
Milk	9
Bread	10
Butter	10
Cookies	5

Frequent 1 – items sets are those items which are bought atleast 33% of times because that's the minimum support level

Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

2-item Sets	**Frequency**
Milk, Bread	7
Milk, Butter	7
Milk, Cookies	3
Bread, Butter	9
Butter, Cookies	3
Bread, Cookies	4

Frequent 2-item Sets	Frequency
Milk, Bread	7
Milk, Butter	7
Bread, Butter	9
Bread, Cookies	4

Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

Milk, Bread, Butter, Cookies

3-item Sets	**Frequency**
Milk, Bread, Butter	6
Milk, Bread, Cookies	1
Bread, Butter, Cookies	3
Milk, Butter, Cookies	2

Frequent 3-item Sets	Frequency
Milk, Bread, Butter	6

Association Rule Mining - Subset Creation

- Frequent 3-Item Set = I => {Milk, Bread, Butter}
- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
- How to form Association Rule…?
  - For every non-empty subset S of I, the association rule is,
    - **S → (I-S)**
    - **If support(I) / support(S) >= min_confidence**

Association Rule Mining - Subset Creation

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
  - Min_Support = 30% and Min_Confidence = 60%
- Rule 1: {Milk} → {Bread, Butter} {S=50%, C=66.67%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk) = $\frac{6/12}{9/12}$ = 6/9 = 66.67% > 60%
  - Valid
- Rule 2: {Bread} → {Milk, Butter} {S=50%, C=60%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Bread) = 6/10 = 60% >= 60%
  - Valid

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
  - Min_Support = 30% and Min_Confidence = 60%

- Rule 3: {Butter} → {Milk, Bread} {S=50%, C=60%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Butter) = 6/10 = 60%>=60
  - Valid

- Rule 4: {Milk, Bread} → {Butter} {S=50%, C=85.7%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk, Bread) = 6/7 = 85.7% > 60%
  - Valid

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
  - Min_Support = 30% and Min_Confidence = 60%

- Rule 5: {Milk, Butter} → {Bread} {S=50%, C=85.7%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk, Butter) = 6/7 = 85.7% >= 60%
  - Valid

- Rule 6: {Bread, Butter} → {Milk} {S=50%, C=66.67%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Bread, Butter) = 6/9 = 66.67%>=60
  - Valid

Consider the following transactions.
Apply the association rule mining to get the association rules with min support of 2 and confidence of 50%

TID	List of Items IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Transactions List

TID	List of Items IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

1-item Sets	Frequency
I1	6
I2	7
I3	5
I4	4
I5	2

Frequent 1-item Sets	Frequency
I1	6
I2	7
I3	5
I4	4

Transactions List

TID	List of Items IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

2-item Sets	Frequency
I1, I2	4
I1, I3	4
I1, I4	1
I1, I5	2
I2, I3	3
I2, I4	2
I2, I5	2
I3, I4	0
I3, I5	1
I4, I5	0

Frequent 2-item Sets	Frequency
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	3
I2, I4	2
I2, I5	2

Transactions List

I1, I2, I3, I4, I5

TID	List of Items IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

3-item Sets	Frequency
I1, I2, I3	2
I1, I2, I4	0
I1, I2, I5	2
I1, I3, I4	1
I1, I3, I5	1
I1, I4, I5	0
I2, I3, I4	0
I2, I3, I5	1
I2, I4, I5	0
I3, I4, I5	0

Frequent 3-item Sets	Frequency
I1, I2, I3	2
I1, I2, I5	2

Transactions List

TID	List of Items IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

I1, I2, I3, I5

4-item Sets	Frequency
I1, I2, I3, I5	1

Frequent 4-item Sets	Frequency
Not Possible	

- Frequent 3-Item Set = I => {1, 2, 3} and {1, 2, 5}
- Min_Support = 2 = 2/9 = 22.22% and Min_Confidence = 50%

- Non-Empty subset are
  - {{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}}
  - {{1}, {2}, {5}, {1, 2}, {1, 5}, {2, 5}}

- How to form Association Rule…?
  - For every non-empty subset S of I, the association rule is,
    - **S → (I-S)**
    - **If support(I) / support(S) >= min_confidence**

- Frequent 3-Item Set = I => {1, 2, 3}
- Non-Empty subset are
  - {{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}}

- Rule 1: {1} → {2, 3} {S= 22.22 %, C=33.34%}
  - Support = 2/9 = 22.22%
  - Confidence = Support (1, 2, 3)/Support(1) = $\frac{2/9}{6/9}$ = 2/6 = 33.34% < 50%
  - Invalid Rule

- Rule 2: {2} → {1, 3} {S= 22.22 %, C=28.57%}
  - Support = 2/9 = 22.22 %
  - Confidence = Support (1, 2, 3)/Support(2) = 2/7 = 28.57% < 50%
  - Invalid Rule

- Frequent 3-Item Set = I => {1, 2, 3}
- Non-Empty subset are
  - {{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}}

- Rule 3: {3} → {1, 2} {S= 22.22 %, C=40%}
  - Support = 2/9 = 22.22 %
  - Confidence = Support (1, 2, 3)/Support(3) = 2/5 = 40% < 50%
  - Invalid Rule

- Rule 4: {1, 2} → {3} {S= 22.22 %, C=50%}
  - Support = 2/9 = 22.22 %
  - Confidence = Support (1, 2, 3)/Support(1, 2) = 2/4 = 50% >= 50%
  - Valid Rule

- Frequent 3-Item Set = I => {1, 2, 3}
- Non-Empty subset are
  - {{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}}

- Rule 5: {1, 3} → {2} {S= 22.22 %, C=50%}
  - Support = 2/9 = 22.22 %
  - Confidence = Support (1, 2, 3)/Support(1, 3) = 2/4 = 50% >= 50%
  - Valid Rule

- Rule 6: {2, 3} → {1} {S= 22.22 %, C=66.67%}
  - Support = 2/9 = 22.22 %
  - Confidence = Support (1, 2, 3)/Support(2, 3) = 2/3 = 66.67% >= 50%
  - Valid Rule

Candidate Elimination Algorithm

https://www.geeksforgeeks.org/ml-candidate-elimination-algorithm/

Candidate Elimination Algorithm – Explained

Initialize the generic and specific boundary

For each training example *d*, do:

***If d* is *positive* example**

- **Remove from *G* any hypothesis *h* inconsistent with *d***
- **For each hypothesis *s* in *S* not consistent with *d*:**
  - **Remove *s* from *S***
  - **Add to *S* all minimal generalizations of *s* consistent with *d***

***If d* is *negative* example**

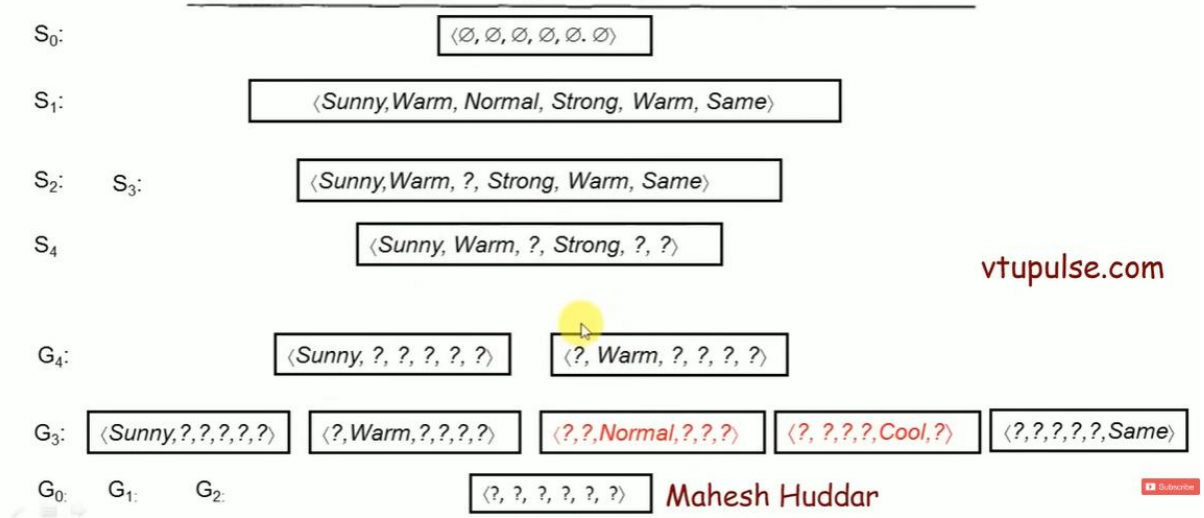
- **Remove from *S* any hypothesis *h* inconsistent with *d***
- **For each hypothesis *g* in *G* not consistent with *d*:**
  - **Remove *g* from *G***
  - **Add to *G* all minimal specializations of *g* consistent with *d***

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
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

S_0: ⟨∅, ∅, ∅, ∅, ∅, ∅⟩

G_0: ⟨?, ?, ?, ?, ?, ?⟩

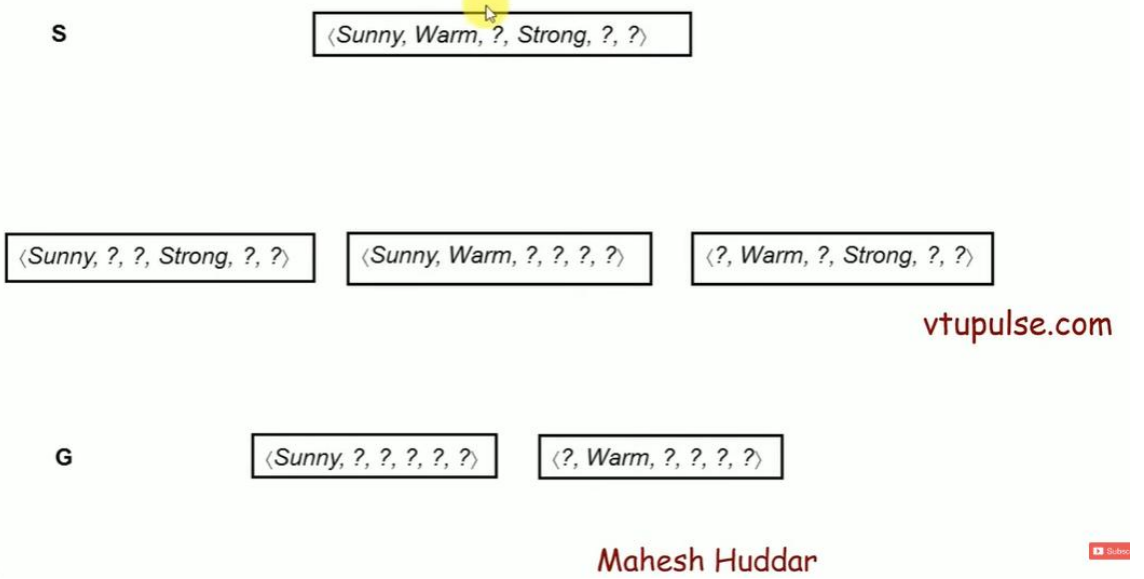
Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes



vtupulse.com

Mahesh Huddar

Learned Version Space by Candidate Elimination Algorithm



vtupulse.com

Mahesh Huddar

Candidate Elimination Algorithm Solved Example - 2

S0: (0, 0, 0)

S1: (0, 0, 0)

S2: (0, 0, 0)

S3: (Small, Red, Circle)

S4: (Small, Red, Circle)

S5: (Small, ?, Circle)

S: G: (Small, ?, Circle)

G5: (Small, ?, Circle)

G4: (Small, ?, Circle)

G3: (Small, ?, Circle)

G2: (Small, Blue, ?) (Small, ?, Circle) (?, Blue, ?) (Big, ?, Triangle) (?, Blue, Triangle)

G1: (Small, ?, ?) (?, Blue, ?) (?, ?, Triangle)

G0: (?, ?, ?)

SIze	Color	Shape	Class / Label
Big	Red	Circle	No
Small	Red	Triangle	No
Small	Red	Circle	Yes
Big	Blue	Circle	No
Small	Blue	Circle	Yes

Candidate Elimination Algorithm Solved Example - 3

- S0: (0, 0, 0, 0, 0)
- S1: (0, 0, 0, 0, 0)
- S2: (Many, Big, No, Exp, Many)
- S3: (Many, ?, No, Exp, ?)
- S4: (Many, ?, No, ?, ?)

Example	Citations	Size	InLibrary	Price	Editions	Buy
1	Some	Small	No	Affordable	One	No
2	Many	Big	No	Expensive	Many	Yes
3	Many	Medium	No	Expensive	Few	Yes
4	Many	Small	No	Affordable	Many	Yes

Final Hypothesis Set: (Many, ?, No, ?, ?) (Many, ?, ?, ?, ?)

- G4: (Many,?,?,?, ?)
- G3: (Many,?,?,?, ?) (?,?,?,Exp,?)
- G2: (Many,?,?,?, ?) (?, Big,?,?,?) (?,?,?,Exp,?) (?,?,?,?,Many)
- G1: (Many,?,?,?, ?) (?, Big,?,?,?) (?,Medium,?,?,?) (?,?,?,Exp,?) (?,?,?,?,Many) (?,?,?,?,Few)
- G0: (?, ?, ?, ?, ?)

Candidate Elimination Algorithm - Solved Example - 4

S0: (0, 0, 0, 0, 0)

S1: (Circular, Large, Light, Smooth, Thick)

S2: (Circular, Large, Light, ?, Thick)

S3: (Circular, Large, Light, ?, Thick)

S4: (?, Large, Light, ?, Thick)

Example	Shape	Size	Color	Surface	Thickness	Target Concept
1	Circular	Large	Light	Smooth	Thick	Malignant (+)
2	Circular	Large	Light	Irregular	Thick	Malignant (+)
3	Oval	Large	Dark	Smooth	Thin	Benign (-)
4	Oval	Large	Light	Irregular	Thick	Malignant (+)

G4: (?, ?, Light, ?, ?) (?, ?, ?, Irregular, ?) (?, ?, ?, ?, Thick)

G3: ~~(Circular, ?, ?, ?, ?)~~ (?, ?, Light, ?, ?) (?, ?, ?, Irregular, ?) (?, ?, ?, ?, Thick)

G2: (?, ?, ?, ?, ?)

G1: (?, ?, ?, ?, ?)

G0: (?, ?, ?, ?, ?)

Subscribe to Mahesh Huddar Visit: www.vtupulse.com

Candidate Elimination Algorithm - Solved Example - 5

S0: (0, 0, 0, 0, 0)

S1: (Round, Triangle, Round, Purple, Yes)

S2: (Round, Triangle, Round, Purple, Yes)

S3: (?, Triangle, Round, ?, Yes)

S4: (?, Triangle, Round, ?, Yes)

S5: (?, ?, Round, ?, Yes)

G5: (?, ?, Round, ?, Yes)

Ex	Eyes	Nose	Head	Fcolor	Hair	Smile
1	Round	Triangle	Round	Purple	Yes	Yes
2	Square	Square	Square	Green	Yes	No
3	Square	Triangle	Round	Yellow	Yes	Yes
4	Round	Triangle	Round	Green	No	No
5	Square	Square	Round	Yellow	Yes	Yes

G4: (Square, Triangle, ?, ?, ?) (?, Triangle, Square, ?, ?) (?, Triangle, ?, Yellow, ?) (?, Triangle, ?, Purple, ?) (?, Triangle, ?, ?, yes)
(Square, ?, Round, ?, ?) (?, Square, Round, ?, ?) (?, ?, Round, Yellow, ?) (?, ?, Round, Purple, ?) (?, ?, Round, ?, Yes)

G3: (?, Triangle, ?, ?, ?) (?, ?, Round, ?, ?)

G2: (Round, ?, ?, ?, ?) (?, Triangle, ?, ?, ?) (?, ?, Round, ?, ?) (?, ?, ?, Purple, ?)

G1: (?, ?, ?, ?, ?)

G0: (?, ?, ?, ?, ?)

Subscribe to Mahesh Huddar Visit: www.vtupulse.com

KMeans Clustering Algorithm

https://www.gatevidyalay.com/tag/k-means-clustering-numerical-example-pdf/

K-Means Algorithm for Clustering

Here is the pseudocode for implementing a K-means algorithm.

Input: Algorithm K-Means (K number of clusters, D list of data points)

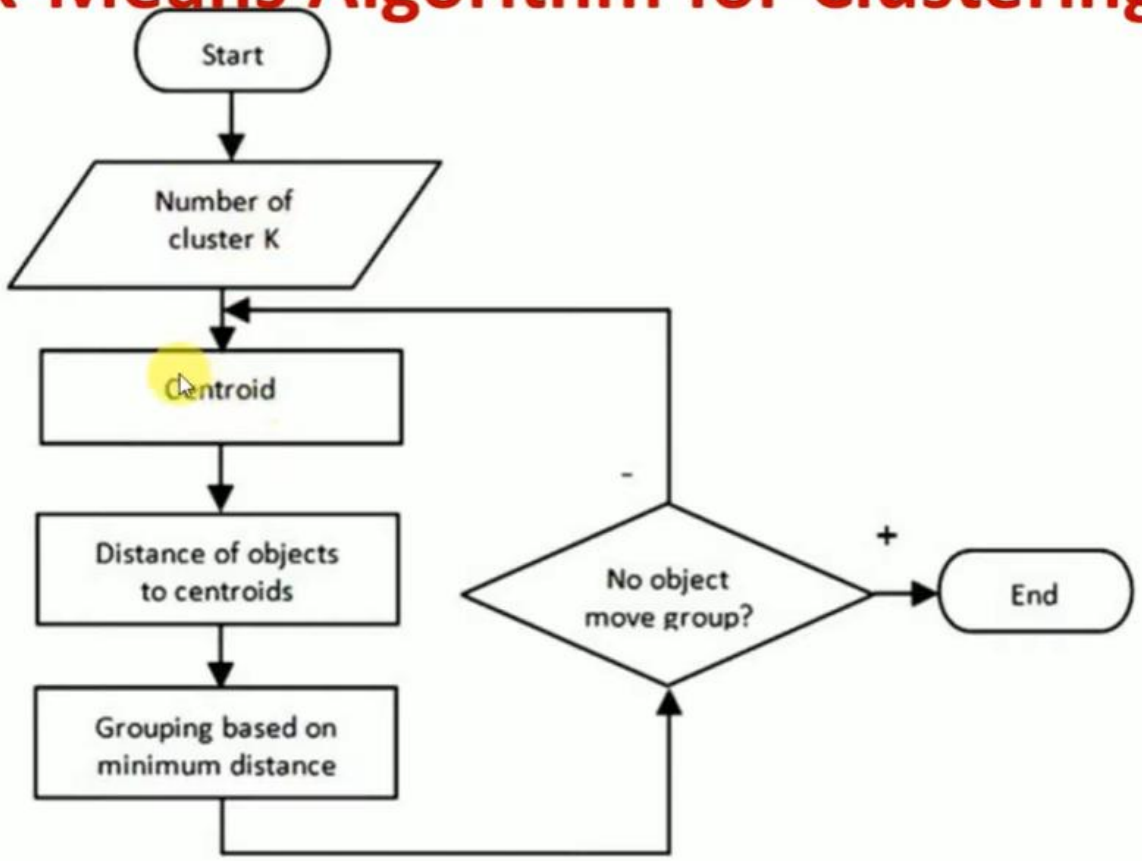
1. Choose K number of random data points as initial centroids (cluster centers).

2. Repeat till cluster centers stabilize:

   a. Allocate each point in D to the nearest of Kth centroids.

   b. Compute centroid for the cluster using all points in the cluster.

K-Means Algorithm for Clustering



Advantages and Disadvantages of K-Means Algorithm

Advantages of K-Means Algorithm

1. K-means algorithm is simple, easy to understand, and easy to implement.
2. It is also efficient, in which the time taken to cluster K-means rises linearly with the number of data points.
3. No other clustering algorithm performs better than K-means.

Disadvantages of K-Means Algorithm

1. The user needs to specify an initial value of K.
2. The process of finding the clusters may not converge.
3. It is not suitable for discovering clusters that are not hyper ellipsoids or hyper spheres).

K-Means Clustering – Solved Example

- Suppose that the data mining task is to cluster points into three clusters,
- where the points are
- $A1(2, 10)$, $A2(2, 5)$, $A3(8, 4)$, $B1(5, 8)$, $B2(7, 5)$, $B3(6, 4)$, $C1(1, 2)$, $C2(4, 9)$.
- The distance function is Euclidean distance.
- Suppose initially we assign $A1$, $B1$, and $C1$ as the center of each cluster, respectively.

K-Means Clustering – Solved Example

Initial Centroids:
A1: (2, 10)
B1: (5, 8)
C1: (1, 2)

New Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			2	10	5	8	1	2		
A1	2	10	0.00		3.61		8.06		1	
A2	2	5	5.00		4.24		3.16		3	
A3	8	4	8.49		5.00		7.28		2	
B1	5	8	3.61		0.00		7.21		2	
B2	7	5	7.07		3.61		6.71		2	
B3	6	4	7.21		4.12		5.39		2	
C1	1	2	8.06		7.21		0.00		3	
C2	4	9	2.24		1.41		7.62		2	

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-Means Clustering – Solved Example

Current Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

New Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			2	10	6	6	1.5	1.5		
A1	2	10	0.00		5.66		6.52		1	1
A2	2	5	5.00		4.12		1.58		3	3
A3	8	4	8.49		2.83		6.52		2	2
B1	5	8	3.61		2.24		5.70		2	2
B2	7	5	7.07		1.41		5.70		2	2
B3	6	4	7.21		2.00		4.53		2	2
C1	1	2	8.06		6.40		1.58		3	3
C2	4	9	2.24		3.61		6.04		2	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-Means Clustering – Solved Example

Current Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

New Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

Data Points			Distance to 3	Distance to 9.5	Distance to 6.5	Distance to 5.25	Distance to 1.5	Distance to 3.5	Cluster	New Cluster
A1	2	10	1.12		6.54		6.52		1	1
A2	2	5	4.61		4.51		1.58		3	3
A3	8	4	7.43		1.95		6.52		2	2
B1	5	8	2.50		3.13		5.70		2	1
B2	7	5	6.02		0.56		5.70		2	2
B3	6	4	6.26		1.35		4.53		2	2
C1	1	2	7.76		6.39		1.58		3	3
C2	4	9	1.12		4.51		6.04		1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-Means Clustering – Solved Example

Current Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

Data Points			Distance to 3.67	Distance to 9	Distance to 7	Distance to 4.33	Distance to 1.5	Distance to 3.5	Cluster	New Cluster
A1	2	10	1.94		7.56		6.52		1	1
A2	2	5	4.33		5.04		1.58		3	3
A3	8	4	6.62		1.05		6.52		2	2
B1	5	8	1.67		4.18		5.70		1	1
B2	7	5	5.21		0.67		5.70		2	2
B3	6	4	5.52		1.05		4.53		2	2
C1	1	2	7.49		6.44		1.58		3	3
C2	4	9	0.33		5.55		6.04		1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

When centroids are not given, you can assign centroids randomly

Example 2

K-Means Clustering using L1 Distance

- Consider the 5 data points shown below:

P1: (1, 2, 3)

P2: (0, 1, 2)

P3: (3, 0, 5)

P4: (4, 1, 3)

P5: (5, 0, 1)

- Apply the **Kmeans** clustering algorithm, to group those data points into 2 clusters, using the L1 distance measure.
- Consider the initial centroids are C1: (1, 0, 0) and C2: (0, 1, 1).

K-Means Clustering using L1 Distance

- L1 distance is just manhattan distance: sum of differences in each dimension – **ITERATION 1**

Data Point	C1: (1, 0, 0)	C2: (0, 1, 1)	Cluster
P1: (1, 2, 3)	5	4	C2
P2: (0, 1, 2)	4	1	C2
P3: (3, 0, 5)	7	8	C1
P4: (4, 1, 3)	7	6	C2
P5: (5, 0, 1)	5	6	C1

L1 distance is just manhattan distance: sum of differences in each dimension - **ITERATION 2**

Data Point	C1: (4, 0, 3)	C2: (1.6, 1.3, 2.6)	Cluster
P1: (1, 2, 3)	5	1.7	C2
P2: (0, 1, 2)	6	2.5	C2
P3: (3, 0, 5)	3	5.3	C1
P4: (4, 1, 3)	1	3.1	C1
P5: (5, 0, 1)	3	6.3	C1

L1 distance is just manhattan distance: sum of differences in each dimension - **ITERATION 3**

Data Point	C1: (4, 0.33, 3)	C2: (0.5, 1.5. 2.5)	Cluster
P1: (1, 2, 3)	4.67	1.5	C2
P2: (0, 1, 2)	5.67	1.5	C2
P3: (3, 0, 5)	3.33	6.5	C1
P4: (4, 1, 3)	0.67	4.5	C1
P5: (5, 0, 1)	3.33	7.5	C1

Clusters using a Single Link Technique Agglomerative Hierarchical Clustering

Clusters using a Single Link Technique Example - 1

Sample No.	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Step 1: Compute the distance matrix

- So we have to find the Euclidean distance between each and every points.
- Let $A(x_1, y_1)$ and $B(x_2, y_2)$ are two points.
- Then Euclidean distance between

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Sample No.	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

$$d(p_1, p_2) = \sqrt{(0.22 - 0.40)^2 + (0.38 - 0.53)^2}$$
$$= 0.23$$
$$d(p_1, p_3) = \sqrt{(0.35 - 0.40)^2 + (0.32 - 0.53)^2}$$
$$= 0.22$$
$$d(p_2, p_3) = \sqrt{(0.35 - 0.22)^2 + (0.32 - 0.38)^2}$$
$$= 0.14$$

Clusters using a Single Link Technique Example - 1

Sample No.	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & 0.22 & 0.14 & 0 & & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 & \\ P6 & 0.24 & 0.24 & 0.10 & 0.22 & 0.39 & 0 \end{pmatrix}$$

Step 2: Merging the two closest members.

- Here the **minimum value is 0.10** and hence we combine P3 and P6 (as 0.10 came in the P6 row and P3 column).
- Now, form clusters of elements corresponding to the minimum value and update the distance matrix.

Now we will update the Distance Matrix:

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & 0.22 & 0.14 & 0 & & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 & \\ P6 & 0.24 & 0.24 & 0.10 & 0.22 & 0.39 & 0 \end{pmatrix} \quad \begin{pmatrix} & P1 & P2 & P3, P6 & P4 & P5 \\ P1 & 0 & & & & \\ P2 & 0.23 & 0 & & & \\ P3, P6 & 0.22 & 0.14 & 0 & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 \end{pmatrix}$$

(P3, P6)

Now we will update the Distance Matrix:

$$\begin{pmatrix} & P1 & P2 & P3, P6 & P4 & P5 \\ P1 & 0 & & & & \\ P2 & 0.23 & 0 & & & \\ P3, P6 & 0.22 & 0.14 & 0 & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 \end{pmatrix} \quad \begin{pmatrix} & P1 & P2 & P3, P6, P4 & P5 \\ P1 & 0 & & & \\ P2 & 0.23 & 0 & & \\ P3, P6, P4 & 0.22 & 0.14 & 0 & \\ P5 & 0.34 & 0.14 & 0.28 & 0 \end{pmatrix}$$

{(P3, P6), P4}

Now we will update the Distance Matrix:

$$\begin{pmatrix} & P1 & P2 & P3, P6, P4 & P5 \\ P1 & 0 & & & \\ P2 & 0.23 & 0 & & \\ P3, P6, P4 & 0.22 & 0.14 & 0 & \\ P5 & 0.34 & 0.14 & 0.28 & 0 \end{pmatrix} \quad \begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

{(P3, P6), P4} and **(P2, P5)**

Now we will update the Distance Matrix:

$$\begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

$$\begin{pmatrix} & P1 & P2, P5, P3, P6, P4 \\ P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{pmatrix}$$

[{(P3, P6), P4}, (P2, P5)]

Now we will update the Distance Matrix:

$$\begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

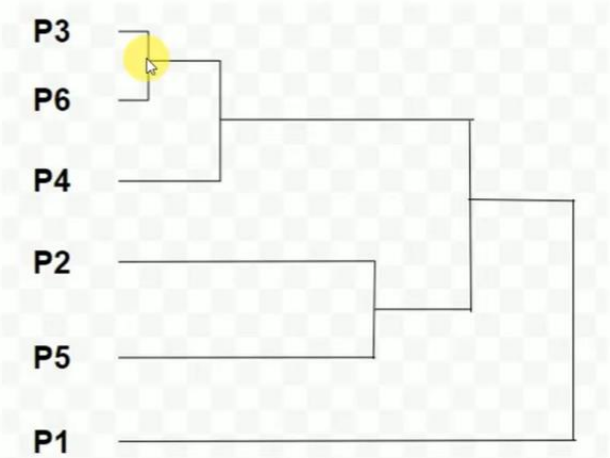
$$\begin{pmatrix} & P1 & P2, P5, P3, P6, P4 \\ P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{pmatrix}$$

[{(P3, P6), P4}, (P2, P5)]

[{(P3, P6), P4}, (P2, P5)] , P1

So now we have reached to the solution, the dendrogram for those question will be as follows:

[{(P3, P6), P4}, (P2, P5)] , P1



Dendogram of the cluster formed

Agglomerative Hierarchical Clustering Single link Complete link Clustering

Agglomerative Hierarchical Clustering Solved Example

- Consider the following set of 6 one dimensional data points:
- 18, 22, 25, 42, 27, 43
- Apply the **agglomerative hierarchical clustering** algorithm to build the hierarchical clustering **dendogram**.
- Merge the clusters using **Min distance** and update the proximity matrix accordingly.
- Clearly show the **proximity matrix** corresponding to each iteration of the algorithm.

- Step – 1

	18	22	25	27	42	43
18	0	4	7	9	24	25
22	4	0	3	5	20	21
25	7	3	0	2	17	18
27	9	5	2	0	15	16
42	24	20	17	15	0	1
43	25	21	18	16	1	0

(42, 43)

- Step – 2

	18	22	25	27	42, 43
18	0	4	7	9	24
22	4	0	3	5	20
25	7	3	0	2	17
27	9	5	2	0	15
42, 43	24	20	17	15	0

(42, 43), (25, 27)

Step – 3

	18	22	25, 27	42, 43
18	0	4	7	24
22	4	0	3	20
25, 27	7	3	0	15
42, 43	24	20	15	0

(42, 43), ((25, 27), 22)

Step – 4

	18	22, 25, 27	42, 43
18	0	4	24
22, 25, 27	4	0	15
42, 43	24	15	0

(42, 43), (((25, 27), 22), 18)

p – 5

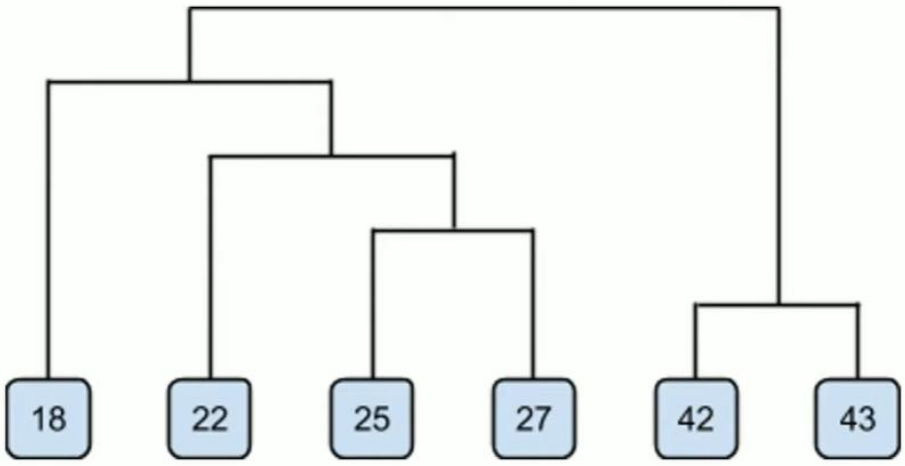
	18, 22, 25, 27	42, 43
18, 22, 25, 27	0	15
42, 43	15	0

((42, 43), (((25, 27), 22), 18))

Step – 6

	18, 22, 25, 27, 42, 43
18, 22, 25, 27, 42, 43	0

Dendrogram ((42, 43), (((25, 27), 22), 18))



Solved Example Complete Linkage

- Given a one-dimensional data set {1, 5, 8, 10, 2}, use the agglomerative clustering algorithms with the complete link with Euclidean distance to establish a hierarchical grouping relationship.
- By using the cutting threshold of 5, how many clusters are there?

$$Euclidean\ distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$Euclidean\ distance = \sqrt{(x_2 - x_1)^2}$$

- In order to use the agglomerative algorithm,
- we need to calculate the distance matrix.
- One-dimensional data set {1, 5, 8, 10, 2}

	1	5	8	10	2
1	0	4	7	9	1
5	4	0	3	5	3
8	7	3	0	2	6
10	9	5	2	0	8
2	1	3	6	8	0

	1	2	3	4	5
1	0	4	7	9	1
2	4	0	3	5	3
3	7	3	0	2	6
4	9	5	2	0	8
5	1	3	6	8	0

- d(2, {1,5}) = max{ d(2,1), d(2,5) } = max {4, 3} = 4
- d(3, {1,5}) = max{ d(3,1), d(3,5) } = max {7, 6} = 7
- d(4, {1,5}) = max{ d(4,1), d(4,5) } = max {9, 8} = 9
- Let the 1st column (row) denote the distances between this cluster and other points, we have the following distance matrix:

	1	2	3	4	5
1	0	4	7	9	1
2	4	0	3	5	3
3	7	3	0	2	6
4	9	5	2	0	8
5	1	3	6	8	0

	1,5	2	3	4
1,5	0	4	7	9
2	4	0	3	5
3	7	3	0	2
4	9	5	2	0

- From the above distance matrix, we can see the distance between points 3 and 4 is smallest.
- Hence, they merge together to form a cluster {3, 4}.
- Using the complete link, we have the distance between different points/clusters as follows:
- d({1,5}, {3, 4}) = max{ d({1,5}, 3), d({1,5}, 4) } = max{ 7, 9} = 9
- d(2, {3,4}) = max{ d(2,3), d(2,4) } = max {3, 5} = 5
- Thus, we can update the distance matrix, where row 2 corresponds to point 2, rows 1 and 3 correspond to clusters {1, 5} and {3, 4}, as follows:

	1,5	2	3	4
1,5	0	4	7	9
2	4	0	3	5
3	7	3	0	2
4	9	5	2	0

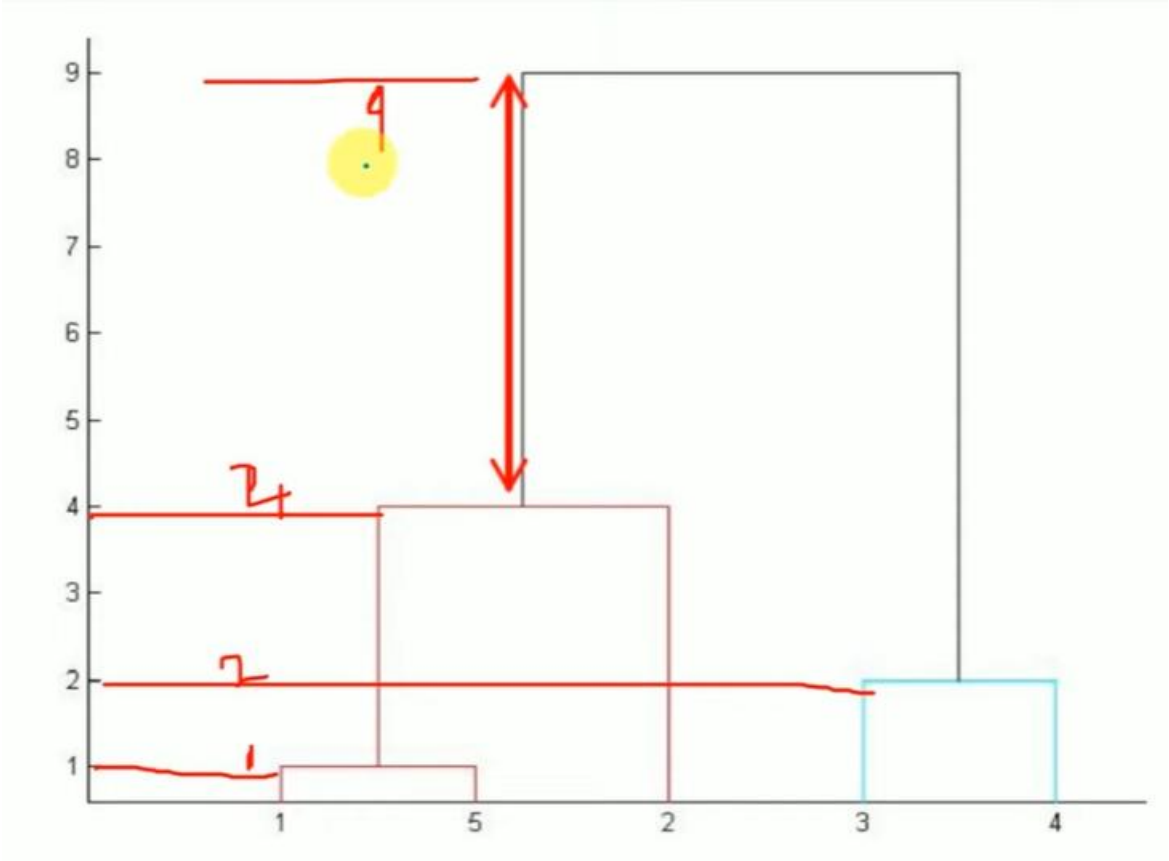
	1,5	2	3,4
1,5	0	4	9
2	4	0	5
3,4	9	5	0

- Following the same procedure, we merge point 2 with the cluster {1, 5} to form {1, 2, 5} and update the distance matrix as follows:

	1,5	2	3,4
1,5	0	4	9
2	4	0	5
3,4	9	5	0

	[1,5],2	[3,4]
[1,5],2	0	9
[3,4]	9	0

- After increasing the distance threshold to 9, all clusters would merge.



DBSCAN Clustering Algorithm

DBSCAN Clustering Algorithm Solved Example – 1

- Apply the DBSCAN algorithm to the given data points and
- Create the clusters with
- minPts = 4 and
- epsilon (ε) = 1.9.

Data Points:

P1: (3, 7)	P2: (4, 6)
P3: (5, 5)	P4: (6, 4)
P5: (7, 3)	P6: (6, 2)
P7: (7, 2)	P8: (8, 4)
P9: (3, 3)	P10: (2, 6)
P11: (3, 5)	P12: (2, 4)

- Use Eucladian distance and calculate the distance between each points.

$$Distance(A(x_1, y_1), B(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

DBSCAN Clustering Algorithm Solved Example – 1

P1: (3, 7)
P2: (4, 6)
P3: (5, 5)
P4: (6, 4)
P5: (7, 3)
P6: (6, 2)
P7: (7, 2)
P8: (8, 4)
P9: (3, 3)
P10: (2, 6)
P11: (3, 5)
P12: (2, 4)

minPts = 4 and epsilon (ε) = 1.9

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0											
P2	1.41	0										
P3	2.83	1.41	0									
P4	4.24	2.83	1.41	0								
P5	5.66	4.24	2.83	1.41	0							
P6	5.83	4.47	3.16	2.00	1.41	0						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

DBSCAN Clustering Algorithm Solved Example – 1

minPts = 4 and epsilon (ε) = 1.9

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0											
P2	1.41	0										
P3	2.83	1.41	0									
P4	4.24	2.83	1.41	0								
P5	5.66	4.24	2.83	1.41	0							
P6	5.83	4.47	3.16	2.00	1.41	0						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

P1: P2, P10
P2: P1, P3, P11
P3: P2, P4
P4: P3, P5
P5: P4, P6, P7, P8
P6: P5, P7
P7: P5, P6
P8: P5
P9: P12
P10: P1, P11
P11: P2, P10, P12
P12: P9, P11

If points are less than minpts its called as a noise

DBSCAN Clustering Algorithm Solved Example – 1

P1: P2, P10
P2: P1, P3, P11
P3: P2, P4
P4: P3, P5
P5: P4, P6, P7, P8
P6: P5, P7
P7: P5, P6
P8: P5
P9: P12
P10: P1, P11
P11: P2, P10, P12
P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9

Point	Status	
P1	Noise	Border
P2	Core	
P3	Noise	Border
P4	Noise	Border
P5	Core	
P6	Noise	Border
P7	Noise	Border
P8	Noise	Border
P9	Noise	
P10	Noise	Border
P11	Core	
P12	Noise	Border

DBSCAN Clustering Algorithm Solved Example – 1

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

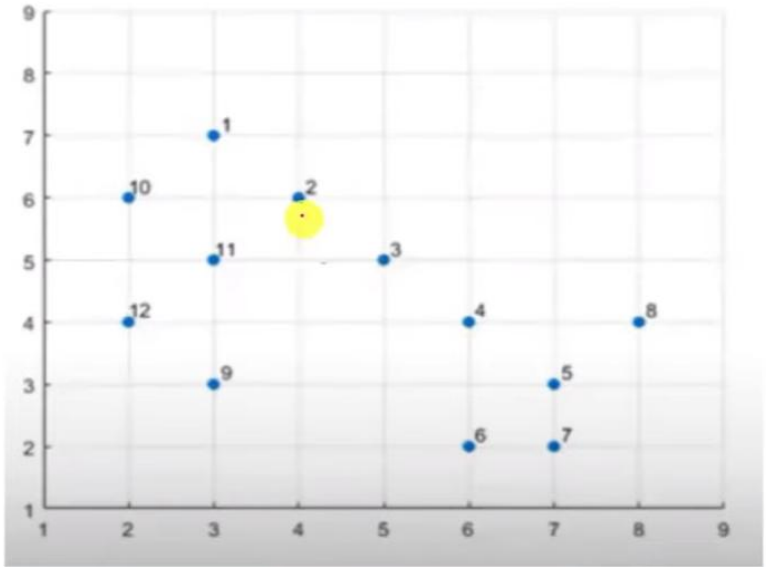
P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9



DBSCAN Clustering Algorithm Solved Example – 1

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

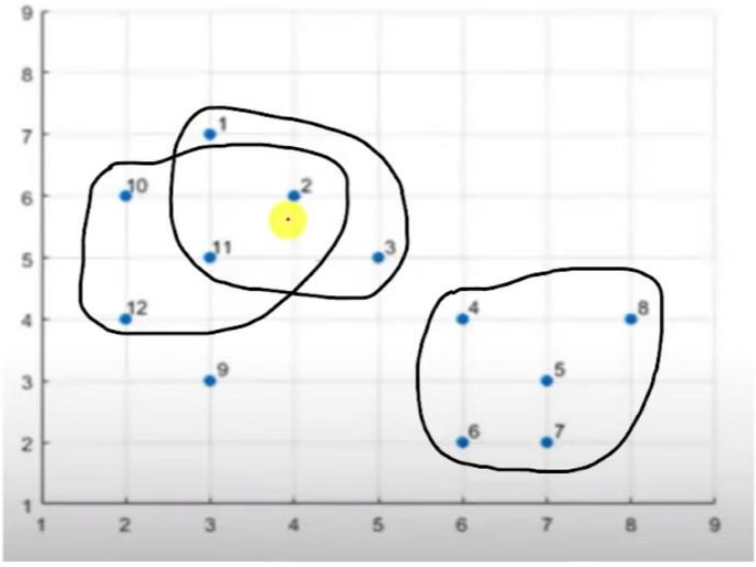
P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9



Apply the DBSCAN algorithm with similarity threshold of 0.8 (using the similarity matrix) to the given data points and MinPts>=2 (Minimum required points in a cluster) what are core, border and noise (outliers) in the set of points given in table.

	P1	P2	P3	P4	P5
P1	1.00	0.10	0.41	0.55	0.35
P2	0.10	1.00	0.64	0.47	0.98
P3	0.41	0.64	1.00	0.44	0.85
P4	0.55	0.47	0.44	1.00	0.76
P5	0.35	0.98	0.85	0.76	1.00

minPts = 2 and Similarity Index = 0.8

	P1	P2	P3	P4	P5
P1	1.00	0.10	0.41	0.55	0.35
P2	0.10	1.00	0.64	0.47	0.98
P3	0.41	0.64	1.00	0.44	0.85
P4	0.55	0.47	0.44	1.00	0.76
P5	0.35	0.98	0.85	0.76	1.00

P1: -

P2: P5

P3: P5

P4: -

P5: P2, P3

Point	Status
P1	Noise
P2	Core
P3	Core
P4	Noise
P5	Core

No Border Points in the given dataset

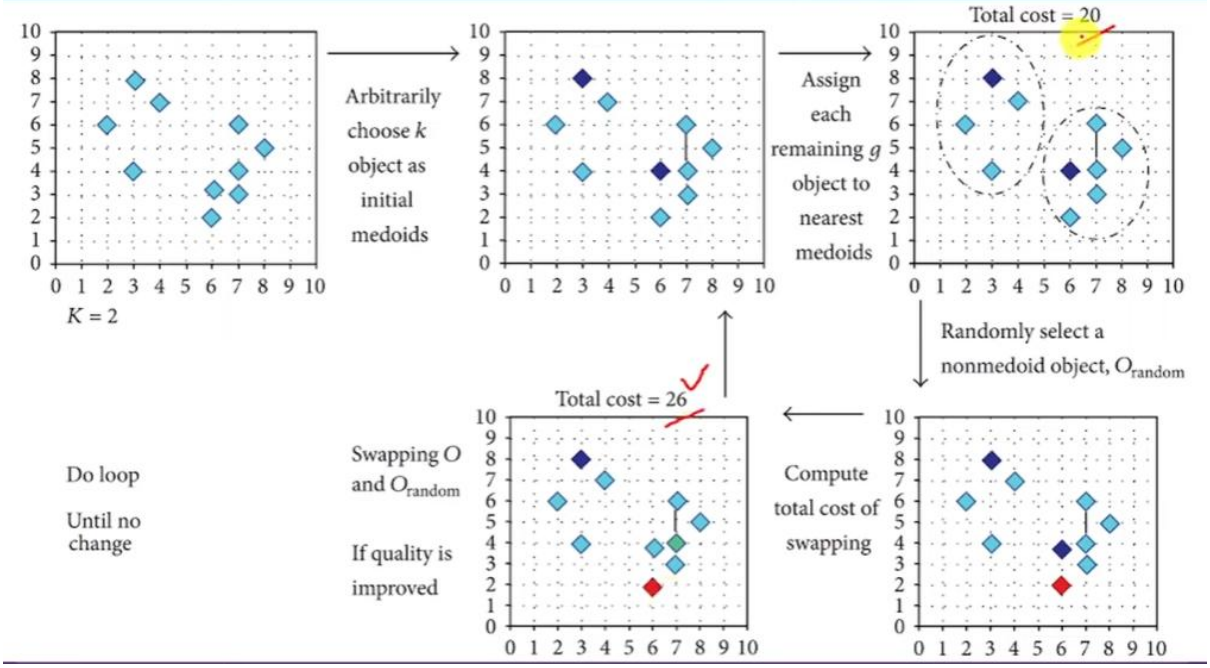
K-Medoid Clustering Algorithm

K-Medoid vs K-Means Clustering

- Partitioning Around Medoids or the K-medoids algorithm is a partitional clustering algorithm which is slightly modified from the K-means algorithm.
- In K-means algorithm, they choose means as the centroids but in the K-medoids, data points are chosen to be the medoids.

K-Medoid Clustering Algorithm

1. Initially select **k random** points as the medoids from the given **n data points** of the data set.
2. **Associate each data point** to **the closest medoid** by using any of the most common **distance metrics.**
3. **Calculate the cost** as the total sum of the distances (also called dissimilarities) of the data points from the assigned medoid. $c = \sum_{Ci} \sum_{Pi \in Ci} |Pi - Ci|$
4. **Swap one medoid point** with a **non-medoid point** and recalculate the cost.
5. If the **calculated cost** with the **new medoid point** is **more** than the previous cost, we **undo the swap**, and the algorithm converges else; **we repeat step 4.**



Advantages:

- It is simple to understand and easy to implement.
- K-Medoid Algorithm is fast and converges in a fixed number of steps.
- K-Medoid is less sensitive to outliers than other partitioning algorithms.

Disadvantages:

- The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrarily shaped) groups of objects.
- It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.

K-Medoid Clustering – Solved Example – 1

i	x	y
X1	2	6
X2	3	4
X3	3	8
X4	4	7
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6

- Apply K-Medoid clustering algorithm to form two clusters.
- Use Manhattan distance to find the between data point and medoid.

K-Medoid Clustering – Solved Example – 1

Step 1

- Select two medoids
- C1=(3, 4)
- C2=(7, 4)
- $Manhattan\ Dist = |x_1 - x_2| + |y_1 - y_2|$
- $Mdist[(2,6),(3,4)] = |2-3| + |6-4| = 3$
- $Mdist[(3,4),(3,4)] = |3-3| + |4-4| = 0$

i	x	y	C1	C2	Cluster
X1	2	6	3	7	C1
X2	3	4	0	4	C1
X3	3	8	4	8	C1
X4	4	7	4	6	C1
X5	6	2	5	3	C2
X6	6	4	3	1	C2
X7	7	3	5	1	C2
X8	7	4	4	0	C2
X9	8	5	6	2	C2
X10	7	6	6	2	C2

K-Medoid Clustering – Solved Example – 1

- C1: {(2,6), (3,4), (3,8), (4,7)}
- C2: {(6, 2), (6, 4), (7, 3), (7, 4), (8, 5), (7,6)}
- **Calculate the Total Cost**
- $\boldsymbol{Cost(c, x)} = \sum_i |\boldsymbol{c_i} - \boldsymbol{x_i}|$
- $Total\ Cost = \{Cost\big((3,4),(2,6)\big) + Cost\big((3,4),(3,8)\big) + Cost\big((3,4),(4,7)\big) + Cost\big((7,4),(6,2)\big) + Cost\big((7,4),(6,4)\big) + Cost\big((7,4),(7,3)\big) + Cost\big((7,4),(8,5)\big) + Cost\big((7,4),(7,6)\big)\}$
- $\boldsymbol{Total\ Cost} = \mathbf{3+4+4+2+3+1+1+2} = \mathbf{20}$

K-Medoid Clustering – Solved Example – 1

Step 3

- **Randomly select one non-medoid point and recalculate the cost.**
- C1=(3, 4) and C2=(7, 4)
- **O=(7, 3)**
- **Swap C2 with O**
- **New Medoids**
- **C1=(3, 4) and O=(7, 3)**

i	x	y	C1	O	Cluster
X1	2	6			
X2	3	4			
X3	3	8			
X4	4	7			
X5	6	2			
X6	6	4			
X7	7	3			
X8	7	4			
X9	8	5			
X10	7	6			

K-Medoid Clustering – Solved Example – 1

Step 3

- New Medoids
- C1=(3, 4) and O=(7, 3)
- $Manhattan\ Dist = |x_1 - x_2| + |y_1 - y_2|$
- $Mdist[(2,6),(7,3)] = |2-7| + |6-3| = 8$

i	x	y	C1	O	Cluster
X1	2	6	3	8	C1
X2	3	4	0	5	C1
X3	3	8	4	9	C1
X4	4	7	4	7	C1
X5	6	2	5	2	O
X6	6	4	3	2	O
X7	7	3	5	0	O
X8	7	4	4	1	O
X9	8	5	6	3	O
X10	7	6	6	3	O

K-Medoid Clustering – Solved Example – 1

Step 3

- New Medoids
- C1=(3, 4) and O=(7, 3)
- $Manhattan\ Dist = |x_1 - x_2| + |y_1 - y_2|$
- $Mdist[(2,6),(7,3)] = |2-7| + |6-3| = 8$

i	x	y	C1	O	Cluster
X1	2	6	3	8	C1
X2	3	4	0	5	C1
X3	3	8	4	9	C1
X4	4	7	4	7	C1
X5	6	2	5	2	O
X6	6	4	3	2	O
X7	7	3	5	0	O
X8	7	4	4	1	O
X9	8	5	6	3	O
X10	7	6	6	3	O

- New Cluster are
- C1: {(2,6), (3,4), (3,8), (4,7)}
- O: {(6, 2), (6, 4), (7, 3), (7, 4), (8, 5), (7,6)}

- C1: {(2,6), (3,4), (3,8), (4,7)}
- O: {(6, 2), (6, 4), (7, 3), (7, 4), (8, 5), (7,6)}
- **Calculate the Total Cost**
- $\boldsymbol{Cost(c,x)} = \sum_i |\boldsymbol{c_i} - \boldsymbol{x_i}|$
- $Current\ Total\ Cost = \{Cost\big((3,4),(2,6)\big) + Cost\big((3,4),(3,8)\big) + Cost\big((3,4),(4,7)\big) + Cost\big((7,3),(6,2)\big) + Cost\big((7,3),(6,4)\big) + Cost\big((7,3),(7,4)\big) + Cost\big((7,3),(8,5)\big) + Cost\big((7,3),(7,6)\big)\}$
- $\boldsymbol{Current\ Total\ Cost} = 3 + 4 + 4 + 2 + 2 + 1 + 3 + 3 = 22$

Step 4

- **Cost of Swapping of medoid C2 with O**
- $S = Current\ Total\ Cost - Previous\ Total\ Cost$
- $S = 22 - 20 = 2 > 0$
- **Hence Swapping C2 with O is not a good Idea.**
- **Final Medoids are C1 =(3, 4) and C2=(7, 4)**
- **Clusters are**
- **C1: {(2,6), (3,4), (3,8), (4,7)}**
- **C2: {(6, 2), (6, 4), (7, 3), (7, 4), (8, 5), (7,6)}**

Solved example 2

https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/

Types of Regression Models

- Regression modeling is a process of determining a relationship between one or more independent variables and one dependent or output variable.
- **Examples:**
  1. Predicting the height of a person given the age of the person.
  2. Predicting the price of the car given the car model, year of manufacturing, mileage, engine capacity, etc.

- Based on the type of functions used to represent the relationship between the dependent or output variable and independent variables, the regression models are categorized into four types. The regression models are,
  1. Simple Linear Regression
  2. Multiple Regression
  3. Polynomial Regression
  4. Logistic Regression

---

1. Simple Linear Regression

- Assume that there is only one independent variable x. If the relationship between x (independent variable) and y (dependent or output variable) is modeled by the relation,

$$y = a + bx$$

- then the regression model is called a linear regression model.

2. Multiple Regression

- Assume that there are multiple independent variables *say* x_1, x_2, ...x_n. If the relationship between independent variables x and dependent or output variable y is modeled by the relation,

$$y=a_0+a_1*x_1+a_2*x_2+.....+a_n*x_n$$

- then the regression model is called a multiple regression model.

3. Polynomial regression

- Assume that there is only one independent variable x. If the relationship between independent variables x and dependent or output variable y is modeled by the relation,

$$y=a_0+a_1*x+a_2*x^2+.....+a_n*x^n$$

- for some positive integer n>1, then we have a polynomial regression.

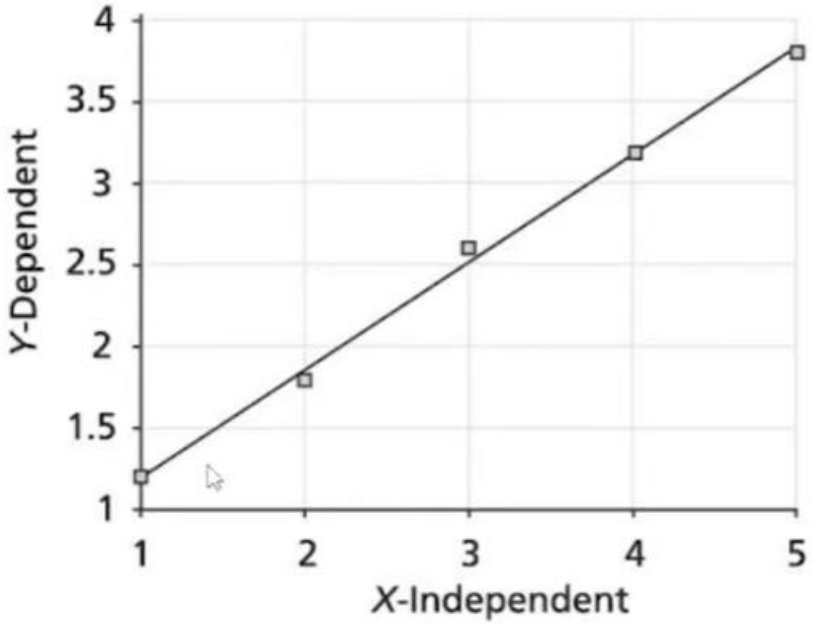
4. Logistic Regression

- Logistic regression is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature.

Linear Regression Algorithm

- Let us consider an example where the five weeks' sales data (in Thousands) is given as shown in Table.
- Apply linear regression technique to predict the 7th and 12th week sales.

x_i (Week)	y_j (Sales in Thousands)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8



- Linear regression equation is given by
- $y = a_0 + a_1 * x + e$
- *where*
- $a_1 = \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{\overline{x^2} - \bar{x}^2}$
- $a_0 = \bar{y} - a_1 * \bar{x}$

x_i (Week)	y_j (Sales in Thousands)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8

- Here, there are 5 items, i.e., i = 1, 2, 3, 4, 5.

	x_i (Week)	y_j (Sales in Thousands)	x_i^2	$x_i * y_j$
	1	1.2	1	1.2
	2	1.8	4	3.6
	3	2.6	9	7.8
	4	3.2	16	12.8
	5	3.8	25	19
Sum	15	12.6	55	44.4
Average	$\bar{x} = 3$	$\bar{y} = 2.52$	$\overline{x^2} = 11$	$\overline{xy} = 8.88$

- $\bar{x} = 3$ $\bar{y} = 2.52$ $\overline{x^2} = 11$ $\overline{xy} = 8.88$

- $a_1 = \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{\overline{x^2} - \bar{x}^2} = \frac{8.88 - 3 * 2.52}{11 - 3^2} = 0.66$
- $a_0 = \bar{y} - a_1 * \bar{x} = 2.52 - 0.66 * 3 = 0.54$
- **Regression equation is**
- $y = a_0 + a_1 * x$
- $y = 0.54 + 0.66 * x$

- **Regression equation is**
- $y = a_0 + a_1 * x$
- $y = 0.54 + 0.66 * x$
- The predicted 7th week sale (when x = 7) is,
- **y = 0.54 + 0.66 x 7 = 5.16**
- the predicted 12th week sale (when x = 12) is,
- **y = 0.54 + 0.66 x 12 = 8.46**

Matrix method

Linear Regression – Solved Example – Matrix Method

- Here, the independent variable X is be given as:
- X^T = [1 2 3 4]
- The dependent variable is given as follows:
- Y^T = [1 3 4 8]
- The data can be given in matrix form as follows:

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix}, \qquad Y = \begin{pmatrix} 1 \\ 3 \\ 4 \\ 8 \end{pmatrix}$$

x_i (Week)	y_j (Sales in Thousands)
1	1
2	3
3	4
4	8

The first column can be used for setting bias.

- The regression is given as:

$$\boldsymbol{a} = ((\boldsymbol{X^T X})^{-1} \boldsymbol{X^T})\boldsymbol{Y}$$

- The computation order of this equation is shown step by step as:

1. Computation of $(X^T X) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 4 & 10 \\ 10 & 30 \end{pmatrix}$

2. Computation of matrix inverse of $(X^T X)^{-1} = \begin{pmatrix} 4 & 10 \\ 10 & 30 \end{pmatrix}^{-1} = \begin{pmatrix} 1.5 & -0.5 \\ -0.5 & 0.2 \end{pmatrix}$

3. Computation of $((X^T X)^{-1} X^T) = \begin{pmatrix} 1.5 & -0.5 \\ -0.5 & 0.2 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0.5 & 0 & -0.5 \\ -0.3 & -0.1 & 0.1 & 0.3 \end{pmatrix}$

4. Finally, $((X^T X)^{-1} X^T) Y = \begin{pmatrix} 1 & 0.5 & 0 & -0.5 \\ -0.3 & -0.1 & 0.1 & 0.3 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \\ 4 \\ 8 \end{pmatrix} = \begin{pmatrix} -1.5 \\ 2.2 \end{pmatrix} \begin{pmatrix} Intercept \\ slope \end{pmatrix}$

- **Regression equation is**
- $\boldsymbol{y = a_0 + a_1 * x}$
- $\boldsymbol{y = -1.5 + 2.2 * x}$
- The predicted 5th week sale (when x = 5) is,
- $y = -1.5 + 2.2 * 5 = \mathbf{9.5}$

Multiple Linear Regression

Multiple Linear Regression – Solved Example

- In linear regression model we have one dependent and one independent variable.
- Multiple regression model involves multiple predictors or independent variables and one dependent variable.
- This is an extension of the linear regression problem.

- The multiple regression of two variables x1 and x2 is given as follows:

$$y = f(x_1, x_2)$$

$$y = a_0 + a_1 x_1 + a_2 x_2$$

- In general, this is given for 'n' independent variables as:

$$y = f(x_1, x_2, \ldots . x_n)$$

$$y = a_0 + a_1 x_1 + a_2 x_2 + \ldots \ldots + a_n x_n + \varepsilon$$

Logistic Regression Algorithm

- Linear regression predicts the numerical response but is not suitable for predicting the categorical variables.
- When categorical variables are involved, it is called classification problem.
- Logistic regression is suitable for binary classification problem.

Logistic Regression – Algorithm & Solved Example

For example, the following scenarios are instances of predicting categorical variables.

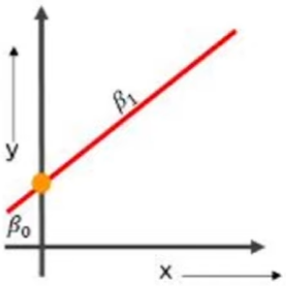
1. Is the mail spam or not spam? The answer is yes or no. Thus, categorical dependent variable is a binary response of yes or no.
2. If the student should be admitted or not is based on entrance examination marks. Here, categorical variable response is admitted or not.
3. The student being pass or fail is based on marks secured.

How Does the Logistic Regression Algorithm Work?

- Consider the following example:
- An organization wants to determine an employee's salary increase based on their performance.
- For this purpose, a linear regression algorithm will help them decide.
- Plotting a regression line by considering the employee's performance as the independent variable, and the salary increase as the dependent variable will make their task easier.

- Now, what if the organization wants to know whether an employee would get a promotion or not based on their performance?
- The above linear graph won't be suitable in this case.
- As such, we clip the line at zero and one, and convert it into a sigmoid curve (S curve).
- Based on the threshold values, the organization can decide whether an employee will get a salary increase or not.

- To understand logistic regression, let's go over the odds of success.
- $Odds\ (\theta) = \frac{Probability\ of\ an\ event\ happening}{Probability\ of\ an\ event\ not\ happening}$
- $Odds\ (\theta) = \frac{p}{1-p}$
- The values of odds range from **zero to ∞** and the values of probability lies between **zero and one**.



- Consider the equation of a straight line:
- $\boldsymbol{y = \beta 0 + \beta 1 * x}$
- Now to predict the odds of success, we take log on odds formula:

$$\log\left(\frac{p(x)}{1-P(x)}\right) = \beta_0 + \beta_1 x$$

- Exponentiating both the sides, we have:

$$e^{\ln(x)} = x$$

$$e^{ln}\left(\frac{p(x)}{1-p(x)}\right) = e^{\beta_0+\beta_1 x}$$

$$\left(\frac{p(x)}{1-p(x)}\right) = e^{\beta_0+\beta_1 x}$$

- $Let\ Y = e^{\beta 0+\beta 1 * x}$
- $Then \quad \frac{p(x)}{1-p(x)} = Y$
- $p(x) = Y(1 - p(x))$
- $p(x) = Y - Y(p(x))$
- $p(x) + Y(p(x)) = Y$
- $p(x)(1+Y) = Y$
- $p(x) = \frac{Y}{1+Y}$

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

The equation of the sigmoid function is:

$$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

- The sigmoid curve obtained from the above equation is as follows:

- The student dataset has entrance mark based on the historic data of those who are selected or not selected.
- Based on the logistic regression, the values of the learnt parameters are $\beta_0 = 1$ *and* $\beta_1 = 8$.
- Assuming marks of x = 60, compute the resultant class.

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

$$\beta_0 + \beta_1 x = 481$$

$$p(x) = \frac{1}{1 + e^{-481}} = 0.44$$

- If we assume the threshold value as 0.5, then it is observed that 0.44 < 0.5, therefore, the candidate with marks 60 is not selected.

- The dataset of pass or fail in an exam of 5 students is given in the table.
- Use logistic regression as classifier to answer the following questions.

1. Calculate the probability of pass for the student who studied 33 hours.
2. At least how many hours student should study that makes he will pass the course with the probability of more than 95%.

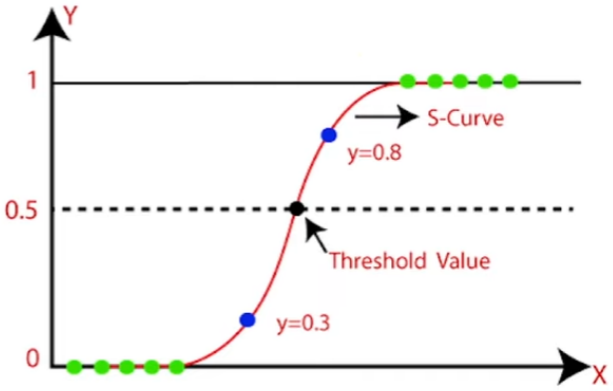
Hours Study	Pass (1) / Fail (0)
29	0
15	0
33	1
28	1
39	1

Assume the model suggested by the optimizer for odds of passing the course is,

$$\log(odds) = -64 + 2 * hours$$

- We use Sigmoid Function in logistic regression

- $s(x) = \frac{1}{1+e^{-x}}$



1. Calculate the probability of pass for the student who studied 33 hours.

Hours Study	Pass (1) / Fail (0)
29	0
15	0
33	1
28	1
39	1

$$s(x) = \frac{1}{1+e^{-x}}$$

- $p = \frac{1}{1+e^{-z}}$
- $z = -64 + 2 * 33 = -64 + 66 = 2$
- $p = \frac{1}{1+e^{-2}} = 0.88$
- That is, if student studies 33 hours, then there is **88% chance** that the student will pass the exam

$$\log(odds) = z = -64 + 2 * hours$$

---

2. At least how many hours student should study that makes he will pass the course with the probability of more than 95%.

Hours Study	Pass (1) / Fail (0)
29	0
15	0
33	1
28	1
39	1

- $p = \frac{1}{1+e^{-z}} = 0.95$
- $0.95 * (1 + e^{-z}) = 1$
- $0.95 * e^{-z} = 1 - 0.95$
- $e^{-z} = \frac{0.05}{0.95} = 0.0526$
- $\ln(e^{-z}) = \ln(0.0526)$

$$\ln(e^x) = x$$

$$-z = ln(0.0526) = -2.94$$

$$z = 2.94$$

- $z = 2.94$
- $\log(odds) = z = -64 + 2 * hours$
- $2.94 = -64 + 2 * hours$
- $2 * hours = 2.94 + 64$
- $2 * hours = 66.94$
- $hours = \frac{66.94}{2}$
- $\boldsymbol{hours = 33.47\ Hours}$

Hours Study	Pass (1) / Fail (0)
29	0
15	0
33	1
28	1
39	1

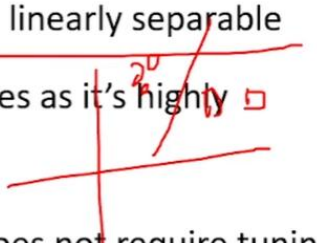
- The student should study **at least 33.47 hours**, so that he will pass the exam with more than 95% probability

Applications

- Linear regression predicts the numerical response but is not suitable for predicting the categorical variables.
- When categorical variables are involved, it is called classification problem.
- Logistic regression is suitable for binary classification problem.
- Here, the output is often a categorical variable.
- The independent variables can be nominal, ordinal, or of interval type.

- Is the mail spam or not spam? The answer is yes or no. Thus, categorical dependent variable is a binary response of yes or no.
- If the student should be admitted or not is based on entrance examination marks. Here, categorical variable response is admitted or not.
- Ecommerce companies can identify buyers if they are likely to purchase a certain product
- Companies can predict whether they will gain or lose money in the next quarter, year, or month based on their current performance

- Logistic regression performs better when the data is linearly separable
- It does not require too many computational resources as it's highly interpretable
- There is no problem scaling the input features—It does not require tuning
- It is easy to implement and train a model using logistic regression
- It gives a measure of how relevant a predictor (coefficient size) is, and its direction of association (positive or negative)



Linear Regression	Logistic Regression
Used to solve regression problems	Used to solve classification problems
The response variables are continuous in nature	The response variable is categorical in nature
It helps estimate the dependent variable when there is a change in the independent variable	It helps to calculate the possibility of a particular event taking place
It is a straight line	It is an S-curve (S = Sigmoid)

Feature Selection

- Machine learning (ML) is a subfield of artificial intelligence that allows computers to learn without being explicitly programmed.
- Usually, machine learning model is built with help of dataset.
- A dataset is usually represented in a tabular form: rows, and columns are features.

Address	Number of Rooms	House Age	owner	price

- In machine learning, feature selection selects the most relevant subset of features from the original feature set by dropping redundant, noisy, and irrelevant features.
- Feature selection is used today in many applications like Object detection, NLP, remote sensing, image retrieval, etc.

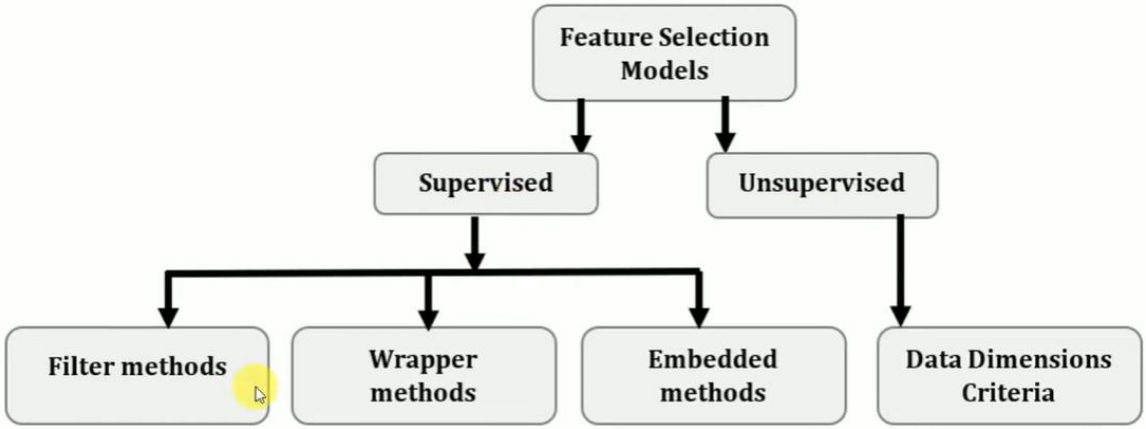
- If there are too many features, our model can may become weak or generate some misleading patterns.
- That happens because, usually, some features aren't correlated with the target variable and represent noise.
- If our model outputs predictions based on such features, its accuracy is likely to be subpar.
- Also, a dataset with many columns slows down the training process.

- For example, the number of rooms and address are relevant for predicting the sale price of a house, but the current owner's name isn't.

Address	Number of Rooms	House Age	owner	price

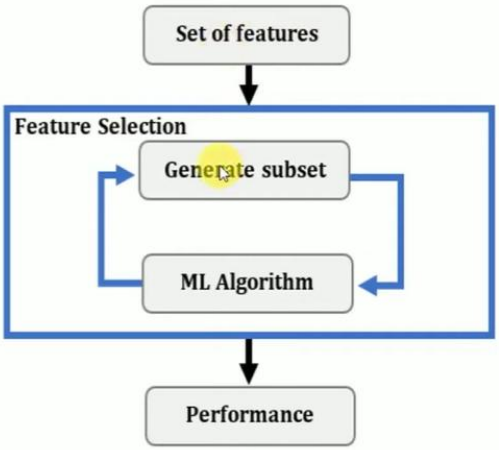
- It can confuse the learning algorithm to let the name affect the sale price prediction, which is likely to lead to wrong results and make the obtained model imprecise.

- There are several methods of doing feature selection:



- **Unsupervised Methods**
- Unsupervised feature selection methods are applied to unlabeled data.
- An unsupervised selection method rates each feature dimension according to a number of factors, including entropy, variance, and the capacity to maintain local similarity.
- **Supervised Methods**
- On the other hand, we use supervised feature selection methods on labeled data.
- They determine the features that are expected to maximize the supervised model's performance.
- Supervised feature selection methods can be split into three primary categories based on the feature selection strategy.

- **Wrapper Methods**
- We use a wrapper method after choosing the ML algorithm to use.
- For each feature subset, we estimate the algorithm's performance by training and evaluating it using only the features in a subset.
- Then, we add or remove features based on the estimate.
- This is an iterative process.



We use a greedy strategy to form feature subsets.

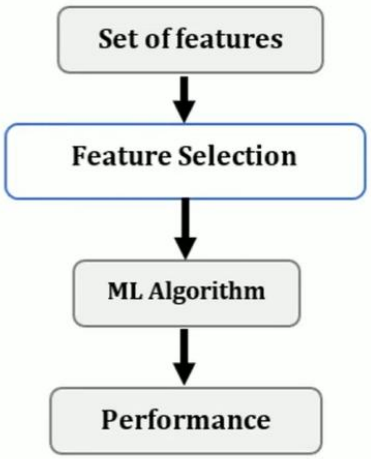
- In **forward wrapper methods**, we start from an empty feature set and add the feature maximizing the performance in each step until no substantial improvement is observed.
- So, if there are **n** features, we build **n** ML models in the first iteration.
- Then, we select the feature corresponding to the model with the best performance.
- In the second iteration, we repeat the process with the remaining **n-1** features.
- We continue like this as long as there's a significant performance improvement between models with which we end successive iterations.

We use a greedy strategy to form feature subsets.

- **Backward methods** work the opposite way.
- They start from the full feature set and remove them one by one.
- Finally, stepwise methods reconsider features.
- So, in each iteration, they can remove a feature previously added as well as add a feature discarded in a previous step.

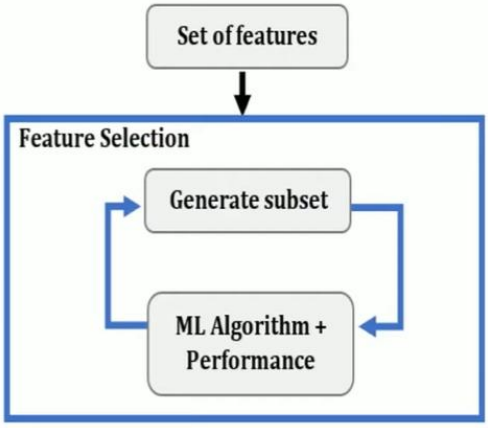
Filter Methods

- Filter methods use statistical tools to select feature subsets based on their relationship with the target.
- These methods remove features with low correlation with the target variable before training the final ML model.
- In doing so, they compute correlation and estimate the strength of the relationship using the, and other statistical tools.
- Chi-Square Test, Information Gain, Fisher's Score, Pearson correlation, ANOVA, variance thresholding



- **Intrinsic (or Embedded) Methods**

- Here selection of feature happens simultaneously with and is performed implicitly by the ML algorithm of our choice.
- During training, some steps of the ML algorithm do feature selection:
- For instance, this is the case with decision trees.
- At each node split, they choose the best feature to split the data by.
- Those choices represent feature selection.



Feature selection methods allow us to:

- Reduce overfitting as less redundant data means less chance to make decisions based on noise;
- Improve accuracy by removing misleading and unimportant data;
- Reduce training time since data with fewer columns mean faster training.
- However, feature selection methods are hard to apply to high-dimensional data.
- The more features we have, the longer it takes for selection to complete.
- Also, there's the risk of overfitting when there aren't enough observations.

Forward Feature Selection Subset Selection Dimensionality Reduction

Subset selection

- In machine learning subset selection or feature selection is the process of selecting a subset of relevant features for use in model construction.
- The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information.
- There are mainly two approaches to subset selection.
  - forward selection and
  - backward selection.

Forward Feature Selection

$x_1\ x_2\ x_3\ t$

We use the following notations:

n	:	number of input variables
$x_1, \ldots, x_n$	:	input variables
F_i	:	a subset of the set of input variables
$E(F_i)$	:	error incurred on the validation sample when only the inputs in F_i are used

1. Set $F_0 = \varnothing$ and $E(F_0) = \infty$.
2. For $i = 0, 1, \ldots,$ repeat the following until $E(F_{i+1}) \geq E(F_i)$:
   (a) For all possible input variables x_j, train the model with the input variables $F_i \cup \{x_j\}$ and calculate $E(F_i \cup \{x_j\})$ on the validation set.
   (b) Choose that input variable x_m that causes the least error $E(F_i \cup \{x_j\})$:

$$m = \arg\min_j E(F_i \cup \{x_j\})$$

   (c) Set $F_{i+1} = F_i \cup \{x_m\}$.
3. The set F_i is outputted as the best subset.

Remarks

1. In this procedure, we stop if adding any feature does not decrease the error E. We may even decide to stop earlier if the decrease in error is too small, where there is a user-defined threshold that depends on the application constraints.
2. This process may be costly because to decrease the dimensions from n to k, we need to train and test the system $n + (n-1) + (n-2) + \ldots + (n-k)$ times, which is $O(n^2)$.

Backward Feature Selection

We use the following notations:

n	:	number of input variables
$x_1, \ldots, x_n$	:	input variables
F_i	:	a subset of the set of input variables
$E(F_i)$	:	error incurred on the validation sample when only the inputs in F_i are used

1. Set $F_0 = \{x_1, \ldots, x_n\}$ and $E(F_0) = \infty$.
2. For $i = 0, 1, \ldots$, repeat the following until $E(F_{i+1}) \geq E(F_i)$:
   (a) For all possible input variables x_j, train the model with the input variables $F_i - \{x_j\}$) and calculate $E(F_i - \{x_j\})$ on the validation set.
   (b) Choose that input variable x_m that causes the least error $E(F_i - \{x_j\})$:

$$m = \arg\min_j E(F_i - \{x_j\})$$

   (c) Set $F_{i+1} = F_i - \{x_m\}$.
3. The set F_i is outputted as the best subset.

How to plot ROC Curve

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.

This curve plots two parameters:

- True Positive Rate
- False Positive Rate

Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90				
2	P	0.80				
3	N	0.70				
4	P	0.60				
5	P	0.55				
6	N	0.54				
7	N	0.53				
8	N	0.51				
9	P	0.50				
10	N	0.40				

P → P
N → P

TP = True Positive

FP = False Positive

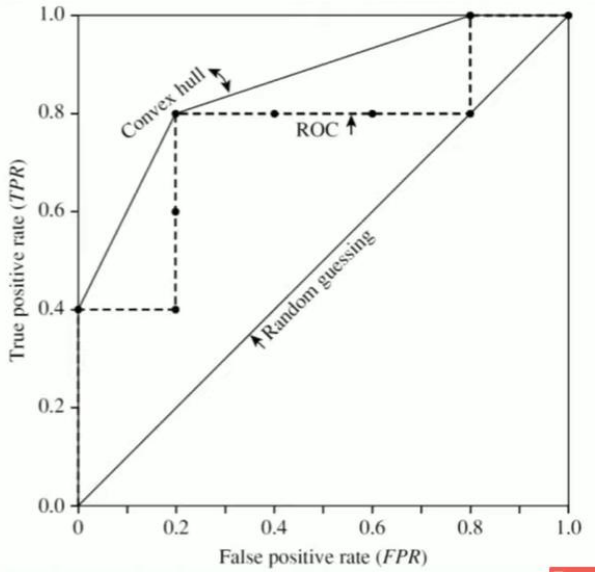
TPR = True Positive Rate

$$TPR = \frac{TP}{P}$$

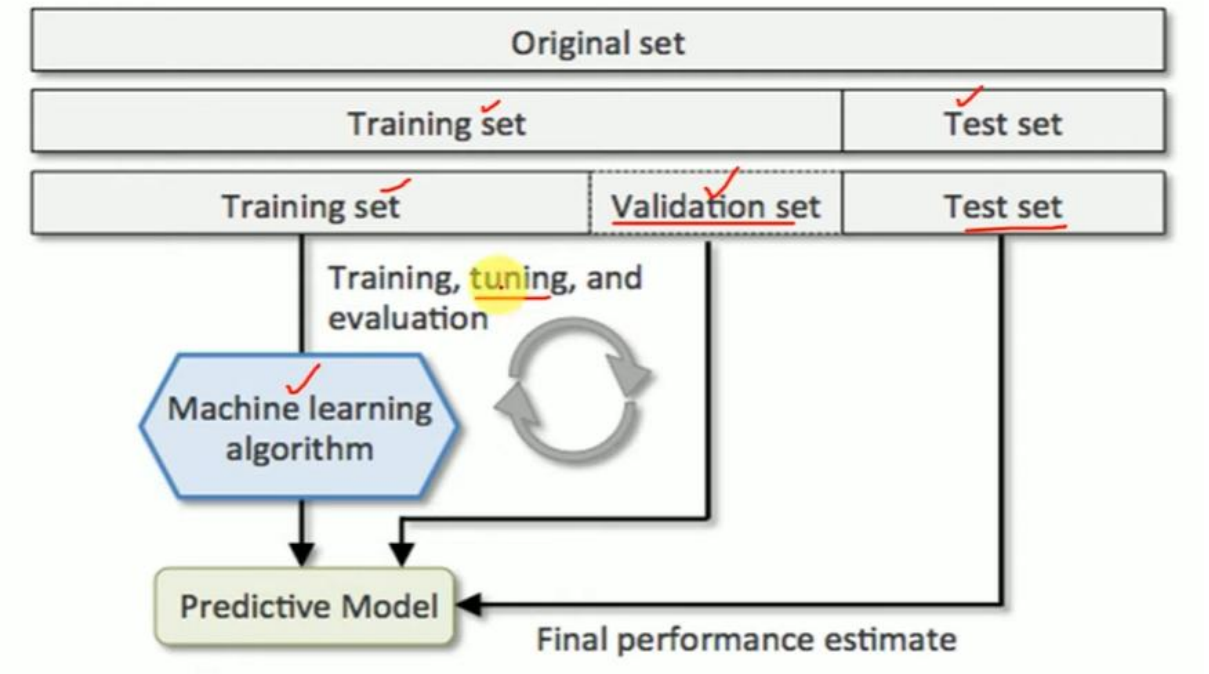
FPR = False Positive Rate

$$FPR = \frac{FP}{N}$$

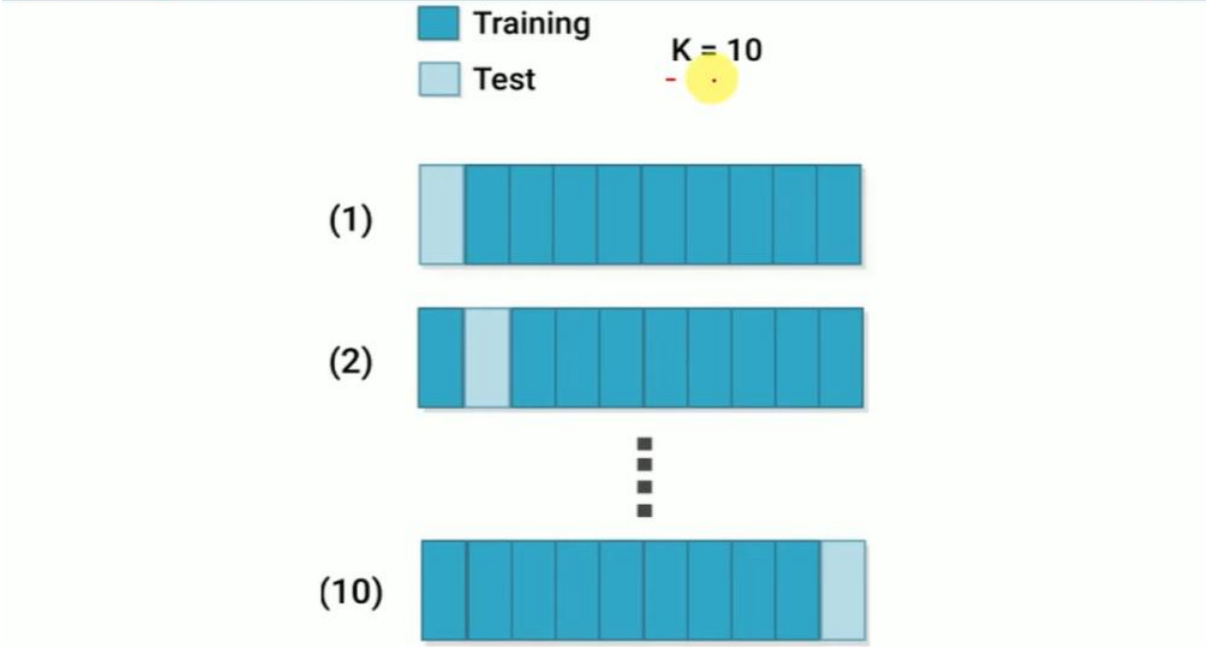
Tuple	Class	Prob	TP	FP	TPR	FPR
1	P	0.90	1	0	0.2	0
2	P	0.80	2	0	0.4	0
3	N	0.70	2	1	0.4	0.2
4	P	0.60	3	1	0.6	0.2
5	P	0.55	4	1	0.8	0.2
6	N	0.54	4	2	0.8	0.4
7	N	0.53	4	3	0.8	0.6
8	N	0.51	4	4	0.8	0.8
9	P	0.50	5	4	1.0	0.8
10	N	0.40	5	5	1.0	1.0

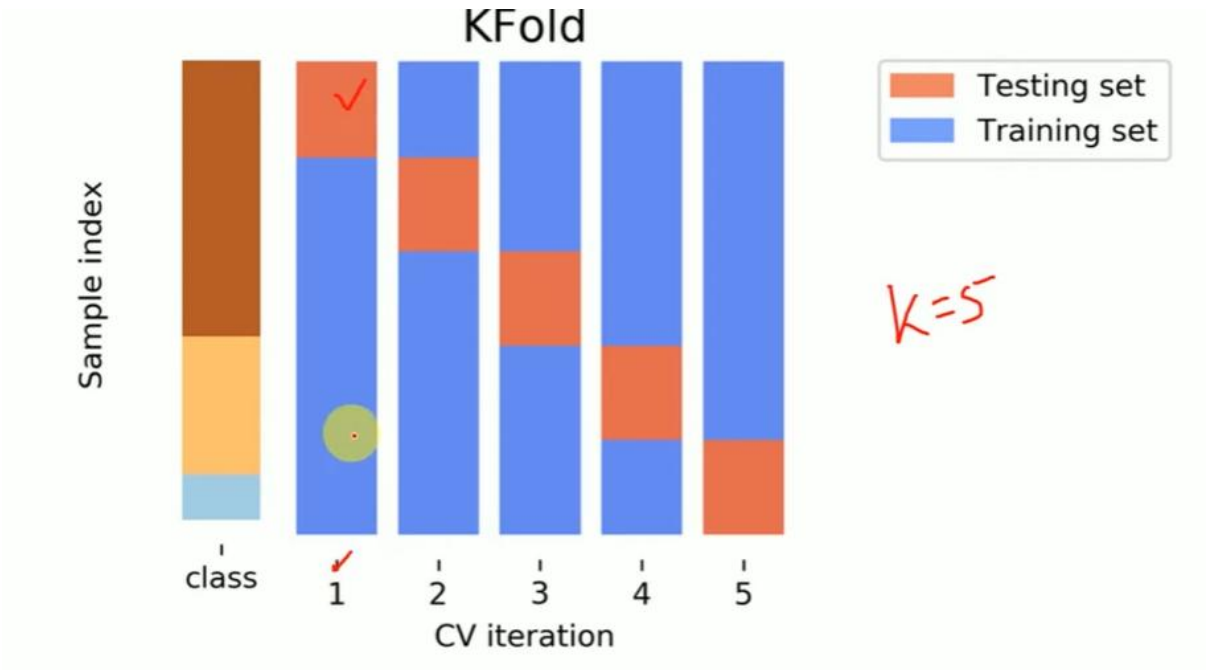


K-Fold Cross Validation, Stratified K-Fold,

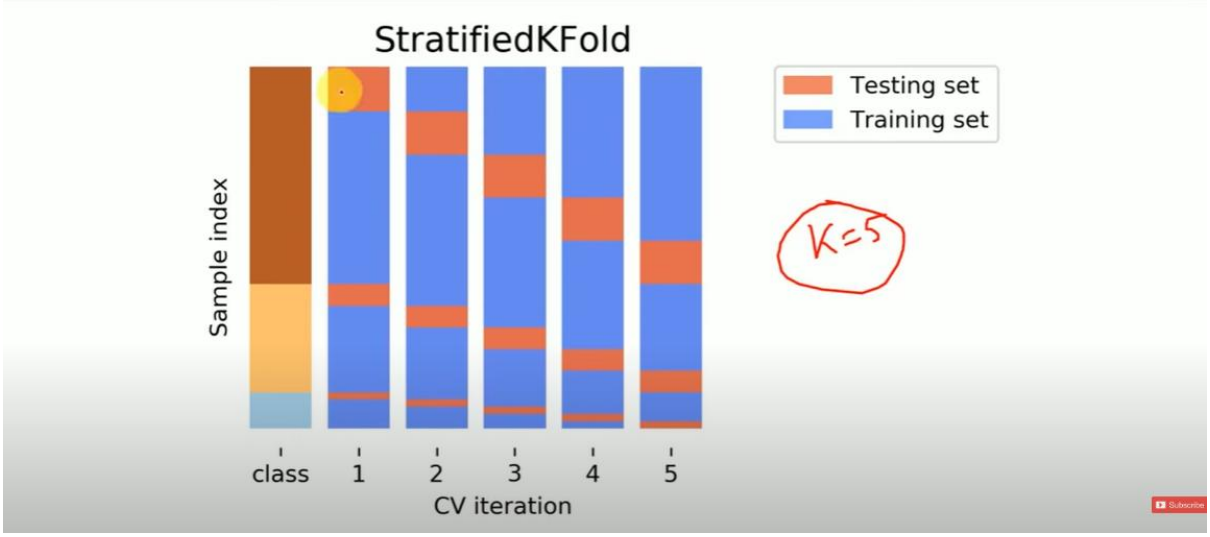


K-Fold Cross Validation in Machine Learning





Stratified K-folds Cross Validation in Machine Learning



Data Smoothing Methods

- The binning method can be used for smoothing the data.
- Mostly data is full of noise.
- Data smoothing is a data pre-processing technique used to remove the noise from the data set.

There are mainly three techniques to do Data Smoothing

1. Smoothing the data by equal frequency bins
2. Smoothing the data by bin means
3. Smoothing the data by bin boundaries

Data Smoothing by Equal Frequency Bins

Example

- Unsorted data for price in dollars
- Before sorting: 8 16, 9, 15, 21, 21, 24, 30, 26, 27, 30, 34
- First of all, sort the data
- After Sorting: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

Sorted Data: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

- **Bin 1: 8, 9, 15, 16**
- **Bin 2: 21, 21, 24, 26,**
- **Bin 3: 27, 30, 30, 34**

Data Smoothing by Bin Means

Sorted Data: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

- **Bin 1: 8, 9, 15, 16**
- **Mean of Bin 1: (8+ 9 + 15 +16 / 4) = 12**
- **Bin 1 = 12, 12, 12, 12**

Sorted Data: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

- **Bin 2: 21, 21, 24, 26,**
- **Mean of Bin 2: (21 + 21 + 24 + 26 / 4) = 23**
- **Bin 2 = 23, 23, 23, 23**

Sorted Data: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

- **Bin 3: 27, 30, 30, 34**
- **Mean of Bin 3: (27 + 30 + 30 + 34 / 4) = 30**
- **Bin 3 = 30, 30, 30, 30**

Data Smoothing by Bin Boundary

Sorted Data: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

- **Before bin Boundary:** Bin 1: 8, 9, 15, 16
- Here, 9 is near to 8, so 9 will be treated as 8. 15 is nearer to 16 and farther away from 8. So, 15 will be treated as 16.
- **After bin Boundary:** Bin 1: 8, 8, 16, 16

Data Smoothing by Bin Boundary

Sorted Data: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

- **Before bin Boundary:** Bin 2: 21, 21, 24, 26,
- **After bin Boundary:** Bin 2: 21, 21, 26, 26,
- **Before bin Boundary:** Bin 3: 27, 30, 30, 34
- **After bin Boundary:** Bin 3: 27, 27, 27, 34

Min-Max Normalization

Data Normalization or Scaling

There are mainly four techniques to do Data Normalization or Scaling

1. Min-Max Normalization
2. Z-Score normalization using mean and standard deviation
3. Z-Score using mean and mean absolute deviation
4. Normalization by decimal scaling

Min- Max Normalization or Scaling

Min-Max Normalization

$$V = \frac{x - min}{max - min}$$

min = 200 and Max= 1000

$$V = \frac{200 - 200}{1000 - 200} = 0$$

$$V = \frac{300 - 200}{1000 - 200} = 0.125$$

$$V = \frac{400 - 200}{1000 - 200} = 0.25$$

$$V = \frac{600 - 200}{1000 - 200} = 0.5$$

$$V = \frac{1000 - 200}{1000 - 200} = 1$$

Data(v)
200
300
400
600
1000

Z-Score Normalization or Scaling

Z-Score Normalization

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

$$\textbf{Mean} = \frac{(200 + 300 + 400 + 600 + 1000)}{5} = 500$$

$$\textbf{Standard Deviation} = \sqrt{\frac{\sum(x_i - \mu)^2}{n}}$$

$$= \sqrt{\frac{(200 - 500)^2 + (300 - 500)^2 + (400 - 500)^2 + (600 - 500)^2 + (1000 - 500)^2}{5}}$$

$$= 282.8$$

Data(v)
200
300
400
600
1000

Z-Score Normalization or Scaling

Z-Score Normalization

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

$$\textbf{Mean} = \frac{(200 + 300 + 400 + 600 + 1000)}{5} = 500$$

$$\textbf{Standard Deviation} = \sqrt{\frac{\sum(x_i - \mu)^2}{n}}$$

$$= \sqrt{\frac{(200 - 500)^2 + (300 - 500)^2 + (400 - 500)^2 + (600 - 500)^2 + (1000 - 500)^2}{5}}$$

$$= 282.8$$

Data(v)
200
300
400
600
1000

Z-Score Normalization or Scaling

Z-Score Normalization

$$z = \frac{(x - \mu)}{\sigma}$$

$$V = \frac{200 - 500}{282.8} = -1.06$$

$$V = \frac{300 - 500}{282.8} = -0.707$$

$$V = \frac{400 - 500}{282.8} = -0.354$$

$$V = \frac{600 - 500}{282.8} = 0.354$$

$$V = \frac{1000 - 500}{282.8} = 1.77$$

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

$\textbf{Mean} = 500$

$\textbf{Standard Deviation} = 282.8$

Data(v)	Normalized Data(v)
200	-1.06
300	-0.707
400	-0.354
600	0.354
1000	1.77

Z-Score Normalization – Mean Absolute Deviation

Z-Score Normalization

$$z = \frac{x - \mu}{A}$$

μ = Mean

A = Mean Absolute Deviation

$$\textbf{Mean} = \frac{(200 + 300 + 400 + 600 + 1000)}{5} = 500$$

$$\textbf{Mean Absolute Deviation} = A = \frac{|200 - 500| + |300 - 500| + \ldots + |1000 - 500|}{5} = 240$$

Data(v)
200
300
400
600
1000

Z-Score Normalization – Mean Absolute Deviation

Z-Score Normalization

$$z = \frac{(x - \mu)}{A}$$

$$V = \frac{200 - 500}{240} = -1.25$$

$$V = \frac{300 - 500}{240} = -0.833$$

$$V = \frac{400 - 500}{240} = -0.417$$

$$V = \frac{600 - 500}{240} = 0.417$$

$$V = \frac{1000 - 500}{240} = 2.08$$

$$z = \frac{x - \mu}{A}$$

μ = Mean

A = Mean Absolute Deviation

***Mean* = 500**

***Mean Absolute Deviation* = 240**

Data(v)	Normalized Data(v)
200	-1.25
300	-0.833
400	-0.417
600	0.4117
1000	2.08

Normalization using Decimal Scaling

Normalization using Decimal Scaling

- **Find Value of j,**
- The smallest integer j such that $Max\left(\frac{v_i}{10^j}\right) \leq 1$

$j = 3$

- $\frac{200}{10^3} \leq 1$
- $\frac{1000}{10^3} \leq 1$

Data(v)
200
300
400
600
1000

Normalization using Decimal Scaling

Normalization using Decimal Scaling

- **Find Value of j,**
- The smallest integer j such that $Max\left(\frac{v_i}{10^j}\right) \leq 1$
- $\frac{200}{10^3} = 0.2$
- $\frac{300}{10^3} = 0.3$
- $\frac{400}{10^3} = 0.4$
- $\frac{600}{10^3} = 0.6$
- $\frac{1000}{10^3} = 1$

Data(v)
200
300
400
600
1000

Partitioning in Data Mining

Data Partitioning into Bins

Partitioning is a unsupervised technique of converting Numerical data to categorical data.

There are mainly three unsupervised techniques to do Data Partitioning into Bins

1. Equal Frequency Partitioning
2. Equal Width Partitioning
3. Clustering base Partitioning

Data Partitioning - Equal Frequency Partitioning

Equal-frequency (Equidepth) partitioning

- Partition the data into Equidepth bins of depth 4:
- Data: 5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215
- Bin 1: 5, 10, 11, 13
- Bin 2: 15, 35, 50, 55
- Bin 3: 72, 92, 204, 215

Data Partitioning - Equal Width Partitioning

- In Equal width, we divide the data in equal widths.
- In order to calculate the width, we have the formula.

$$width = \frac{\max - \min}{N} = \frac{215 - 5}{3} = 70$$

Data:
5, 10, 11, 13,
15, 35, 50,
55, 72, 92,
204, 215

75+70
145
70
215

- We divide the data with three categories:
- Bin 1: 5, 10, 11, 13, 15, 35, 50, 55, 72
- Bin 2: 92
- Bin 3: 204, 215

K-means clustering is done before

How to find the Entropy and Information Gain in Decision Tree

ENTROPY AND INFORMATION GAIN

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

ENTROPY AND INFORMATION GAIN

ENTROPY MEASURES HOMOGENEITY OF EXAMPLES

- Entropy measures the *impurity* of a collection of examples. It depends from the distribution of the random variable *p*.
  - *S* is a collection of training examples $Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$
  - p_+ the proportion of positive examples in *S*
  - p_- the proportion of negative examples in *S*

Examples

Entropy ([14+, 0–]) = – 14/14 log_2 (14/14) – 0 log_2 (0) = 0

Entropy ([9+, 5–]) = – 9/14 log_2 (9/14) – 5/14 log_2 (5/14) = 0.94

Entropy ([7+, 7–]) = – 7/14 log_2 (7/14) – 7/14 log_2 (7/14) = 1/2 + 1/2 = 1

INFORMATION GAIN MEASURES THE EXPECTED REDUCTION IN ENTROPY

- Given entropy as a measure of the impurity in a collection of training examples, the ***information gain,*** is simply the expected reduction in entropy caused by partitioning the examples according to an attribute.
- More precisely, the information gain, ***Gain(S, A)*** of ***an*** attribute **A,** relative to a collection of examples ***S,*** is defined as,

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- where ***Values(A)*** is the set of all possible values for attribute A, and S_v, is the subset of S for which attribute A has value v (i.e., $S_v = \{s \in S | A(s) = v\}$)

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+, 5-] \\
S_{Weak} &\leftarrow [6+, 2-] \\
S_{Strong} &\leftarrow [3+, 3-] \\
Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) \\
&\quad - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}$$

- Given

$$p_1 = 0.1,\ p_2 = 0.2,\ p_3 = 0.3\ and\ p_4 = 0.4$$

- Find the Entropy ?

- $Entropy = -\sum_{i=1}^{n} p_i \log_2(p_i)$

$p_1 = 0.1,\ p_2 = 0.2,\ p_3 = 0.3\ and\ p_4 = 0.4$

- $Entropy = -p_1 * \log_2(p_1) - p_2 * \log_2(p_2) - p_3 * \log_2(p_3) - p_4 * \log_2(p_4)$
- $Entropy = -0.1 * \log_2(0.1) - 0.2 * \log_2(0.2) - 0.3 * \log_2(0.3) - 0.4 * \log_2(0.4)$
- $Entropy = -0.1 * (-3.322) - 0.2 * (-2.322) - 0.3 * (-1.736) - 0.4 * (-1.322)$
- $Entropy = 0.3322 + 0.4644 + 0.5208 + 0.5288$
- $Entropy = 1.8462$

$$\log_2(0.1) = \frac{\log(0.1)}{\log(2)} = \frac{-1}{0.3010} = -3.322$$

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribute: Outlook

$Values\ (Outlook) = Sunny, Overcast, Rain$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$

$S_{Sunny} \leftarrow [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5} log_2 \frac{2}{5} - \frac{3}{5} log_2 \frac{3}{5} = 0.971$

$S_{Overcast} \leftarrow [4+, 0-]$ $\quad Entropy(S_{Overcast}) = -\frac{4}{4} log_2 \frac{4}{4} - \frac{0}{4} log_2 \frac{0}{4} = 0$

$S_{Rain} \leftarrow [3+, 2-]$ $\quad Entropy(S_{Rain}) = -\frac{3}{5} log_2 \frac{3}{5} - \frac{2}{5} log_2 \frac{2}{5} = 0.971$

$$Gain\ (S, Outlook) = Entropy(S) - \sum_{v \in \{Sunny, Overcast, Rain\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Outlook) = Entropy(S) - \frac{5}{14} Entropy(S_{Sunny}) - \frac{4}{14} Entropy(S_{Overcast}) - \frac{5}{14} Entropy(S_{Rain})$$

$$Gain(S, Outlook) = 0.94 - \frac{5}{14} 0.971 - \frac{4}{14} 0 - \frac{5}{14} 0.971 = 0.2464$$

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$

$S_{Hot} \leftarrow [2+, 2-]$ $\quad Entropy(S_{Hot}) = -\frac{2}{4} log_2 \frac{2}{4} - \frac{2}{4} log_2 \frac{2}{4} = 1.0$

$S_{Mild} \leftarrow [4+, 2-]$ $\quad Entropy(S_{Mild}) = -\frac{4}{6} log_2 \frac{4}{6} - \frac{2}{6} log_2 \frac{2}{6} = 0.9183$

$S_{Cool} \leftarrow [3+, 1-]$ $\quad Entropy(S_{Cool}) = -\frac{3}{4} log_2 \frac{3}{4} - \frac{1}{4} log_2 \frac{1}{4} = 0.8113$

$$Gain\ (S, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Temp) = Entropy(S) - \frac{4}{14} Entropy(S_{Hot}) - \frac{6}{14} Entropy(S_{Mild}) - \frac{4}{14} Entropy(S_{Cool})$$

$$Gain(S, Temp) = 0.94 - \frac{4}{14} 1.0 - \frac{6}{14} 0.9183 - \frac{4}{14} 0.8113 = 0.028$$

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribute: H

$Values\ (Humidity) = High, Normal$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$

$S_{High} \leftarrow [3+, 4-]$ $\quad Entropy(S_{High}) = -\frac{3}{7} log_2 \frac{3}{7} - \frac{4}{7} log_2 \frac{4}{7} = 0.9852$

$S_{Normal} \leftarrow [6+, 1-]$ $\quad Entropy(S_{Normal}) = -\frac{6}{7} log_2 \frac{6}{7} - \frac{1}{7} log_2 \frac{1}{7} = 0.5916$

$$Gain\ (S, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Humidity) = Entropy(S) - \frac{7}{14} Entropy(S_{High}) - \frac{7}{14} Entropy(S_{Normal})$$

$$Gain(S, Humidity) = 0.94 - \frac{7}{14} 0.9852 - \frac{7}{14} 0.5916 = 0.1516$$

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribute: Wind

$Values\ (Wind) = Strong, Weak$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$

$S_{Strong} \leftarrow [3+, 3-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [6+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{6}{8} log_2 \frac{6}{8} - \frac{2}{8} log_2 \frac{2}{8} = 0.8113$

$$Gain\ (S, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Wind) = Entropy(S) - \frac{6}{14} Entropy(S_{Strong}) - \frac{8}{14} Entropy(S_{Weak})$$

$$Gain(S, Wind) = 0.94 - \frac{6}{14}\ 1.0 - \frac{8}{14}\ 0.8113 = 0.0478$$

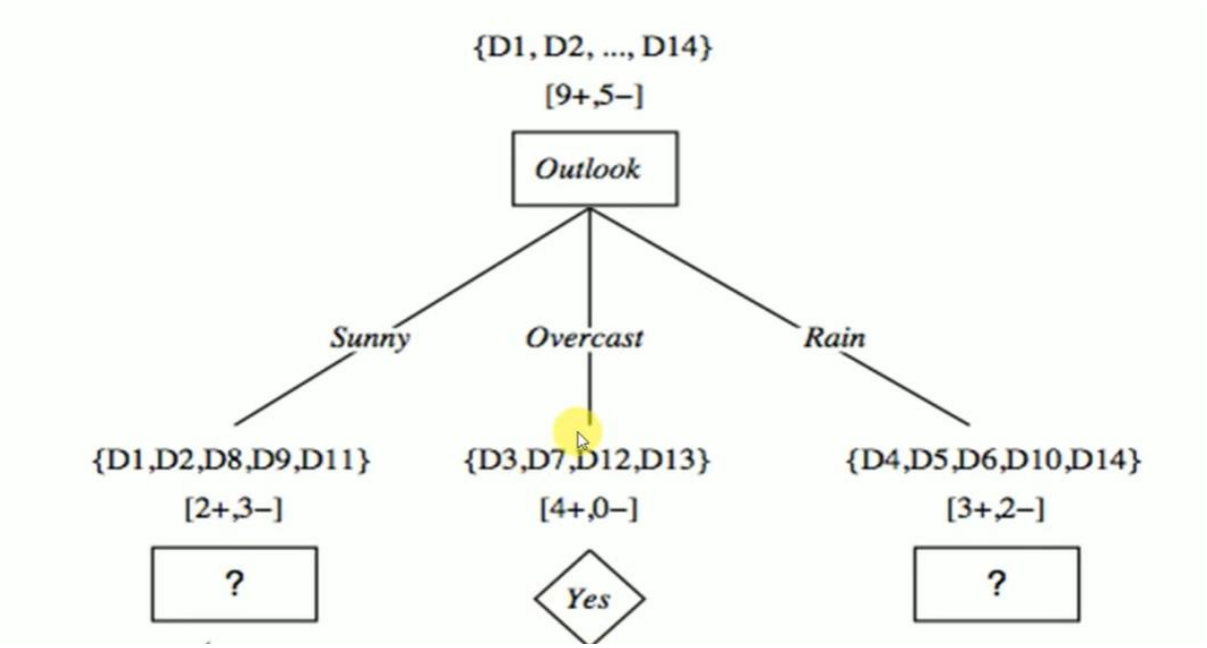
Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

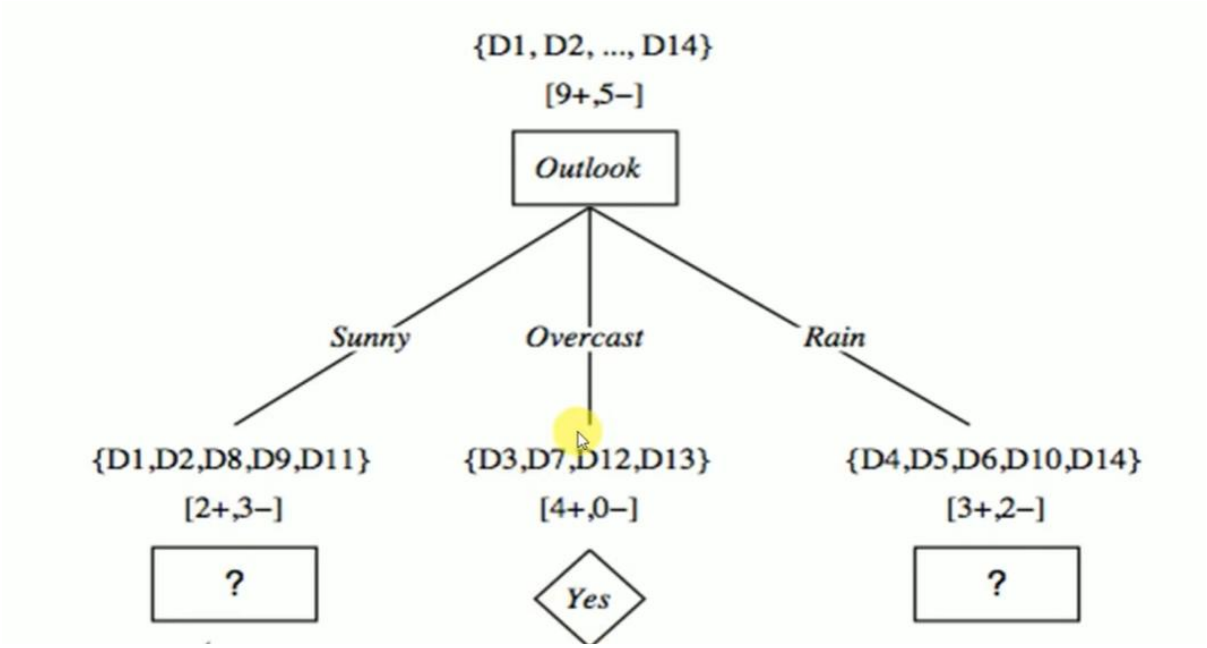
$$Gain(S, Outlook) = 0.2464$$

$$Gain(S, Temp) = 0.0289$$

$$Gain(S, Humidity) = 0.1516$$

$$Gain(S, Wind) = 0.0478$$


{D1, D2, ..., D14}
[9+,5−]
Outlook
Sunny
Overcast
Rain
{D1,D2,D8,D9,D11}
[2+,3−]
?
{D3,D7,D12,D13}
[4+,0−]
Yes
{D4,D5,D6,D10,D14}
[3+,2−]
?


{D1, D2, ..., D14}
[9+,5−]
Outlook
Sunny
Overcast
Rain
{D1,D2,D8,D9,D11}
[2+,3−]
?
{D3,D7,D12,D13}
[4+,0−]
Yes
{D4,D5,D6,D10,D14}
[3+,2−]
?

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 2-]$ $\quad Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Mild}) = 1.0$

$S_{Cool} \leftarrow [1+, 0-]$ $\quad Entropy(S_{Cool}) = 0.0$

$$Gain\ (S_{Sunny}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Temp) = Entropy(S) - \frac{2}{5}Entropy(S_{Hot}) - \frac{2}{5}Entropy(S_{Mild}) - \frac{1}{5}Entropy(S_{Cool})$$

$$Gain(S_{sunny}, Temp) = 0.97 - \frac{2}{5}0.0 - \frac{2}{5}1 - \frac{1}{5}0.0 = 0.570$$

Day	Temp	Humidity	Wind	Play Tennis
Dl	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
Dl1	Mild	Normal	Strong	Yes

Attribute: Humidity

$Values\ (Humidity) = High, Normal$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{high} \leftarrow [0+, 3-]$ $\quad Entropy(S_{High}) = 0.0$

$S_{Normal} \leftarrow [2+, 0-]$ $\quad Entropy(S_{Normal}) = 0.0$

$$Gain\ (S_{Sunny}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Humidity) = Entropy(S) - \frac{3}{5}Entropy(S_{High}) - \frac{2}{5}Entropy(S_{Normal})$$

Day	Temp	Humidity	Wind	Play Tennis
Dl	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
Dl1	Mild	Normal	Strong	Yes

Attribute: Wind

$Values\ (Wind) = Strong, Weak$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Strong} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [1+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$

$$Gain\ (S_{Sunny}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

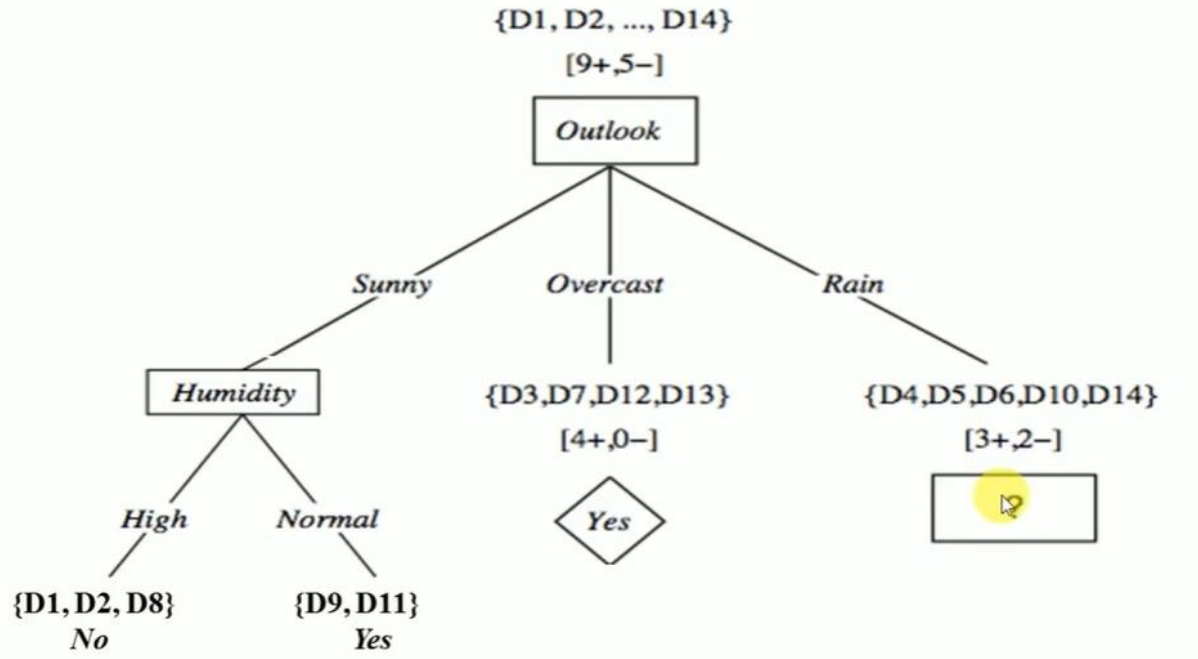
$$Gain(S_{sunny}, Wind) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}0.918 = 0.0192$$

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

$$Gain(S_{sunny}, Temp) = 0.570$$

$$Gain(S_{sunny}, Humidity) = 0.97$$

$$Gain(S_{sunny}, Wind) = 0.0192$$



Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Temp

$$Values\ (Temp) = Hot, Mild, Cool$$

$$S_{Rain} = [3+, 2-] \qquad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$$

$$S_{Hot} \leftarrow [0+, 0-] \qquad Entropy(S_{Hot}) = 0.0$$

$$S_{Mild} \leftarrow [2+, 1-] \qquad Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$$

$$S_{Cool} \leftarrow [1+, 1-] \qquad Entropy(S_{Cool}) = 1.0$$

$$Gain\ (S_{Rain}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Temp) = Entropy(S) - \frac{0}{5}Entropy(S_{Hot}) - \frac{3}{5}Entropy(S_{Mild}) - \frac{2}{5}Entropy(S_{Cool})$$

$$Gain(S_{Rain}, Temp) = 0.97 - \frac{0}{5}0.0 - \frac{3}{5}0.918 - \frac{2}{5}1.0 = 0.0192$$

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
Dl0	Mild	Normal	Weak	Yes
Dl4	Mild	High	Strong	No

Attribute: Humidity

$$Values\ (Humidity) = High, Normal$$

$$S_{Rain} = [3+, 2-] \qquad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$$

$$S_{High} \leftarrow [1+, 1-] \qquad Entropy(S_{High}) = 1.0$$

$$S_{Normal} \leftarrow [2+, 1-] \qquad Entropy(S_{Normal}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$$

$$Gain\ (S_{Rain}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Humidity) = Entropy(S) - \frac{2}{5}Entropy(S_{High}) - \frac{3}{5}Entropy(S_{Normal})$$

$$Gain(S_{Rain}, Humidity) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}0.918 = 0.0192$$

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
Dl0	Mild	Normal	Weak	Yes
Dl4	Mild	High	Strong	No

Attribute: Wind

$$Values\ (wind) = Strong, Weak$$

$$S_{Rain} = [3+, 2-] \qquad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$$

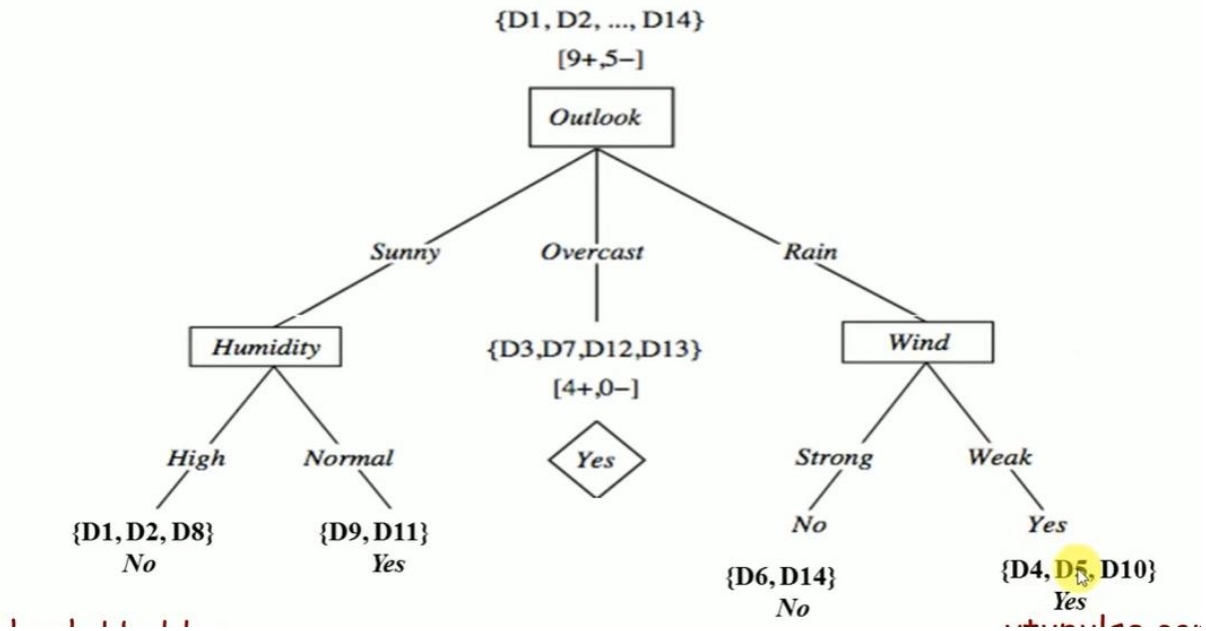
$$S_{Strong} \leftarrow [0+, 2-] \qquad Entropy(S_{Strong}) = 0.0$$

$$S_{Weak} \leftarrow [3+, 0-] \qquad Entropy(S_{weak}) = 0.0$$

$$Gain\ (S_{Rain}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

$$Gain(S_{Rain}, Wind) = 0.97 - \frac{2}{5}0.0 - \frac{3}{5}0.0 = 0.97$$



Decision Tree Algorithm – ID3 Solved Example

1. What is the entropy of this collection of training examples with respect to the target function classification?
2. What is the information gain of *a1* and *a2* relative to these training examples?
3. Draw decision tree for the given dataset.

Instance	Classification	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

Instance	Classification	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

Attribute: a1

$Values\ (a1) = T, F$

$S = [3+, 3-]$ $\quad Entropy(S) = 1.0$

$S_T = [2+, 1-]$ $\quad Entropy(S_T) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$S_F \leftarrow [1+, 2-]$ $\quad Entropy(S_F) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$

Example - 2
Decision Tree Algorithm – ID3
Solved Example

$$Gain\ (S, a1) = Entropy(S) - \sum_{v \in \{T,F\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, a1) = Entropy(S) - \frac{3}{6}Entropy(S_T) - \frac{3}{6}Entropy(S_F)$$

$$Gain(S, a1) = 1.0 - \frac{3}{6} * 0.9183 - \frac{3}{6} * 0.9183 = 0.0817$$

Instance	Classification	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

Attribute: a2

$Values\ (a2) = T, F$

$S = [3+, 3-]$ $\quad Entropy(S) = 1.0$

$S_T = [2+, 2-]$ $\quad Entropy(S_T) = 1.0$

$S_F \leftarrow [1+, 1-]$ $\quad Entropy(S_F) = 1.0$

Example - 2
Decision Tree Algorithm – ID3
Solved Example

$$Gain\ (S, a2) = Entropy(S) - \sum_{v \in \{T,F\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, a2) = Entropy(S) - \frac{4}{6}Entropy(S_T) - \frac{2}{6}Entropy(S_F)$$

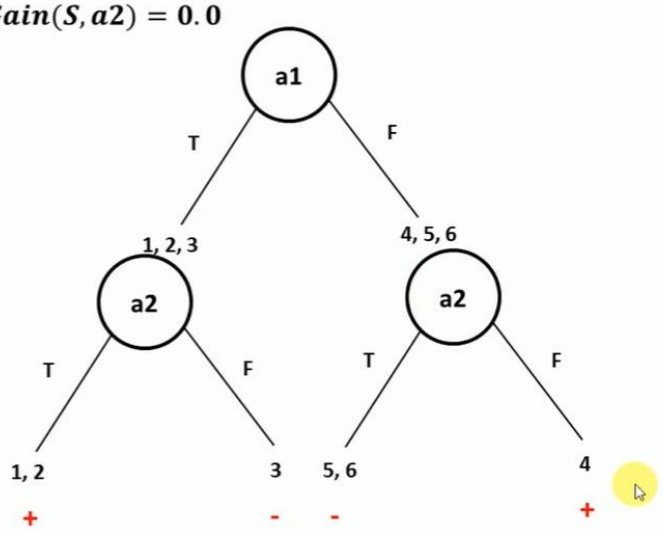
$$Gain(S, a2) = 1.0 - \frac{4}{6} * 1.0 - \frac{2}{6} * 1.0 = 0.0$$

Instance	Classification	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

Example - 2
Decision Tree Algorithm – ID3
Solved Example

$Gain(S, a1) = 0.0817$ – *Maximum Gain*

$Gain(S, a2) = 0.0$



Optics Clustering Algorithm

Ordering points to identify clustering structure

Optics

- DBSCAN and Optics Algorithms are ***density based*** algorithms.
- Unlike K means or K medoid, who find clusters in ***spherical shape***, density-based clustering methods have been developed to discover clusters with arbitrary shape.
- These algorithms decides clusters based on density of data objects.
- They try to locate clusters as dense area and non dense area as noise.
- DBSCAN and Optics algorithms use two important parameter ***Minimum Points(MinPts) and Threshold value Eps (ε).***

- OPTICS produces ***a set or ordering*** of density-based clusters.
- It can constructs different clusters simultaneously.
- The objects should be processed in a specific order.
- This order selects an object that is density-reachable with respect to the lowest value so that clusters with higher density (lower) will be finished first.
- For each object, ***Core-distance and Reachability-distance*** is stored.

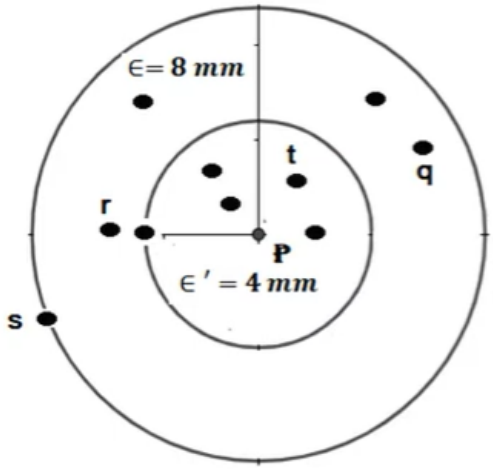
Core-distance of an object:

The core-distance of an object p is the smallest value that makes p a core object.

If p is not a core object, the core distance of p is undefined.

Reachability-distance of an object: The reachability-distance of an object q with respect to another object p is the greater value of the core-distance of p and the Euclidean distance between p and q.

If p is not a core object, the reachability-distance between p and q is undefined.



MinPts=6 Eps(ε) =8mm

Dist(p, q)=7mm

Dist(p, r)=5mm

Dist(p, s)=8mm

Dist(p, t)=3mm

P is core object. As no. of objects are more than equal to 6 within a radius of 8mm.

Core distance of p (ε') =4mm. As p have 6 objects (including p) at distance of 4mm.

Reachability-distance of an object (q, r, s and t)

Max(core distance of p, dist(p, q))= max(4,7)=7

Max(core distance of p, dist(p, r))= max(4,5)=5

Max(core distance of p, dist(p, s))= max(4,8)=8

Max(core distance of p, dist(p, t))= max(4,3)=4

Regularization Lasso vs Ridge vs Elastic Net Overfitting Underfitting Bias & Variance

Regularization in Machine Learning

- While developing machine learning models you must have encountered a situation in which the **training accuracy** of the model is **high** but the **validation accuracy** or the **testing accuracy** is **too low**.
- This is the case which is popularly known as **overfitting** in the domain of machine learning.

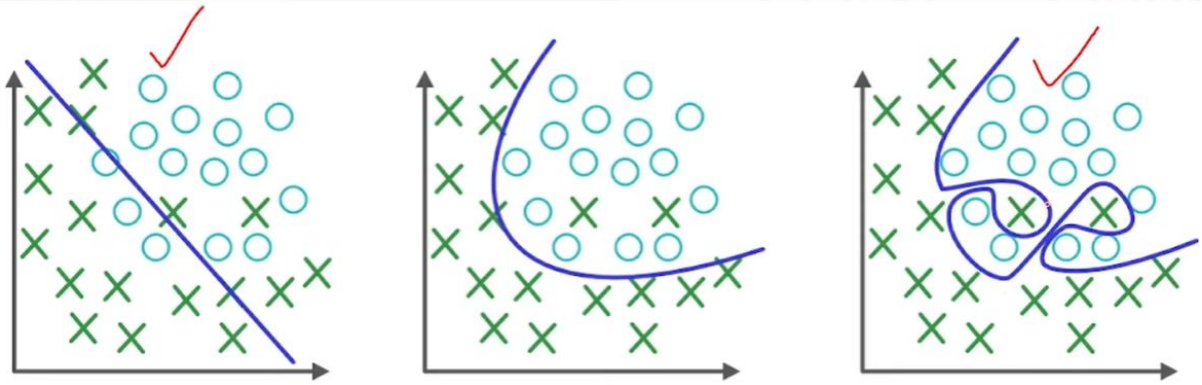
Regularization in Machine Learning

What are Overfitting?

- Overfitting is a phenomenon that occurs when a Machine Learning model is constrained to the training set and not able to perform well on unseen data.
- That is when our model learns the noise in the training data as well.
- This is the case when our model memorizes the training data instead of learning the patterns in it.

What are Underfitting?

- In the case of the underfitting model is unable to perform well even on the training data hence we cannot expect it to perform well on the validation data.
- This is the case when we are supposed to increase the complexity of the model or add more features to the feature set.



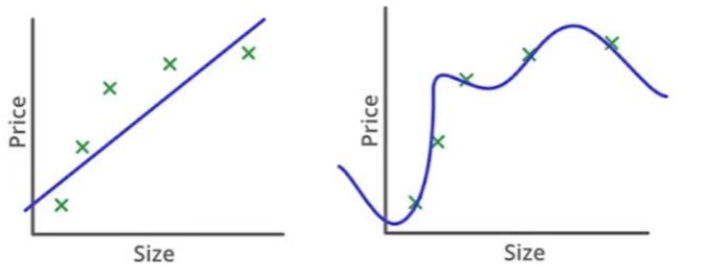
Under-fitting
(too simple to explain the variance)

Appropirate-fitting

Over-fitting
(forcefitting--too good to be true)

What are Bias and Variance?

- Bias refers to the errors which occur when we try to fit a statistical model on real-world data which does not fit perfectly well on some mathematical model.
- If we use a way too simplistic a model to fit the data then we are more probably face the situation of High Bias which refers to the case when the model is unable to learn the patterns in the data at hand and hence performs poorly.
- Variance implies the error value that occurs when we try to make predictions by using data that is not previously seen by the model.
- There is a situation known as high variance that occurs when the model learns noise that is present in the data.



$\theta_0 + \theta_1 x$

High bais (underfit)

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance (overfit)

- **Finding a proper balance between the two that is also known as the Bias Variance Tradeoff can help us prune the model from getting overfitted to the training data.**

- Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting.

The commonly used regularization techniques are :

- Lasso Regularization – L1 Regularization
- Ridge Regularization – L2 Regularization
- Elastic Net Regularization – L1 and L2 Regularization

Lasso Regression

- A regression model which uses the L1 Regularization technique is called LASSO (Least Absolute Shrinkage and Selection Operator) regression.
- Lasso Regression adds the "absolute value of magnitude" of the coefficient as a penalty term to the loss function(L).
- Lasso regression also helps us achieve feature selection by penalizing the weights to approximately equal to zero if that feature does not serve any purpose in the model.

Lasso Regression

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{m}|w_i|$$

- *where,*
- $\boldsymbol{m}$ – *Number of Features*
- $\boldsymbol{n}$ – *Number of Examples*
- $\boldsymbol{y_i}$ – *Actual Target Value*
- $\boldsymbol{\hat{y}_i}$ – *Predicted Target Value*

Ridge Regression

- A regression model that uses the **L2 regularization** technique is called **Ridge regression**.
- **Ridge regression** adds the "*squared magnitude*" of the coefficient as a penalty term to the loss function(L).

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^{m} w_i^2$$

Elastic Net Regression

- This model is a combination of L1 as well as L2 regularization.
- That implies that we add the absolute norm of the weights as well as the squared measure of the weights.
- With the help of an extra hyperparameter that controls the ratio of the L1 and L2 regularization.

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \left((1-\alpha)\sum_{i=1}^{m}|w_i| + \alpha \sum_{i=1}^{m} w_i^2\right)$$

Z-Score based Outlier or Anomaly detection and Removal

Z-Score based Outlier or Anomaly detection

- Outliers are the values in dataset which standouts from the rest of the data.
- The outliers can be a result of error in reading, fault in the system, manual error or misreading.
- Types of outlier detection methods
  1. IQR - Interquartile Range
  2. Z-Score

- Suppose we have a dataset of daily sales revenue for a retail store over the past 30 days:
- [100, 150, 120, 125, 140, 130, 110, 135, 130, 150, 140, 100, 95, 80, 120, 125, 130, 100, 140, 135, 130, 145, 110, 120, 130, 135, 140, 125, 130, 120]
- We want to identify any days where the sales revenue is significantly different from the other days, which may indicate an anomaly or outlier.

Step 1: Calculate mean and standard deviation

- We first calculate the mean and standard deviation of the dataset:
- [100, 150, 120, 125, 140, 130, 110, 135, 130, 150, 140, 100, 95, 80, 120, 125, 130, 100, 140, 135, 130, 145, 110, 120, 130, 135, 140, 125, 130, 120]
- $Mean: \mu = \frac{\sum_{i=1}^{30} x_i}{n} = \frac{100+150+120+\cdots+130+120}{30} = 123.5$
- $Standard\ deviation:$
- $\sigma = \sqrt{\frac{\sum_{i=1}^{30}(x_i-\mu)^2}{n-1}} = \sqrt{\frac{(100-123.5)^2+(150-123.5)^2+\cdots+(120-123.5)^2}{30-1}} = 20.0$

- **Step 2:** Calculate z-scores
- We then calculate the z-score for each data point, which represents how many standard deviations away from the mean the data point is:
- $z = \frac{x-\mu}{\sigma}$
- $z_1 = \frac{100-123.5}{20} = -1.17$
- [-1.17, 0.14, -0.28, -0.16, 0.56, 0.08, -0.89, 0.32, 0.08, 0.14, 0.56, -1.17, -1.45, -2.22, -0.28, -0.16, 0.08, -1.17, 0.56, 0.32, 0.08, 0.86, -0.89, -0.28, 0.08, 0.32, 0.56, -0.16, 0.08, -0.28]

- **Step 3:** Set a threshold
- We set a threshold for what we consider an anomaly.
- Usually z-score =3 is considered as a cut-off value to set the limit. which captures 99.7% of the data in a normal distribution
- Therefore, any z-score greater than +3 or less than -3 is considered as outlier which is pretty much similar to standard deviation method.

- **Step 4:** Identify anomalies
- [-1.17, 0.14, -0.28, -0.16, 0.56, 0.08, -0.89, 0.32, 0.08, 0.14, 0.56, -1.17, -1.45, -2.22, -0.28, -0.16, 0.08, -1.17, 0.56, 0.32, 0.08, 0.86, -0.89, -0.28, 0.08, 0.32, 0.56, -0.16, 0.08, -0.28]

No Outliers found in the given data

How to build a decision Tree for Boolean Function

(a) $A \wedge \neg B$

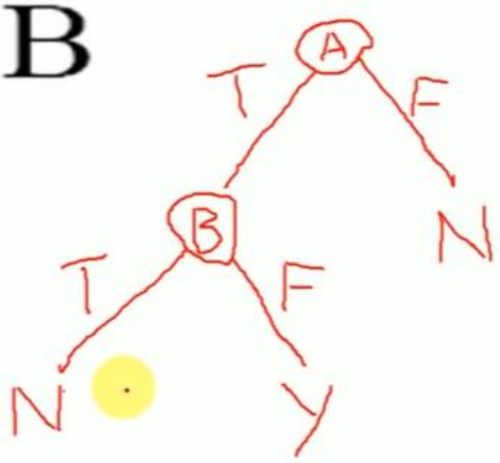
(b) $A \vee [B \wedge C]$

(c) A XOR B

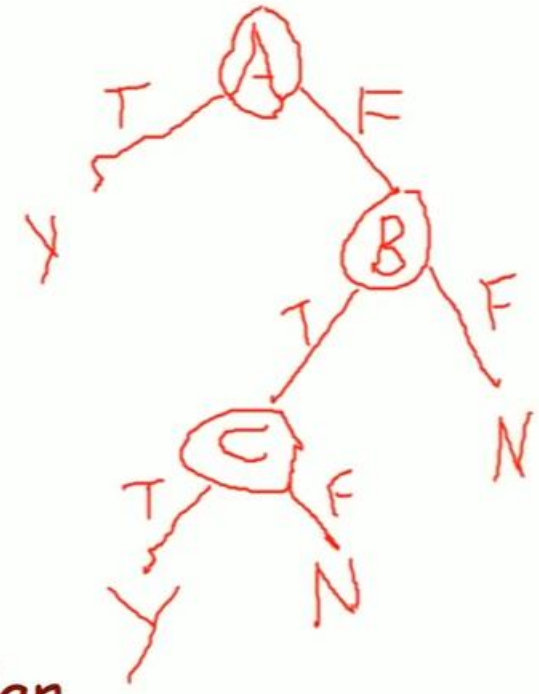
(d) $[A \wedge B] \vee [C \wedge D]$

- Every Variable in Boolean function such as A, B, C etc. has two possibilities that is True and False
- Every Boolean function is either True or False
- If the Boolean function is true we write YES (Y)
- If the Boolean function is False we write NO (N)

(a) $A \wedge \neg B$

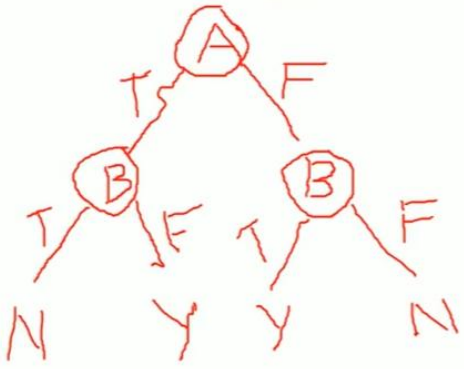


(b) $A \vee [B \wedge C]$

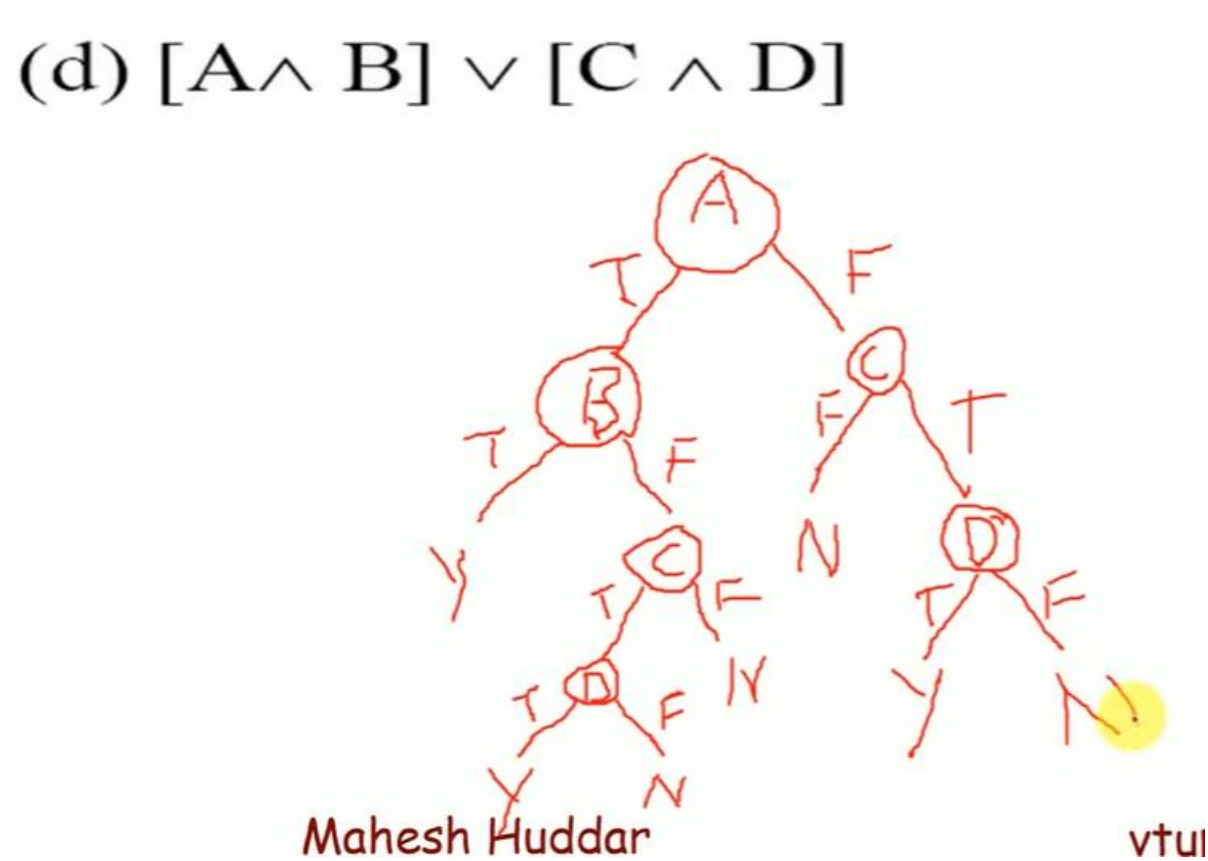


Mahesh Huddar

(c) A XOR B $= (A \wedge \neg B) \vee (\neg A \wedge B)$



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0



Linear Discriminant Analysis

⇒ Compute the Linear Discriminant Projection for the 2D dataset.

$X_1 = (x_1, x_2) = \{(4,1), (2,4), (2,3), (3,6), (4,4)\}$

$X_2 = (x_1, x_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$

Sol:

Step1: Compute mean of Class1 & Class2

$$SW = S1 + S2$$

$$\mu_1 = \left\{ \frac{4+2+2+3+4}{5}, \frac{1+4+3+6+4}{5} \right\}$$

$$\mu_1 = \{3, 3.6\}$$

$$S1 = \Sigma (x - \mu_1)(x - \mu_1)^T$$

$$(x - \mu_1) = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ -2.6 \end{bmatrix} \begin{bmatrix} 1 & -2.6 \end{bmatrix} = \begin{bmatrix} 1 & -2.6 \\ -2.6 & 6.76 \end{bmatrix}$$

$$\begin{bmatrix} -1 \\ 0.4 \end{bmatrix} \begin{bmatrix} -1 & 0.4 \end{bmatrix} = \begin{bmatrix} 1 & -0.4 \\ -0.4 & 0.16 \end{bmatrix}$$

$$\begin{bmatrix} -1 \\ -0.6 \end{bmatrix} \begin{bmatrix} -1 & -0.6 \end{bmatrix} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 0.36 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 2.4 \end{bmatrix} \begin{bmatrix} 0 & 2.4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 5.76 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0.4 \end{bmatrix} \begin{bmatrix} 1 & 0.4 \end{bmatrix} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 0.16 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix}$$

⇒ Compute the Linear Discriminant Projection for the 2D dataset.

$X_1 = (x_1, x_2) = \{(4,1), (2,4), (2,3), (3,6), (4,4)\}$

$X_2 = (x_1, x_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$

Sol:

Step 1: Compute mean of Class 1 & Class 2

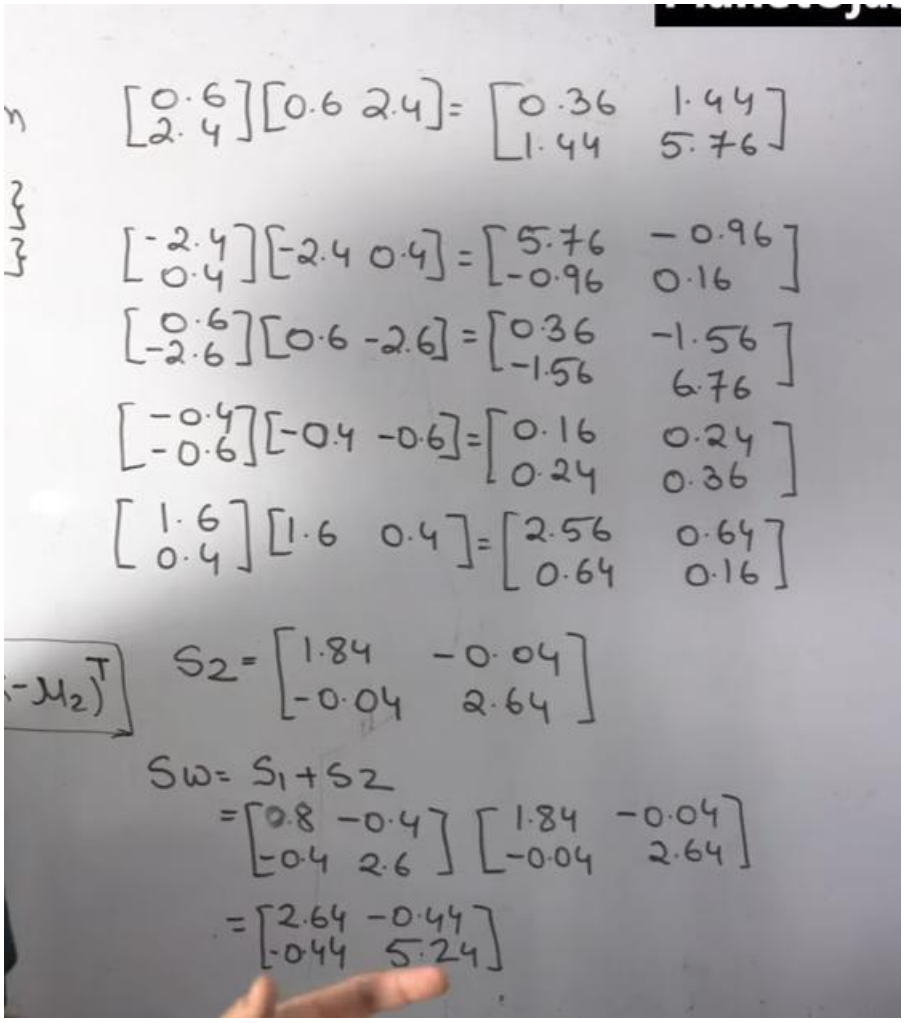
$$S_W = S_1 + S_2$$

$$\mu_2 = \left\{ \frac{9+6+9+8+10}{5}, \frac{10+8+5+7+8}{5} \right\}$$

$\mu_1 = \{3, 3.6\}$ $\mu_2 = \{8.4, 7.6\}$

$$S_2 = \sum (x - \mu_2)(x - \mu_2)^T$$

$$(X_2 - \mu_2) = \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix}$$


$\begin{bmatrix} 0.6 \\ 2.4 \end{bmatrix}\begin{bmatrix} 0.6 & 2.4 \end{bmatrix} = \begin{bmatrix} 0.36 & 1.44 \\ 1.44 & 5.76 \end{bmatrix}$
$\begin{bmatrix} -2.4 \\ 0.4 \end{bmatrix}\begin{bmatrix} -2.4 & 0.4 \end{bmatrix} = \begin{bmatrix} 5.76 & -0.96 \\ -0.96 & 0.16 \end{bmatrix}$
$\begin{bmatrix} 0.6 \\ -2.6 \end{bmatrix}\begin{bmatrix} 0.6 & -2.6 \end{bmatrix} = \begin{bmatrix} 0.36 & -1.56 \\ -1.56 & 6.76 \end{bmatrix}$
$\begin{bmatrix} -0.4 \\ -0.6 \end{bmatrix}\begin{bmatrix} -0.4 & -0.6 \end{bmatrix} = \begin{bmatrix} 0.16 & 0.24 \\ 0.24 & 0.36 \end{bmatrix}$
$\begin{bmatrix} 1.6 \\ 0.4 \end{bmatrix}\begin{bmatrix} 1.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 2.56 & 0.64 \\ 0.64 & 0.16 \end{bmatrix}$
$S_2 = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$
$S_W = S_1 + S_2$
$= \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix}\begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$
$= \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.24 \end{bmatrix}$

⇒ Compute the Linear Discriminant Projection for the 2D dataset.

$X_1 = (x_1, x_2) = \{(4,1), (2,4), (2,3), (3,6), (4,4)\}$

$X_2 = (x_1, x_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$

Sol:

Step1: Compute mean of Class1 & Class2

$\mu_1 = \{3, 3.6\}$ $\mu_2 = \{8.4, 7.6\}$

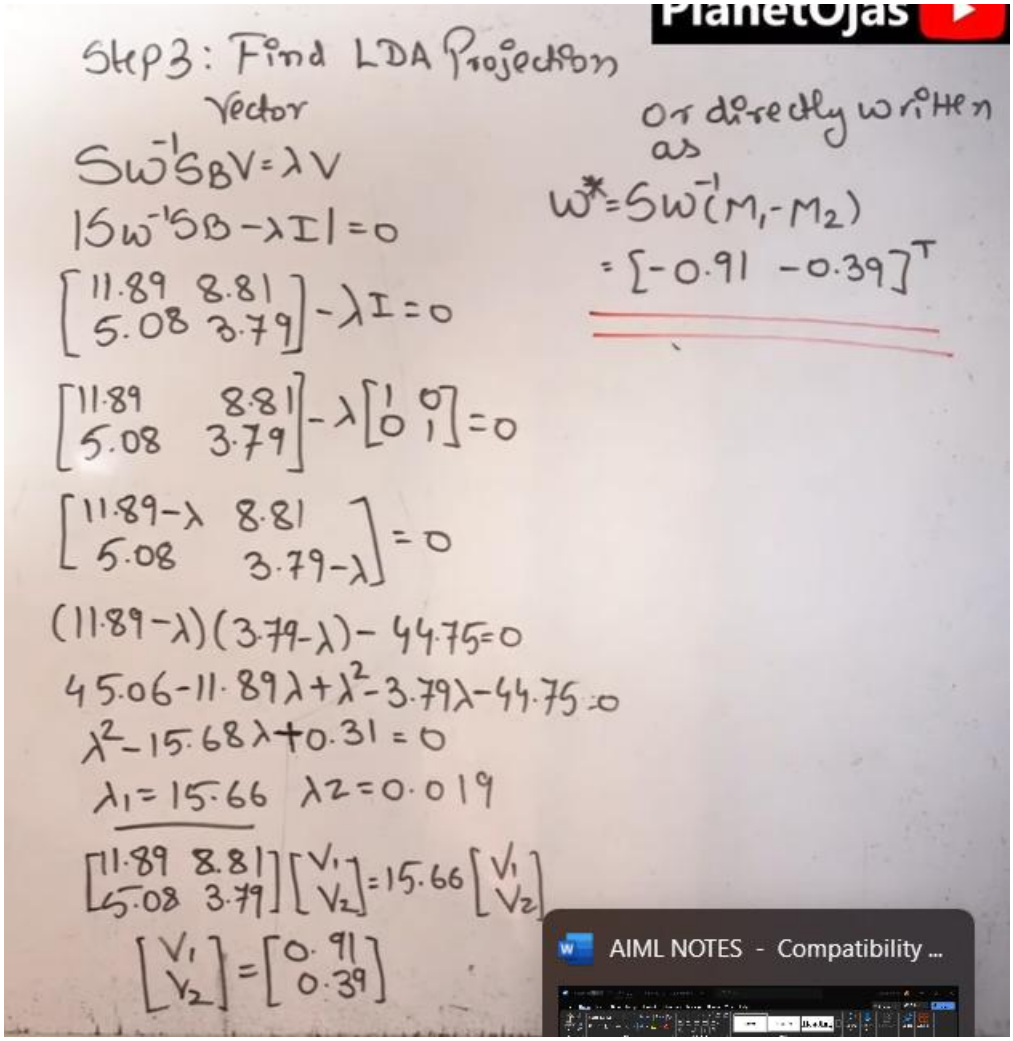
$SW = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.24 \end{bmatrix}$

Step2: Compute between Class Scatter Matrix

$$SB = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$\mu_1 - \mu_2 = \begin{bmatrix} 3 \\ 3.36 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} = \begin{bmatrix} -5.4 \\ -4 \end{bmatrix}$$

$$SB = \begin{bmatrix} -5.4 \\ -4 \end{bmatrix} \begin{bmatrix} -5.4 & -4 \end{bmatrix} = \begin{bmatrix} 29.16 & 21.60 \\ 21.60 & 16 \end{bmatrix}$$



KNN Classifier to classify New Instance IRIS

KNN Classifier Solved Example - 1

Sepal Length	Sepal Width	Species
5.3	3.7	Setosa
5.1	3.8	Setosa
7.2	3.0	Virginica
5.4	3.4	Setosa
5.1	3.3	Setosa
5.4	3.9	Setosa
7.4	2.8	Virginica
6.1	2.8	Verscicolor
7.3	2.9	Virginica
6.0	2.7	Verscicolor
5.8	2.8	Virginica
6.3	2.3	Verscicolor
5.1	2.5	Verscicolor
6.3	2.5	Verscicolor
5.5	2.4	Verscicolor

Sepal Length	Sepal Width	Species
5.2	3.1	?

Step 1: Find Distance

Distance (Sepal Length, Sepal Width) = $\sqrt{(x-a)^2 + (y-b)^2}$

Distance (Sepal Length, Sepal Width) = $\sqrt{(5.2-5.3)^2 + (3.1-3.7)^2}$

Distance (Sepal Length, Sepal Width) = 0.608

Sepal Length	Sepal Width	Species	Distance
5.3	3.7	Setosa	0.608

KNN Classifier Solved Example - 1

Sepal Length	Sepal Width	Species	Distance	Rank
5.3	3.7	Setosa	0.608	3
5.1	3.8	Setosa	0.707	6
7.2	3.0	Virginica	2.002	13
5.4	3.4	Setosa	0.36	2
5.1	3.3	Setosa	0.22	1
5.4	3.9	Setosa	0.82	8
7.4	2.8	Virginica	2.22	15
6.1	2.8	Verscicolor	0.94	10
7.3	2.9	Virginica	2.1	14
6.0	2.7	Verscicolor	0.89	9
5.8	2.8	Virginica	0.67	5
6.3	2.3	Verscicolor	1.36	12
5.1	2.5	Verscicolor	0.60	4
6.3	2.5	Verscicolor	1.25	11
5.5	2.4	Verscicolor	0.75	7

Step 2: Find Rank

Least Distance = Least Rank

KNN Classifier Solved Example - 1

Sepal Length	Sepal Width	Species	Distance	Rank
5.3	3.7	Setosa	0.608	3
5.1	3.8	Setosa	0.707	6
7.2	3.0	Virginica	2.002	13
5.4	3.4	Setosa	0.36	2
5.1	3.3	Setosa	0.22	1
5.4	3.9	Setosa	0.82	8
7.4	2.8	Virginica	2.22	15
6.1	2.8	Verscicolor	0.94	10
7.3	2.9	Virginica	2.1	14
6.0	2.7	Verscicolor	0.89	9
5.8	2.8	Virginica	0.67	5
6.3	2.3	Verscicolor	1.36	12
5.1	2.5	Verscicolor	0.60	4
6.3	2.5	Verscicolor	1.25	11
5.5	2.4	Verscicolor	0.75	7

Step 3: Find the Nearest Neighbor

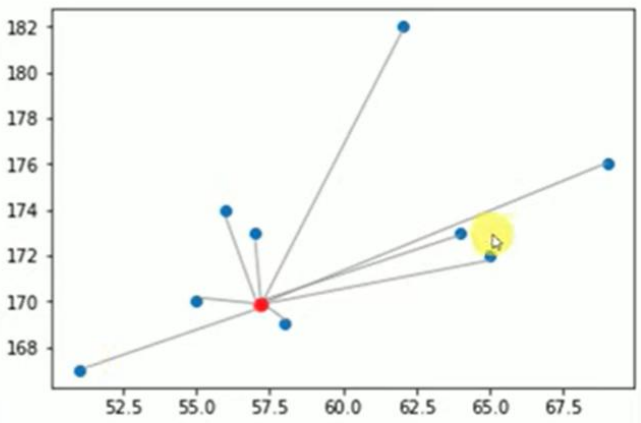
If k = 1 – Setosa

If k = 2 – Setosa

If k = 5 – Setosa

K Nearest Neighbor Algorithm - Solved Example - 2

Height (CM)	Weight (KG)	Class
167	51	Underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	Underweight
169	58	Normal
173	57	Normal
170	55	Normal
170	57	?



Height (CM)	Weight (KG)	Class
167	51	Underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	Underweight
169	58	Normal
173	57	Normal
170	55	Normal
170	57	?

THE DISTANCE FORMULA

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$d_1 = \sqrt{(170-182)^2 + (57-57)^2}$

$d_2 = \sqrt{(170-180)^2 + (57-62)^2}$

Height (CM)	Weight (KG)	Class	Distance	Rank
169	58	Normal ✓	1.4	1 ✓
170	55	Normal ✓	2	2 ✓
173	57	Normal ✓	3	3 ✓
174	56	Underweight ✓	4.1	4 ✓
167	51	Underweight ✓	6.7	5 ✓
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

- If K=1, Normal
- If K=2, Normal
- If K=3, Normal
- If K=4, Normal
- If K=5, Normal

K-Nearest Neighbors Algorithm

	Pepper	Ginger	Chilly	Liked
A	True	True	True	False
B	True	False	False	True
C	False	True	True	False
D	False	True	False	True
E	True	False	False	True

Solved Example 3

K-Nearest Neighbors Algorithm Solved Example - 3

- The "Restaurant A" sells burger with optional flavors: Pepper, Ginger and Chilly.
- Every day this week you have tried a burger (A to E) and kept a record of which you liked.
- Using Hamming distance, show how the 3NN classifier with majority voting would classify

{ pepper: false, ginger: true, chilly : true}

K-Nearest Neighbors Algorithm Solved Example - 3

	Pepper	Ginger	Chilly	Liked
A	True	True	True	False
B	True	False	False	True
C	False	True	True	False
D	False	True	False	True
E	True	False	False	True

New Example - Q: pepper: false, ginger: true, chilly : true

K-Nearest Neighbors Algorithm Solved Example - 3

- But How to calculate the distance for attributes with nominal or categorical values.
- Here we can use Hamming distance to find the distance between the categorical values.
- Let x1 and x2 are the attribute values of two instances.
- Then, in hamming distance, if the categorical values are same or matching that is x1 is same as x2 then distance is 0, otherwise 1.
- For example,
- If value of **x1 is blue** and **x2 is also blue** then the distance between x1 and x2 is **0**.
- If value of **x1 is blue** and **x2 is red** then the distance between x1 and x2 is **1**.

K-Nearest Neighbors Algorithm Solved Exampl

	Pepper	Ginger	Chilly	Liked	Distance
A	True	True	True	False	1 + 0 + 0 = 1
B	True	False	False	True	1 + 1 + 1 = 3
C	False	True	True	False	0 + 0 + 0 = 0
D	False	True	False	True	0 + 0 + 1 = 1
E	True	False	False	True	1 + 1 + 1 = 3

New Example - Q: pepper: false, ginger: true, chilly : true

Use Hamming Distance and

K-Nearest Neighbors Algorithm Solved Example - 3

	Pepper	Ginger	Chilly	Liked	Distance	3NN
A	True	True	True	False	1 + 0 + 0 = 1	2
B	True	False	False	True	1 + 1 + 1 = 3	
C	False	True	True	False	0 + 0 + 0 = 0	1
D	False	True	False	True	0 + 0 + 1 = 1	2
E	True	False	False	True	1 + 1 + 1 = 3	

New Example - Q: pepper: false, ginger: true, chilly : true

Confusion matrixs

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All
- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class
  - **Sensitivity**: True Positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity**: True Negative recognition rate
    - **Specificity = TN/N**

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?
- Perfect score is 1.0

$$recall = \frac{TP}{TP + FN}$$

- Inverse relationship between precision & recall
- ***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- ***F_β*:** weighted measure of precision and recall
  - assigns ß times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$