INDUCTIVE BIAS:-

→ Remarks on CE and VS algoritm:-

1) Will the CE algorithm gives us Correct hypothesis ?

2) what training Example should the learner request next?

→ Inductive learning:

from Examples we derive Rules (Real Time Experience).

(ex):- House Building - Cement ratio we dont know
- So first water mixed more dilute
- so less " mixed becomes more

By the way of Experience we will Come to know how much ratio have to mix.

" various Example - got Experience, and applied to New Ex".

- Deductive learning:

- Already Existing rules are applied to our Examples

(ex) known Civil Engineer knows how to build, Cement, water mix.

* Biased hypothesis Space:

Does not Consider all types of training Examples.

Bias means Showing Partiality, Difference.

" All types of Ex Not Consider. So what is the Solution means include all hypothesis."

Ex :

$$\text{Sunny} \overset{\text{And}}{\wedge} \text{warm} \wedge \text{normal} \wedge \text{strong} \wedge \text{cool} \wedge \text{change} \Rightarrow \text{yes}.$$

Then only Player going the Enjoy spot.

" " " " " $\vee$ $\Rightarrow$ NO.

minor changes in one attribute

also players can't Enjoy spot.

* Because of machine learned and habituated the word Change, Simple ($\vee$ Change) changes it will tell No.
* Thats why Biased Not Suitable.

Unbiased hypothesis Space:

Representing Set of Examples, from Tabulation

Possible instance $= 3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$.

Target Concepts : 2^{96} (huge)

(practically Not possible), to learn those many Ex, Thats why Not go for unbiased

* Idea of Inductive Bias :-

Making of addressing Capable of Inductive Bias

Again.

(Ex) We are CSE Engg, But I have to Construct Building, we don't know how to build, By the way we Searched Civil Engg Not available immediatly, What will do, we will Start to do Construct with ideas.

* learner generalizes beyond the observed training Ex to infer New Ex.

$\boxed{\text{`}>\text{' - inductively inferred from}}$

$x > y$ = y is inductively inferred from x

x is Predefined ~~Example~~. ~~Based~~ in the System, from the x you are defining y.

$\boxed{x \text{ giving o/p for } y.}$

<u>Ex</u> :- Learning Alg = L

Training data $D_c = \{x_1, c(x)\}$

New instance $= x_i$ (Now my Task is to classify x_i)

↓

Represented as $L(x_i, D_c)$ ← How do you obtain Result of x_i

$(D_c \wedge x_i) > L(x_i, D_c)$.

already in Predefined System

(i.e) $L(x_i, D_c)$ inductively inferred in $(D_c \wedge x_i)$.

—x—

Trouble free

Decision Tree learning:-

* Mainly used in tree structured Classification and Regression

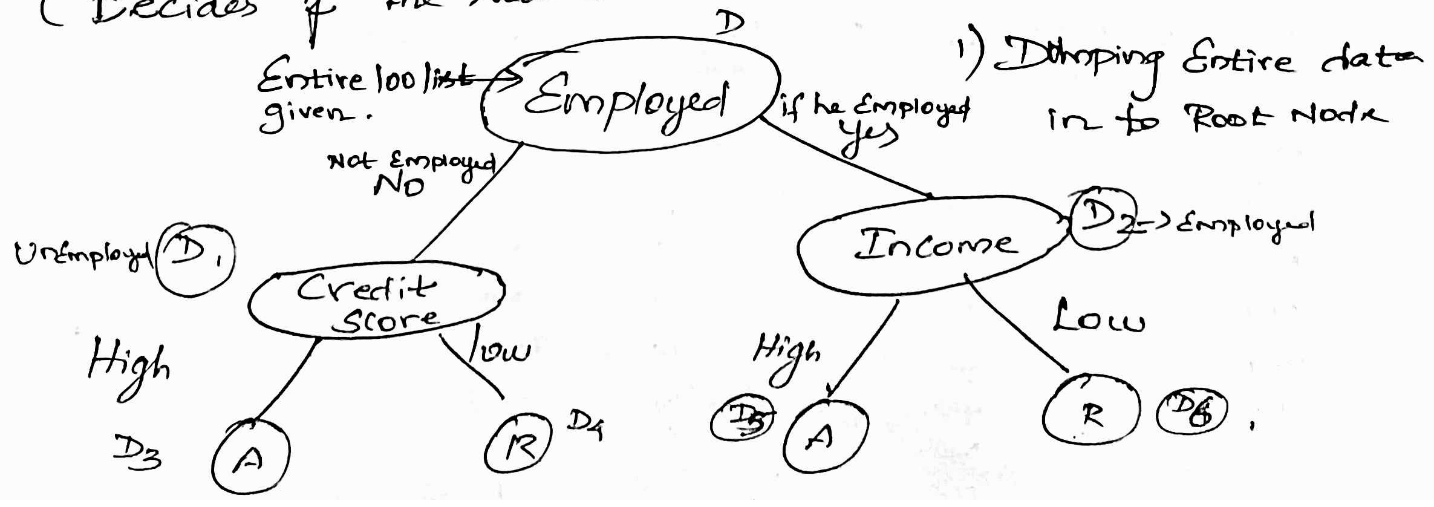
* Classification Consist of Many Algorithms. One of the Algorithm is decision tree (Tree Based).

* Dataset $\xrightarrow{\text{given}}$ Algorithm $\longrightarrow$ Classifies the data
(By using decision tree Alg).

* When we given a data to classifier, it will Say which Class the data belongs to (yes class, No class, (+)ve, (-)ve) Based on Ex.

* 2 Types of Nodes:-
  1) Decision Node (root node - where the Branch is begin)
  2) Leaf Node (you cannot have further Branches (i.e) last Row)

Ex. Loan System
(Decides if the loan should be approved / Rejected)



* 4 data sets we got i.e (D_3, D_4, D_5, D_6), whom you have to approved / Rejected.

Algorithm :

1) In the given data set, choose a target attribute (age of Employee Income status)

2) Calculate information gain of target attribute

$$\text{Information gain} = \frac{-P}{P+N} \log_2 \left(\frac{P}{P+N}\right) - \frac{N}{P+N} \log_2 \left(\frac{N}{P+N}\right)$$

3. for remaining attributes, find Entropy

$$\text{Entropy} = IG \times \text{Probability}$$

How many Times occur.

$$\underset{\text{Entropy}}{E(A)} = \sum \frac{P_i + N_i}{P+N} I(P_i N_i)$$

4. Calculate Gain = $IG - E(A)$

Based on the gain Construct decision tree

— x —

Example & Algorithm of Decision Tree learning:-

Age	Competetion	Type	Profit.
old	yes	S/W	Down
old	NO	S/W	Down
old	NO	H/W	Down
mid	yes	S/W	Down
mid	yes	H/W	Down
mid	NO	H/W	UP
mid	NO	S/W	UP

new	yes	S/W	UP
new	NO	H/W	UP
new	No	S/W	UP

Step 1 :-

Target attribute = Profit.

Step 2 :- Information Gain

$$I_G = \frac{-P}{P+N} \log_2 \left(\frac{P}{P+N}\right) - \frac{N}{P+N} \log_2 \left(\frac{N}{P+N}\right)$$

$\Rightarrow P = \text{Count (down)} = 5$

$N = \text{Count (up)} = 5$

$$= \frac{-5}{10} \log_2 \left(\frac{5}{10}\right) - \frac{5}{10} \log \left(\frac{5}{10}\right)$$

$$= -\left(\frac{1}{2} \log_2 (2^{-1}) + \frac{1}{2} \log_2 (2^{-1})\right)$$

$$= -\left(\frac{1}{2} \times -1 \log_2^2 + \frac{1}{2} \times -1 \log_2^2\right)$$

$\therefore \log_n a^m = m \log a$

$\therefore \log_2^2 = 1$

$$= -\left(\frac{1}{2} \times -1 + \frac{1}{2} \times -1\right)$$

$$= -(-1) = 1$$

$$\boxed{IG = 1}$$

Step 3 :- Calculate entropy for Remaining attributes

$$E(A) = \sum \frac{P_i + N_i}{P+N} I(P_i N_i)$$ [IG × probability]

To find the Entropy for Non-Target attributes.

Remaining attributes are Age, Competition, Type.

(Age) :-

1) Prepare a table for Each Attribute

rows - Values of undertaken attribute (old, mid, New)

Columns - Values of target Attribute (down, UP)

	down	UP
old	3	0
mid	2	2
new	0	3

Entropy = IG x Probability

P = down Count

N = UP Count

$$\text{Information Gain (Old)} = -\left(\frac{3}{3}\log\left(\frac{3}{3}\right) + \frac{0}{3}\log\left(\frac{0}{3}\right)\right) = 0$$

P = 3
N = 0

$\therefore \log(1) = 0$

$$\text{Probability} = \frac{3}{10}$$

[In (Ex) Tabulation, How many Age Count 10, from that How many Age (Old) = 3]

$$\boxed{\text{Entropy (Old)} = 0 \times \frac{3}{10} = 0}$$

$$IG(\text{mid}) = -\left(\frac{2}{4}\log\left(\frac{2}{4}\right) + \frac{2}{4}\log\left(\frac{2}{4}\right)\right) = 1$$

Probability = 4/10

$$\boxed{\text{Entropy (mid)} = 1 \times \frac{4}{10} = 0.4}$$

~~IG (mid)~~

$$I_G(\text{new}) = -\left(\frac{0}{3}\log\left(\frac{0}{3}\right) + \frac{3}{3}\log\left(\frac{3}{3}\right)\right] = 0$$

Probability = 3/10

$$\boxed{\text{Entropy (new)} = 0 \times 3/10 = 0}$$

Entropy (Age) = E(o) + E(m) + E(N) = 0 + 0.4 + 0 = 0.4

Step 4:- Entropy = IG - E(A) = 1 - 0.4 = 0.6

Like that for Competition & Type.

—×—

In the Same way, Calculate Gain for Other attributes.

Gain (Competetion) = 0.124

Gain (Type) = 0

Gain (Age) = 0.6

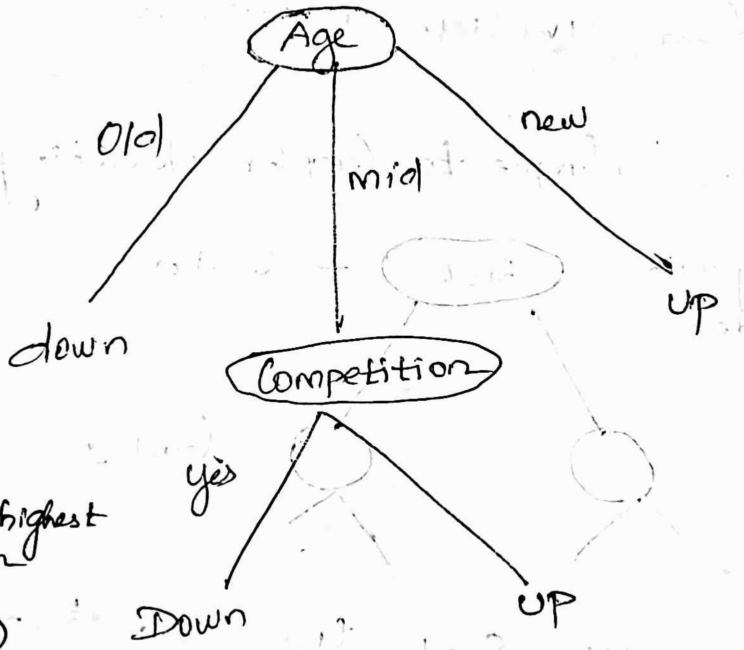
Highest Gain → root Node
(Age) = 0.6.

Age → High value (0.6)

Old → all down

mid → Some down & Some up.

new → all UP



Competition = 0.124

Why Competition → next highest gain

Why not Other (Type)

Type gain value is 0, No Requirement, ignore

—×—