(1) (a) Terms related to DBSCAN clustering algorithm

(i) Core object :- An object or point is said to be a core object if there are at least "minpts" No. of objects in the ε - neighbourhood of an object or point

Given, ε = parameter or radius of a circle drawn with any object as the centre point.

minpts : Minimum No. of points (an integer value)

p is core point/object if it has more than minpts No. of points in its ε - neighbourhood.

(ii) Border object :- An object is said to be a border point/object if there are less than minpts No. of points in its ε - neighbourhood but the point itself is in the vicinity of a core object or a core point.

p = core obj

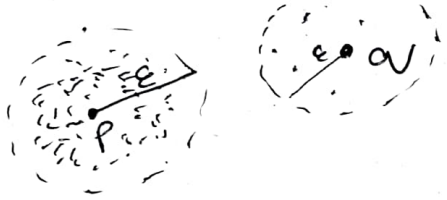
q = border object

(P.T.O)

(iii) Noise object :- A point or an object that is neither a border object nor a core object is a noise point.

In other words, it has less than minpts No. of objects in its ε-neighbourhood and it is also not near to any core point

(Also, No point is directly density reachable from a core point)



p = core object
q = noise object.

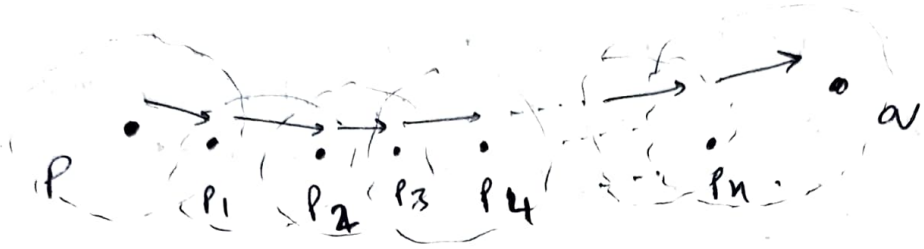
(iv) Density Reachability :- Point p is said to be density reachable from another point q if there are points such as $p_1, p_2, p_3, \ldots, p_n$ in between p and q such that those n points are directly density reachable with one other, then point p is said to be density reachable from point q.

Point p and point q may be core points or border points.

p_1 should be directly density reachable from p.

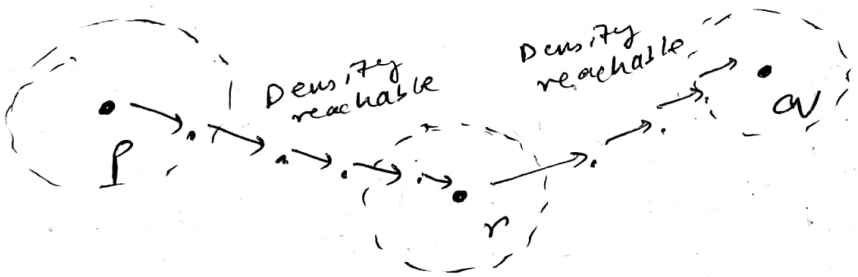
p_n should be directly density reachable from q.

(P.T.O)



point p is density reachable to q.

(v) Density connectivity :- Two points, p and q are said to be density connected if there is another point r, such that p is density reachable from r and q is density reachable from r, then points p and q are density connected to each other.



p and q are density connected to each other.

(P.T.O)

(b) The conditions to be satisfied by DBSCAN clustering algorithm are :-

(i) Maximality condition :- clusters formed by DBSCAN clustering algorithm should have as maximum no. of points possible in the clusters formed, so as to have dense clusters. Also, the algorithm should include as many points as possible in the clusters.

(ii) Connectivity condition :- Points that are produced / put together in a cluster should be density reachable or density connected to each other, such that the algorithm produces dense regions as a cluster and sparse regions as noise. This ensures that density based clustering is done properly.

DBSCAN algorithm :-

(i) Select a point p.
(ii) Traverse all points Density Reachable from p.
(iii) If p = core object, points form a cluster.
(iv) else if p = border object or p does not have minpts no. of points in its ε - neighbourhood, then skip p and goto next point.
(v) Continue doing so till all points are exhausted.

(P. T. O)

(c) **parameters :-**

eps (ε) ⟶ the chosen radius value that dictates how big or small the ε – neighbourhood of a point should be.

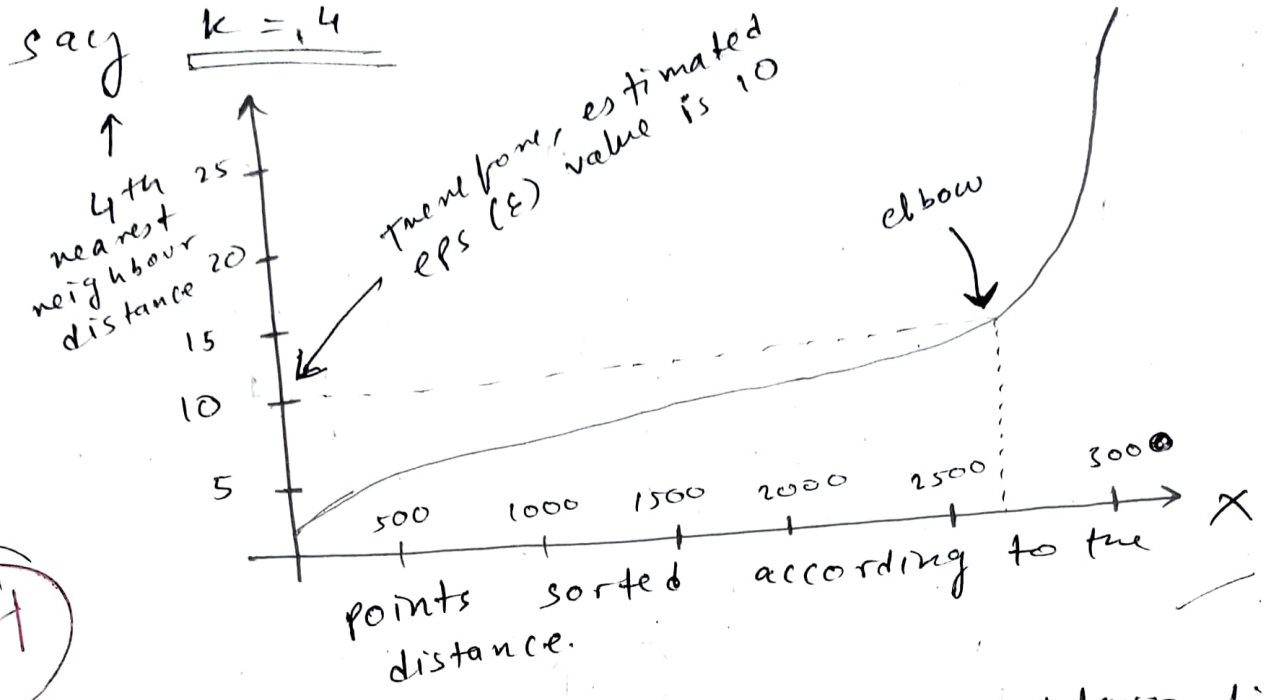
minpts ⟶ chosen value that dictates the min. NO. of points that should be in a ε – neighbourhood to form a dense region.

Choosing minpts. :- An arbitrary minpts value is chosen by the user and given to the DBSCAN algorithm such that we can estimate an ε(eps) value corresponding to a given minpts value.

Choosing eps (ε) :- from given minpts, we assign k = minpts. (chosen).

↳ NOW, we try to find the k^{th} nearest neighbour distance for all given points in the point set and try to plot the graph vs the NO. of points sorted according to the distance

(P.T.O)



(4)

Now, we plot the kth nearest neighbour distance vs the points sorted according to their distance.

Now, then we use the elbow finding method in the graph to select the point of sharp change in the graph.

Corresponding y value to the point of sharp change in the elbow-finding method will give us the appropriate eps (ε) value, that will result in optimal clustering in the DBSCAN algorithm.

(Please Turn Over)

(P.T.O)

(2) (a) External cluster validation :-

(i) Purity :- Purity is the simplest measure of cluster validation where in we find the dominant class in a cluster and find its cardinality. Then we divide the cardinality of the dominant cluster to the no. of points in the cluster to give us the purity measure for an i^{th} cluster.

$$\text{purity}(i) = \frac{1}{n} \times \max_j (n_{ij}) ;$$

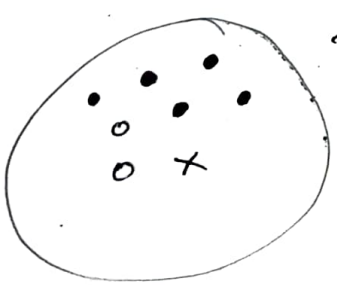
purity measure of i^{th} cluster

n = No. of points in i^{th} cluster

$\max_j (n_{ij})$ = cardinality of dominant class in the i^{th} cluster.

Examples :-

● → class A
o → class B
× → class C



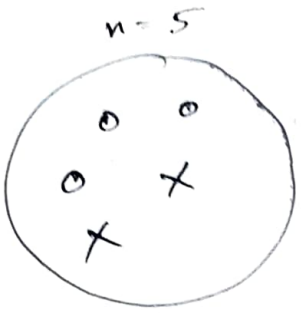
cluster 1

$n = 8$

$$\text{purity} = \frac{1}{8} \times \max(5, 2, 1)$$

$$\text{purity} = \frac{1}{8} \times 5 = \frac{5}{8}$$

dominant class is class A
Cardinality of class A is 5 :-



cluster 2

$$\text{purity} = \frac{1}{5} \times \max(0, 3, 2)$$

$$= \frac{3}{5}$$

dominant class is class B.
so, cardinality of B = 3

---

ii) **Rand index** :- It is a measure, similar to the confusion matrix where in we plot the no. of points in actual clusters to the predicted clusters and try to analyse how much of it is correctly clustered or classified.

3

Predicted clusters → / Actual clusters ↓	predicted ↑ No. of points present in cluster (predicted)	predicted ↑ No. of points not present in cluster.
Actual No. of points in cluster	20 (A)	42 (B)
Actual No. of points not in cluster	45 (C)	30 (D)

A = True positive
B = False negative
C = False positive
D = True negative

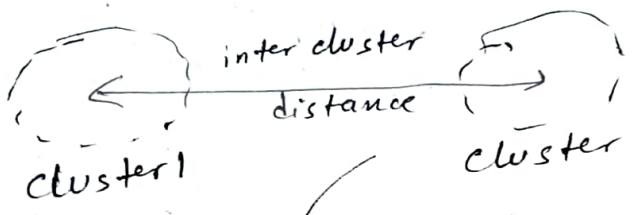
RAND index

$$\text{Rand index (R.I)} = \frac{A + D}{A + B + C + D} = \frac{20 + 30}{20 + 42 + 45 + 30}$$

$$= 0.365$$

(P.T.O)

(b) Inter cluster Distance refers to the measure of the distance between two distinct clusters. (5)



Two inter cluster distance are as follows :-

(i) Centroid Linkage Distance :-
Distance between the centroids of two clusters, given as

$$\delta_1 = d(\bar{v}_S, \bar{v}_T) \quad \text{where}$$

$$\bar{v}_S = \frac{1}{|S|} \cdot \sum \phi(x)$$

$$\bar{v}_T = \frac{1}{|T|} \sum \phi(y)$$

where
S = cluster 1
T = cluster 2
$\bar{v}_S$ = Centroid of cluster 1
$\bar{v}_T$ = Centroid of cluster 2
x = points in cluster 1 $(x \in S)$
y = points in cluster 2 $(y \in T)$

(P-T·O)

(ii) Simple Linkage dist :- The smallest inter cluster distance in between two clusters among their nearest points, given by

$$\delta_2 = \min \left(d(x, y) \right)$$

where $x \in S$, $y \in T$.

(iii) Average Linkage distance :- Average distance computed across / in between all the points in cluster 1 and cluster 2.

5

$$\therefore \delta_3 = \frac{1}{|S| \times |T|} \times \sum_x \sum_y d(x, y)$$

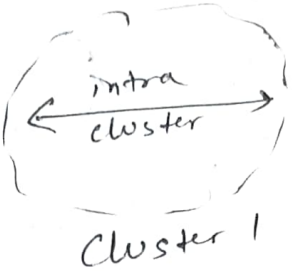
where S = cluster 1
T = cluster 2
$x \in S$, $y \in T$.

Total no. of points in cluster 1 and cluster 2 computed each-one-one, is $|S| \times |T|$.

$d(x, y)$ → distance between object x and object y.

(P. T. O)

Intra cluster distance refers to the width of a single cluster.



Two Types of Intra cluster distance :-

(i) Complete diameter distance :-

Maximum distance between any 2 given points in a cluster. which is equal to the largest diameter of the cluster.

Given by,

$$D_1 = \max\left(d(x, y)\right),$$

where $x \in S$, $y \in S$

(ii) Average diameter distance :- From every point x in cluster S, calculate distance of every other point y and find the average.

Total no of points $|S| \times (|S|-1)$

Given by

$$D_2 = \frac{1}{|S| \cdot (|S|-1)} \times \sum_{x,y} d(x, y)$$

$d(x, y) \rightarrow$ distance between x and y.

(P-T.O)

(iii) Centroid Diameter distance :- Distance of all points x from the centroid $\bar{v}$, divided by half of the cardinality of cluster S.

$$\text{Distance} = \frac{2}{|S|} \times \sum d(x, \bar{v})$$

(A3)

where $\bar{v} = \frac{1}{|S|} \sum x$

(c) Dunn's cluster validation index is an index that tries to maximize the inter cluster distance (δ) and tries to minimize the intra cluster distance (Δ), so that a cluster is well formed and distinct.

Dunn's index is basically, the ratio of the intercluster distance (δ) to the maximum intra cluster distance (Δ).

Dunn's index :- (Mathematically,)

$$DI = \min_{1 \le i \le c} \left\{ \min_{1 \le j \le c} \left\{ \frac{\delta(x_i, x_j)}{\max_{1 \le k \le c} (\Delta(x_k))} \right\} \right\}$$

where x_i = i^{th} cluster
x_j = j^{th} cluster
x_k = k^{th} cluster.
c = Total No. of clusters.

The maximum intra cluster distance among all the clusters is fixed and then it is used as denominator and the numerator is the current inter cluster distance between X_i and X_j.

32

Since (δ) is larger and (Δ) is smaller,

So, Dunn's index produces value ≥ 1.

Larger the value of Dunn's index, better is the clustering.

Smaller the value of Dunn's index, worse is the clustering

This is the usefulness of Dunn's index in clustering problems.

Dunn's index can be used to ~~decide~~ rate / give an index for how good the classification or the clustering is and can be used as a benchmark to improve ~~decisions and~~ algorithms.

(P.T.O)

3) (a) Let X and Y be two items.

Then,

the support can be defined as

$$\text{support} = \frac{(X \cup Y).\text{count}}{n}$$

where, n is the total No. of transactions,

and $(X \cup Y).\text{count}$ is the No. of transactions in which X and Y occur together.

Support can be defined as the ratio of the No. of transactions in which both item X and Y appear together to the total No. of transactions

Confidence can be defined as,

$$\text{confidence} = \frac{(X \cup Y).\text{count}}{X.\text{count}}$$

where $X.\text{count}$ = No. of transactions in which only X appears.

$(X \cup Y).\text{count}$ = No. of transactions in which both X and Y appear.

(P.T.O)

Confidence can be defined as the ratio of the No. of transactions in which both item X and Y appear to the No. of transactions in which only item X appears.

Confidence can be thought of as conditional probability $P(Y/X)$.

4

An item is referred to as an frequent item set, if its support value is greater than or equal to the given minimum support (minsup) value. (support $\geq$ minsup)

An association rule is referred to as an important rule, when its confidence value is greater than or equal to the minimum confidence (minconf) value. (conf $\geq$ minconf)

(b) Given Transactions,

T_1 :- bread, butter

T_2 :- bread, milk, butter

T_3 :- bread, jelly, butter

T_4 :- bread, coke

T_5 :- bread, milk

T_6 :- coke, milk.

minsup = 30%.

minconf = 80%.

n = Total No. of transactions

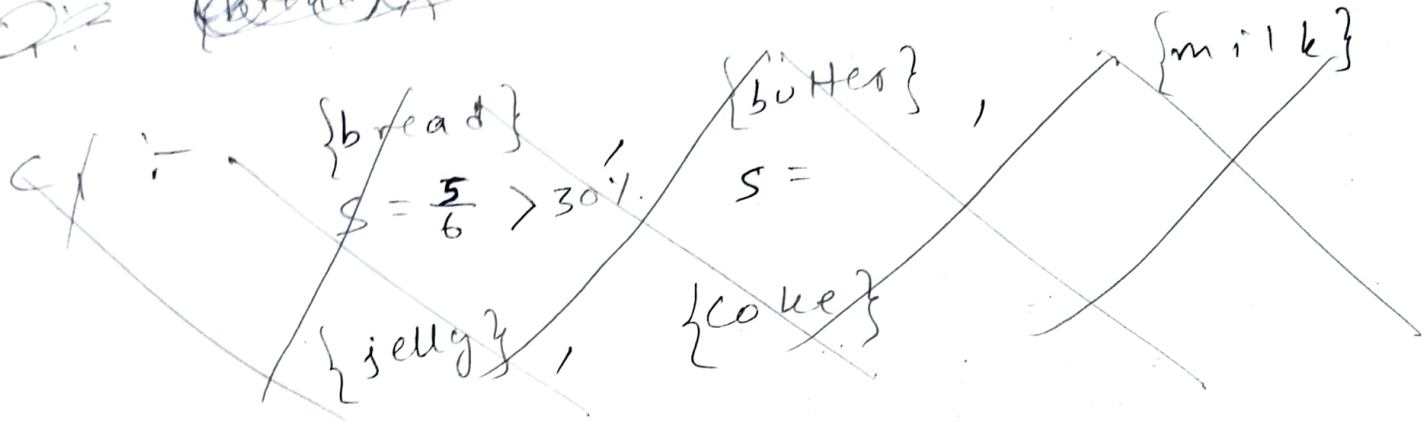
n = 6

Using Apriori Algorithm

Candidate key / frequent item set generation :-

s = support value.

Q:2 ~~{bread}, {butter}~~



C_1 :-
- ✓ {bread} ⟶ $s = \frac{5}{6} > 30\%$. (valid)
- ✓ {butter} ⟶ $s = \frac{3}{6} > 30\%$ (valid)
- ✓ {milk} ⟶ $s = \frac{3}{6} > 30\%$. (valid)
- × {jelly} ⟶ $s = \frac{1}{6} < 30\%$. (invalid)
- ✓ {coke} ⟶ $s = \frac{2}{6} > 30\%$. (valid)

F_1 :- {bread}, {butter}, {milk}, {coke}
(based on support values)

Now,

(P.T.O)

C_2 :- ✓ {bread, butter} → $s = \frac{3}{6} > 30\%$. (valid)

✓ {bread, milk} → $s = \frac{2}{6} > 30\%$. (valid)

× {bread, coke} → $s = \frac{1}{6} < 30\%$. (invalid)

× {butter, milk} → $s = \frac{1}{6} < 30\%$. (invalid)

× {butter, coke} → $s = \frac{0}{6} < 30\%$. (invalid)

× {milk, coke} → $s = \frac{1}{6} < 30\%$ (invalid)

F_2 :- {bread, butter}, {bread, milk}

↑ generate

C_3 :- {bread, butter, milk}

$s = \frac{1}{6} < 30\%$ (invalid)

only 1 candidate possible from candidate generation

Also, we can see that the subset {butter, milk} is not present in F_2. So this combination is not valid in C_3.

So,

F_3 :- ϕ (null set)

So, the Final Frequent itemset can be written as

$$F = \{\{bread, butter\}, \{bread, milk\}\}$$

Now, $minconf = 80\%$.

Association rules :-

	$X \longrightarrow Y$		confidence.
×	bread	butter	$c = \frac{3}{5} < 80\%$ (invalid)
✓	butter	bread	$c = \frac{3}{3} = 100\% > 80\%$ (valid)
×	bread	milk	$c = \frac{2}{5} < 80\%$ (invalid)
×	milk	bread	$c = \frac{2}{3} < 80\%$ (invalid)

5+1

So, as per $minconf = 80\%$, only one association rule can be deemed as important rule.

So,

Important Association Rule :- Butter → Bread

with confidence $= 100\% > minconf = 80\%$

(c) The major draw backs of apriori algorithm are as follows:-

(i) Lack of data or incomplete dataset can result in improper association rules that may not reflect the real scenario.

(ii) Improper tuning of min support and min. confidence values can lead to improper association rules.

(iii) Too much variations in transactions may lead to improper extraction of frequent itemsets that will result in wrong association rules.

(iv) Apriori algorithm can sometimes give wrong association rules when there are conflicting transactions.

2

(P.T.O)

(a) From the Given Dataset in the question-paper, we can calculate the entropy of the class variable as

$$I(p,n) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

5½

Now, p = Yes class value. $p = 6$
n = No class value $n = 4$ } $p+n = 10$.

$$\therefore I(6,4) = -\frac{6}{10}\log_2\left(\frac{6}{10}\right) - \frac{4}{10}\log_2\left(\frac{4}{10}\right)$$

$$= -\left[\frac{6}{10}\times\left(\log_2 2 + \log_2 3 - \log_2 10\right) + \frac{4}{10}\times\left(2\log_2 2 - \log_2 10\right)\right]$$

Now,

$$\log_2 10 = \log_2 2 * \log_2 5$$

$$= 1 + 2.3 = 3.3$$

$$\Rightarrow I(6/4) = -\left[\frac{6}{10}(1 + 1.6 - 3.3) + \frac{4}{10} \times [2 - 3.3]\right)$$

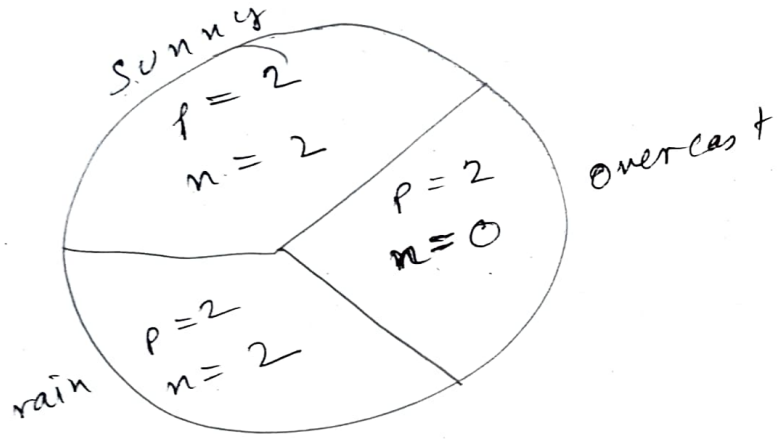
$$= -\left[0.6 \times -0.7 + 0.4 \times -1.3\right]$$

$$= 0.42 + 0.52$$

$$= 0.94$$

Now,

For attribute outlook :-



Outlook.
(classes)

p for sunny => No. of yes outcome for class sunny of attribute outlook

n for sunny => No. of NO outcome for class sunny of attribute outlook

Thus,

E(outlook) = Expected entropy for attribute outlook

Similarly for all classes, we find out p and n.

Now,

$$E(\text{outlook}) = \sum_i \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$= \frac{4}{10} \times I(2,2) + \frac{2}{10} \times I(2,0) + \frac{4}{10} \times I(2,2)$$

Now,

$$I(2,0) = -\frac{2}{2}\log_2 1 - 0$$

$$I(2,0) = 0$$

$$I(2,2) = -\frac{2}{4}\log_2 \frac{2}{4} - \frac{2}{4}\log_2 \frac{2}{4}$$

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

$$= -\left[\log_2 2 - \log_2 4\right]$$

$$= -[1-2]$$

$$I(2,2) = 1$$

$$\Rightarrow E(\text{outlook}) = \frac{4}{10} \times 1 + 0 + \frac{4}{10} \times 1$$

$$= \frac{8}{10} = 0.8$$

$$\Rightarrow \boxed{E(\text{outlook}) = 0.8}$$

for Attribute Humidity :

Humidity
(2 classes)

Now,

$$I(4,1) = -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}$$

$$= -\left[\frac{4}{5}\times\left[2\log_2 2 - \log_2 5\right] + \frac{1}{5}\left[\log_2 1 - \log_2 5\right]\right]$$

$$= -\left[\frac{4}{5}\times\left[2 - 2.3\right] + \frac{1}{5}\times -2.3\right]$$

$$= +\frac{4}{5}\times 0.3 + \frac{1}{5}\times 2.3$$

$$0.7$$

$$I(2,3) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$$

$$= -\left[\frac{2}{5}\times(\log 2 - \log 5) + \frac{3}{5}(\log 3 - \log 5)\right]$$

$$= \frac{2}{5}\times 1.3 + \frac{3}{5}\times 0.7$$

$$0.94$$

Now

$$I(\text{humidity}) = \frac{5}{10}\times I(4,1) + \frac{5}{10}\times I(2,3)$$

Examination Roll No. 510819011 Sheet No. 1

Subject Code CS-4261

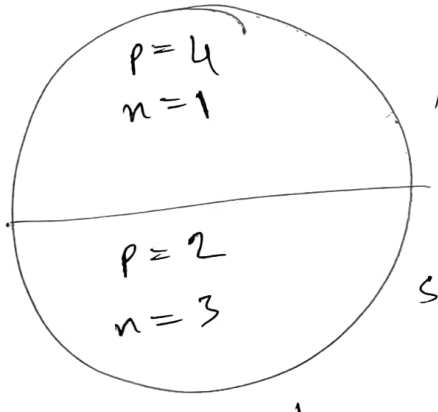
11/5/20

Signature(s) of the Invigilator with date

$$E(\text{humidity}) = 0.5 \times 0.7 + 0.5 \times 0.94$$

$$= 0.82$$

for attribute wind,



$$E(\text{wind}) = \frac{5}{10} \times I(2,3) + \frac{5}{10} \times I(4,1)$$

using previously calculated values,

$$E(\text{wind}) = 0.5 \times 0.94 + 0.5 \times 0.7$$

$$= 0.82$$

NOW,
we have

$$E(\text{outlook}) = 0.8$$

$$E(\text{humidity}) = 0.82$$

$$E(\text{wind}) = 0.82$$

(P-T·O)

Now, information gain for each attribute

Gain (outlook) = I(p, n) − E(outlook)

= 0.94 − 0.8

gain(outlook) = 0.14

gain (humidity) = 0.94 − 0.82

= 0.12.

gain (wind) = I(p, n) − E(wind)

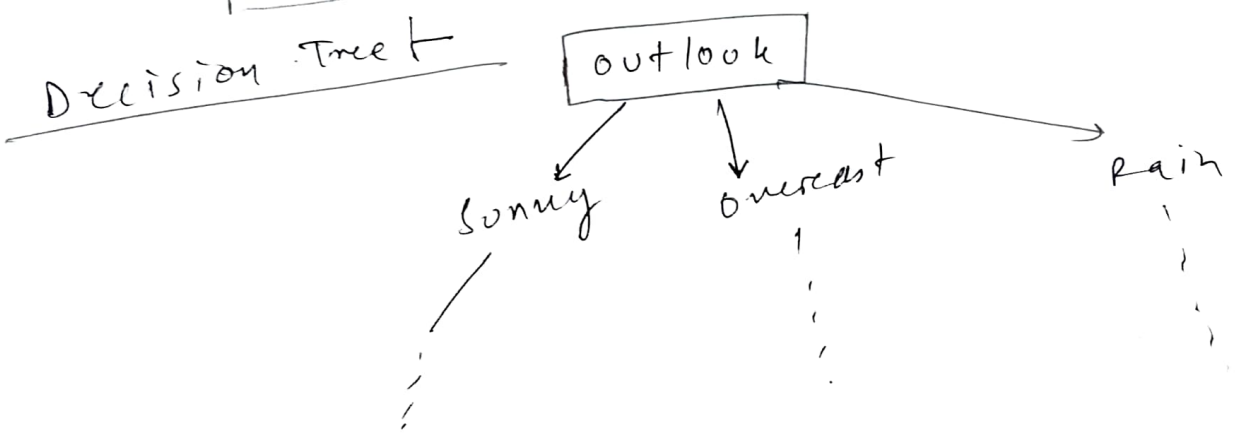
= 0.94 − 0.82

= 0.12.

Now, we have maximum information gain for attribute outlook

So, outlook is the root for the decision tree.

Decision Tree :-



(b) The possible terminating criteria for a decision tree algorithm are:

(i) All attributes exhausted and no more attributes are left to be added to tree.

(ii) The tree is complete without taking into consideration, all possible attributes, i.e., some attributes influence the final outcome so less that they are not needed in the decision tree.

3

(iii) from given attributes, no complete decision tree is possible, i.e very few attributes but too many variations in tuples to be factored in.

(c) The causes of model overfitting are noisy data present in the table or too many variations that do not influence a proper outcome.

Overfitting can be defined as high training accuracy but low validation accuracy.

The overfitting problem in decision trees can be solved using

(i) Pre pruning (ii) Post-pruning

(P.T.O)

Pre pruning :- we prune the branches in a decision tree, while we are creating it. i.e, we perform pruning and creation simultaneously till we get a final tree.

3

Post pruning :- we create a complete decision tree first, then we prune branches, depending on conflicts present in tree and table (misclassifications)

---

(5) (a) Bayesian classification

$$P(C_i/X) = P(C_i) \times P(X/C_i)$$

Given tuple,

$X = (\text{outlook} = \text{overcast}, \text{humidity} = \text{high}, \text{wind} = \text{weak})$

$P(C_i = \text{Yes}) = \frac{6}{10}$

$P(C_i = \text{NO}) = \frac{4}{10}$

Examination Roll No. 510519011 Sheet No. 2

Subject Code CS 4261

11/5/23

Signature(s) of the Invigilator with date

C_i = yes

$P(\text{outlook} = \text{overcast} / \text{yes}) = \frac{2}{6}$

$P(\text{humidity} = \text{high} / \text{yes}) = \frac{2}{6}$

$P(\text{wind} = \text{weak} / \text{yes}) = \frac{4}{6}$

4/2

Now,

$P(X / C_i = \text{yes}) = \frac{2}{6} \times \frac{2}{6} \times \frac{4}{6}$

$= \frac{16}{256} = \frac{2}{27}$

$= 0.074$

$P(X / C_i = \text{NO}) = \frac{1}{7} \times \frac{4}{7} \times \frac{2}{7}$

$= 0.023$

C_i = NO

$P(\text{outlook} = \text{overcast} / \text{NO}) = \frac{0}{4}$

$P(\text{humidity} = \text{high} / \text{NO}) = \frac{3}{4}$

$P(\text{wind} = \text{weak} / \text{NO}) = \frac{1}{4}$

Here we have zero probability problem.

So we use Laplacian correction (+1 to all cases)

$P(\text{overcast} / \text{NO}) = \frac{1}{7}$

$P(\text{high} / \text{NO}) = \frac{4}{7}$

$P(\text{weak} / \text{NO}) = \frac{2}{7}$

Total 3 cases = so denominator +3 in all

numerator +1 in all..

Using Laplacian Correction to avoid zero probability problem.

Now,
for given tuple X,

$$P(C_i = Yes / X) = P(C_i = Yes) \times P(X / C_i = Yes)$$

$$= \frac{6}{10} \times 0.074$$

$$= 0.044$$

$$P(C_i = NO / X) = P(C_i = NO) \times P(X / C_i = NO)$$

$$= \frac{4}{10} \times 0.023$$

$$= 0.0092.$$

So, For C_i = Yes, we have the higher probability,

So,
predicted class Label = YES

for
given tuple X = (outlook = overcast, humidity = high, wind = weak)

(P. T. O)

(b) The main error in bayesian classification problem is the zero-probability error, where we may get a value 0 corresponding to a probability.

[But upon multiplication, we can get the total class value probability as zero, whereas it may not be true in the real scenario.]

This is the zero probability problem.

Such a problem can be corrected using Laplacian correction, where in we add +1 to all possible probabilities that may be affected by a zero value.

Suppose,

$P(A) = \frac{0}{7}$

$P(B) = \frac{3}{7}$

$P(C) = \frac{4}{7}$

To avoid zero probability, we use Laplacian correction and add 1 to all numerator probabilities.

So denominator becomes (+3) for all cases as we add (+1) to 3 numerators in 3 probabilities.

After Laplacian correction,

$P(A) = \frac{1}{10}$; $P(B) = \frac{4}{10}$ / $P(C) = \frac{5}{10}$

(c) If humidity attribute had continuous values, then we would have had to convert the continuous values to a set of discrete class values using various discretization techniques in data pre processing. Then, we could have applied decision Tree / Bayesian Algorithm as Decision tree works only on discrete values.