Assignment-2

Name : Faisal Ahmed

ID : 21301186

Section : 07

Course : CSE422 (IBU)

3)

(a) Encoding: Represent coloured graph with red, blue and green colour of 7 vertices

A → red, B → blue, C → green, D → green

E → red, ~~ttt~~ F → green, G → blue

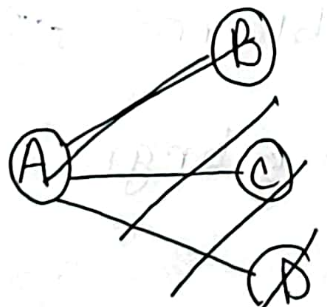
Initially the state will be:

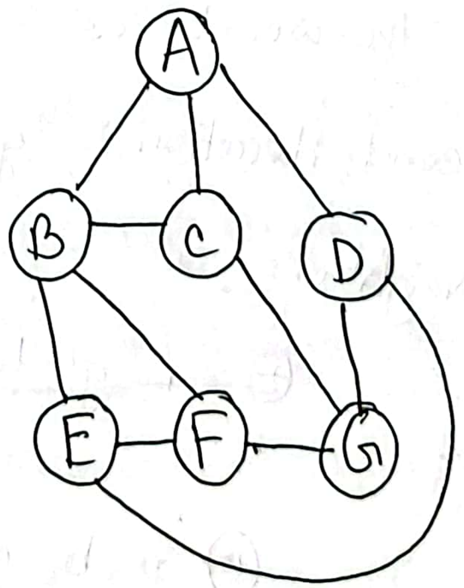
[red, blue, green, green, red, green, blue]

Neighbor:

Goal: The goal is to minimize the number of colours in the graph.

(b)





Fitness function → (Total edges − Total conflicted edges)

Best case = (11 − 0) = 11

Hill climbling algorithm

Initially : g b r bb g r

Fitness : (11 - 3) = 8

First iteration :

Neighboring saples =)

(i) g b r gbrg r ; fitness = (11 - 1) = 10

(ii) g b r b b g r r ; fit ness = (11 - 4) = 7

So, the best neighbor is gbr gbgr and

the worst is gbrbbrrr

Second iteration : " g b r g brgr "

Neighbors :

~~(i) r g b b g b g ; fitness = (11 - 1) = 10~~

(i) r b g g r g b , fitness = (11 - 0) = 11

(ii) r b g g r b b , fitness = (11 - 2) = 9

So, optimal solution : r b g g r g b

e) The problem scenarios of hill climbing:

Local optima: Although we have reached the exact outcome but when the data set is huge the outcome which may look the ultimate solution may not be ~~t~~ our actual target. To solve this problem we can use random restart where we will use random combinations of colour each time to get the exact outcome we require.

Again for the ~~the~~ plateaus and ridge issue where ~~ver~~ there can be a point from which we will get the same fitness from the neighbors and a lot of closed outputs in small range, the neighbor will miss to reach the final outcome everytime hill climbing is applied, we can use problem reformulation and simulated annealing by taking a temp.

Again, stochastic hill climbing will choose the better neighbor randomly which may solve the Local optima issue.

(d) Let, the energy $E = (\text{Fitness} * 10)$

Initial stage = RGBRGBR and $T_i = 100$ (initial)

First iteration

"RGBRGBG"

$\Delta E = E_N - E_C = 9*10 - 8*10 = 10 \cancel{\neq 0} > 0$

So, current sample will be updated as

RGBRGBG

$T_F = T_i \times \alpha^{1} = 100 \times 0.5^{1} = 50$

Second iteration RGBRGBR

$\Delta E = E_N - E_C = (6 \times 10) - 9 \times 10$

$\cancel{=10 \neq 0}$ $-30 < 0$

Current sample will be RGBRGBR

$e^{\frac{-30}{50}} = 0.5488 > rand(0,1) = 0.49$

$T_{F_2} = T_F \times \alpha^2 = 50 \times 0.5^2$

$= 12.5$

So, RGBRGGR accepted

Third iteration, "R B G G R G B"

$\Delta E = E_n - E_c$

$= (11 \times 10) - (9 \times 10)$

$= 20 > 0$

So, $T_{F_3} = T_{F_2} \times \alpha^3 = 1.5625$

So, R B G G R G B is accepted.

So the optimal solution is R B G G R G B

Temperature has a great significance as temperatur decreases the value of the function also reduces as aresult the possibility of picking the childra also reduces. If temperature increase the algorithm will continue picking its neighbor thus it will become an infinite loop ~~whil~~ if temperature continues to increase.

(e) Genetic algorithm:

(i) Initially chromosomes:

R G B R G G R

R G B R G B R

G B R G R B G

R B R G B R B

fitness calculation:

RGBRGGR - 6

RGBRGBR - 8

~~R~~ GBR GRBG - 7

RBR GBRB - 7

So, RGBRGGR will be left out as it has lowest fitness

<u>Crossing over</u>

RGBR GBR → RGBGRBG $\xrightarrow{\text{Mutation}}$ RGBGRBR

GBR GRBG → GBRRGBR $\xrightarrow{\text{Mutation}}$ GBRBGBG

R G B | R G B R

R G R | G B R B

RGBGBRB $\xrightarrow{\text{Mutation}}$ RGBGGRB

RGRRGBR $\xrightarrow{\text{Mutation}}$ RGRBGBR

New population:

RGBGRBR

GBRBGBG

RGBGGRB

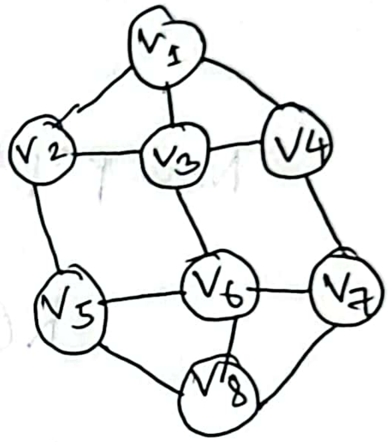
RGRBGBR

Optimal solution here is : RGBGRBR

(f) If all chromosomes in population are same then there will be always same ~~in~~ outcome of crossing over. And the only ~~ea~~ effective way to reach the goal is mutation. So, to reach the goal state it will require much more time then usual.

4) (a) Encoding → 0, 1, 0, 1, 0, 1, 0, 1

where 0 refers to group 1 and 1 refers to group 2.



So, Group 1 : V_1, V_3, V_5, V_7

Group2 : V_2, V_4, V_6, V_8

Neighbor : Neighbor code can be achieve by altering the bits. Such as : [0, 1, 1, 1, 0, 1, 1, 0]

(b) Initially 0, 1, 0, 1, 0, 1, 0, 1

Number of total edges = 11

First Iteration, lets alter V_1, updated set will be

1, 1, 0, 1, 0, 1, 0, 1

Group 1 : V_1, V_2, V_4, V_6, V_8

Group 2 : V_3, V_5, V_7

New total no. of edge = 9

Now, $e^{-\Delta E/T} = e^{-2/50} = 0.96$.

Though, the case has become worse still we will accept it

Second iteration : altering V_6

1, 1, 0, 1, 0, 0, 0, 1

$T = 100 \times 0.5^2 = 25$, edges = 8

So, it has worsen again,

$P = e^{\frac{-(9-8)}{25}} = \overset{0.9608}{\cancel{0.913}}$. As probability is still higher so we will accept it.

Lets flip V_5 now,

Third Iteration , $t = 100 \times 0.5^3 = 12.5$

After altering V_5

1, 1, 0, 1, 1, 0, 0, 1

edges = 7.

So, $e^{\frac{-8-7}{12.5}} = 0.923$.

Still it's higher so we will accept it although has worsen.

Simulated annealing let us choose the worse cases as well because, the case may be temporary worse and going forward we may find the best case.

when the temperature is high it explores the worse cases to see if it can give better output going forward.

If the temperature is increasing at each iteration it will continue to explore the worse cases and it will

Group 1 : ~~V_3, V_5, V_7~~ V_1, V_2, V_4, V_6, V_8

Group 2 : V_3, V_5, V_7

Total no. of present edges = 10.

As it is going against our requirement we will stay with previous assumption

Second iteration,

Let's alter V_6.

Group 1 : V_2, V_4, V_8

Group 2 : V_1, V_3, V_5, V_6, V_7

Total no. of present edges = 9

So, it has make it worse so we will stick to 0, 1, 0, 1, 0, 1, 0, 1

No. of edges = 11

(c) Hill climb will choose the ~~nex~~ ~~fit~~ best neighbor it will encounter fast. It may not be the overall best output which reduce its efficiency. whereas in stochiostic it may initially count the worst cases but eventually it will give us the best outcome. By random restart we will generate a bunch of cases by ~~altering~~ ~~color~~ new initial states and see whethere we are getting more and more edges. If we find the most edges that will be our expected sample

(d) Given, $T(K) = T_0 \alpha^K$

Initially : 0, 1, 0, 1, 0, 1, 0, 1

Total no. of edge = 11

First Iteration :, $T(1) = 100 \times 0.5^1 = 50$

After altering V_1,

not stop, an infinite loop may occur.

(e) Initial population

Chromosome 1 : 0, 1, 0, 1, 0, 1, 0, 1

Chromosome 2 : 1, 1, 0, 1, 0, 0, 0, 1

Chromosome 3 : 1, 1, 0, 1, 1, 0, 0, 1

Chromosome 4 : 1, 0, 0, 1, 1, 0, 0, 1

Fitness are, chromosome 1 : 11

Chromosome 2 : 8

Chromosome 3 : 7

chromosome 4 : 8

Selection : Chromosome 1, 2 and 4 are selected

Crossing over :

0, 1, 0, 1, 0, 1, 0, 1 → 1, 1, 0, 1, 0, 1, 0, 1

1, 1, 0, 1, 0, 0, 0, 1 → 0, 1, 0, 1, 0, 0, 0, 1

0, 1, 0, 1, 0, 1, 0, 1 → 1, 1, 0, 1, 1, 1, 0, 1
1, 1, 0, 1, 1, 0, 0, 1 → 0, 1, 0, 1, 0, 0, 0, 1

Off spring 1: 1, 1, 0, 1, 0, 1, 0, 1

Offspring 2: 0, 1, 0, 1, 0, 0, 0, 1

Offspring 3: 1, 1, 0, 1, 1, 1, 0, 1, offspring4: 0, 1, 0, 1, 0, 0, 0, 1

After mutation

Offspring 1: 1, 1, 0, 1, 0, 1, 0, 0
Offspring 3: 1, 1, 0, 1, 0, 1, 0, 1
Offspring 2: 0, 1, 0, 1, 0, 0, 1, 1
Offspring 4: 0, 1, 0, 1, 1, 0, 0, 1

(f) If the initial population ~~of~~ are same like 0, 1, 0, 1, 0, 1, 0, 1 then we would have got the exact same code after crossing over. So, ~~getting the chance of~~ chance of getting diversified code from crossing over would have been nullified.

Mutation is helpful for finding a better solution. In our problem solution. we can see that after mutation we have got

1, 1, 0, 1, 0, 1, 0, 0

and 0, 1, 0, 1, 0, 0, 1, 1

Both of this samples are different from the the chromosomes of the population. This were not possible to achieve using crossing over. Mutation helps to bring out the characteristics | that even if they are missing in the parents that helps when parents are missing traits which may required in the goal.

1) (a) All cities should be visited. So, to travel the minimum distance all cities should be visited.

So, the initial state would be (A, B, C, D, E, F).

distance would be = $d_{AB} + d_{BC} + d_{CD} + d_{DE}$

$+ d_{EF} + d_{FA}$

Neighbour: The neighbour state will be by ~~changing~~ swapping any values like:

(A, D, C, B, E, F). FA

(b) Initially → A, B, C, D, E, F

distance = 39 + 35 + 80 + 40 + 46 + 41

= 281

First iteration,

(A, D, C, B, F, E)

new distance = $d_{AD} + d_{DC} + d_{CB} + d_{BF} + d_{FE} +$

d_{EA}

$= 53 + 80 + 35 + 36 + 46 + 56$

$= 306$ (worse)

As it is worse we will stay to the previous one

Second iteration,

A, C, B, D, E, F

So distance $= d_{AC} + d_{CB} + d_{BD} + d_{DE} + d_{EF} + d_{FA}$

$= 65 + 35 + 36 + 40 + 46 + 41$

$= 263$ (Better)

So, A, C, B, D, E, F will be the new set.

With hill climbing we can find local optima like the second iteration. However it can be local optima instead of expected outcome. Even we can get plateus like the same distance for different combinations.

(c) First choice hill climbing will choose the better children on first come basis. It will choose the children like 263 from second iteration as it has appeared fast.

Stochastic hill climbing will see some iterations of the distance value and will choose randomly among the better values. Random restart will randomly start with randomly generated combination and it will not declare as goal if it founds a better value it will iterate a number of time and among those iterations best value will be declared as ~~output~~ optimal combination.

(d) $T(K) = T_0 \times \alpha^k$

$T_0 = 100$, $\alpha = 0.5$

Initially: 1, 2, 3, 4, 5, 6

distance = 281

Iteration 1,

(A, D, C, B, F, E)

$T(1) = 100 \times 0.5^1 = 50$

new distance: 306 (worse)

$P = e^{-\Delta E/T} = e^{-\left(\frac{306-281}{50}\right)}$

= 0.6065 which is high chance to be greater than random(0, 1), So, it will be accepted.

Iteration 2: A, C, B, D, E, F

$T(2) = 100 \times 0.5^2 = 25$

New distance = 263 (better)

So, it would be also accepted.

Iteration 3 :

A, C, B, E, D, F

distance = 276 (worse)

$$P = e^{-\Delta E / T} = e^{-\frac{(276-263)}{12.5}}$$

$$= 0.35345$$

which has high ~~chang~~ chance to be less than random (0, 1). So, it will not be accepted.

f) If all the chromosomes in initial population were same like A, B, C, D, E, F then crossing over would not have generated any new variant. After crossing over we would have got the same four new chromosomes which would make it inefficient.

Mutation is helpful in finding a better solution because it ~~would~~ can generate new trait which are absent in the initial chromosomes which are (A, B, C, D, E, F), (A, D, C, B, F, E), (A, C, B, D, E, F), (A, C, B, E, D, F). So, mutation is very important in this case which may provide new ~~traint~~ traits which are absent in the parents variants and which may requires for the optimal goal.

Part 1 (Set 3):

3)

$$\text{Entropy (low temperature)} = -p(\text{fail})\log_2 p(\text{fail}) - p(\text{not fail})\log_2 p(\text{not fail})$$

$$= -\frac{5}{30}\log_2\frac{5}{30} - \frac{25}{30}\log_2\frac{25}{30}$$

$$= 0.4308 + 0.2192$$

$$= 0.65$$

$$\text{Entropy (medium temperature)} = -p(\text{fail})\log_2 p(\text{fail}) - p(\text{not fail})\log_2 p(\text{not fail})$$

$$= -\frac{20}{40}\log_2\frac{20}{40} - \frac{20}{40}\log_2\frac{20}{40}$$

$$= 0.5 + 0.5$$

$$= 1$$

$$\text{Entropy (high temperature)} = -p(\text{fail})\log_2 p(\text{fail}) - p(\text{not fail})\log_2 p(\text{not fail})$$

$$= -\frac{25}{30}\log_2\left(\frac{25}{30}\right) - \frac{5}{30}\log_2\left(\frac{5}{30}\right)$$

$$= 0.2192 + 0.4308$$

$$= 0.65$$

$$\text{Entropy (decision)} = -p(\text{fail})\log_2 p(\text{fail}) - p(\text{not fail})\log_2 p(\text{not fail})$$

$$= -0.5\log_2(0.5) - 0.5\log_2(0.5)$$

$= 0.5 + 0.5$

$= 1$

So, Information gain(temperature) = Entropy (decision) -

P(low temp) * Entropy(low temp)

- p(medium temp) * Entropy(medium temp)

- p(high temp) * Entropy(high temp)

$$= 1 - \left(\frac{30}{100} * 0.65\right) - \left(\frac{40}{100} * 1\right) - \left(\frac{30}{100} * 0.65\right)$$

$= 1 - 0.195 - 0.4 - 0.195$

$= 1 - 0.79 = 0.21$ (Ans)

Now, for maintenance frequency,

Entropy (decision) = 1

Entropy (regular maintenance) = - p(fail) $\log_2$ P(fail) - p(not fail) $\log_2$ P(not fail)

$$= -\frac{10}{60}\log_2\left(\frac{10}{60}\right) - \frac{50}{60}\log_2\left(\frac{50}{60}\right)$$

$= 0.4308 + 0.2192$

$= 0.65$

Entropy(irregular maintenance) ~~$= -p\log_2 p(fail) - p\log_2 p$~~

$= -p(fail)\log_2 p(fail) - p(not\ fail)\log_2 p(not\ fail)$

$= -1\log_2(1) - 0\log_2 p(0)$

$= 0$

So, Information gain (maintenance frequency)

$=$ Entropy(decision) $-$ Entropy(regular maintenance)

$*$ p(regular maintenance) $-$ Entropy(irregular maintenance)

$*$ p(irregular maintenance)

$= 1 - (0.65 * 0.6) - (0 * 0.4)$

$= 1 - 0.39 = 0.61$ (Ans)

Part 2:

Dataset-2: Firstly, splitting based on Workplace Environment,

Entropy(Supportive) $= -p(yes)\log_2 p(yes) - p(No)\log_2 p(No)$

$= -1\log_2 p(1) - 0$

$= 0$

$$\text{Entropy(Hostile)} = -p(\text{Yes})\log_2 p(\text{Yes}) - p(\text{No})\log_2 p(\text{No})$$

$$= 0 - 1\log_2(1)$$

$$= 0$$

$$\text{Entropy(Neutral)} = -p(\text{Yes})\log_2 p(\text{Yes}) - p(\text{No})\log_2 p(\text{No})$$

$$= -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)$$

$$= 0.529 + 0.442 = 0.971$$

$\therefore$ ~~Inf~~ ~~Entropy (Decision)~~

$$\text{Entropy (Feeling safe)} = -p(\text{Yes})\log_2 p(\text{Yes}) - p(\text{No})\log_2 p(\text{No})$$

$$= -\frac{7}{1\cancel{4}}\log_2\left(\frac{7}{1\cancel{4}}\right) - \frac{8}{15}\log_2\left(\frac{8}{1\cancel{4}}\right)$$

$$= \cancel{0.5131 + 0.484}\ 0.5 + 0.5$$

$$\cancel{= 0.9971} \quad = 1$$

So, Information gain(Workplace environment)

$$= \text{Entropy (feeling safe)} - P(\text{Neutral}) * \text{Entropy (Neutral)}$$

$$= \overset{1}{\cancel{0.9971}} - \left(\frac{5}{14} * 0.971\right) = 0.6532$$

Now, splitting based on harrassment policies.

$$\text{Entropy (strong)} = -p(\text{Yes}) \log_2 P(\text{Yes}) - p(\text{No}) \log_2 P(\text{No})$$

$$= -1 \log_2 1 - 0$$

$$= 0$$

$$\text{Entropy (Moderate)} = -p(\text{Yes}) \log_2 P(\text{Yes}) - P(\text{No}) \log_2 P(\text{No})$$

$$= -\frac{2}{5} \log_2 \left(\frac{2}{5}\right) - \frac{3}{5} \log_2 \left(\frac{3}{5}\right)$$

$$= 0.971$$

$$\text{Entropy (Weak)} = p(\text{Yes}) \log_2 P(\text{Yes}) - P(\text{No}) \log_2 P(\text{No})$$

$$= 0 - 1 \log_2 (1)$$

$$= 0$$

So, Information Gain (harrassment policies)

$$= \text{Entropy (feeling safe)} - \{P(\text{Moderate}) * \text{Entropy (Moderate)}\}$$

$$= 1 - \left(\frac{5}{14} * 0.971\right) = 0.6532$$

Now, Splitting based on workplace flexibility.

$$\text{Entropy (High)} = -p(\text{Yes}) \log_2 P(\text{Yes}) - p(\text{No}) \log_2 P(\text{No})$$

$$= -\frac{3}{4} \log_2 \left(\frac{3}{4}\right) - \frac{1}{4} \log_2 \left(\frac{1}{4}\right)$$

$$= 0.3113 + 0.5 = 0.8113$$

$$\text{Entropy(Medium)} = -p(\text{Yes})\log_2 P(\text{Yes}) - p(\text{No})\log_2 P(\text{No})$$

$$= -\frac{4}{5}\log_2\left(\frac{4}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right)$$

$$= 0.2575 + 0.4644$$

$$= 0.7219$$

$$\text{Entropy (Low)} = -p(\text{Yes})\log_2 P(\text{Yes}) - p(\text{No})\log_2 P(\text{No})$$

$$= 0 - 1\log_2(1) = 0$$

So, Information Gain (Workplace flexibility)

$$= \text{Entropy(feeling safe)} - (P(\text{High}) * \text{Entropy (High)})$$
$$- (P(\text{Medium}) * \text{Entropy (Medium)})$$

$$= 1 - \left(\frac{4}{14} * 0.8113\right) - \left(\frac{5}{14} * 0.7219\right)$$

$$= 1 - 0.2318 - 0.2578 = 0.5104$$

Again, splitting based on health benefits

$$\text{Entropy (Yes)} = \cancel{p\log} - p(\text{Yes})\log_2 P(\text{Yes}) - p(\text{No})\log_2 P(\text{No})$$

$$= -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}$$

$$= 0.3113 + 0.5 = 0.8113$$

$$\text{Entropy}(No) = -p(Yes)\log_2 P(Yes) - p(No)\log_2 P(No)$$

$$= -\frac{1}{6}\log_2\left(\frac{1}{6}\right) - \frac{5}{6}\log_2\frac{5}{6}$$

$$= 0.4308 + 0.2192$$

$$= 0.65$$

So, Information Gain (Health benefits)

$$= \text{Entropy (Feeling safe)} - (P(Yes) * \text{Entropy}(Yes))$$
$$- (P(No) * \text{Entropy}(No))$$

$$= 1 - \left(\frac{8}{14} * 0.8113\right) - \left(\frac{6}{14} * 0.65\right)$$

$$= 1 - 0.4636 - 0.2786$$

$$= 0.2578$$

Again splitting based on safety Measures:

$$\text{Entropy}(Yes) = -p(Yes)\log_2 P(Yes) - P(No)\log_2 P(No)$$

$$= -1\log_2 P(1) - 0 = 0$$

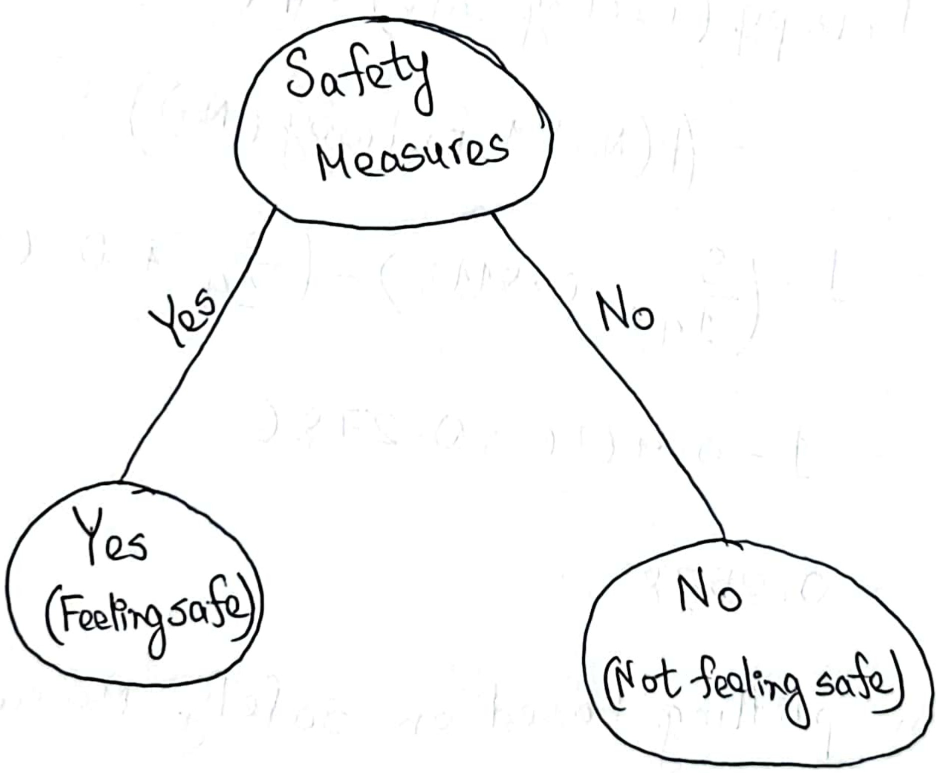
$$\text{Entropy}(No) = -p(Yes)\log_2 P(Yes) - p(No)\log_2 P(No)$$

$$= 0 - 1\log_2 P(1) = 0$$

So, information gain (safety measures)

$= 1 - 0 - 0 = 1$

So, the highest information gain value is of "Safety Measures" attribute. Therefore, the decision tree would be:



Decision tree

Dataset 4:

II Splitting based on Soil type.

$$Entropy(Loam) = -p(Fruit)\log_2 P(fruit) - p(cereal)\log_2 P(cereal) - p(vegetable)\log_2 P(vegetable)$$

$$= -1\log_2(1) - 0 - 0$$

$$= 0$$

$$Entropy(Sandy) = -p(Fruit)\log_2 P(Fruit) - p(cereal) * \log_2 p(cereal) - p(vegetable) * \log_2 p(vegetable)$$

$$= 0 - 1\log_2(1) - 0$$

$$= 0$$

$$Entropy(clay) = -p(Fruit)\log_2 P(Fruit) - p(cereal) * \log_2 P(cereal) - p(vegetable) * \log_2 P(vegetable)$$

$$= 0 - 0 - 1\log_2(1)$$

$$= 0$$

Again, $Entropy(Best\ crop\ to\ grow) = -p(Fruit)\log_2 P(Fruit) - p(cereal)\log_2 p(cereal) - p(vegetable)\log_2 p(vegetable)$

$$= -\frac{5}{15}\log_2\left(\frac{5}{15}\right) - \frac{5}{15}\log_2\left(\frac{5}{15}\right) - \frac{5}{15}\log_2\left(\frac{5}{15}\right)$$

$$= 0.5283 + 0.5283 + 0.5283$$

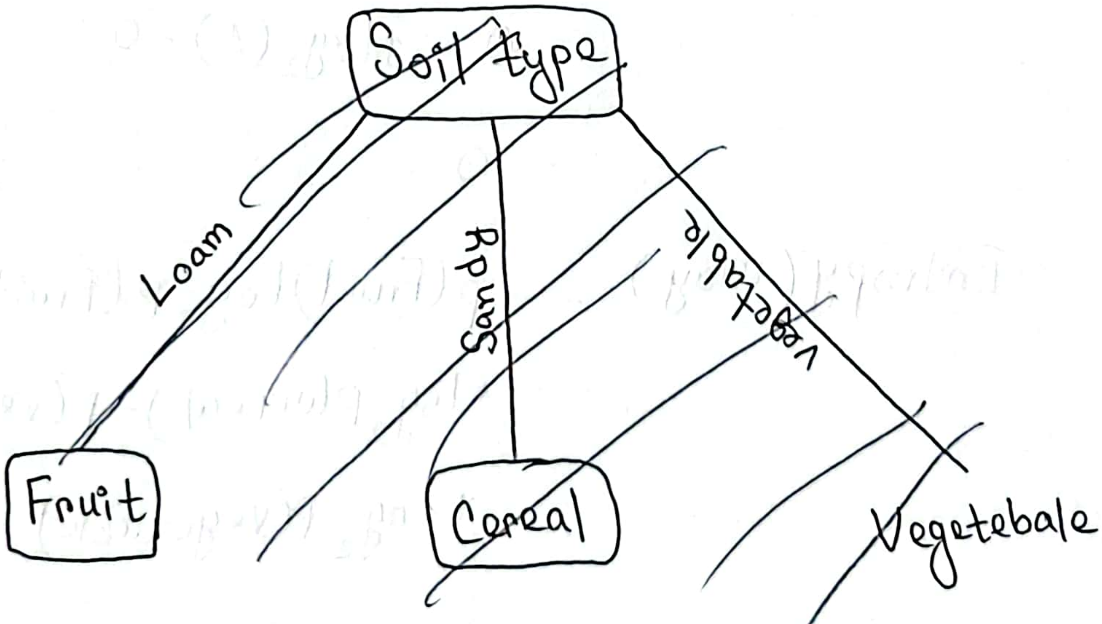
$$= 1.585$$

So, information gain(Soil type)

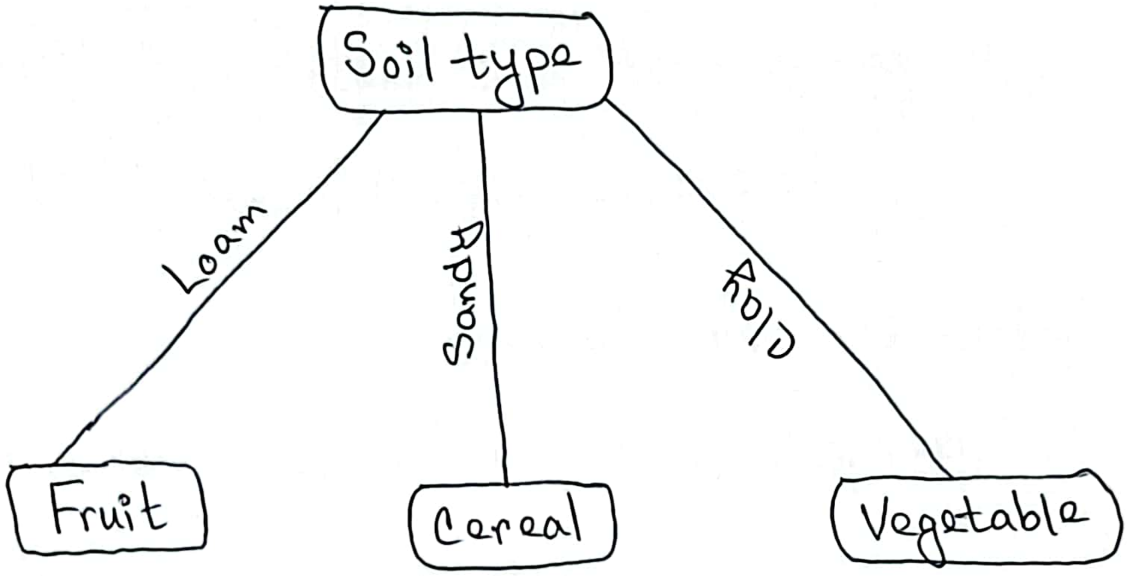
$$= \text{Entropy (Best crop to grow)} - 0 - 0 - 0$$

$$= 1.585$$

So, information gain of soil type has the highest value. So it will be the root node.



So, the decision tree will look like :



Decision Tree

Part 3 (Set 3):

4) Splitting a dataset into training and test sets serves the following purposes:

1) The training set is used to train machiene learning model. The model learns patterns, relationships, decision rules from the data

2) It reduce the risk of overfitting because model performs well on training data but poorly on ~~new~~, unknown data as it memorizes training set rather than learning the pattern.

3) It provides a realistic esmitation of how the model will perform in the real world by evaluating the model's performance on unseen data.

14) The steps to handle continuous attribute are:

① Sort the data in ascending order based on the attribute values.

② Select split points by evaluating midpoint between consecutive attribute values.

③ Calculate information gain with the help of unique categories and selecting highest information gain.

④ Convert continuous attribute into binary or categorical attribute.

⑤ Build the decision tree recursively by treating each continous attribute as categorical one.

16> ID3 handles missing or incomplete data by:

1> Ignoring missing values during split calculation

2> Assign the mode or most common value

3> Assign probabilistic weights

4> Picking an alternate attribute that closely approximates the split of the original attribute.

5> By prunning the missing or incompletedata.

17) Yes, ID3 algorithm can handle multi-class classification problems effectively by the ~~fold~~ following steps:

1) Information gain for multi-class labels.

2) Handling ~~mu~~ splits of multi-class

3) Ending criteria for multi-class classification

4) Nullify overfitting in multi-class problems. by prunning and minimum split size.

18) The primary stopping condition for ID3 algorithm while building decision tree are:

1) If all the instances in current subset belong to the same class, no need to split further. The algorithm stops and creates leaf node by labeling with that class

2) If there are no more remaining attributes to split the data, the algorithm also stops by thaking

that node a leaf node.

3) If a subset data is empty, the algorithm stops and a leaf node is created

4) If the entropy of the subset of data is zero, the algorithm stops by creating a leaf node with that class label.

Set-3 (Bonus)

1) 1.1 Entropy (Dataset) $= -p(\text{vote})\log_2 p(\text{vote})$

$-p(\text{not vote})\log_2 p(\text{not vote})$

$= -\frac{75}{120}\log_2\left(\frac{75}{120}\right) - \frac{45}{120}\log_2\frac{45}{120}$

$= 0.4238 + 0.5306$

$= 0.9544$

1.2 Entropy (18-30 group) $= -p(\text{vote})\log_2 p(\text{vote})$

$-p(\text{not vote})\log_2 p(\text{not vote})$

$= -\frac{30}{50}\log_2\frac{30}{50} - \frac{20}{50}\log_2\frac{20}{50}$

$= 0.4422 + 0.5288$

$= 0.971$

Entropy (31-50 group) $= -p(\text{vote})\log_2 p(\text{vote})$

$-p(\text{not vote})\log_2 p(\text{not vote})$

$= -\frac{35}{40}\log_2\frac{35}{40} - \frac{5}{40}\log_2\frac{5}{40}$

$= 0.1686 + 0.375 = 0.5436$

Entropy (51+ group) = - p(vote) log$_2$ p(vote) - p(not vote) log$_2$ p(not vote)

$= -\frac{10}{30}\log_2\frac{10}{30} - \frac{20}{30}\log_2\frac{20}{30}$

$= 0.5283 + 0.39$

$= 0.9183$

(1.3) Information gain (age group)

= Entropy (dataset) - (Entropy (18-30) * P(18-30)) - (Entropy(31-50) * P(31-50)) - (Entropy (51+) * P(51+))

$= 0.9544 - \left(0.971 \times \frac{50}{120}\right) - \left(0.5436 \times \frac{40}{120}\right) - \left(0.9183 \times \frac{30}{120}\right)$

$= 0.9544 - 0.405 - 0.1812 - 0.2296$

$= 0.1386$ (Ans)

Question 1.2

$$\text{Entropy (low interest)} = -p(\text{vote}) \log_2 p(\text{vote}) - p(\text{not vote}) \log_2 p(\text{not vote})$$

$$= -\frac{15}{60} \log_2 \frac{15}{60} - \frac{45}{60} \log_2 \frac{45}{60}$$

$$= 0.5 + 0.3113$$

$$= 0.8113$$

$$\text{Entropy (Medium interest)} = -p(\text{vote}) \log_2 p(\text{vote}) - p(\text{not vote}) \log_2 p(\text{not vote})$$

$$= -\frac{25}{30} \log_2 \frac{25}{30} - \frac{5}{30} \log_2 \frac{5}{30}$$

$$= 0.2192 + 0.4308$$

$$= 0.65$$

$$\text{Entropy (High interest)} = -1 \log_2 (1) - 0$$

$$= 0$$

Information gain (Interest in politics)

$= \text{Entropy (Dataset)} - (\text{Entropy (Low interest)} * p(\text{Low interest}))$

$- (\text{Entropy (Medium interest)} * p(\text{Medium interest}))$

$= 0.9544 - (0.8113 \times 0.5) - (0.65 \times 0.25)$

$= 0.9544 - 0.40565 - 0.1625 = 0.38625$

Question 1.3: Between "Interest in politics" and "Age group", ~~"Age group"~~ "Interest in politics" attribute should be in the root node of the decision tree because of having higher information gain value i.e. 0.38625 (Ans)

2> (2.1) Entropy (Low credit) $= -p(\text{default}) \log_2 p(\text{default})$

$- p(\text{not default}) \log_2 p(\text{not default})$

$= -\frac{10}{40} \log_2 \left(\frac{10}{40}\right) - \frac{30}{40} \log_2 \left(\frac{30}{40}\right)$

$= 0.5 + 0.3113 = 0.8113$

Entropy (Medium credit) $= -p(\text{Default}) \log_2 p(\text{default})$

$-p(\text{not default}) \log_2 p(\text{not default})$

$= -\frac{5}{35} \log_2\left(\frac{5}{35}\right) - \frac{30}{35} \log_2\left(\frac{30}{35}\right)$

$= 0.4011 + 0.1906$

$= 0.5917$

Entropy (High credit) $= -p(\text{default}) \log_2 p(\text{default})$

$-p(\text{not default}) \log_2 p(\text{not default})$

$= -\frac{1}{25} \log_2\left(\frac{1}{25}\right) - \frac{24}{25} \log_2 \frac{24}{25}$

$= 0.1858 + 0.0565$

$= 0.2423.$

Entropy (Dataset) $= -p(\text{default}) \log_2 p(\text{Default})$

$-p(\text{not default}) \log_2 p(\text{not default})$

$= -\frac{16}{100} \log_2 \frac{16}{100} - \frac{84}{100} \log_2 \frac{84}{100}$

$= 0.4230 + 0.2113 = 0.6343.$

Information gain (credit course)

= Entropy (dataset) - (Entropy (Low credit) * p(Low credit))

- Entropy (medium credit) * p (medium credit))

- Entropy (high credit) * p (high credit))

$$= 0.6343 - \left(0.8113 \times \frac{40}{100}\right) - \left(0.5917 \times \frac{35}{100}\right)$$

$$-\left(0.2423 \times \frac{25}{100}\right)$$

$$= 0.6343 - 0.32452 - 0.2071 - 0.0606$$

$$= 0.04208$$

(2.2) Entropy (Good Loan history) = - p(default) $\log_2$ p(default)

- p (not default) $\log_2$ p(not default)

$$= -\frac{5}{50}\log_2\frac{5}{50} - \frac{45}{50}\log_2\frac{45}{50}$$

$$= 0.3322 + 0.1368$$

$$= 0.469$$

$$\text{Entropy (Bad loan history)} = -p(\text{default}) \log_2 p(\text{default}) - p(\text{not default}) \log_2 p(\text{not default})$$

$$= -\frac{25}{50} \log_2 \frac{25}{50} - \frac{25}{50} \log_2 \frac{25}{50}$$

$$= 0.5 + 0.5 = 1$$

$$\text{Entropy(Dataset)} = -p(\text{default}) \log_2 p(\text{default}) - p(\text{not default}) \log_2 p(\text{not default})$$

$$= -\frac{30}{100} \log_2 \frac{30}{100} - \frac{70}{100} \log_2 \frac{70}{100}$$

$$= 0.5211 + 0.3602$$

$$= 0.8813$$

So, Information gain(Loan History) =

Entropy(Dataset) − (Entropy (Good Loan history) * p(Good Loan history) − (Entropy (Bad Loan history) * p(Bad Loan history)))

$$= 0.8813 - ((0.469 \times 0.5) - (1 \times 0.5))$$

$$= -0.1468 \quad (\text{Ans})$$

Dataset-1

$$\text{Entropy(Participation)} = \cancel{7} - p(\text{Yes})\log_2 p(\text{Yes}) - p(\text{No})\log_2 p(\text{No})$$

$$= -\frac{7}{14}\log_2\left(\frac{7}{14}\right) - \frac{7}{14}\log_2\left(\frac{7}{14}\right)$$

$$= 0.5 + 0.5 = 1$$

Splitting based on Agegroup,

$$\text{Entropy(Young)} = -p(\text{Yes})\log_2 p(\text{Yes}) - p(\text{No})\log_2 p(\text{No})$$

$$= -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$$

$$= 0.4422 + 0.5288$$

$$= 0.971$$

$$\text{Entropy(Middle-aged)} = -p(\text{Yes})\log_2 p(\text{Yes}) - p(\text{No})\log_2 p(\text{No})$$

$$= -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$$

$$= 0.5 + 0.5 = 1$$

$$\text{Entropy(Senior)} = -p(\text{Yes})\log_2 p(\text{Yes}) - p(\text{No})\log_2 p(\text{No})$$

$$= -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$$

$$= 0.5288 + 0.4422 = 0.971$$

So, Information gain (Age group) = Entropy (Participation) − (Entropy(Young) * P(Young)) − (Entropy(Middle-aged) * P(Middle aged)) − (Entropy(Senior) * P(Senior))

$$= 1 - \left(0.971 * \frac{5}{14}\right) - \left(1 \times \frac{4}{14}\right) - \left(0.971 * \frac{5}{14}\right)$$

$$= 1 - 0.3468 - 0.2857 - 0.3468$$

$$= 1 - 0.9793 = 0.0207$$

Now splitting based on Employment status

$$\text{Entropy(Employed)} = -P(\text{Yes})\log_2 P(\text{Yes}) - P(\text{No})\log_2 P(\text{No})$$

$$= \cancel{\frac{7}{14}} - \frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7}$$

$$= 0.4613 + 0.5239$$

$$= 0.9852$$

$$\text{Entropy (Unemployed)} = -p(\text{Yes})\log_2 P(\text{Yes}) - p(\text{No}) \log_2 P(\text{No})$$

$$= -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}$$

$$= 0.5239 + 0.4613$$

$$= 0.9852$$

Information gain (Employment status)

$$= \text{Entropy (participation)} - (\text{Entropy (Employed)} * p(\text{Employed})) - (\text{Entropy (Unemployed)} * P(\text{Unemployed})).$$

$$= 1 - (0.9852 \times 0.5) - (0.9852 \times 0.5)$$

$$= 0.0148.$$

Splitting based on Interest in community events.

$$\text{Entropy (High)} = -p(\text{Yes})\log_2 P(\text{Yes}) - p(\text{No})\log_2 P(\text{No})$$

$$= -1\log 1 - 0 = 0$$

$$\text{Entropy (Medium)} = -p(\text{Yes})\log_2 P(\text{Yes}) - p(\text{No})\log_2 P(\text{No})$$

$$= -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$$

$$= 0.5+0.5 = 1$$

$$\text{Entropy } (\cancel{\text{High}}\ \text{Low}) = -p(\text{Yes})\log_2 P(\text{Yes}) - p(\text{No})\log_2 P(\text{No})$$

$$= -0 - 1\log 1$$

$$= 0$$

So, Information gain (Interest in community events)

$$= \text{Entropy(Participation)} - (\text{Entropy(High)} * P(\text{High})) - (\text{Entropy(Medium)} * P(\text{Medium})) - (\text{Entropy (Low)} * P(\text{Low}))$$

$$= 1 - 0 - \left(1 * \frac{4}{14}\right) - 0$$

$$= 1 - 0.2857 = 0.7143$$

So, The root node will be Interest in community event

We can see that for high and low, we are getting ultimate output but for medium there is contradictory output. So, for medium of Interest in community events we do splitting,

Firstly with Employment status.

~~P(Em~~ Entropy(~~employment~~ employed)

$$= -p(Yes)\log_2 P(Yes) - p(No)\log_2 P(No)$$

$$= -1\log 1 - 0 = 0$$

$$Entropy(Unemployed) = -p(Yes)\log_2 P(Yes) - p(No)\log_2 P(No)$$

$$= -0 - 1\log 1 = 0$$

So, information gain(Employment status/Medium)

$$= 1 - 0 - 0 = 1$$

So, Employment status has the highest information gain given, Interest in community events as medium, So, the decision tree would be:

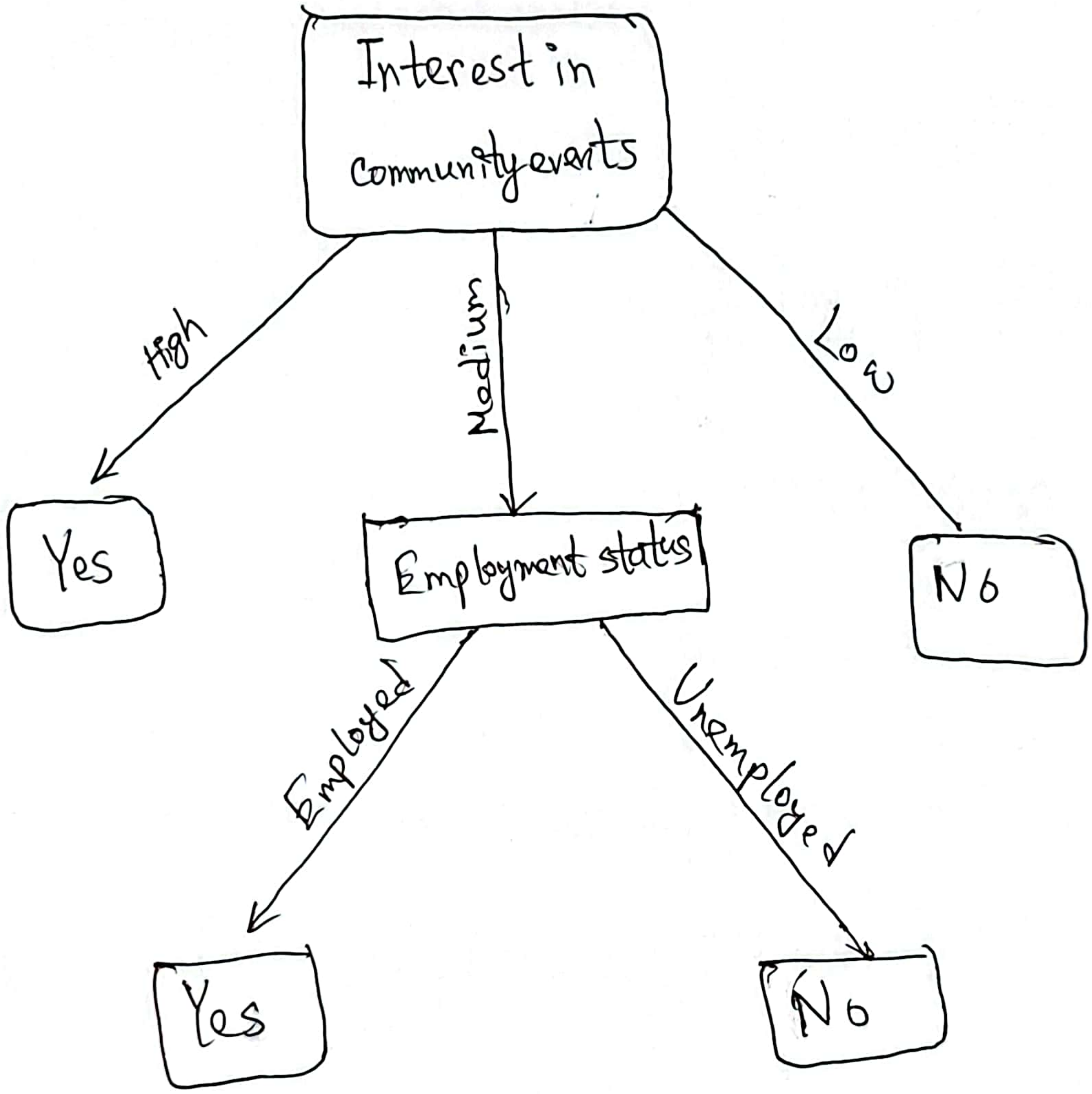


Fig: Decision tree

Dataset 3:

Entropy (Successful career) = $-p(Yes)\log_2 p(Yes)$

$-p(No)\log_2 p(No)$

$= -\frac{8}{14}\log_2\frac{8}{14} - \frac{6}{14}\log_2\frac{6}{14}$

$= 0.4613 + 0.5239$

$= 0.9852$

Splitting based on Internship experience :

Entropy (Yes) = $-p(Yes)\log_2 p(Yes) - p(No)\log_2 p(No)$

$= -1\log_2 1 - 0$

$= 0$

Entropy (No) = $-p(Yes)\log_2 p(Yes) - p(No)\log_2 p(No)$

$= -0 - 1\log_2 1 = 0$

So, Information gain (Internship experience) =

Entropy (successful career) − (Entropy(Yes) * P(Yes)) − (Entropy(No) * P(No)) = 0.9852 − 0 − 0 = 0.9852

So, "Internship Experience" attribute has the highest possible information gain value. ~~So~~, Therefore it will be the root node of our decision tree

~~Information~~

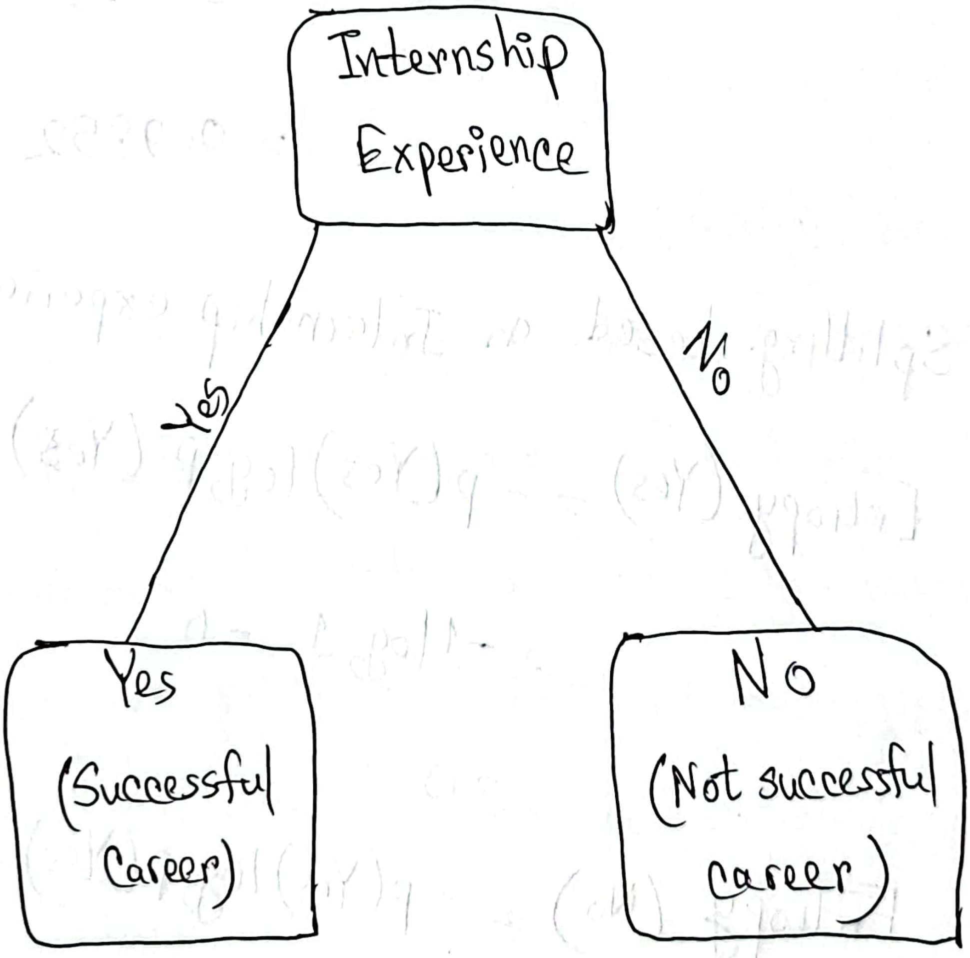


Fig : Decision tree

Part 3

1> Supervised learning deals with predicting outputs on the basis of input-output pairs, while unsupervised learning ~~et~~ emphasizes on finding patternsor groups in unlabeled data.

2> A training dataset is a set of data that is used to machiene learning model how to make decisions. It is very important in machiene learning. Because it helps model to learn relationships, patterns and correlations between input features and output labels. It helps to make accurate predictions on new dat that was not part of training process. It helps to reduce the error between its prediction and actual outcome.

3> The process of how a machiene learning model learn from data are:

(i) Input data is providel into the model

(ii) Initially the model makes prediction

(iii) Loss is being calculated by comparing its prediction with actual outcome.

(iv) By the help of optimization technique, model parameters are adjusted

(v) The model continues to improve its performance by continuing cycle

(vi) Finally it generalize for the new and unseen data.

5) Overfitting occurs when a model learns the random fluctuations and noise alongside with underlying patterns. This problem can be solved by simplifying the model, using more data in the dataset, prunning the part which bears les importance, injecting small amount of noise to the training data etc.

6> Loss function plays a very crucial role in machiene learning. It helps to detect the accuracy of the model and ~~th~~ to which extent the parameters of the model should be changed. Besides by reducing the loss function a model becomes better gradually.

7>

Classification	Regression
1> Output is discrete categories	1> Output is continuous numerical values
2> Goal is to assign a input to a class	2> Goal is to predict a value

8> The main objective of the ID3 algorithm ~~is to cer~~ in decision tree learning is to classify data by choosing splits that maximize value of information gain by reducing entropy and increasing homogeneity of the output subset. The ultimate goal is to make an efficient tree that can accurately classify unseen data.

9> The steps of creating a decision tree using ID3 algorithm are:

1> Calculating the data set's entropy

2> Calculating the information gain of each attribute with the help of entropy of unique category of that particular attribute.

3> Comparing the information gain values of all the attributes and taking the attribute having highest information gain value as the root node.

4> Constructing the decision tree from the root node by forming sub tree by following the cycle untill all the leaf node value becomes the targetted values.

10> Entropy is used to measure the impurity of dataset. The aim is to split the dataset that the entropy reduce which results in increasing homogeneity. So, by reducing the entropy, efficient decision tree for the new datas can build.

11> At first the ID3 algorithm calculate entropy for each unique category of an attribute then calculate the information gain by the help of those entropy of that attribute. The more the information gain value of attributes, the more suitable for splitting the dataset.

12> Information gain is a measure to quantify the effectiveness of an attribute in classifying the dataset. It is being calculated with the help of entropy values of the unique categories of each attribute. Later on the basis of highest information gain values root node is selected for the decision tree and the ~~co~~ tree continues the process

13) The ID3 algorithm prefers attributes with the highest information gain values because it is calculated by eliminating the entropy i.e. impurity of dataset from the actual expected results. So, more the information gain value, more the homogeneity of subsets.

~~14)~~ 15) Although ID3 is an efficient model, it has some drawbacks for large datasets, which are:

1) It needs huge memory usage for all the entropy and information gain values

2) It struggles with continuous features as it is designed for categorical attributes

3) It is biased towards attributes with mor values and in large datasets, this biasness may lead to suboptimal splits and less generalizable model.