Data Mining

Category	
Files	
Created	@October 30, 2022 7:53 PM
Reminder	
Status	Open
URL	
Updated	@December 20, 2022 7:41 PM

▼ Sessional 1

High level topics -

- Frequency Pattern Growth tree
- Decision Tree
- Apriori Algoithm
- naive bayes

Support = How frequently the combination of X and y occur in a transaction database

Freq(x,y) / N

Confidence = How frequently the combination of X and Y occurs in those transaction which X occurs in a transaction database.

Freq(x,y) / freq(x)

where

Freq(x,Y) is the combination of items X and Y in a Market basket transaction database.

N is the total no of transaction

https://www.chegg.com/flashcards/chapter-6-data-mining-basic-algorithms-and-evaluation-methods-51ddbb08-0be1-454f-8b12-c9a57fc8279e/deck

https://t4tutorials.com/support-confidence-minimum-support-frequent-itemset-in-data-mining/

https://www.javatpoint.com/data-cleaning-in-data-mining

▼ Associate Rule Mining

https://medium.com/analytics-vidhya/association-rule-feature-mining-arm-in-network-intrusion-detection-system-6174a06789f0

https://www.geeksforgeeks.org/association-rule/

Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. A typical example is a Market Based Analysis.

Market Based Analysis is one of the key techniques used by large relations to show associations between items.It allows retailers to identify relationships between the items that people buy together frequently.

Data Cleansing -

- **Remove duplicate or irrelevant observations**
- **Fix structural errors**
- **Filter unwanted outliers**
- **Handle missing data**
- **Validate and QA**

▼ Decision tree

https://www.geeksforgeeks.org/decision-tree-introduction-example/

https://www.tutorialspoint.com/data_mining/dm_dti.htm#:~:text=A decision tree is a,tree is the root node.

▼ Apriori Algo

https://towardsdatascience.com/data-mining-market-basket-analysis-with-apriori-algorithm-970ff256a92c

https://t4tutorials.com/apriori-algorithm-in-data-mining-with-examples/

https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

▼ FP (Frequent Pattern)Growth

- https://www.mygreatlearning.com/blog/understanding-fp-growth-algorithm/
- https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/
- https://www.youtube.com/watch?v=7oGz4PCp9jI&ab_channel=MaheshHuddar

▼ Improve Apriori Algorithm

https://towardsdatascience.com/market-basket-analysis-multiple-support-frequent-item-set-mining-584a311cae66

https://www.tutorialspoint.com/how-can-we-further-improve-the-efficiency-of-apriori-based-mining

https://www.engati.com/glossary/apriori-algorithm

https://www.youtube.com/watch?v=mgUTxzEUyNk&ab_channel=YaminiRichhariya

- hashing
- partitioning
- sampling
- transaction reduction

▼ naive bayes

https://www.youtube.com/watch?v=mzPHmNm_NrM&ab_channel=5MinutesEngineering

https://www.youtube.com/watch?v=Nw27UVn9d0Y&list=PLnX8rkvPNtpXbEbuzGv74uQV-aSUt4KnG&ab_channel=PremnArya

https://www.youtube.com/playlist?list=PLV8vIYTIdSnb4H0JvSTt3PyCNFGGlO78u

notes-

- https://mrcet.com/downloads/digital_notes/CSE/IV Year/DATA WAREHOUSING&DATA MINING.pdf
- https://www.vssut.ac.in/lecture_notes/lecture1428550844.pdf

▼ Sessional 2

Bayesian Belief Networks -
https://www.youtube.com/watch?v=c5eaKymcwiU&t=2s&ab_channel=EasyEngineeringClasses

https://www.youtube.com/watch?v=88Rg58t9114&ab_channel=SimplyComputerScience

k means -

https://www.youtube.com/watch?v=YWgcKSa_2ag&list=PL0s3O6GgLL5fuVR545mzIuCkgGLi02fkN&index=8&ab_channel=Lastmomenttuitions

https://www.youtube.com/watch?v=CLKW6uWJtTc&t=125s&ab_channel=5MinutesEngineering

https://www.youtube.com/watch?v=JUXJSbfqeHg&ab_channel=RANJIRAJ

k mediod -

https://www.youtube.com/watch?v=AUriFHKw0TU&ab_channel=Lastmomenttuitions

https://www.youtube.com/watch?v=RqT_h5vX5P0&ab_channel=HelpingTutorials

dbscan -

https://www.youtube.com/watch?v=1RDyVAoFKgY&ab_channel=UnfoldDataScience

https://www.youtube.com/watch?v=e9WjLdTm3KU&list=PLPN-43XehstOe0CxcXaYeLTFpgD2IiluP&index=24&ab_channel=RANJIRAJ

https://www.youtube.com/watch?v=ZkyQ4rNIFvE&ab_channel=5MinutesEngineering

https://www.youtube.com/watch?v=8if4g3rsTMg&ab_channel=Lastmomenttuitions

optics -

https://www.youtube.com/watch?v=9Zy-JfgAHRc&ab_channel=DhanyajaN

random links -

https://analyticsindiamag.com/apriori-vs-fp-growth-in-market-basket-analysis-a-comparative-guide/

https://textbooks.elsevier.com/manualsprotectedtextbooks/9780123814791/Instructor's_manual.pdf

https://towardsdatascience.com/almost-everything-you-need-to-know-about-decision-trees-with-code-dc026172a284

Answers to semester questions -

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/2993b14e-1b3c-410b-9d14-30b02fe66c0d/DM_Paper_2018_Sol.pdf

▼ 2019 -

▼ Ans 1 (a) - Describe three challenges to data mining regarding data mining methodology and user interaction issues. -

There are several challenges that can arise in data mining, both in terms of the methodology used and in terms of user interaction issues. Here are three examples:

1. Data quality: One challenge in data mining is ensuring that the data being analyzed is of high quality. This can be a problem if the data is incomplete, noisy, or biased, as it can lead to incorrect conclusions being drawn from the analysis. To address this challenge, it is important to carefully pre-process the data and apply techniques such as missing data imputation and outlier detection to ensure that the data is clean and ready for analysis.
2. Algorithmic bias: Another challenge in data mining is the potential for algorithmic bias, which occurs when the algorithms used to analyze the data produce results that are biased or unfair. This can be a problem if the data used to train the algorithms is biased, or if the algorithms themselves are designed to produce biased results. To address this challenge, it is important to carefully select and evaluate the algorithms used in data mining, and to ensure that they are free from biases that could affect the results of the analysis.
3. User interaction issues: Data mining can also be challenged by user interaction issues, such as a lack of understanding or expertise on the part of the user. For example, the user may not have a strong understanding of the data mining process or may not know how to interpret the results of the analysis. To address this challenge, it is important to provide users with the necessary training and support to help them understand and effectively use the results of the data mining process.

Overall, these are just a few examples of the many challenges that can arise in data mining. To be successful, it is important to carefully consider and address these challenges to ensure that the data mining process is effective and produces accurate and useful results.

▼ Ans 1 (b) - Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8), calculates euclidean, manhatten and miyosaki distance

- (22, 1, 42, 10) and (20, 0, 36, 8):

(a) Compute the **Euclidean** distance between the two objects.

$$= \sqrt{(22-20)^2 + (1-0)^2 + (42-36)^2 + (10-8)^2}$$

$$= \sqrt{45}$$

$$= 6.7082$$

- (22, 1, 42, 10) and (20, 0, 36, 8):

(b) Compute the **Manhattan** distance between the two objects

$$= |22-20| + |1-0| + |42-36| + |10-8|$$

$$= 11$$

- (22, 1, 42, 10) and (20, 0, 36, 8):

(c) Compute the **Minkowski** distance between the two objects, using h = 3

$$\sqrt[3]{|22-20|^3 + |1-0|^3 + |42-36|^3 + |10-8|^3}$$

$$= \sqrt[3]{233}$$

$$= 6.1534$$

▼ Ans 1 (b`) - Suppose that for two vectors A and B, we know that their Euclidean distance is less than d. What can be said about their Manhattan distance?

If the Euclidean distance between two vectors A and B is less than d, it does not necessarily follow that the Manhattan distance between the vectors is also less than d. The Manhattan distance between two vectors is defined as the sum of the absolute differences of the corresponding components, while the Euclidean distance is defined as the square root of the sum of the squares of the differences of the corresponding components.

For example, consider the vectors A = (1, 2) and B = (3, 4). The Euclidean distance between A and B is sqrt((3-1)^2 + (4-2)^2) = 2.828, which is less than 3. However, the Manhattan distance between A and B is |3-1| + |4-2| = 4, which is not less than 3.

In general, the Manhattan distance can be larger than the Euclidean distance, as it does not take into account the geometric relationships between the components of the vectors. However, it is also possible for the Manhattan distance to be smaller than the Euclidean distance, depending on the specific values of the components of the vectors.

Hint:

$$\overbrace{(\Delta x)^2 + (\Delta y)^2}^{\text{square of the Euclidean distance}} \le (\Delta x)^2 + 2|\Delta x \Delta y| + (\Delta y)^2 = \overbrace{(|\Delta x| + |\Delta y|)^2}^{\text{square of the Manhattan distance}}$$

Furthermore, since the square of a real number is non-negative,

$$(\Delta x)^2 - 2|\Delta x \Delta y| + (\Delta y)^2 = (|\Delta x| - |\Delta y|)^2 \ge 0$$

we can add $(|\Delta x| + |\Delta y|)^2$ to both sides of (2) to get

$$2\overbrace{\left[(\Delta x)^2 + (\Delta y)^2\right]}^{\text{square of the Euclidean distance}} \ge \overbrace{(|\Delta x| + |\Delta y|)^2}^{\text{square of the Manhattan distance}}$$

▼ Ans 2 (a) i - Give a short example to show that items in a strong association rule actually may be negatively correlated.

here is an example of items in a strong association rule that may be negatively correlated:

Imagine that we are analyzing data on shopping habits at a grocery store. One of the association rules we discover is:

Rule: If a customer buys a bag of chips, they are very likely to also buy a bottle of soda.

Support: 0.8 (this means that 80% of the time, customers who buy chips also buy soda) Confidence: 0.9 (this means that 90% of the time, customers who buy chips also buy soda)

However, we also know that chips and soda are often considered unhealthy snacks, and there is a trend in the market towards buying healthier options. In this case, even though the association rule shows a strong relationship between buying chips and buying soda, the items may actually be negatively correlated in terms of healthiness.

In other words, even though customers who buy chips are very likely to also buy soda, the negative perception of these items as unhealthy may lead to a decrease in their overall sales over time. This shows that even though items in a strong association rule may seem closely related, there can be other factors at play that influence their relationship.

another answer -

Example 1. Let us consider an example from market basket data. In
this example we want to study the purchase of cow's milk (CM) versus
soy milk (SM) in a grocery store. Table 1.2 gives us the data collected
from 100 baskets in the store. In Table 1.2 "CM" means the basket
contains cow's milk and "¬ CM" means the basket does not contain
cow's milk. The same applies for soy milk.
In this data, let us find the positive association rules in the "supportconfidence" framework. The
association rule "SM ⇒ CM" has 20%
support and 25% confidence (support(SM ∧ CM)/support(SM)). The
association rule "CM ⇒ SM" has 20% support and 50% confidence (support(SM ∧ CM)/support(CM)).
The support is considered fairly high for
both rules. Although we may reject the first rule on the confidence basis, the second rule seems a valid
rule and may be considered in the

data analysis. However, when a statistical significance test is considered, such as statistical correlation between the SM and CM items, one
would find that the two items are actually negatively correlated. This
6
shows that the rule “CM ⇒ SM” is misleading. This example shows not
only the importance of considering negative association rules, but also
the importance of statistical significance of the patterns discovered.

also this -

6.3.2 From Association Analysis to Correlation Analysis

As we have seen above, the support and confidence measures are insufficient at filtering out uninteresting association rules. To tackle this weakness, a correlation measure can be used to augment the support-confidence framework for association rules. This leads to *correlation rules* of the form

$$A \Rightarrow B \ [support, \ confidence, \ correlation]. \tag{6.7}$$

That is, a correlation rule is measured not only by its support and confidence but also by the correlation between itemsets A and B. There are many different correlation measures from which to choose. In this section, we study several correlation measures to determine which would be good for mining large data sets.

Lift is a simple correlation measure that is given as follows. The occurrence of itemset A is **independent** of the occurrence of itemset B if $P(A \cup B) = P(A)P(B)$; otherwise, itemsets A and B are **dependent** and **correlated** as events. This definition can easily be extended to more than two itemsets. The **lift** between the occurrence of A and B can be measured by computing

$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}. \tag{6.8}$$

If the resulting value of Equation (6.8) is less than 1, then the occurrence of A is *negatively correlated with* the occurrence of B, meaning that the occurrence of one likely leads to the absence of the other one. If the resulting value is greater than 1, then A and B are *positively correlated*, meaning that the occurrence of one implies the occurrence of the other. If the resulting value is equal to 1, then A and B are *independent* and there is no correlation between them.

Equation (6.8) is equivalent to $P(B|A)/P(B)$, or $conf(A \Rightarrow B)/sup(B)$, which is also referred as the *lift* of the association (or correlation) rule $A \Rightarrow B$. In other words, it assesses the degree to which the occurrence of one “lifts” the occurrence of the other. For example, if A corresponds to the sale of computer games and B corresponds to the sale of videos, then given the current market conditions, the sale of games is said to increase or “lift” the likelihood of the sale of videos by a factor of the value returned by Equation (6.8).

Let’s go back to the computer game and video data of Example 6.7.

▼ Ans 2 (a) ii - association rules

(b) The following contingency table summarizes supermarket transaction data, where hot dogs refers to the transactions containing hot dogs, hot dogs refers to the transactions that do not contain hot dogs, hamburgers refers to the transactions containing hamburgers, and hamburgers refers to the transactions that do not contain hamburgers.

	hot dogs	$\overline{\textit{hot dogs}}$	Σ_{row}
hamburgers	2000	500	2500
$\overline{\textit{hamburgers}}$	1000	1500	2500
Σ_{col}	3000	2000	5000

Answer:

(a) Suppose that the association rule "*hotdogs* $\Rightarrow$ *hamburgers*" is mined. Given a minimum support threshold of 25% and a minimum confidence threshold of 50%, is this association rule strong?
For the rule, support $= 2000/5000 = 40\%$, and confidence $= 2000/3000 = 66.7\%$. Therefore, the association rule is strong.

(b) Based on the given data, is the purchase of *hotdogs* independent of the purchase of *hamburgers*? If not, what kind of *correlation* relationship exists between the two?
$corr_{\{hotdog,hamburger\}} = P(\{hot\ dog,\ hamburger\})/(P(\{hot\ dog\})\ P(\{hamburger\}))$=0.4/(0.5 $\times$ 0.6) = 1.33 > 1. So, the purchase of hotdogs is NOT independent of the purchase of hamburgers. There exists a POSITIVE correlation between the two.

2 Question 2

2.1 Description

The following contingency table summarizes supermarket transaction data, where *hot dogs* refer to the transactions containing hot dogs, $\overline{hot\ dogs}$ refers to the transactions that do not contain hot dogs, *hamburgers* refers to the transactions containing hamburgers, and $\overline{hamburgers}$ refers to the transactions that do not contain hamburgers.

	hot dogs	$\overline{hot\ dogs}$	Σ_{row}
hamburgers	2000	500	2500
$\overline{hamburgers}$	1000	1500	2500
Σ_{col}	3000	2000	5000

Figure 4: Contingency table

1. Suppose that the association rule "*hot dogs* $\Rightarrow$ *hamburgers*" is mined. Given a minimum support threshold of 25% and a minimum confidence threshold of 50% , is this association rule strong?
2. Based on the given data, is the purchase of *hot dogs* independent of the purchase of *hamburgers*? If not, what kind of *correlation* relationship exists between the two?
3. Compare the use of the *all_confidence, max_confidence, Kulczynski*, and *cosine* measures with *lift* and *correlation* on the given data.

2.2 Answer

1. In order for the association rule to be strong the support should be greater then the minimum support threshold and the confidence should be greater then the minimum confidence threshold.

In our case,

$$sup = \frac{\sigma(hotdog, Hamburger)}{|T|} = \frac{2}{5} = 40\%$$

$$conf = \frac{\sigma(hotdog, Hamburger)}{\sigma(hotdog)} = \frac{2}{3} = 66.7\%$$

Both are greater then their thresholds so we can say that the association rule is strong.

2. In order to check the dependence and correlation between two associations, we should calculate their $lift$. Two itemsets are independent when the occurrence of one(A) is independent of the occurrence of the other(B). That occurs when,

$$P(A \cup B) = P(A)\,P(B)$$

And this means that the $lift$ equals to 1 calculated by the following equation.

$$lift = \frac{P(AB)}{P(A)P(B)}$$

In case the $lift$ is greater then 1, then the itemsets are positively correlated and if it is less then 1, the itemsets are negatively correlated. So,

$$lift = \frac{P(hotdog \cup hamburger)}{P(hotdog)P(hamburger)} = \frac{2000/5000}{(3000/5000)(2500/5000)} = 1.33$$

Then we can say that the lift is greater then 1 and the itemsets are positively correlated.

3. In order to calculate *all_confidence, max_confidence, Kulczynski*, and *cosine* measures with *lift* and *correlation* we will use the following equations.

$$AllConf = \frac{sup(AB)}{max(sup(A), sup(B))}$$

$$MaxConf = max\left(\frac{sup(AB)}{sup(A)}, \frac{sup(AB)}{sup(B)}\right)$$

$$Kulc = \frac{sup(AB)}{2}\left(\frac{1}{sup(A)} + \frac{1}{sup(B)}\right)$$

$$Cosine = \frac{sup(AB)}{\sqrt{sup(A)sup(B)}}$$

Lift will be calculated by using the equation from the previous sub-question.

▼ Ans 2 (b) - apriiori, fp tree

2. (b) A database has five transactions. Let min sup = 55% and min conf = 75%.

TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y }
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

(a) Find all frequent itemsets using Apriori and FP-growth, respectively. Compare the efficiency of the two mining processes.
(b)) List all the strong association rules (with support s and confidence c) matching the following metarule, where X is a variable representing customers, and itemi denotes variables representing items (e.g., "A," "B,"):

Similar question -

A database has 5 transactions. Let min sup = 60% and min conf = 80%.

TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y }
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I ,E}

a) Find all frequent itemsets using Apriori and FB–growth.
b) List all of the strong association rules (with support s and confidence c) matching the following metarule, where X is a variable representing customers, and item i denotes variables representing items (e.g., "A", "B", etc.):

$$\forall x \in transaction,\ buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3) \quad [s, c]$$

Apriori algorithm

Apriori:

C1 =

m	3
o	3
n	2
k	5
e	4
y	3
d	1
a	1
u	1
c	2
i	1

L1 =

m	3
o	3
k	5
e	4
y	3

C2 =

mo	1
mk	3
me	2
my	2
ok	3
oe	3
oy	2
ke	4
ky	3
ey	2

L2 =

mk	3
ok	3
oe	3
ke	4
ky	3

C3=

oke	3
key	2

L3=

oke	3

- Finally resulting in the complete set of frequent itemsets:

{ e, k, m, o, y, ke, oe, mk, ok, ky, oke }

FB–Growth algorithm

1. Scan DB once, find frequent 1–itemset (single item pattern) their support => 3

M	3
O	3
N	2
K	5
E	4
Y	3
D	1
A	1
U	1
C	2
I	1

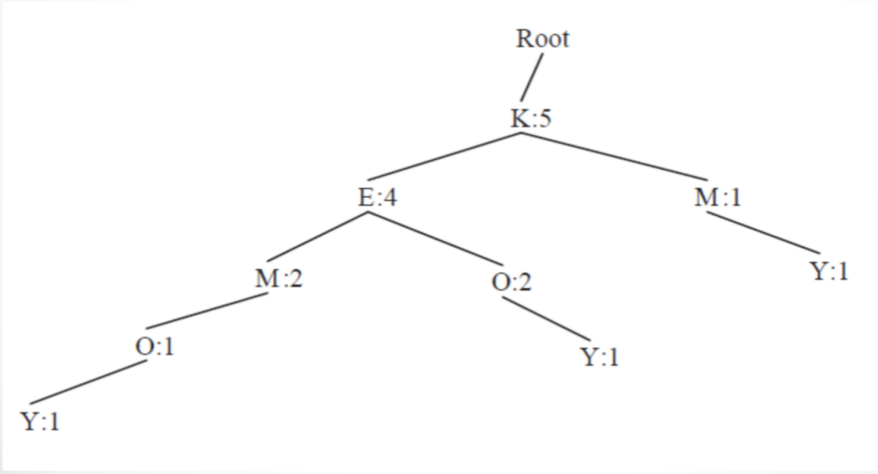
After checking support

K	5
E	4
M	3
O	3
Y	3

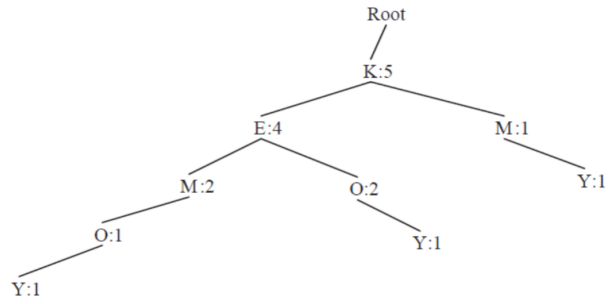
TID	items bought	(ordered) Frequent items
T100	{M, O, N, K, E, Y}	K,E,M,O,Y
T200	{D, O, N, K, E, Y }	K,E,O,Y
T300	{M, A, K, E}	K,E,M
T400	{M, U, C, K, Y}	K, M, Y
T500	{C, O, O, K, I ,E}	K,E,O

FB–Growth algorithm

- Generate FB–tree



- Generate FB–tree – order table



item	conditional pattern base	conditional tree	frequent pattern
y	{ {k,e,m,o:1}, {k,e,o:1}, {k,m:1} }	k:3	{k,y:3}
o	{ {k,e,m:1}, {k,e:2}	k:3,e:3	{k,o:3}, {e,o:3}, {k,e,o:3}
m	{ {k,e:2}, {k:1} }	k:3	{k,m: 3}
e	{ {k:4} }	k:4	{ k,e:4 }

- $buys(X,k) \Lambda buys(X,o) => buys(X, e)$ [60%,100%]
- $buys(X,e) \Lambda buys(X,o) => buys(X, k)$ [60%,100%]

▼ Ans 2 (b`) - Explain FP-Tree algorithm using an example.

The FP-Tree (Frequent Pattern Tree) algorithm is a data structure and algorithm for mining frequent patterns in a large data set. It is used in association rule mining and is particularly effective for finding frequent itemsets in transactional data.

To illustrate how the FP-Tree algorithm works, let's consider the following example:

Suppose we have a data set of 1000 transactions, each with a list of items purchased. We want to use the FP-Tree algorithm to find the most frequent itemsets in the data.

1. First, we compute the frequency of each item in the data set and sort the items in decreasing order of frequency. This helps to reduce the size of the FP-Tree and improve the efficiency of the algorithm.
2. Next, we build the FP-Tree by inserting the items in each transaction into the tree in the order of their frequency. For example, if the most frequent item is item A, it will be inserted as the root of the tree. If item B is the second most frequent item, it will be inserted as a child of the root, and so on.
3. As the FP-Tree is built, we also maintain a list of the frequency of each item in the tree. This helps to identify the frequent itemsets in the data.
4. Once the FP-Tree is built, we can use it to mine the frequent itemsets by traversing the tree and combining the items at each node. For example, if item A and item B are both frequent items, we can combine them to form the frequent itemset {A, B}.

In this example, the FP-Tree algorithm was able to efficiently mine the frequent itemsets in the data by building a tree structure and maintaining the frequency of each item. This can be useful for identifying patterns and associations in transactional data.

https://www.javatpoint.com/fp-growth-algorithm-in-data-mining#:~:text=For I3%2C the prefix path,I2%2C I1%3A4}.

https://www.mygreatlearning.com/blog/understanding-fp-growth-algorithm/

https://www.geeksforgeeks.org/ml-frequent-pattern-growth-algorithm/

▼ Ans 3 (a) - k means

3. (a) Cluster the following eight points (with (x, y) representing locations) into three clusters:

A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).

The distance function between two points a = (x1, y1) and b = (x2, y2) is defined as-

P(a, b) = |x2 – x1| + |y2 – y1|

Use K-Means Algorithm to find the three cluster centers after the second iteration.

https://www.gatevidyalay.com/k-means-clustering-algorithm-example/

▼ Ans 3 (b) - k mediod

https://www.youtube.com/watch?v=RqT_h5vX5P0&ab_channel=HelpingTutorials

https://medium.com/@ali.soleymani.co/beyond-scikit-learn-is-it-time-to-retire-k-means-and-use-this-method-instead-b8eb9ca9079a

▼ Ans 3 (b`) - Explain hierarchical clustering algorithm using an example.

Hierarchical clustering is a type of unsupervised machine learning algorithm that is used to group data points into clusters based on their similarity. There are two main types of hierarchical clustering algorithms: agglomerative and divisive. Agglomerative hierarchical clustering starts with each data point in its own cluster and combines them into larger clusters as the algorithm progresses. Divisive hierarchical clustering starts with all of the data points in a single cluster and divides them into smaller clusters as the algorithm progresses.

To illustrate how hierarchical clustering works, let's consider the following example:

Suppose we have a data set with 8 samples and 2 features. We want to use hierarchical clustering to group the samples into clusters based on their similarity.

1. First, we compute the pairwise distances between all of the samples using a distance metric such as Euclidean distance. This results in a 8x8 distance matrix, as shown below:
2. Next, we apply the hierarchical clustering algorithm to the distance matrix. In agglomerative hierarchical clustering, we start by assigning each sample to its own cluster. In divisive hierarchical clustering, we start by assigning all of the samples to a single cluster.
3. We then iteratively combine or divide the clusters based on the distances between the samples. For example, in agglomerative hierarchical clustering, we may combine the two samples that are closest to each other into a single cluster. In divisive hierarchical clustering, we may divide the samples into two clusters based on the distance between them.
4. This process continues until all of the samples are grouped into a single cluster or a pre-determined number of clusters. The resulting cluster structure can be visualized using a dendrogram, as shown below:

In this example, hierarchical clustering was able to group the samples into two clusters based on their similarity. This can be useful for identifying patterns and structure in the data and for grouping similar samples together.

https://www.kdnuggets.com/2019/09/hierarchical-clustering.html

https://medium.com/@rohanjoseph_91119/learn-with-an-example-hierarchical-clustering-873b5b50890c

https://www.javatpoint.com/hierarchical-clustering-in-machine-learning

▼ Ans 4 (a) - Explain non-linear SVM in detail

Support Vector Machines (SVMs) are a type of supervised machine learning algorithm that can be used for classification and regression tasks. They work by finding a hyperplane in the feature space that maximally separates the different classes in the data.

Non-linear SVMs are an extension of linear SVMs that can handle data that is not linearly separable. They do this by mapping the data to a higher-dimensional space where it may be linearly separable.

To understand how non-linear SVMs work, let's consider the following example:

Suppose we have a data set with 1000 samples and 2 features. The data is not linearly separable, as shown in the following figure:

1. First, we map the data to a higher-dimensional space using a non-linear function, such as a polynomial or a radial basis function. In this higher-dimensional space, the data may now be linearly separable.
2. Next, we apply a linear SVM to the mapped data to find the hyperplane that maximally separates the different classes. This hyperplane will be a non-linear boundary in the original feature space.
3. We can then use the non-linear SVM to make predictions on new data by mapping it to the higher-dimensional space and applying the hyperplane to make a decision.

In summary, non-linear SVMs work by mapping the data to a higher-dimensional space where it may be linearly separable, and then applying a linear SVM to find the separating hyperplane. This allows them to handle data that is not linearly separable in the original feature space.

https://linguisticmaz.medium.com/support-vector-machines-explained-ii-f2688fbf02ae

https://www.youtube.com/watch?v=qjErFs0tepw&ab_channel=RANJIRAJ

https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm#:~:text=Non-linear SVM%3A Non-,as Non-linear SVM classifier.

▼ Ans 4 (b) - Illustrate using an example how dimensionality gets reduced in PCA.

Principal component analysis (PCA) is a technique for reducing the dimensionality of a data set by projecting the data onto a lower-dimensional space. This is done by finding the directions in the data that have the highest variance and retaining only the most important dimensions.

To illustrate how dimensionality gets reduced in PCA, let's consider the following example:

Suppose we have a data set with 1000 samples and 10 features. We want to use PCA to reduce the dimensionality of the data to 3 dimensions.

1. First, we standardize the data by subtracting the mean and dividing by the standard deviation of each feature. This ensures that all of the features are on the same scale and makes it easier to compare the importance of each feature.
2. Next, we compute the covariance matrix of the data, which is a 10x10 matrix that measures the relationships between the different features.
3. We then compute the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors are the directions in the data that have the highest variance, and the eigenvalues are the corresponding magnitudes of the variance.
4. We sort the eigenvalues in decreasing order and select the top 3 eigenvectors, which correspond to the 3 dimensions with the highest variance.
5. We project the data onto the 3 selected eigenvectors by multiplying the data by the matrix of eigenvectors. This results in a new 3-dimensional data set with the same 1000 samples but only 3 features.

In this example, we were able to reduce the dimensionality of the data from 10 dimensions to 3 dimensions by selecting the 3 eigenvectors with the highest variance. This can be useful for reducing the complexity of the data and improving the performance of machine learning algorithms.

https://www.kdnuggets.com/2020/05/dimensionality-reduction-principal-component-analysis.html

https://towardsdatascience.com/principal-component-analysis-for-dimensionality-reduction-115a3d157bad

https://medium.com/analytics-vidhya/dimensionality-reduction-principal-component-analysis-d1402b58feb1

▼ Ans 4 (b`) - Classify a Red Domestic SUV using Naive Bayesian Classifier.

https://www.youtube.com/watch?v=fOK9DiKUGYs&ab_channel=MaheshHuddar

▼ Ans 5 (a) - Decision Tree

5 (a). The following table consists of training data from an employee database. The data have been generalized. For example, “31 ... 35” for age represents the age range of 31 to 35. For a given row entry, count represents the number of data tuples having the values for department, status, age, and salary given in that row.

department	*status*	*age*	*salary*	*count*
sales	senior	31...35	46K...50K	30
sales	junior	26...30	26K...30K	40
sales	junior	31...35	31K...35K	40
systems	junior	21...25	46K...50K	20
systems	senior	31...35	66K...70K	5
systems	junior	26...30	46K...50K	3
systems	senior	41...45	66K...70K	3
marketing	senior	36...40	46K...50K	10
marketing	junior	31...35	41K...45K	4
secretary	senior	46...50	36K...40K	4
secretary	junior	26...30	26K...30K	6

Let status be the class label attribute.
(a) How would you modify the basic decision tree algorithm to take into consideration the count of each generalized data tuple (i.e., of each row entry)?
(b) Use your algorithm to construct a decision tree from the given data

(a) How would you modify the basic decision tree algorithm to take into consideration the *count* of each generalized data tuple (i.e., of each row entry)?

The basic decision tree algorithm should be modified as follows to take into consideration the count of each generalized data tuple.

- The count of each tuple must be integrated into the calculation of the attribute selection measure (such as information gain).
- Take the count into consideration to determine the most common class among the tuples.

(b) Use your algorithm to construct a decision tree from the given data.

The resulting tree is:

(salary = 26K...30K:
junior
= 31K...35K:
junior
= 36K...40K:
senior
= 41K...45K:
junior
= 46K...50K (department = secretary:
junior
= sales:
senior
= systems:
junior

= marketing:
senior)
= 66K...70K:
senior)

(c) Given a data tuple with the values *"systems"*, *"junior"*, and *"26...30"* for the attributes *department*, *status*, and *age*, respectively, what would a naïve Bayesian classification of the *salary* for the tuple be?

$P(X|senior) = 0$; $P(X|junior) = 0.018$. Thus, a naïve Bayesian classification predicts "*junior*".

(d) Design a multilayer feed-forward neural network for the given data. Label the nodes in the input and output layers.

No standard answer. Every feasible solution is correct. As stated in the book, discrete-valued attributes may be encoded such that there is one input unit per domain value. For hidden layer units, the number should be smaller than that of input units, but larger than that of output units.

(e) Using the multilayer feed-forward neural network obtained above, show the weight values after one iteration of the backpropagation algorithm, given the training instance *"(sales, senior, 31...35, 46K...50K)"*. Indicate your initial weight values and biases and the learning rate used.

No standard answer. Every feasible solution is correct.

▼ Ans 5 (b) - Using an example, describe in detail Random Forest algorithm.

Random Forest is an ensemble machine learning algorithm that is used for classification and regression tasks. It is a widely used and effective algorithm that is known for its ability to handle large and complex data sets.

An example of how the Random Forest algorithm works is as follows:

Suppose we have a data set with 1000 samples and 20 features. We want to use the Random Forest algorithm to predict whether a customer will purchase a product based on the features in the data.

1. First, the algorithm will randomly select a subset of the data (called a "bootstrap sample") and build a decision tree model on this subset. This process is repeated multiple times, typically using a large number of decision trees (e.g., 100).
2. For each decision tree, the algorithm will select a random subset of the features to consider at each node. This helps to reduce the correlation between the trees and improve the overall accuracy of the model.
3. The predictions from all of the decision trees are then combined using a majority vote or by averaging the predictions. This results in a single prediction for each sample in the data set.
4. The final model can be used to make predictions on new data by inputting the features and using the combined predictions from all of the decision trees to make a final prediction.

In summary, the Random Forest algorithm works by building multiple decision trees on randomly selected subsets of the data and combining their predictions to make a final prediction. This process helps to reduce overfitting and improve the accuracy of the model.

https://www.javatpoint.com/machine-learning-random-forest-algorithm

https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm#:~:text=The following steps explain the,by averaging the decision tree.

https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

https://builtin.com/data-science/random-forest-algorithm

▼ Ans 5 (b`) i - It is important to calculate the worst-case computational complexity of the decision tree algorithm. Given data set, D, the number of attributes, n, and the number of training tuples, |D|, show that the computational cost of growing a tree is at most n × |D| × log(|D|)

The computational cost of growing a decision tree can be expressed in terms of the number of attributes in the data set, the number of training tuples, and the logarithm of the number of training tuples.

To see this, consider the process of building a decision tree. At each node in the tree, the algorithm must consider all of the available attributes and select the one that best splits the data according to some criterion, such as information gain or gini impurity. This process is repeated at each subsequent node in the tree until the data is fully split into leaves.

If there are n attributes in the data set and |D| training tuples, then the number of times this process must be repeated is at most n × |D|, since each tuple must be considered for each attribute.

In addition, the algorithm must also perform a logarithmic number of operations to select the best attribute at each node. This is because the algorithm typically uses a sorting or binary search algorithm to find the best attribute, both of which have a logarithmic computational complexity.

Therefore, the overall computational cost of growing a decision tree is at most n × |D| × log(|D|). This represents the worst-case computational complexity of the algorithm, which occurs when the tree is fully grown and the data is fully split into leaves.

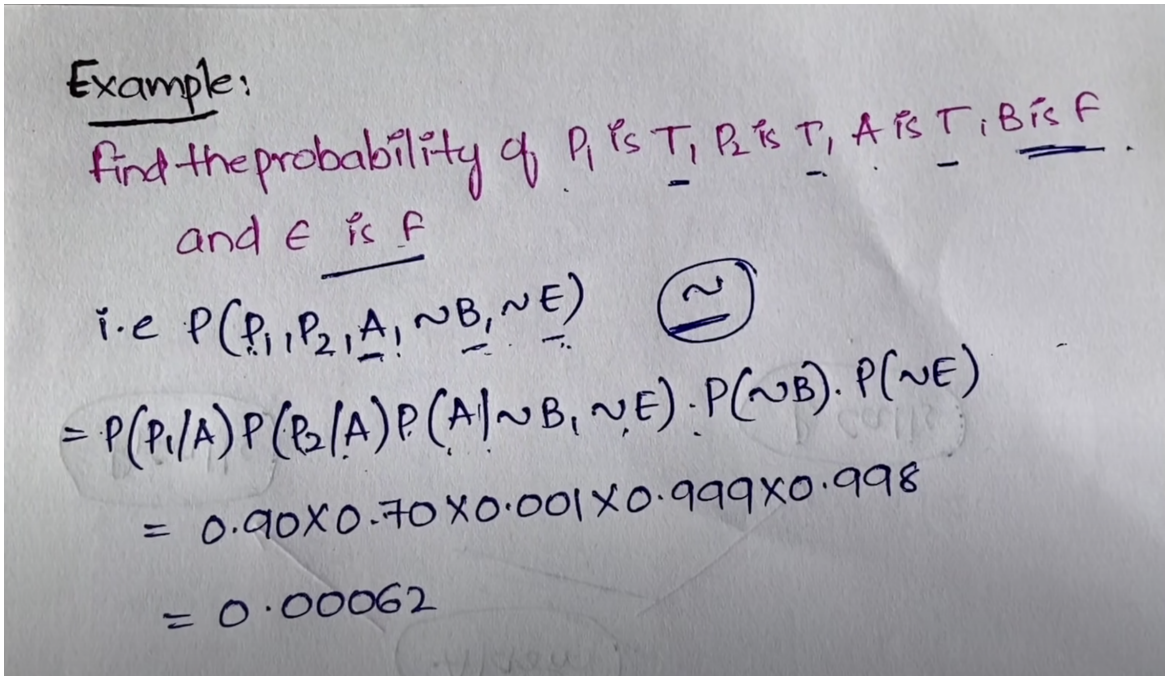
▼ Ans 5 (b`) ii - Why is naive Bayesian classification called “na¨ive”? Briefly outline the major ideas of na¨ive Bayesian classification.

Naive Bayesian classification is called "naive" because it makes the assumption that all of the features in the data are independent of each other. This is a strong assumption that is often not true in real-world data, but it allows for a simple and efficient classification algorithm.

The major ideas of naive Bayesian classification are as follows:

1. Each feature in the data is assumed to be independent of the others.
2. The probability of a particular class label is calculated based on the individual probabilities of each feature, assuming that the features are independent.
3. The class label with the highest probability is chosen as the predicted class for the input data.
4. The probabilities of the individual features and class labels are estimated from the training data using the maximum likelihood estimation method.

Overall, naive Bayesian classification is a simple and efficient algorithm that can be used for classification tasks with relatively small and well-structured data sets. However, it is not as effective on larger or more complex data sets, where the assumption of feature independence may not hold.

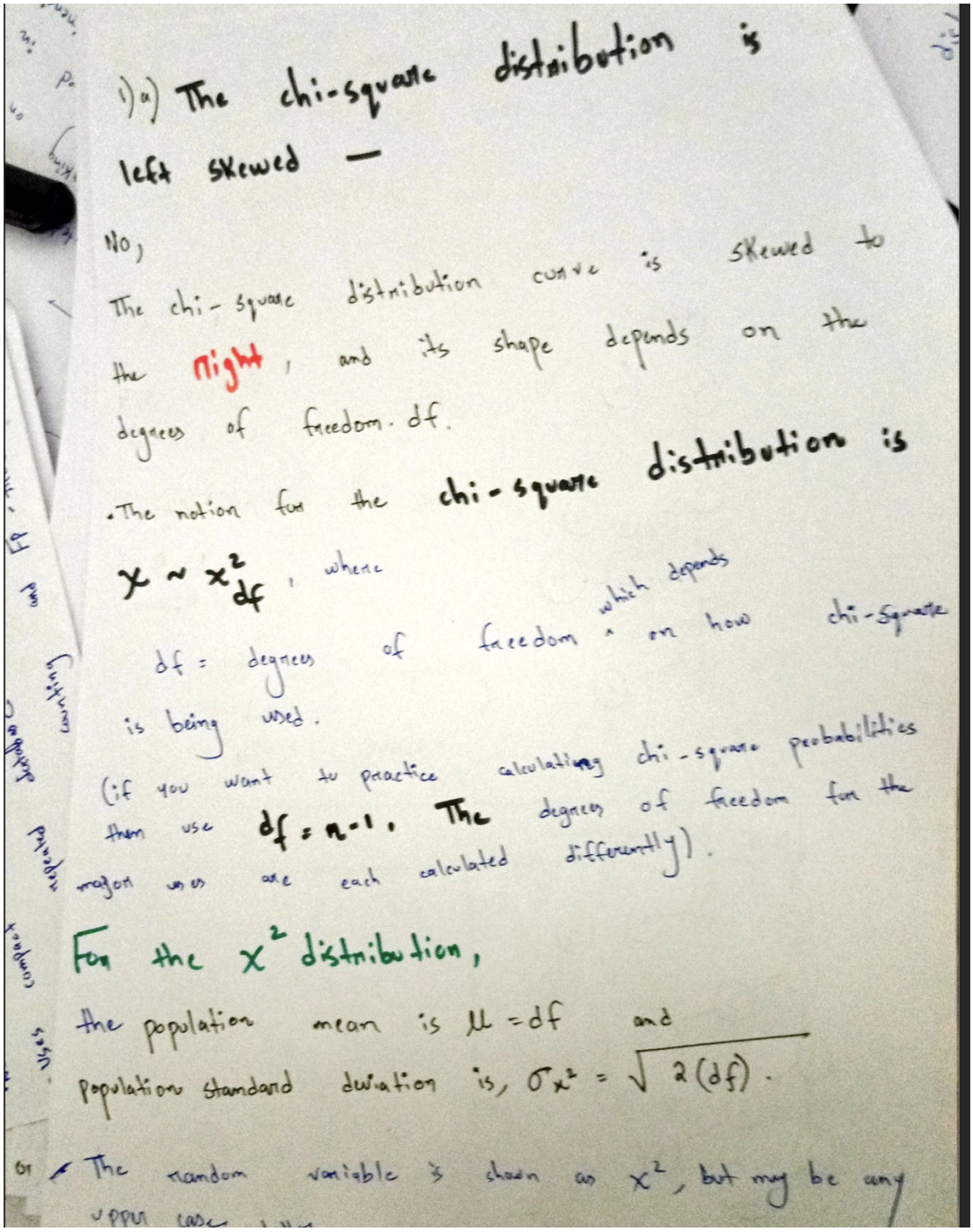


▼ 2020 -

▼ Ans 1 (a) -

Indicate whether the following statement is true or false. Give reasons for your answer. If the statement is false then change the statement to make it true.

1. i) The chi-square distribution is left skewed.
2. ii) Outliers are influential observations
3. iii) The principal components are always mutually exclusive and exhaustive of the variables
4. iv) A 2-nearest neighbor model is more likely to overfit than a 20-nearest neighbor model.
5. v) With k-fold cross-validation, larger k is always better.
6. vi) Overfitting is a danger when learning a classifier, but not when doing unsupervised learning.


1) a) The chi-square distribution is left skewed —

No,

The chi-square distribution curve is skewed to the right, and its shape depends on the degrees of freedom. df.

• The notion for the chi-square distribution is $\chi \sim \chi^2_{df}$, where

df = degrees of freedom, which depends on how chi-square is being used.

(if you want to practice calculating chi-square probabilities then use df = n-1. The degrees of freedom for the major uses are each calculated differently).

For the χ^2 distribution,

the population mean is $\mu = df$ and

population standard deviation is, $\sigma_{\chi^2} = \sqrt{2(df)}$.

or ← The random variable is shown as χ^2, but may be any upper case

1) a) ii) Outliers are influential observations –

An outliers is a point with large residual. Outliers are the data points those diverge by good margin from the overall pattern. It can have extreme X or Y values or both compared to other values.

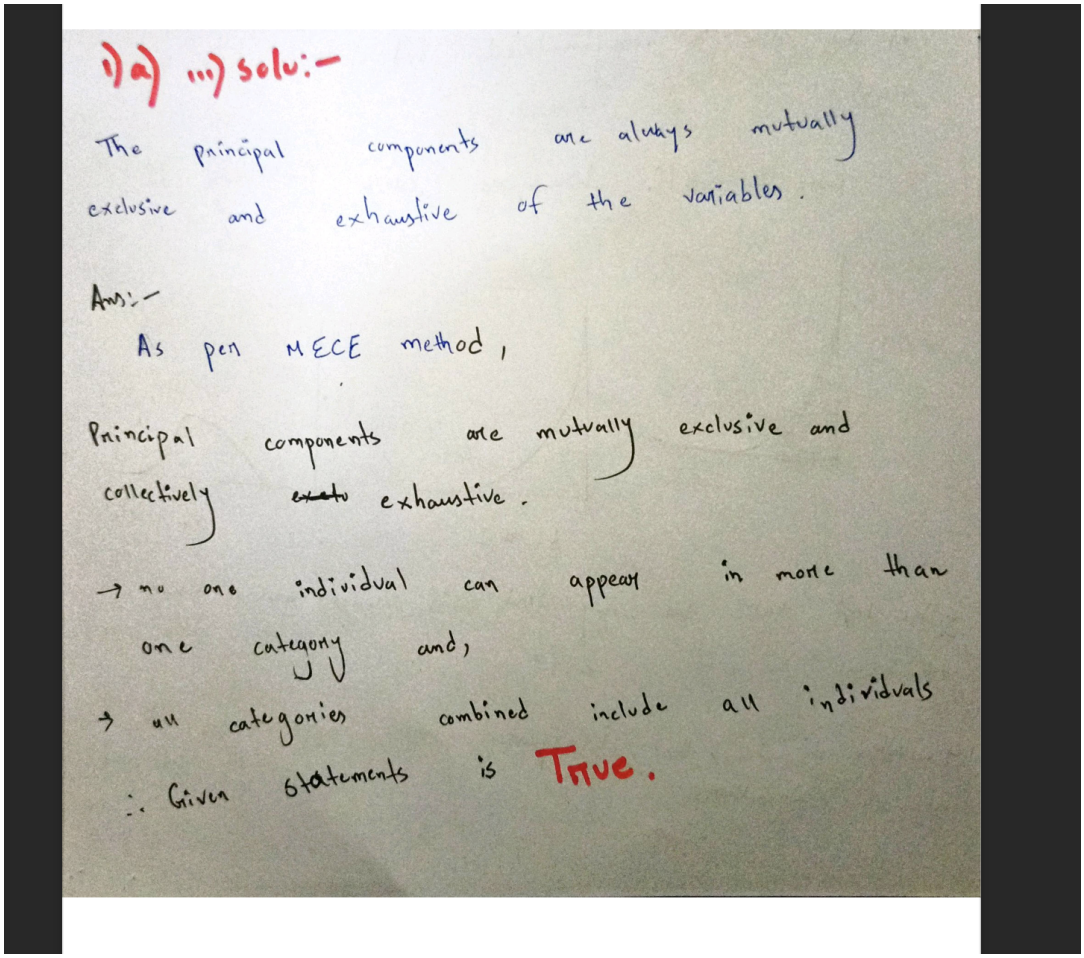
An influential points is a point that has a large impact on the regression.

- It is an outlier that impacts the slope of the regression line.

- To ~~test~~ test the influence of an outlier is to compute the regression equation with and without the outlier.

So,

~~the~~ an outlier is a data point that diverges from an overall pattern in a sample. Therefore, an outlier can certainly influence the relationship b/w the variables ~~and~~ may also exert an influence on the slope of the regression line.



1)a) iii) solu:-

The principal components are always mutually exclusive and exhaustive of the variables.

Ans:-

As per MECE method,

Principal components are mutually exclusive and collectively ~~exato~~ exhaustive.

→ no one individual can appear in more than one category and,

→ all categories combined include all individuals

∴ Given statements is TRUE.

AS per MECE method,

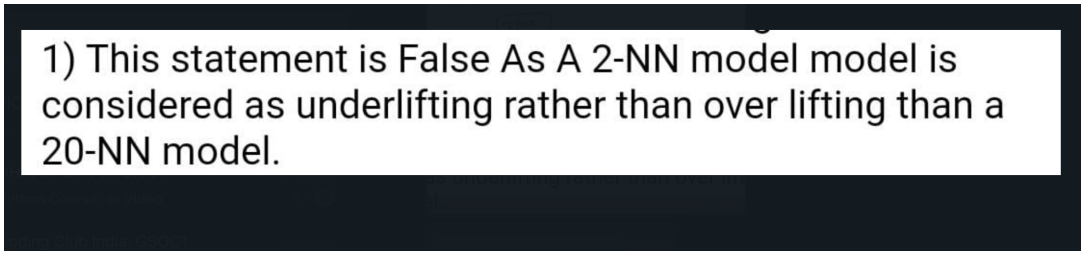
Principle components are mutually exclusive and collectively exhaustive.

→ no one individual can appear in more than one category and,

→ all categories combined include all individuals.

∴ given steetement is

TRue



1. False. The chi-square distribution is generally symmetrical or bell-shaped.
2. True. Outliers are observations that are significantly different from the majority of the data and can have a large influence on the results of statistical analysis.
3. False. The principal components are not necessarily mutually exclusive or exhaustive of the variables. They are linear combinations of the original variables that capture the maximum amount of variation in the data.
4. True. A 2-nearest neighbor model is likely to be more sensitive to the specific characteristics of the training data, while a 20-nearest neighbor model is likely to be more robust to variations in the data. This means that the 2-nearest neighbor model is more likely to overfit to the training data.
5. False. Larger values of k can lead to a more thorough evaluation of the model, but there is no guarantee that a larger value of k will always be better. The optimal value of k will depend on the specific characteristics of the data and the model being evaluated.
6. True. Overfitting is a danger when learning a classifier because the model is trying to fit the training data as closely as possible, potentially at the expense of generalizability to new data. In unsupervised learning, the model is not trying to fit the data in the same way and is not at risk of overfitting.

▼ Ans 1 (b) -

1. For each of the following questions, provide an example of an association rule from the market basket domain that satisfies the following conditions. Also, describe whether such rules are subjectively interesting.

   (a) A rule that has high support and high confidence.
   Answer: Milk ⟶ Bread. Such obvious rule tends to be uninteresting.

   (b) A rule that has reasonably high support but low confidence.
   Answer: Milk ⟶ Tuna. While the sale of tuna and milk may be higher than the support threshold, not all transactions that contain milk also contain tuna. Such low-confidence rule tends to be uninteresting.

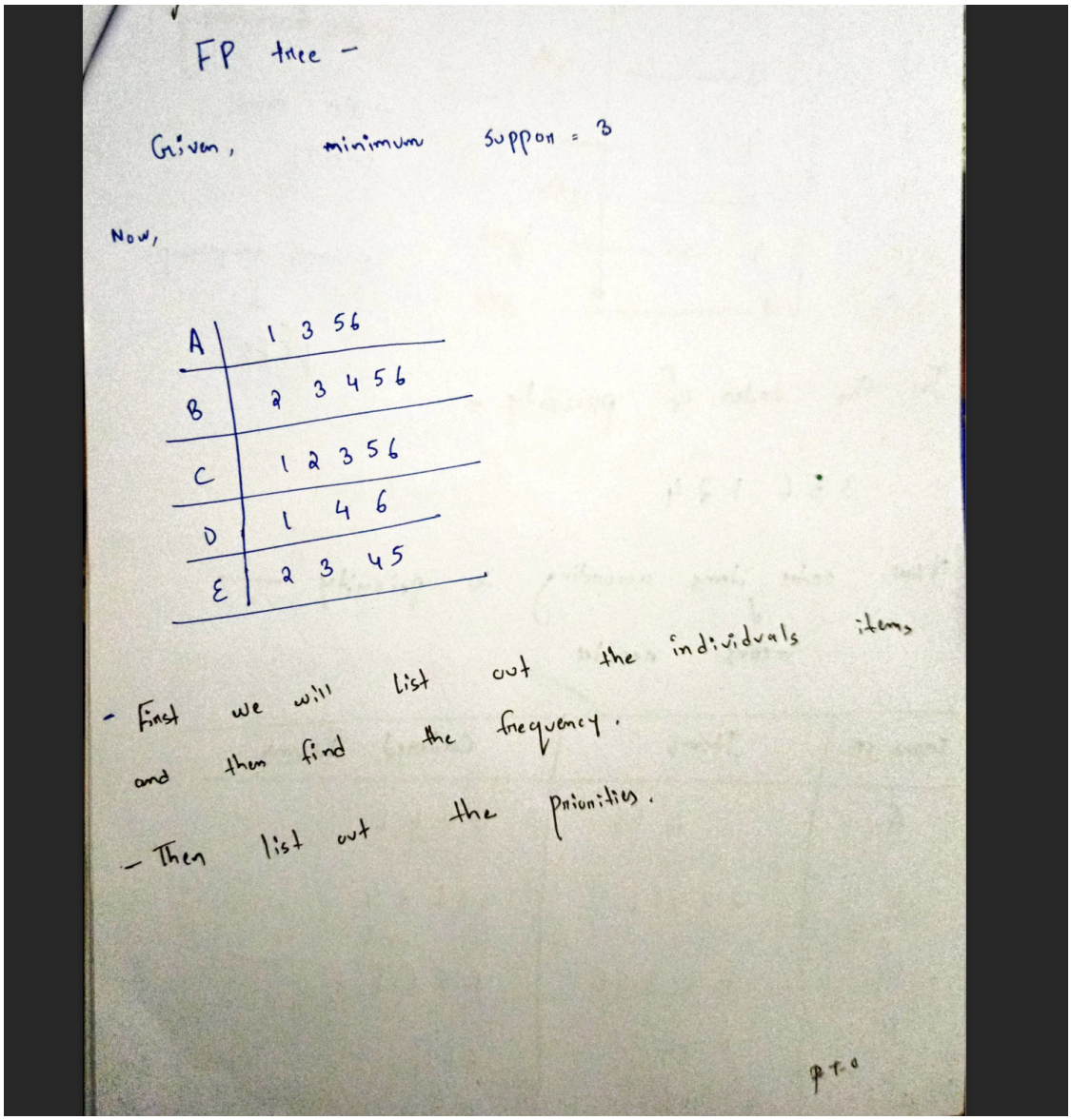
   (c) A rule that has low support and low confidence.
   Answer: Cooking oil ⟶ Laundry detergent. Such low confidence rule tends to be uninteresting.

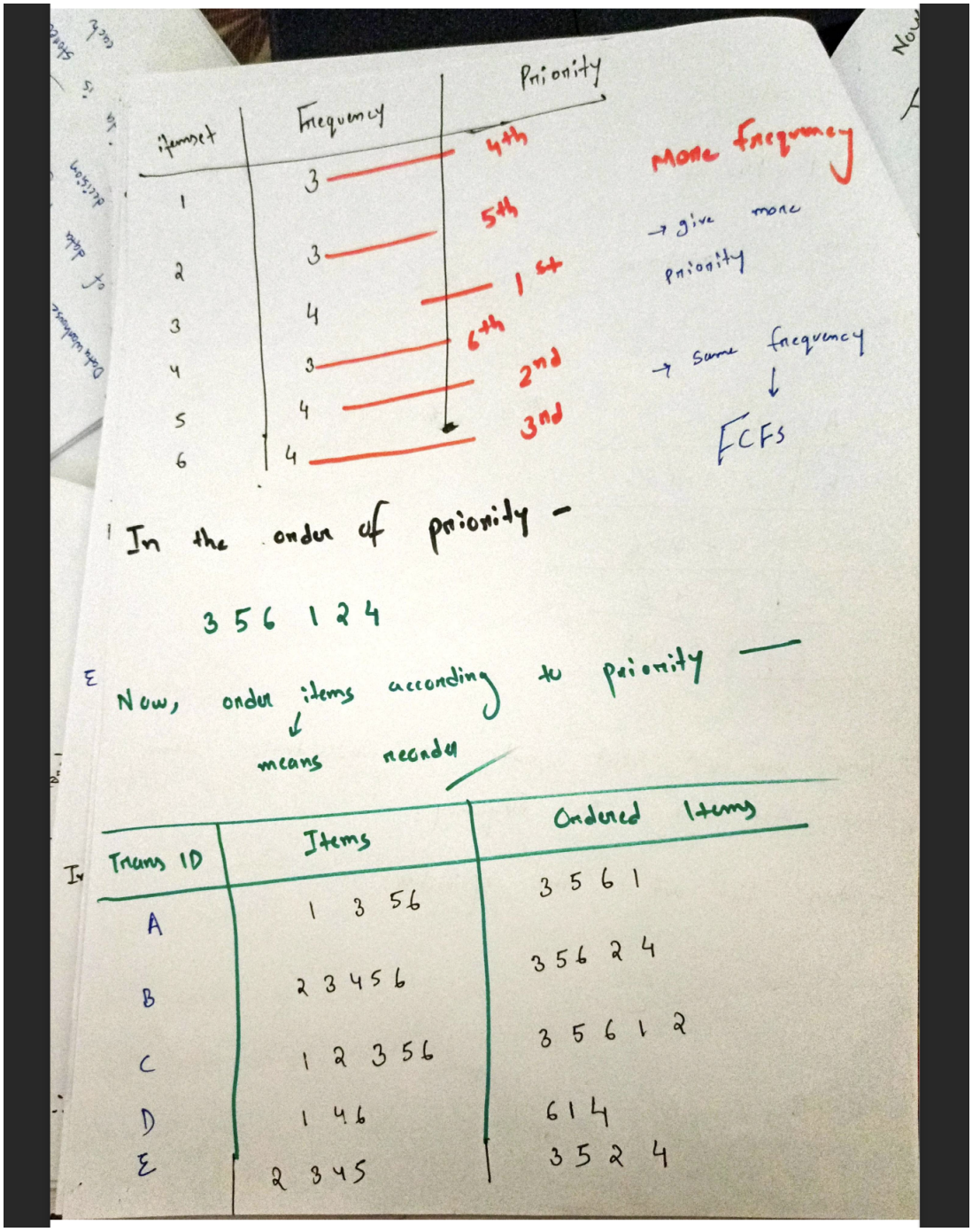
   (d) A rule that has low support and high confidence.
   Answer: Vodka ⟶ Caviar. Such rule tends to be interesting.

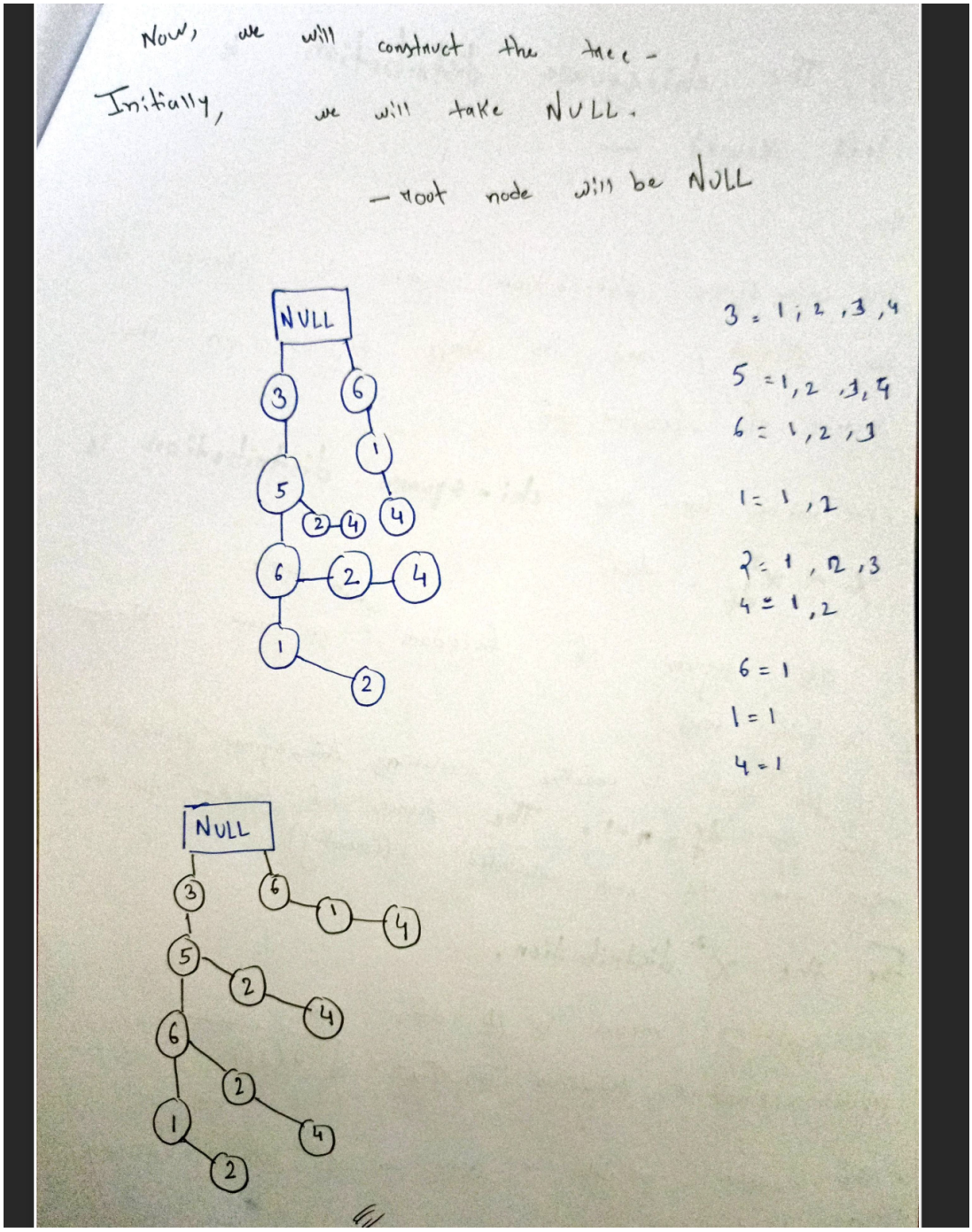
▼ Ans 1 (c) -

Consider the vertical dataset shown below. Assuming minimum support = 3 find all frequent itemsets using FP-tree.

A	B	C	D	E
1	2	1	1	2
3	3	2	4	3
5	4	3	6	4
6	5	5		5
	6	6		

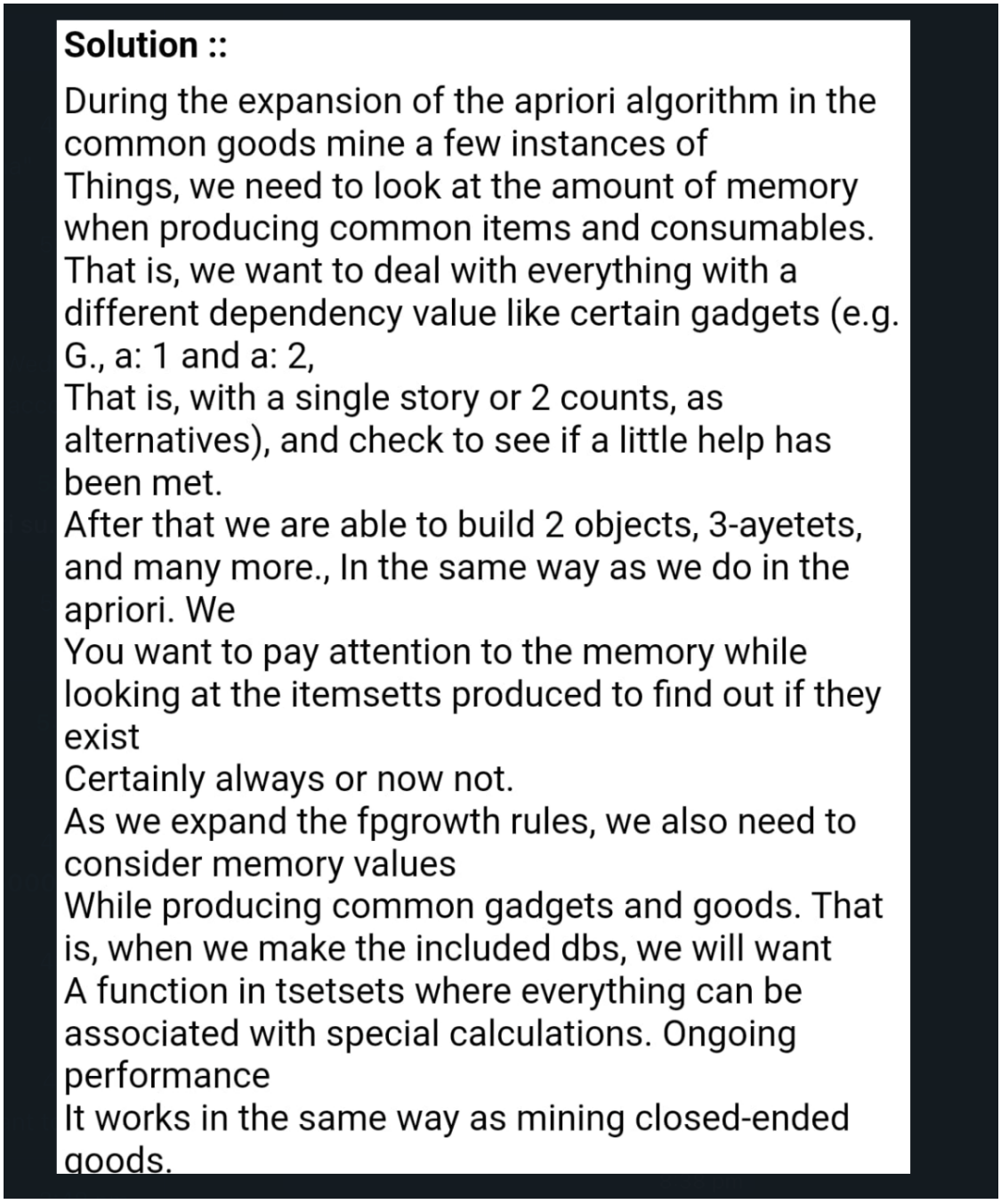



Priority
itemset
Frequency
1
2
3
4
5
6
3
3
4
3
4
4
4th
5th
1st
6th
2nd
3rd
More frequency
→ give more priority
→ same frequency
↓
FCFS
In the order of priority -
3 5 6 1 2 4
Now, order items according to priority
means reorder
Trans ID
Items
Ordered Items
A
1 3 5 6
3 5 6 1
B
2 3 4 5 6
3 5 6 2 4
C
1 2 3 5 6
3 5 6 1 2
D
1 4 6
6 1 4
E
2 3 4 5
3 5 2 4



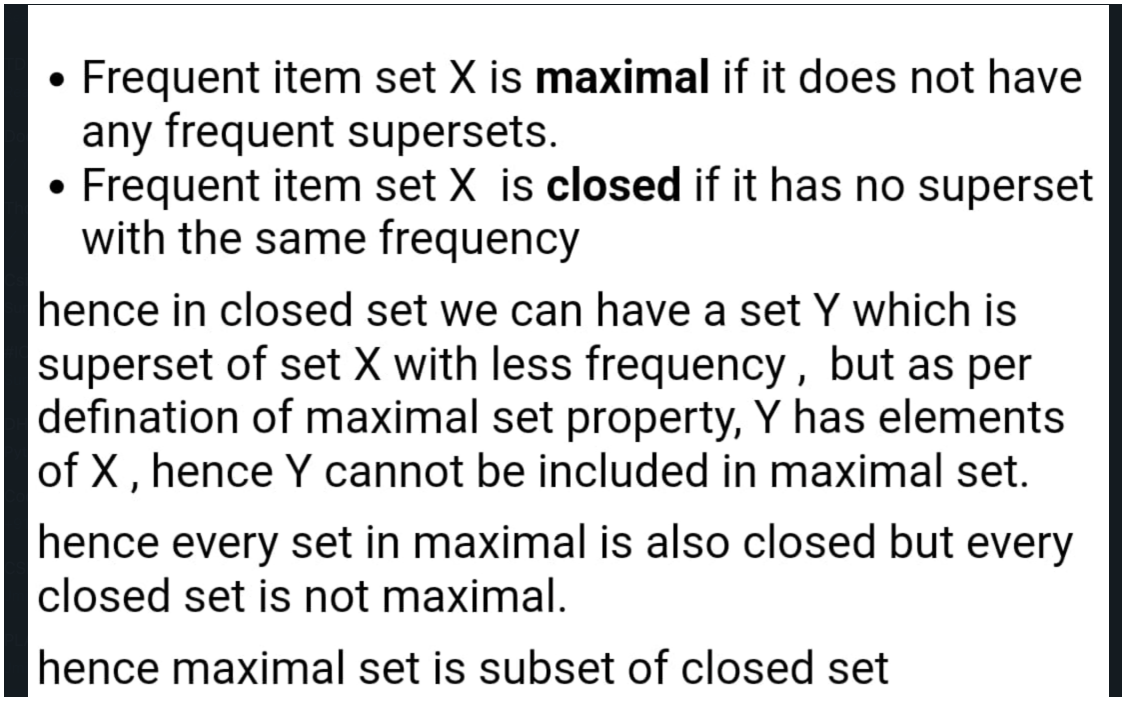
▼ Ans 2 (a) -

Most frequent pattern mining algorithms consider only distinct items in a transaction. However, multiple occurrences of an item in the same shopping basket, such as four cakes and three chocolates, can be important in transactional data analysis. How can one mine frequent itemsets efficiently considering multiple occurrences of items? Propose modifications to the Apriori, to adapt to such a situation.



▼ Ans 2 (b) i -

Let ‘C’ be the set of all closed frequent itemsets and ‘M’ the set of all maximal frequent itemsets for some database. Prove that M □ C.



▼ Ans 2 (c) -

An algorithm has to designed which will display only those frequent items which lies between MaxSupport and MinSupport whose values are supplied by the user. Modify any frequent pattern mining algorithm such that the frequent items are displayed between these two values in decreasing value of their support.

One approach to modifying a frequent pattern mining algorithm to display frequent items between a given range of minimum and maximum support values would be to use a modified version of the Apriori algorithm. The Apriori algorithm is a popular frequent pattern mining algorithm that uses a bottom-up approach to identify frequent itemsets in a dataset. Here is a high-level outline of how the modified Apriori algorithm could work:

Initialize the minimum and maximum support values, and set the current support threshold to the maximum support value.

Scan the dataset and calculate the support for each item. Only items with a support greater than or equal to the current support threshold are considered frequent.

Generate the frequent itemsets by combining the frequent items using the Apriori property (i.e., an itemset is frequent if all of its subsets are frequent).

Sort the frequent itemsets in decreasing order of support.

Display the frequent itemsets that have a support value between the minimum and maximum support values.

Decrement the support threshold and repeat steps 2-5 until the support threshold is less than the minimum support value.

This modified Apriori algorithm would ensure that only frequent items with a support value between the given minimum and maximum values are displayed, in decreasing order of their support.

▼ Ans 3 (a) -

3.a) Consider the following dataset. Compute the information gain of attribute a1 and a2. If we want to find the information gain of a3 what should be done?

Instance	a1	a2	a3	Class
1	T	T	5.0	Y
2	T	T	7.0	Y
3	T	F	8.0	N
4	F	F	3.0	Y
5	F	T	7.0	N
6	F	T	4.0	N
7	F	F	5.0	N
8	T	F	6.0	Y
9	F	T	1.0	N

3) a) solu:-

There are four positive examples i.e 4, Y = 4,

and 5 no i.e N = 5

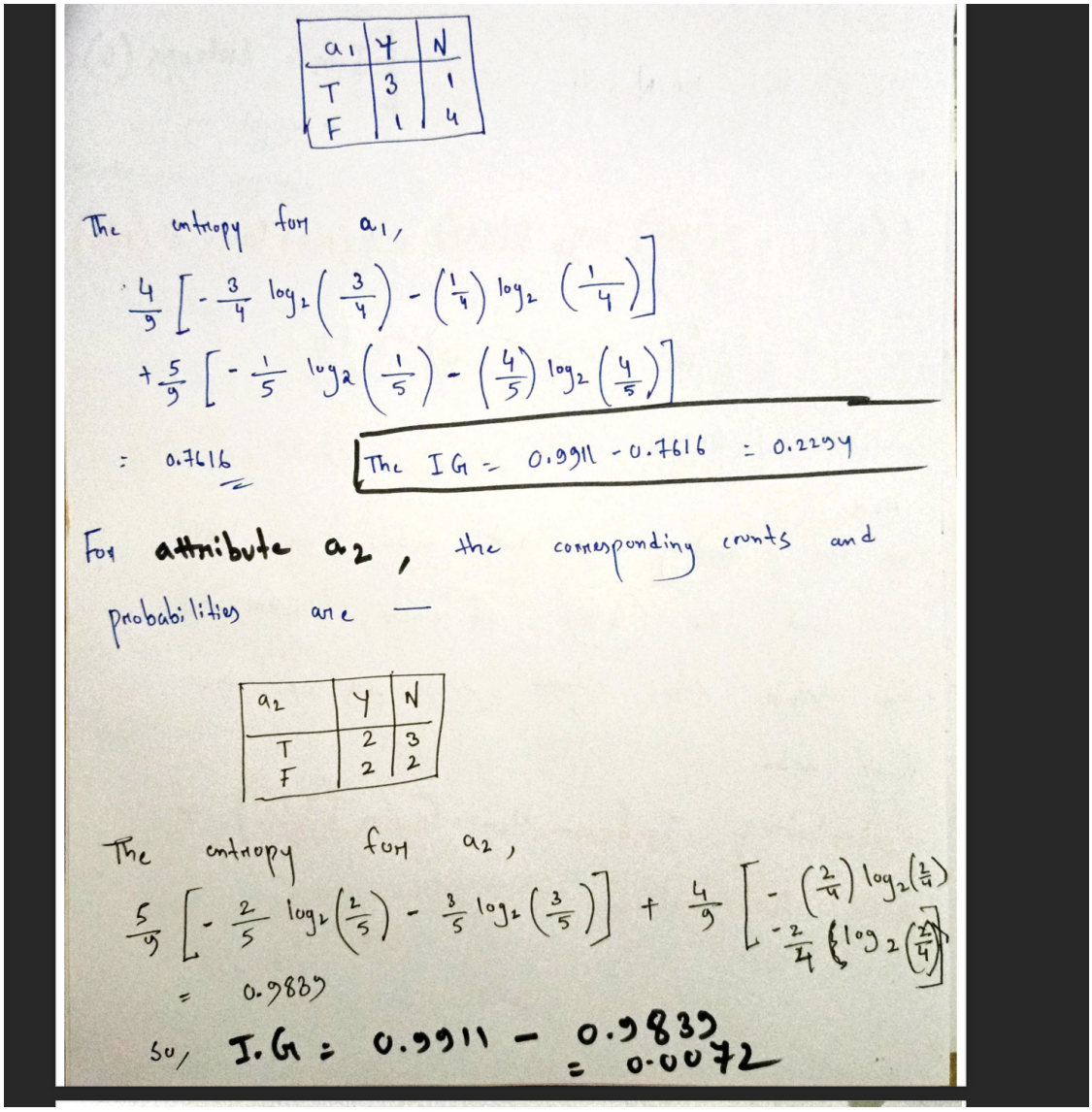
So, Entropy of the training example is

$$-\frac{4}{9}\log_2\left(\frac{4}{9}\right) - \frac{5}{9}\log_2\left(\frac{5}{9}\right)$$

$$= 0.9911$$

For attribute a_1, corresponding counts and probability are-

a_1	Y	N
T	3	1
F	1	4



a_1	Y	N
T	3	1
F	1	4

The entropy for a_1,

$$\frac{4}{9}\left[-\frac{3}{4}\log_2\left(\frac{3}{4}\right)-\left(\frac{1}{4}\right)\log_2\left(\frac{1}{4}\right)\right]$$

$$+\frac{5}{9}\left[-\frac{1}{5}\log_2\left(\frac{1}{5}\right)-\left(\frac{4}{5}\right)\log_2\left(\frac{4}{5}\right)\right]$$

$= 0.7616$

The IG $= 0.9911 - 0.7616 = 0.2294$

For attribute a_2, the corresponding counts and probabilities are —

a_2	Y	N
T	2	3
F	2	2

The entropy for a_2,

$$\frac{5}{9}\left[-\frac{2}{5}\log_2\left(\frac{2}{5}\right)-\frac{3}{5}\log_2\left(\frac{3}{5}\right)\right]+\frac{4}{9}\left[-\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right)\right.$$

$$\left.-\frac{2}{4}\log_2\left(\frac{2}{4}\right)\right]$$

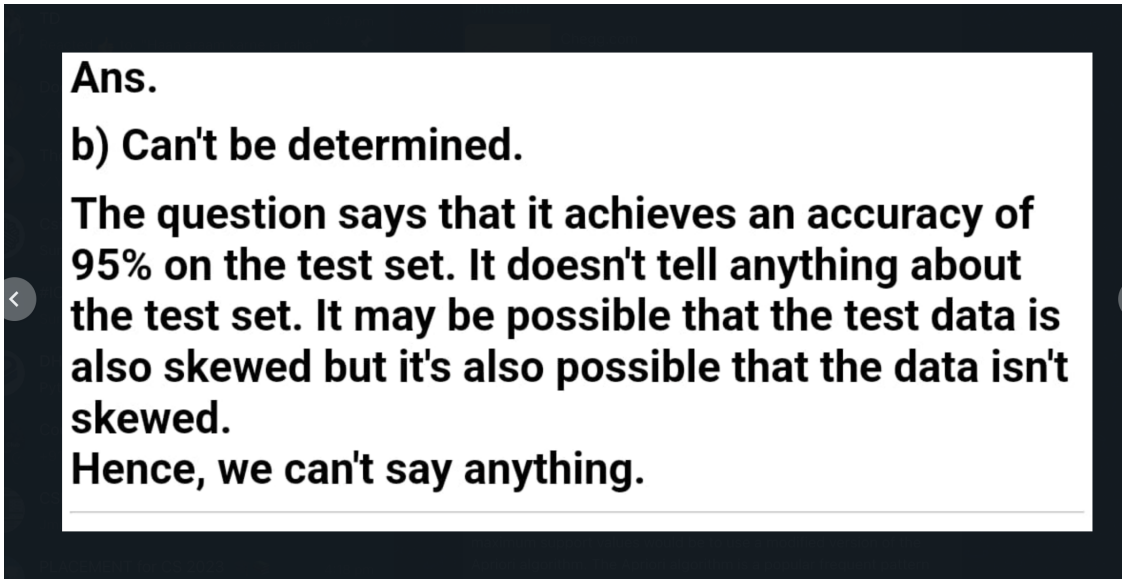
$= 0.9839$

So, I.G $= 0.9911 - 0.9839$

$= 0.0072$

▼ Ans 3 (b) i -

A binary classifier achieves 95% accuracy on a test set consisting of 95% positive and 5% negative instances. If we use the same classifier on a test set composed of 50% positive and 50% negative instances, what can we say on the accuracy of the classifier.



▼ Ans 3 (b) ii -

How boosting improves the accuracy of the classifier? Use a proper example to illustrate your answer.

Boosting is a machine learning technique that can be used to improve the accuracy of a classifier. Boosting works by training a series of weak classifiers, each of which is only slightly better than random guessing, and combining their predictions to form a strong classifier. The weak classifiers are trained iteratively, with the weights of the training examples adjusted at each iteration to give more emphasis to the examples that were misclassified in the previous iteration.

An example of how boosting can improve the accuracy of a classifier is through the use of the AdaBoost (Adaptive Boosting) algorithm. AdaBoost is a popular boosting algorithm that works by training a sequence of decision trees on the training data, with the weights of the training examples adjusted at each iteration to give more emphasis to the examples that were misclassified in the previous iteration. The final classifier is obtained by weighting the predictions of the individual decision trees and combining them using a weighted majority vote.

Suppose we have a dataset with 1000 training examples and 2 features (x1 and x2). We want to use AdaBoost to build a classifier to predict whether a given example belongs to class 0 or class 1. We start by training a decision tree on the training data using the default weights for the training examples (all set to 1/1000). The decision tree misclassifies 30 examples. We then adjust the weights of the misclassified examples so that they have a higher weight in the next iteration. We train a second decision tree on the modified training set and obtain a classifier that misclassifies 25 examples. We repeat this process until we have trained a total of 10 decision trees.

Finally, we combine the predictions of the individual decision trees using a weighted majority vote, with the weights of the decision trees determined by their accuracy on the training data. The resulting classifier has an accuracy of 95%, which is significantly better than the accuracy of the individual decision trees (each with an accuracy of around 50%).

In this example, boosting improved the accuracy of the classifier by training a series of weak classifiers and combining their predictions to form a strong classifier. The weak classifiers were trained iteratively, with the weights of the training examples adjusted at each iteration to give more emphasis to the examples that were misclassified in the previous iteration. This process helped to reduce the error of the classifier and improve its overall accuracy.

▼ Ans 3 (c) -

(c) Given below is the sample dataset for classification of mammals and non-mammals. Classify the test date whose values are **Give birth = Yes, Can Fly = No, Live in Water = Yes, Have Legs = No.**

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals

Example of Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

A: attributes
M: mammals
N: non-mammals

$$P(A|M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A|N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A|M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A|N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

$P(A|M)P(M) > P(A|N)P(N)$
=> Mammals

12

▼ Ans 4 (a) -

Given two examples of Bayesian Belief Network with proper diagram. Using any one of them explain how this network is used to calculate the probabilities.

https://www.upgrad.com/blog/bayesian-network-example/

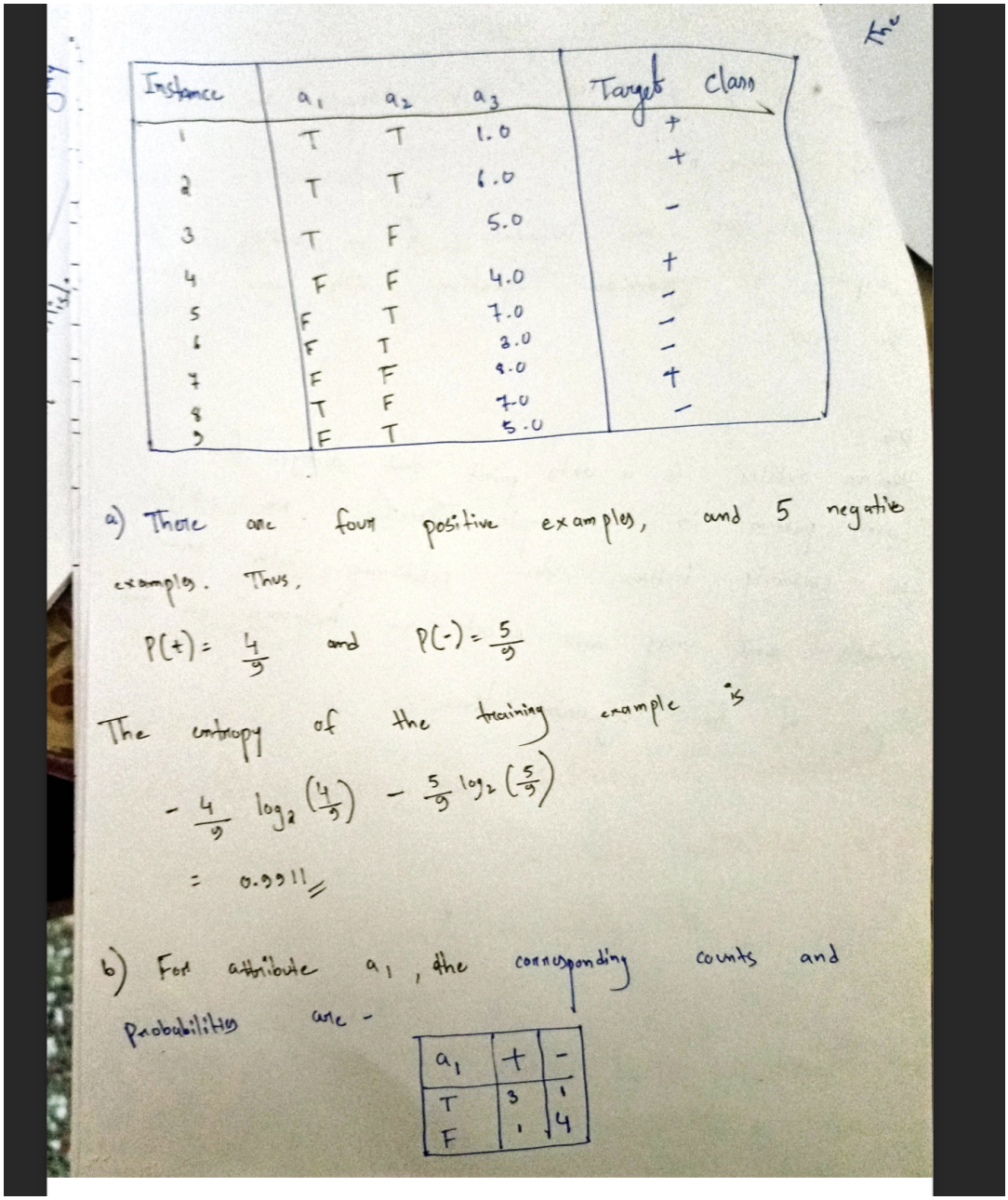
▼ Ans 4 (b) -

Consider the training examples shown below for a binary classification problem.

1. i) What is the entropy of this collection of training examples with respect to the positive class?

2. ii) What are the information gains of a1 and a2 relative to these training examples?
3. iii) What is the best split (between a1 and a2) according to the classification error rate?

Instance	a1	a2	a3	Class
1	T	T	3	+
2	T	T	6	+
3	T	F	5	-
4	F	F	4	+
5	F	T	7	-
6	F	T	4	+
7	F	F	8	-
8	T	F	7	+
9	F	T	6	+
10	F	F	9	-



Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	+
8	T	F	7.0	−
9	F	T	5.0	−

a) There are four positive examples, and 5 negative examples. Thus,

$P(+) = \frac{4}{9}$ and $P(-) = \frac{5}{9}$

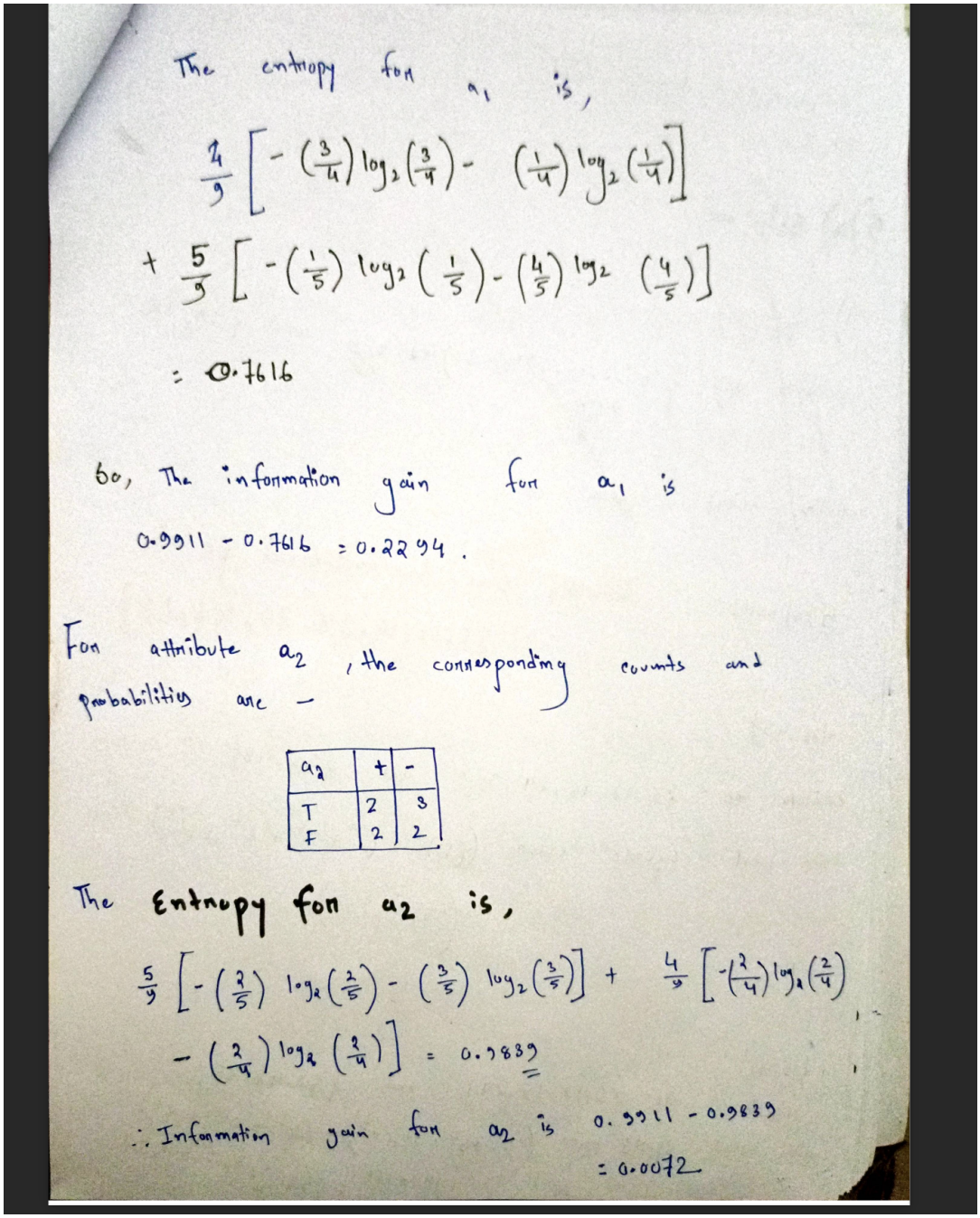
The entropy of the training example is

$$-\frac{4}{9}\log_2\left(\frac{4}{9}\right) - \frac{5}{9}\log_2\left(\frac{5}{9}\right)$$

$$= 0.9911$$

b) For attribute a_1, the corresponding counts and probabilities are -

a_1	+	−
T	3	1
F	1	4



The entropy for a_1 is,

$$\frac{4}{9}\left[-\left(\frac{3}{4}\right)\log_2\left(\frac{3}{4}\right)-\left(\frac{1}{4}\right)\log_2\left(\frac{1}{4}\right)\right]$$

$$+\frac{5}{9}\left[-\left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right)-\left(\frac{4}{5}\right)\log_2\left(\frac{4}{5}\right)\right]$$

$$= 0.7616$$

So, The information gain for a_1 is

$0.9911 - 0.7616 = 0.2294$.

For attribute a_2, the corresponding counts and probabilities are –

a_2	+	-
T	2	3
F	2	2

The **Entropy for a_2 is,**

$$\frac{5}{9}\left[-\left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right)-\left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right)\right]+\frac{4}{9}\left[-\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right)-\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right)\right]=0.9839$$

∴ Information gain for a_2 is $0.9911 - 0.9839$

$= 0.0072$

▼ Ans 4 (c) -

(c) Using decision tree classifier what should be the root of the decision tree for the following data whose classifying attribute is “Class”.

Number	Outlook	Temperature	Humidity	Windy	Class
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Overcast	Mild	High	True	P
12	Overcast	Hot	Normal	False	P

4)c) solv: —

Given, classifying anibute = "Class"

out of 12,

Play, P = 8, No, N = 4

[I.G = Entropy (S) - [weight avg.] × Entropy (each attribute)]

Now,

$$E(S) = -P(Yes)\log_2 P(Yes) - P(No)\log_2 P(No)$$

$$= -\frac{8}{12}\log_2\left(\frac{8}{12}\right) - \left(\frac{4}{12}\right)\log_2\left(\frac{4}{12}\right)$$

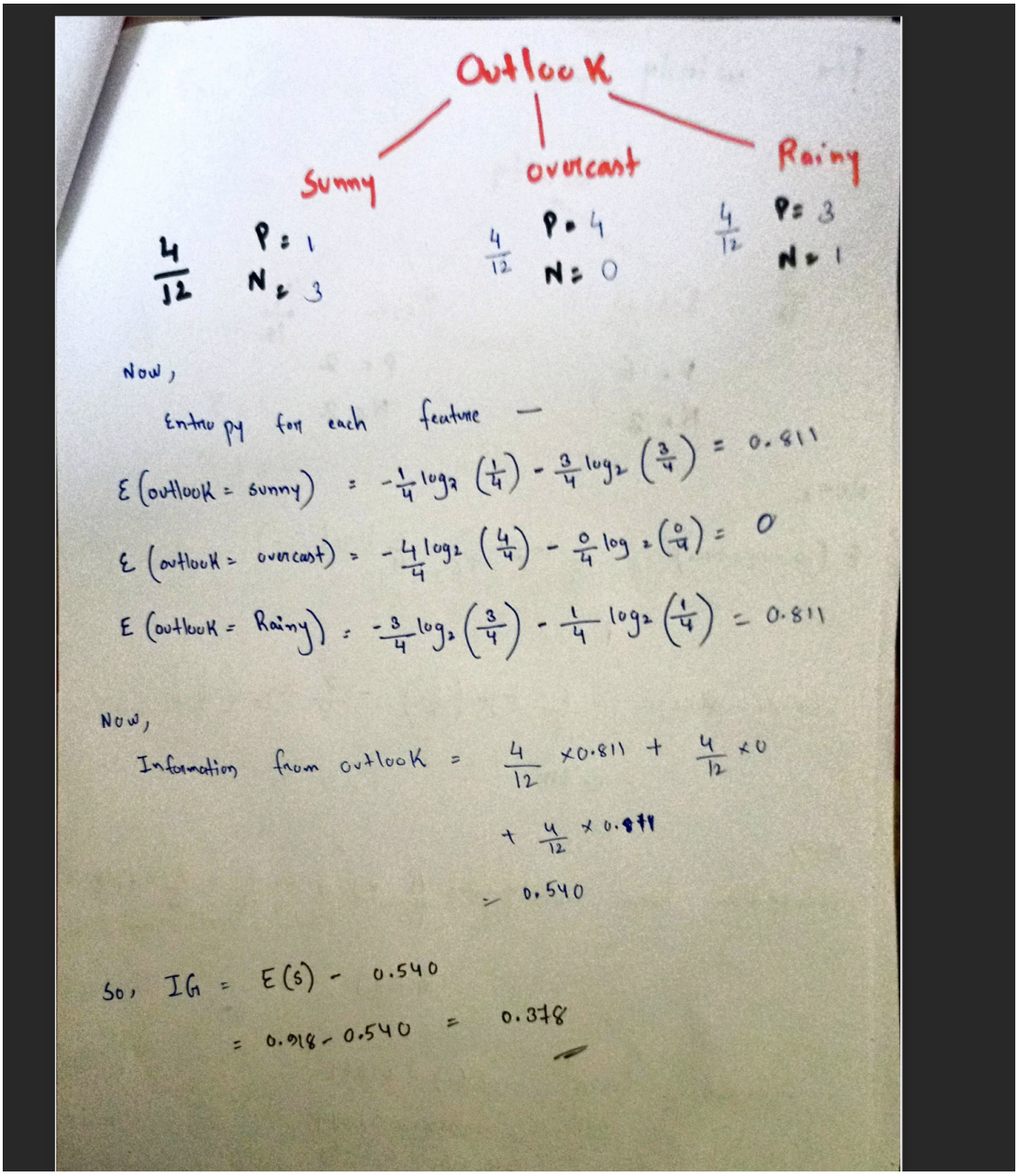
$$= 0.918$$

Now,

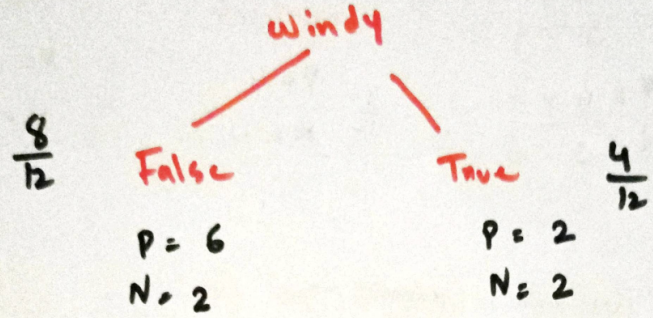
For finding the root node, we have to find the (IG) of each Attribute

- The attribute with highest (IG) will be the Root node.

Calculating Information Gain (IG) of Each Attribute —


Outlook
Sunny
Overcast
Rainy
4/12 P: 1 N: 3
4/12 P: 4 N: 0
4/12 P: 3 N: 1
Now,
Entropy for each feature —
E(outlook = sunny) = -1/4 log2 (1/4) - 3/4 log2 (3/4) = 0.811
E (outlook = overcast) = -4/4 log2 (4/4) - 0/4 log2 (0/4) = 0
E (outlook = Rainy) = -3/4 log2 (3/4) - 1/4 log2 (1/4) = 0.811
Now,
Information from outlook = 4/12 x0.811 + 4/12 x0
+ 4/12 x 0.811
= 0.540
So, IG = E(S) - 0.540
= 0.918 - 0.540 = 0.378

For windy —



Now,

$$E(\text{windy} = \text{true}) = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)$$

$$= 1$$

$$E(\text{windy} = \text{false}) = -\frac{6}{8}\log_2\left(\frac{6}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right)$$

$$= 0.811$$

Now,

Information from windy — $\frac{4}{12} \times 1 + \frac{8}{12} \times 0.811$

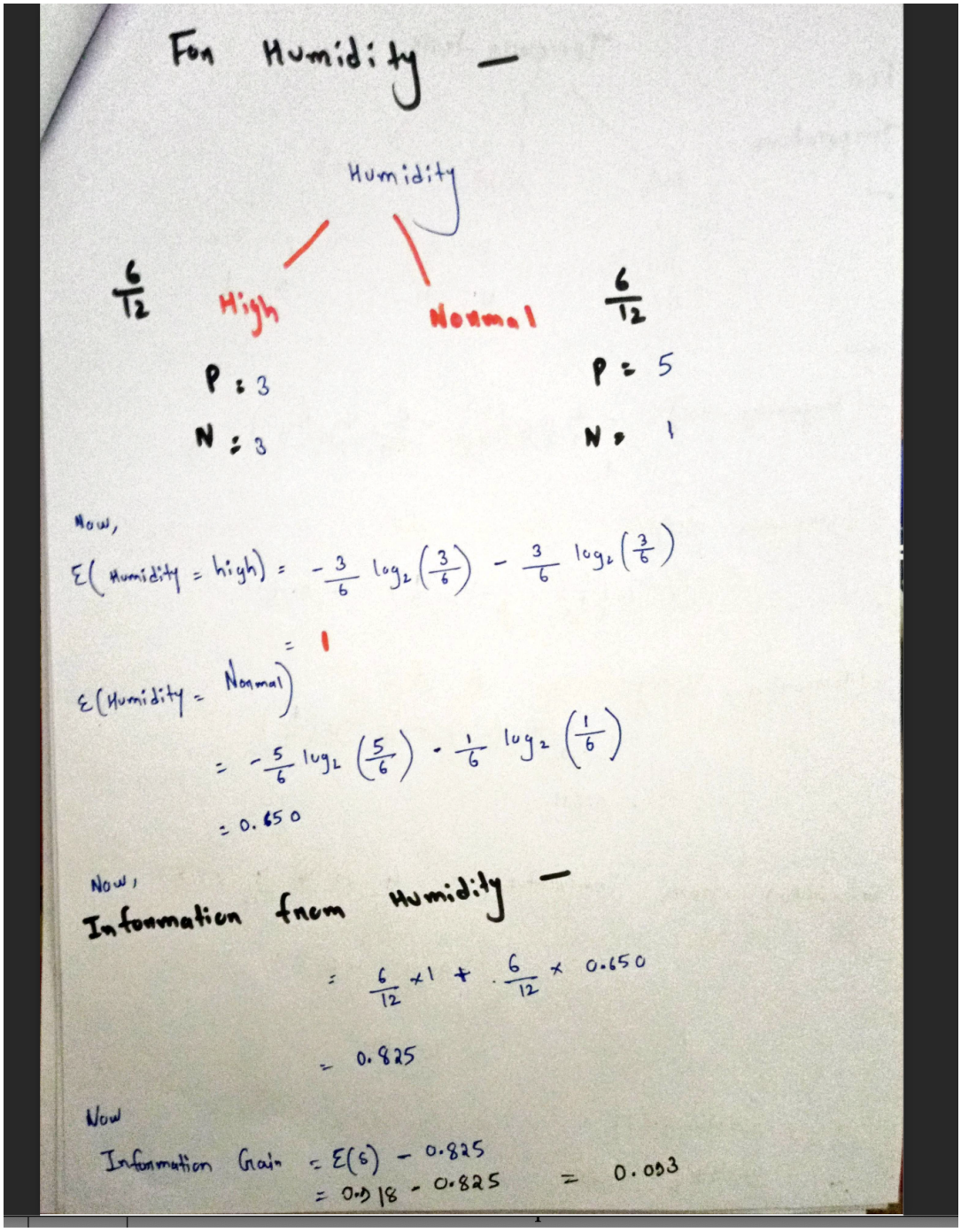
$$= 0.874$$

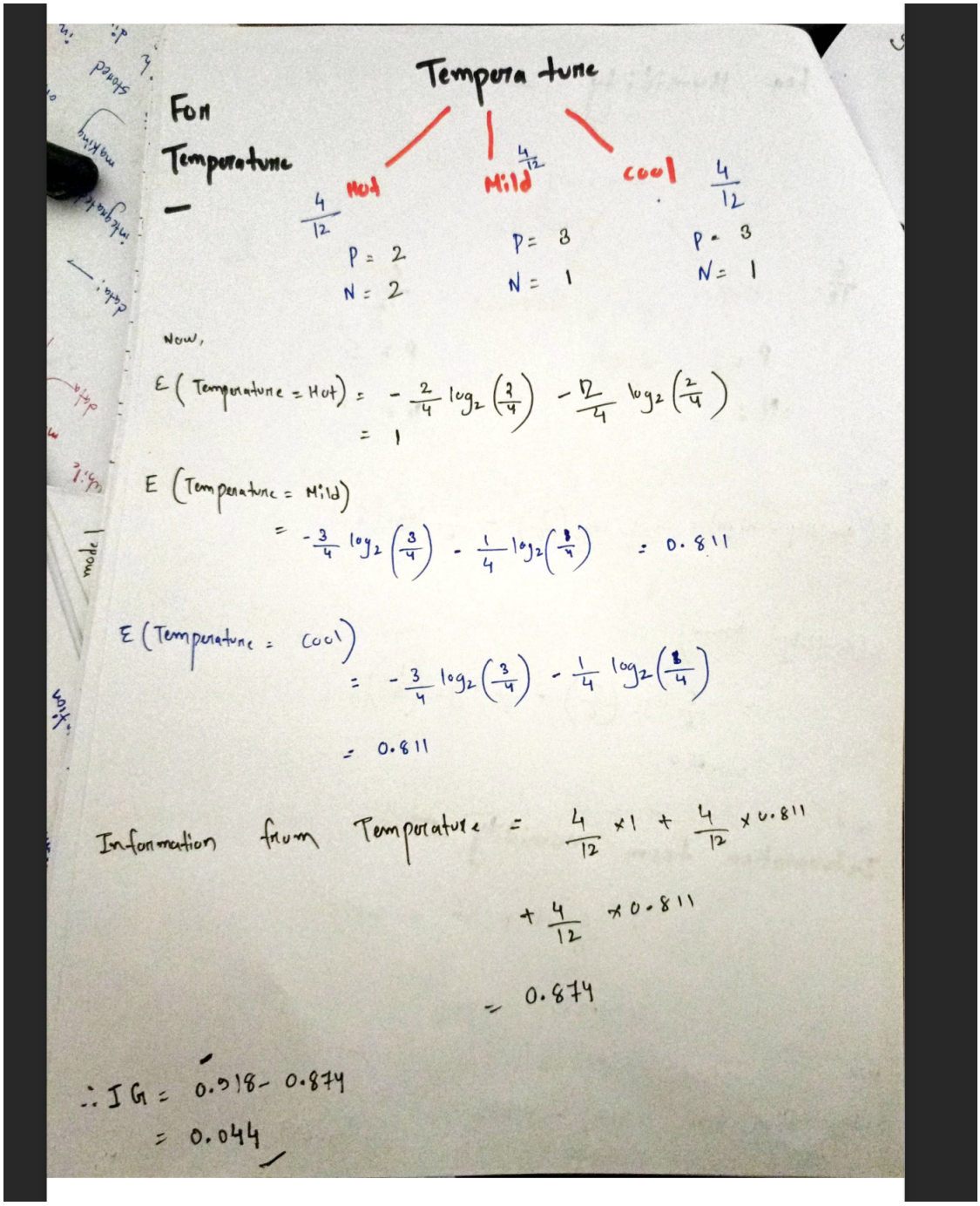
So,

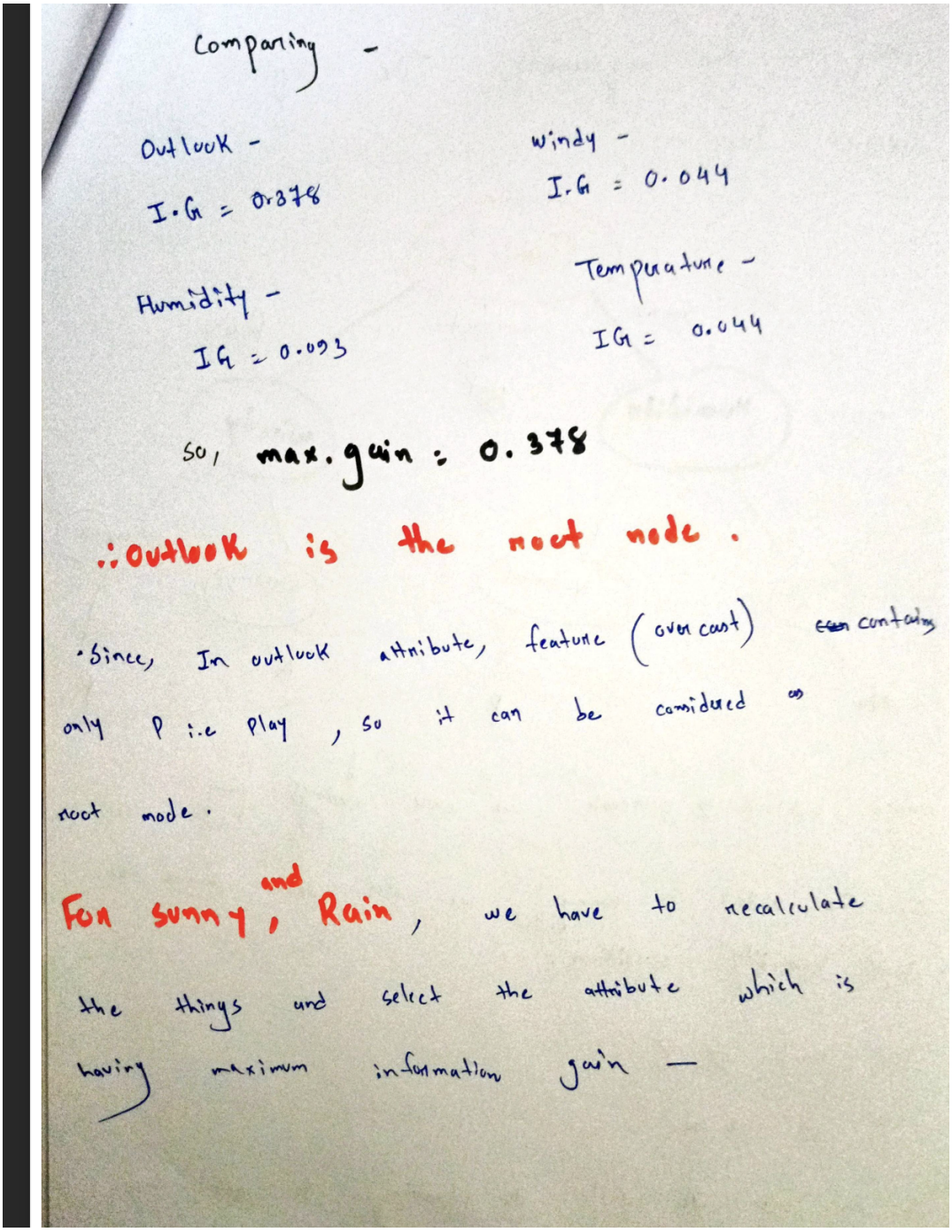
Information Gain $= E(S) - 0.874$

$= 0.918 - 0.874$

$= 0.044$


For Humidity –
Humidity
6/12
High
Normal
6/12
P : 3
N : 3
P : 5
N : 1
Now,
E(Humidity = high) = - 3/6 log2 (3/6) - 3/6 log2 (3/6)
= 1
E(Humidity = Normal)
= - 5/6 log2 (5/6) - 1/6 log2 (1/6)
= 0.650
Now,
Information from Humidity –
= 6/12 x 1 + 6/12 x 0.650
= 0.825
Now
Information Gain = E(S) - 0.825
= 0.918 - 0.825 = 0.093


For Temperature
Temperature
Hot
Mild
cool
4/12
4/12
4/12
P = 2
N = 2
P = 3
N = 1
P = 3
N = 1
Now,
E (Temperature = Hot) = $-\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)$
= 1
E (Temperature = Mild)
$= -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)$ = 0.811
E (Temperature = cool)
$= -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)$
= 0.811
Information from Temperature = $\frac{4}{12} \times 1 + \frac{4}{12} \times 0.811$
$+ \frac{4}{12} \times 0.811$
= 0.874
∴ IG = 0.918 − 0.874
= 0.044



Comparing -

Outlook -

I.G = 0.378

Windy -

I.G = 0.044

Humidity -

IG = 0.093

Temperature -

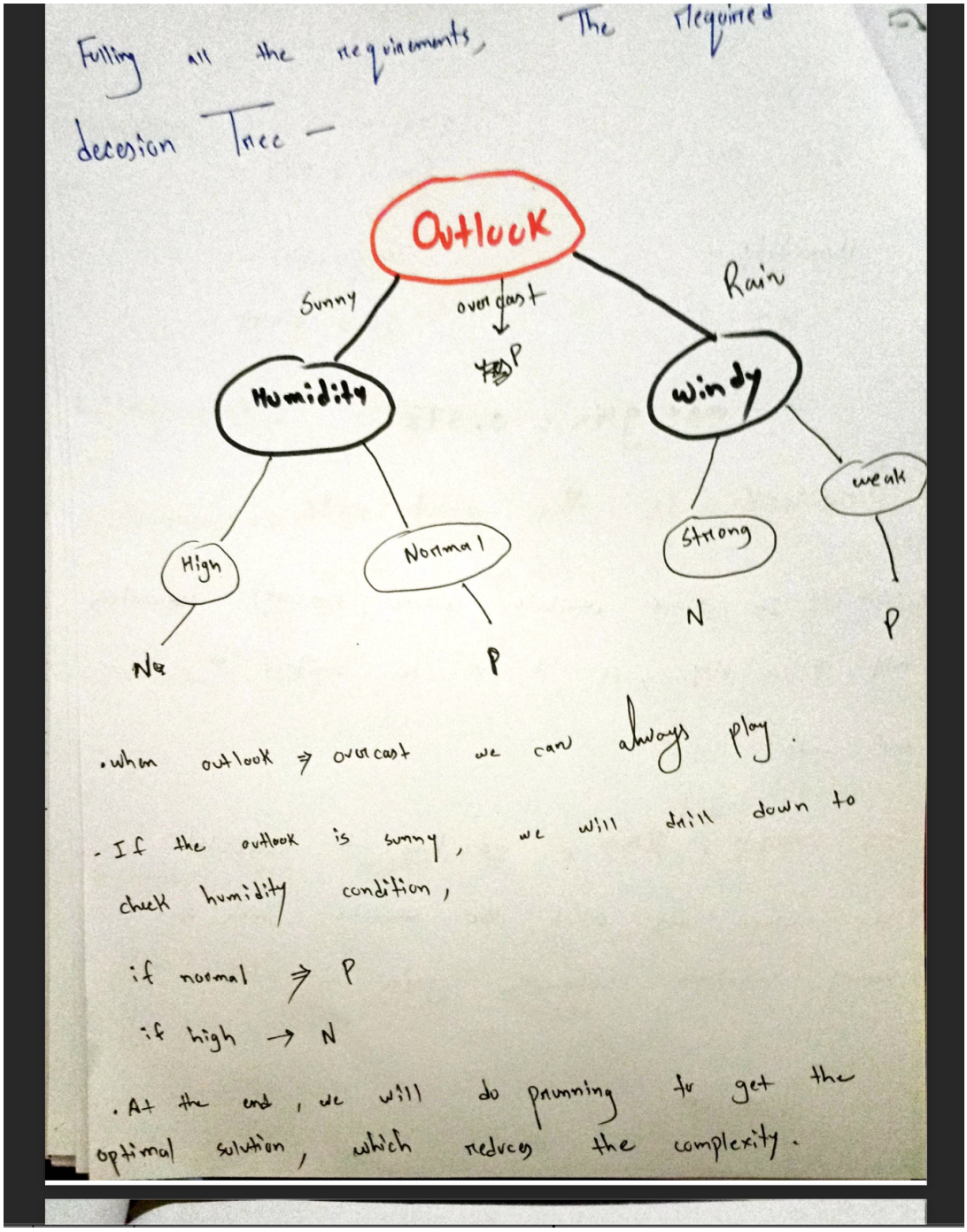
IG = 0.044

So, max. gain : 0.378

∴ Outlook is the root node.

• Since, In outlook attribute, feature (over cast) ~~can~~ contains only P i.e Play, so it can be considered as root node.

For sunny, and Rain, we have to recalculate the things and select the attribute which is having maximum information gain —


Fulling all the requirements, The required
decesion Tree -
Outlook
Sunny
overcast
Rain
P
Humidity
windy
High
Normal
Strong
weak
N
P
N
P
• when outlook ⇒ overcast we can always play.
- If the outlook is sunny, we will drill down to check humidity condition,
if normal ⇒ P
if high → N
• At the end, we will do pruning to get the optimal solution, which reduces the complexity.

▼ Ans 5 (a) -

5.a)	Hierarchical clustering is sometimes used to generate K clusters, K > 1 by taking the clusters at the K^{th} level of the dendrogram. (Root is at level 1.) By looking at the clusters produced in this way, we can evaluate the behavior of hierarchical clustering on different types of data and clusters, and also compare hierarchical approaches to K-means. The following is a set of one-dimensional points: {6, 10, 18, 24, 32, 42, 48, 54}. For each of the following sets of initial centroids, create two clusters by assigning each point to the nearest centroid, and then calculate the total squared error for each set of two clusters. Show both the clusters and the total squared error for each set of centroids. i) (18,40) ii) (15,45) (iii) (10, 32)

- $\{18, 45\}$
  Clusters are $\{6, 12, 18, 24, 30\}$ and $\{42, 48\}$
  with total square error $(12^2 + 6^2 + 0^2 + 6^2 + 12^2) + (3^2 + 3^2) = 360 + 18 = 378$.
- $\{15, 40\}$
  Clusters are $\{6, 12, 18, 24\}$ and $\{30, 42, 48\}$
  with total square error $(9^2 + 3^2 + 3^2 + 9^2) + (10^2 + 2^2 + 8^2) = 180 + 168 = 348$.

i) (18,40)

cluster are - {6, 10, 18, 24} and {32, 42, 48, 54}

with total square error

$= (12^2 + 8^2 + 0^2 + 6^2) + (8^2 + 2^2 + 8^2 + 14^2)$

$= 244 + 328$

$= 572$

ii) (15,45)

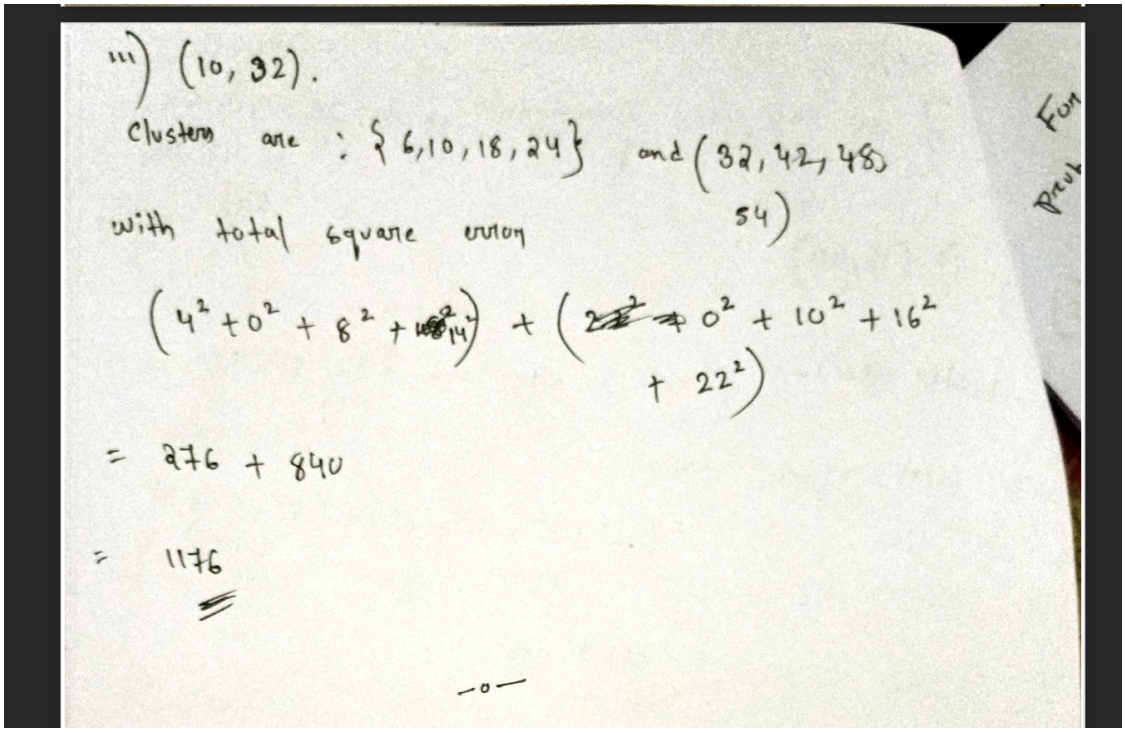
cluster are - {6, 10, 18, 24, 32} and {42, 48, 54}

with total square error

$= (9^2 + 5^2 + 3^2 + 9^2 + 17^2) + (3^2 + 3^2 + 3^2)$

$= 485 + 99$

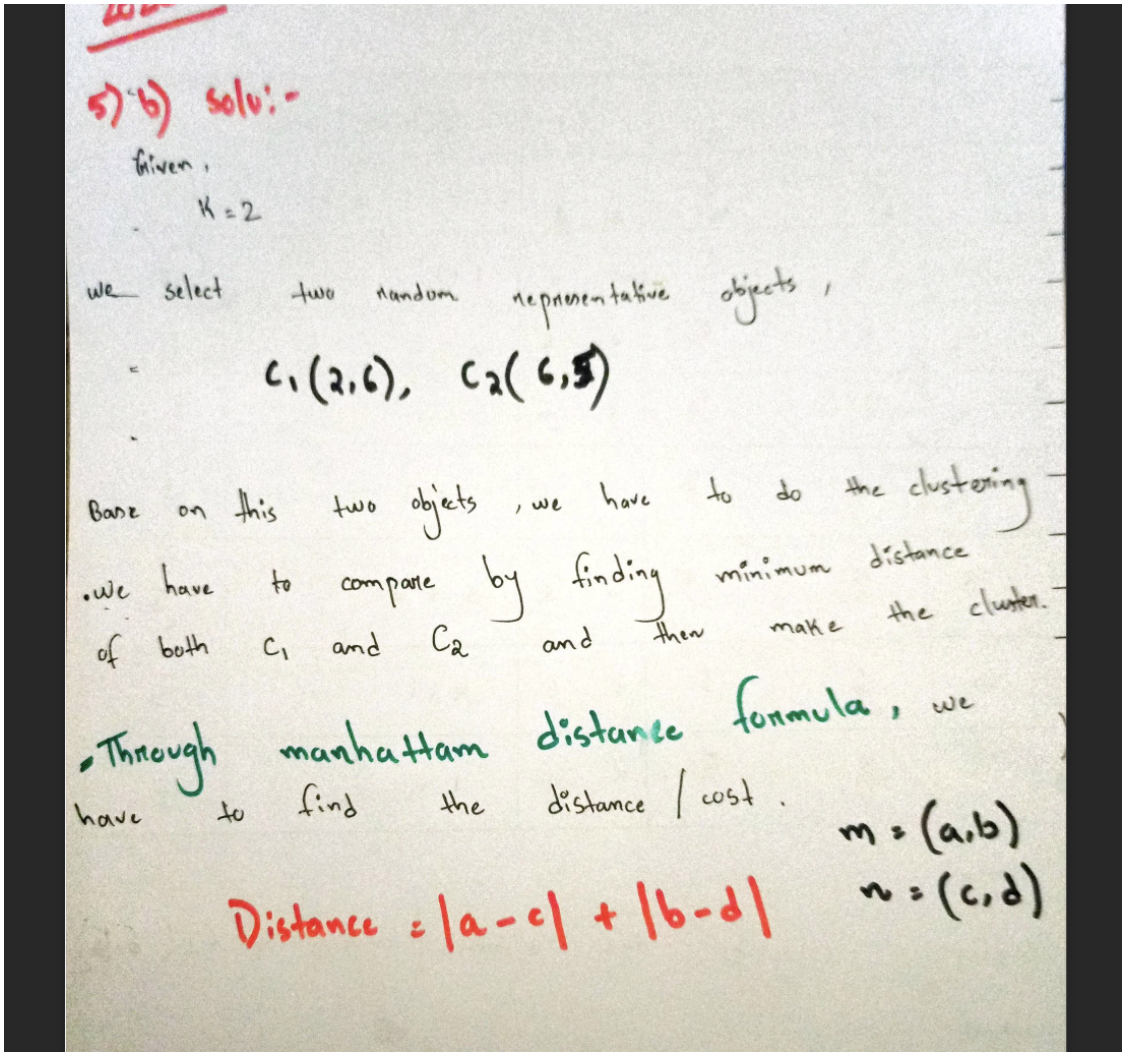
$= 584$



▼ Ans 5 (b) -

(b)	For a given k=2, cluster the following data set using K-mediod clustering algorithm.

Point	X-Axis	Y-Axis
1	7	6
2	2	6
3	3	8
4	8	5
5	7	4
6	4	7
7	6	2
8	7	3
9	6	5
10	3	5



5) b) solu:-

Given,

$K = 2$

We select two random representative objects,

$C_1(2,6)$, $C_2(6,5)$

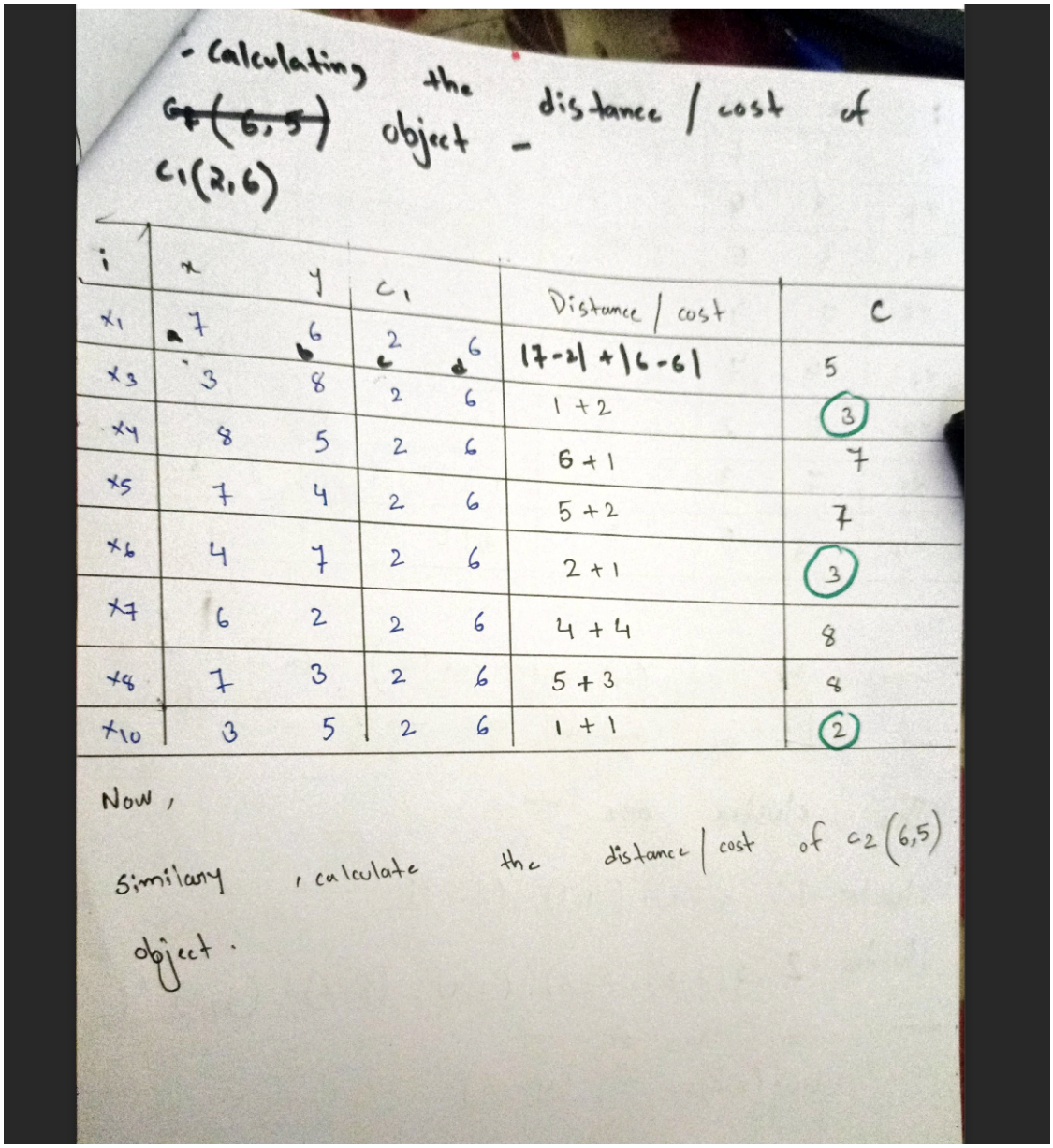
Base on this two objects, we have to do the clustering. We have to compare by finding minimum distance of both C_1 and C_2 and then make the cluster.

- Through manhattam distance formula, we have to find the distance / cost.

$m = (a,b)$

$n = (c,d)$

Distance $= |a-c| + |b-d|$

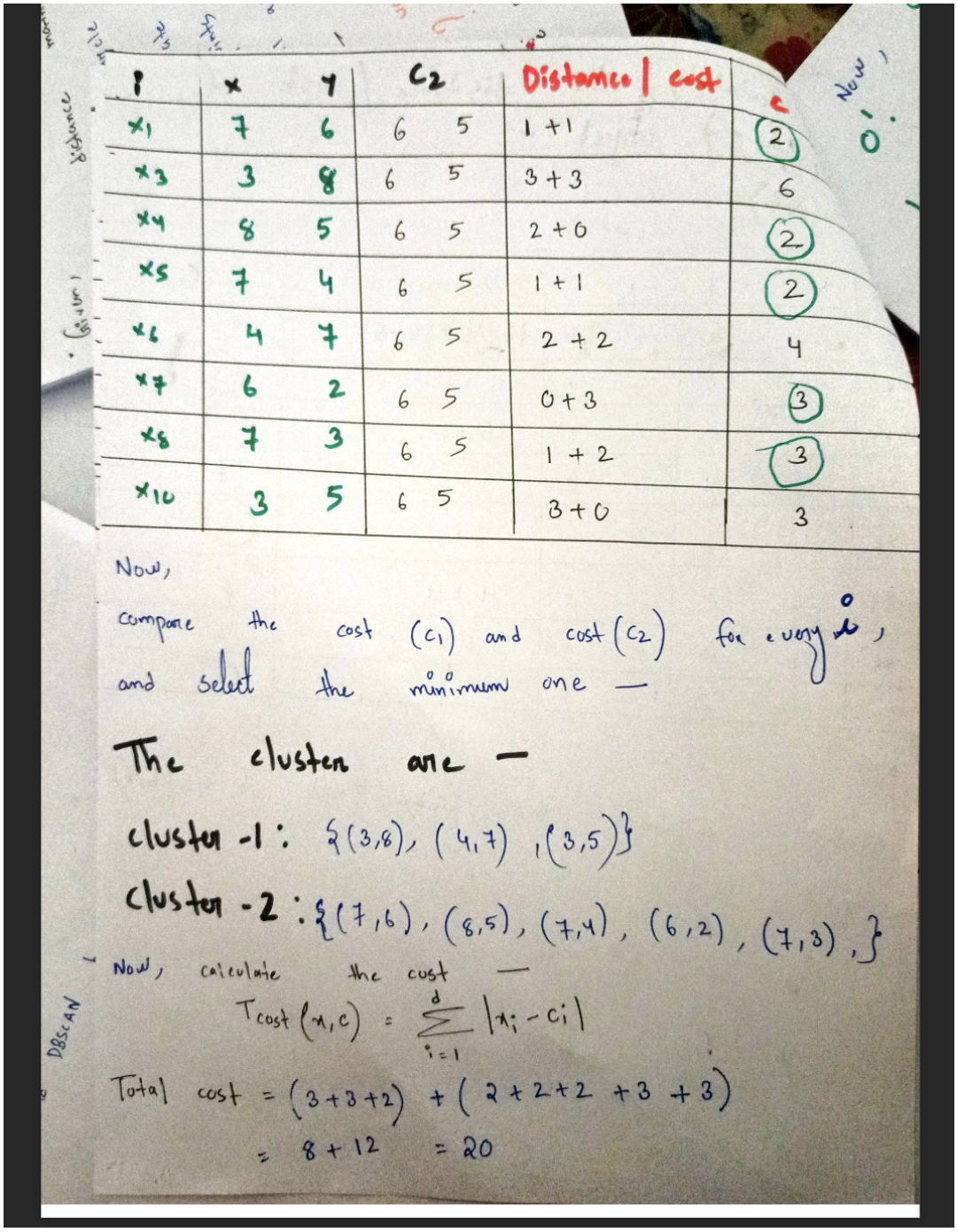


- Calculating the distance / cost of ~~C2(6,5)~~ object - $C_1(2,6)$

i	x	y	c_1		Distance / cost	C
x_1	a 7	b 6	c 2	d 6	$\|7-2\| + \|6-6\|$	5
x_3	3	8	2	6	1 + 2	(3)
x_4	8	5	2	6	6 + 1	7
x_5	7	4	2	6	5 + 2	7
x_6	4	7	2	6	2 + 1	(3)
x_7	6	2	2	6	4 + 4	8
x_8	7	3	2	6	5 + 3	8
x_{10}	3	5	2	6	1 + 1	(2)

Now,

Similarly, calculate the distance / cost of $c_2(6,5)$ object.



P	x	y	C_2	Distance	cost	
x_1	7	6	6 5	1 + 1		2
x_3	3	8	6 5	3 + 3		6
x_4	8	5	6 5	2 + 0		2
x_5	7	4	6 5	1 + 1		2
x_6	4	7	6 5	2 + 2		4
x_7	6	2	6 5	0 + 3		3
x_8	7	3	6 5	1 + 2		3
x_{10}	3	5	6 5	3 + 0		3

Now,

compare the cost (c_1) and cost (c_2) for every i, and select the minimum one —

The clusters are —

cluster -1 : $\{(3,8), (4,7), (3,5)\}$

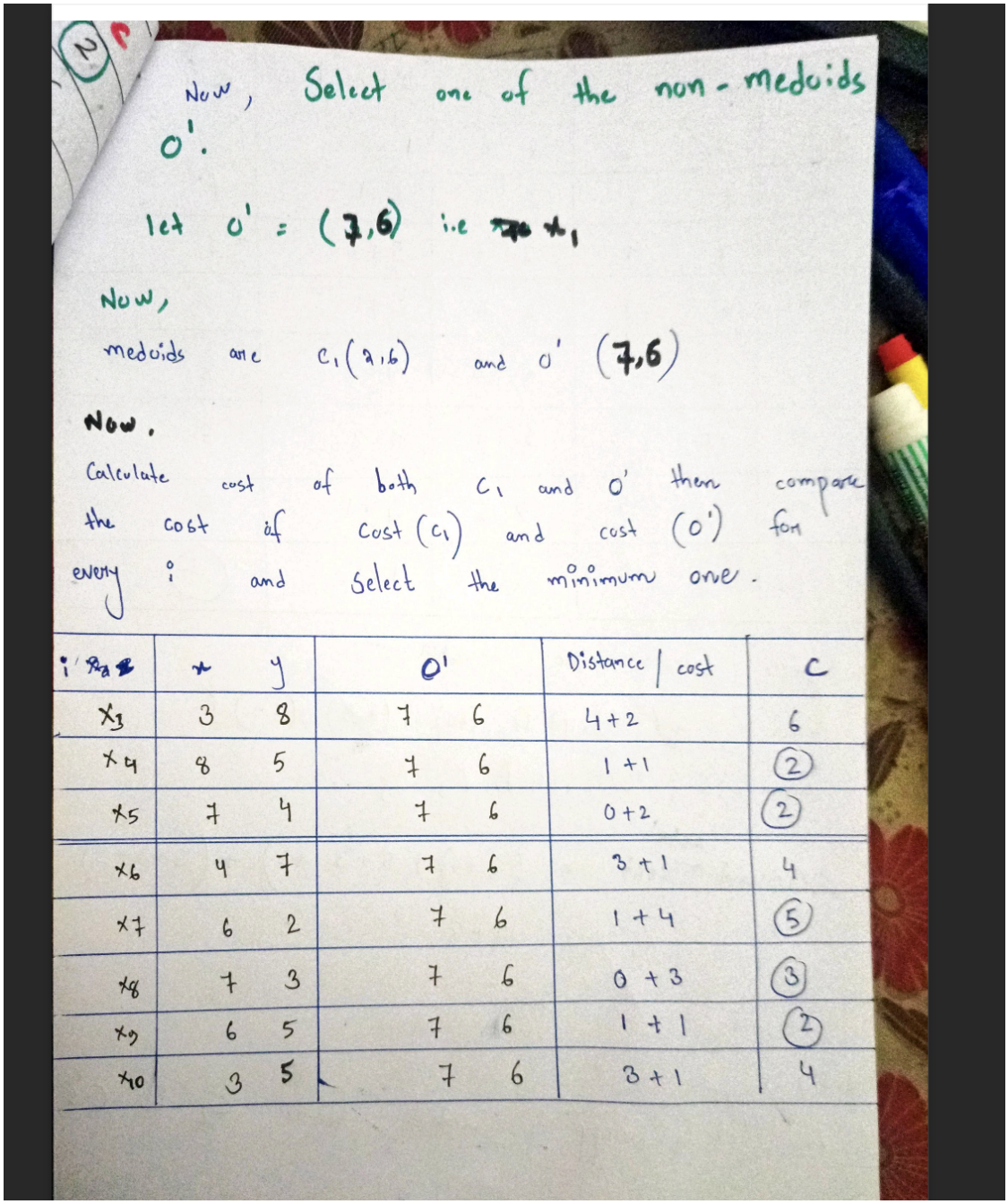
cluster -2 : $\{(7,6), (8,5), (7,4), (6,2), (7,3),\}$

Now, calculate the cost —

$$T_{cost}(x,c) = \sum_{i=1}^{d} |x_i - c_i|$$

Total cost = $(3+3+2) + (2+2+2+3+3)$

$= 8 + 12 = 20$



Now, Select one of the non-medoids O'.

let O' = (7,6) i.e

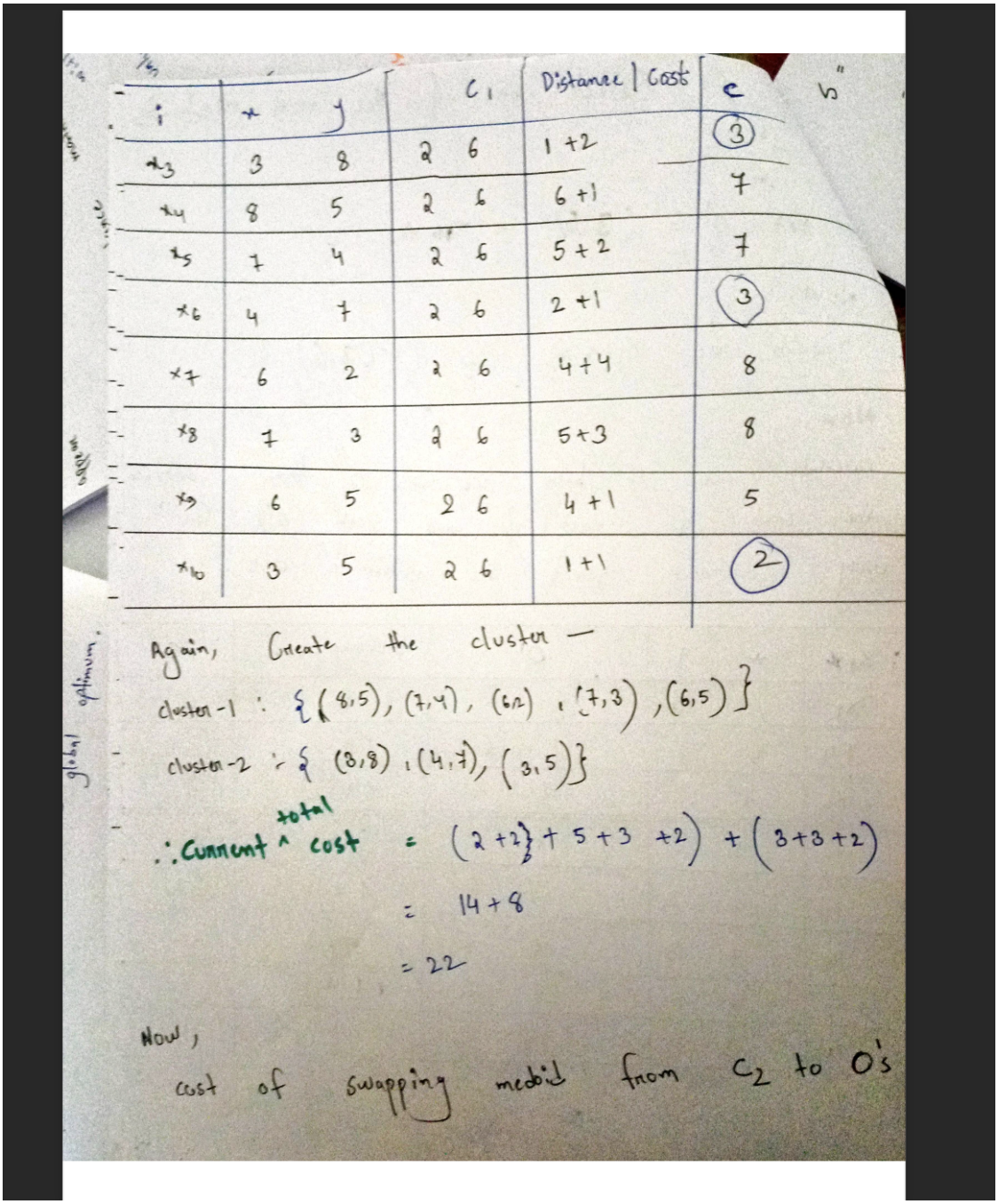
Now,

medoids are $c_1(3,6)$ and O' (7,6)

Now,

Calculate cost of both c_1 and O' then compare the cost if cost (c_1) and cost (O') for every i and Select the minimum one.

i	x	y	O'		Distance / cost	C
x_3	3	8	7	6	4+2	6
x_4	8	5	7	6	1+1	(2)
x_5	7	4	7	6	0+2	(2)
x_6	4	7	7	6	3+1	4
x_7	6	2	7	6	1+4	(5)
x_8	7	3	7	6	0+3	(3)
x_9	6	5	7	6	1+1	(2)
x_{10}	3	5	7	6	3+1	4



i	x	y	c_1	Distance / Cost	c
x_3	3	8	2 6	1 + 2	3
x_4	8	5	2 6	6 + 1	7
x_5	7	4	2 6	5 + 2	7
x_6	4	7	2 6	2 + 1	3
x_7	6	2	2 6	4 + 4	8
x_8	7	3	2 6	5 + 3	8
x_9	6	5	2 6	4 + 1	5
x_{10}	3	5	2 6	1 + 1	2

Again, Create the cluster —

cluster-1 : {(8,5), (7,4), (6,2), (7,3), (6,5)}

cluster-2 : {(3,8), (4,7), (3,5)}

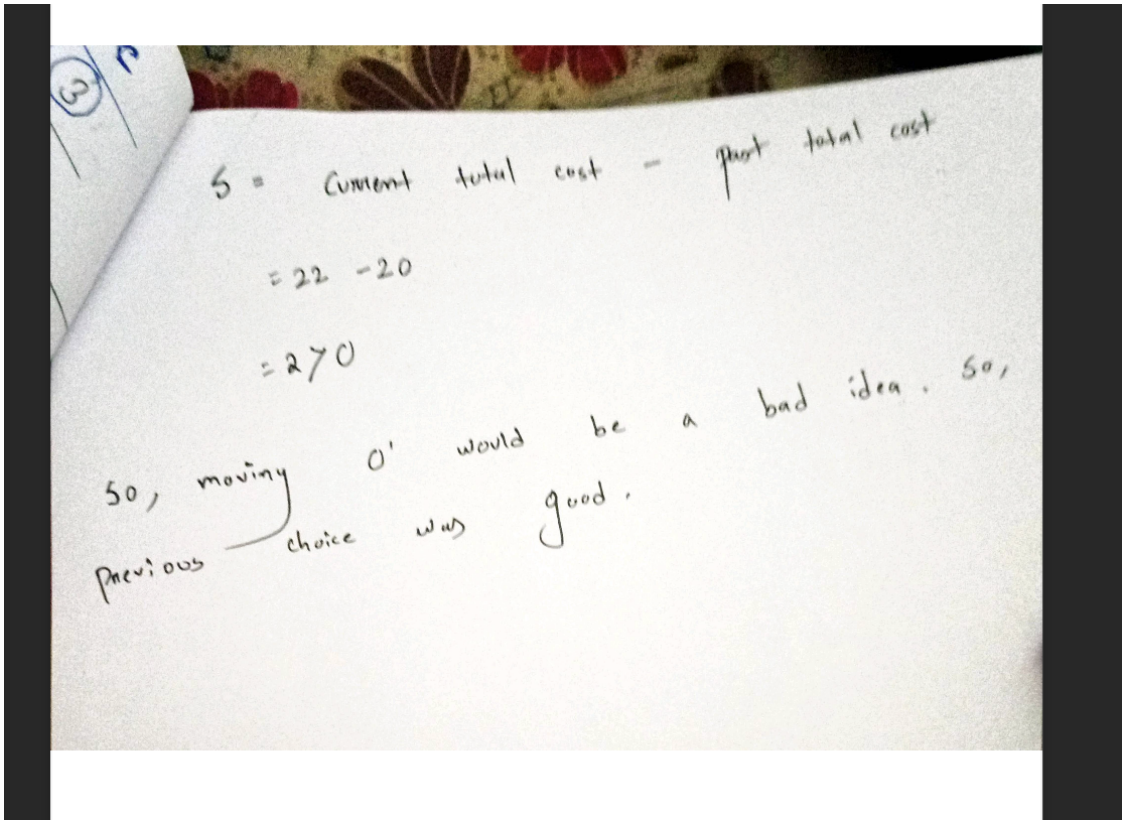
∴ Current total cost = (2 + 2) + 5 + 3 + 2) + (3 + 3 + 2)

= 14 + 8

= 22

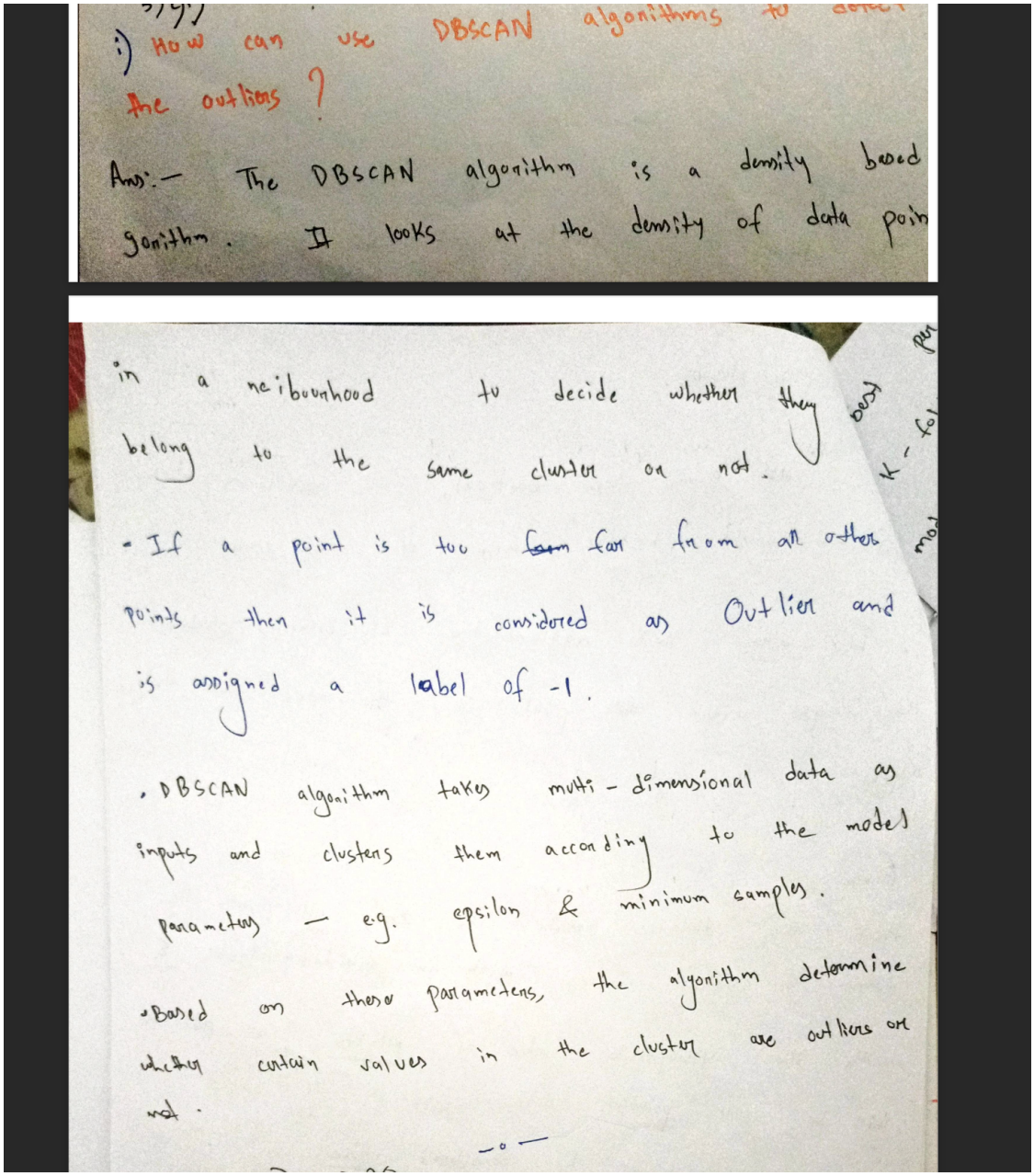
Now,

cost of swapping medoid from c_2 to O's



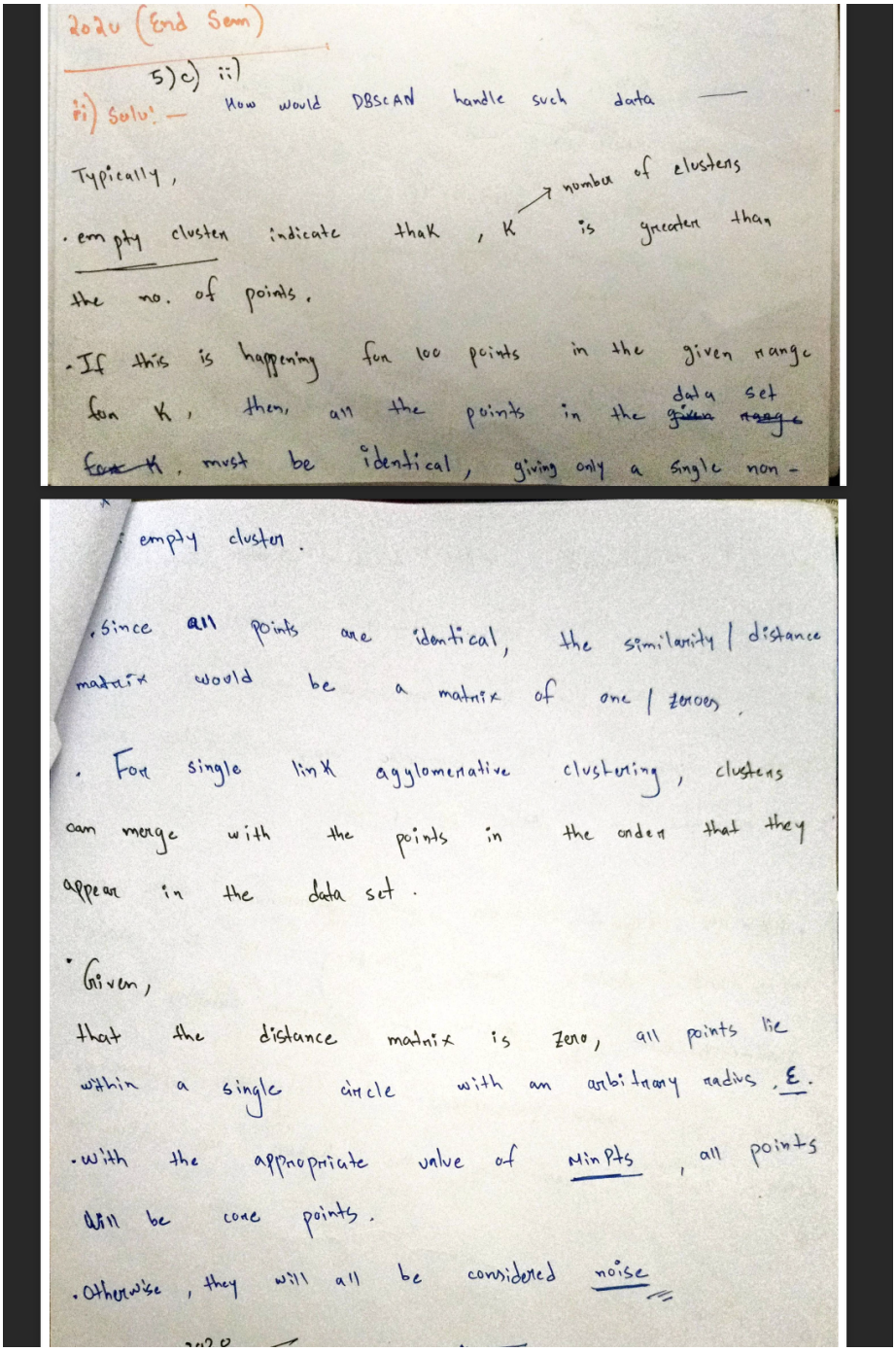
▼ Ans 5 (c) i -

How can use DBSCAN algorithm to detect outliers?



▼ Ans 5 (c) ii -

You are given a data set with 100 records and are asked to cluster this data using K-means algorithm. It is found that for all values of K, 1<=K<=100, the K-means algorithm returns only one non-empty cluster? What is the reason for this.
How would DBSCAN handle such data?



https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7222c2cf-80c0-4792-8cf0-0e85b0211df6/DM_S1_2017_Sol.pdf

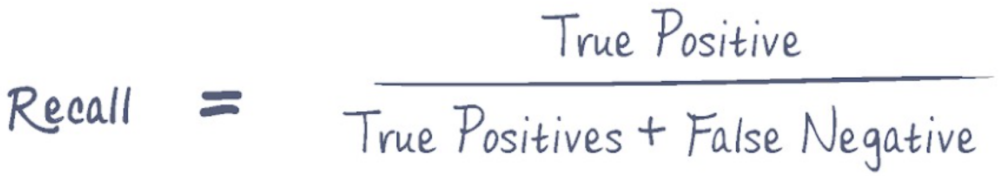
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ce64851c-7001-4b4d-a980-efda0b35fbab/DM_S2_2017_Sol.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/09dd500d-f723-47fb-b2c9-efc159a23262/DM_S2_2021_Sol.pdf

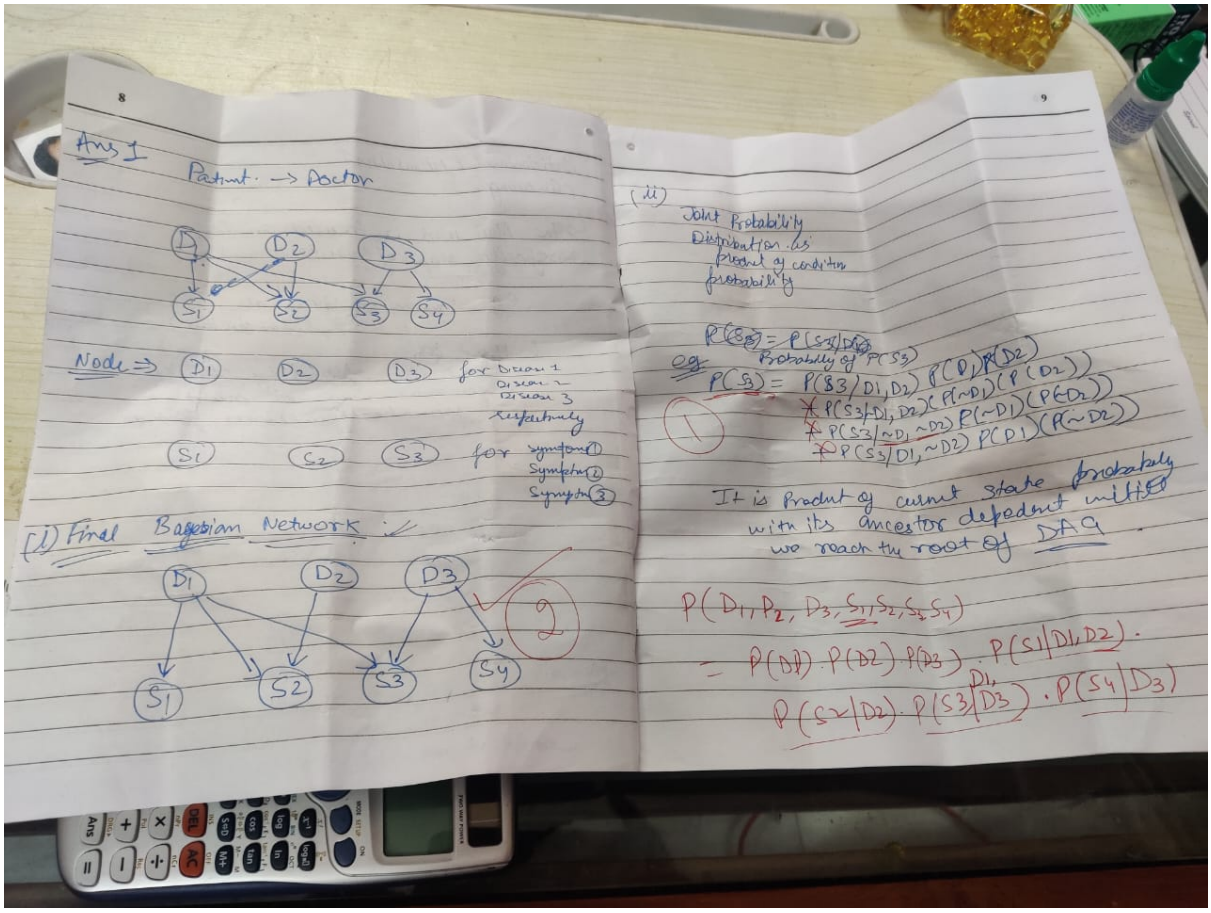
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8be6e111-e65c-4c6c-ab19-84c207523ab0/DM_S1_2022_Sol.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7dcd054f-550f-4bbe-9328-a7f0cdbb17b0/DM_S2_2022_Sol.pdf

notes-



F-Measure = (2 * Precision * Recall) / (Precision + Recall)


8
Ans 1
Patient → Doctor
D1
D2
D3
S1
S2
S3
S4
Node ⇒ D1 D2 D3 for Disease 1 Disease 2 Disease 3 respectively
S1 S2 S3 for symptom 1 Symptom 2 Symptom 3
(i) Final Bayesian Network :
9
(ii)
Joint Probability Distribution is product of condition probability
P(S3) = P(S3/D1,D2) P(D1) P(D2)
Or
P(S3) = P(S3/D1,D2) P(D1)(P(D2))
P(S3/D1,~D2) P(~D1)(P(D2))
P(S3/~D1,~D2) P(D1)(P(~D2))
P(S3/D1,~D2) P(D1)(P(~D2))
It is Product of current state probability with its ancestor dependent until we reach the root of DAG
P(D1,D2,D3,S1,S2,S3,S4)
= P(D1).P(D2).P(D3).P(S1/D1,D2).
P(S2/D2).P(S3/D1,D3).P(S4/D3)

3 Disease and symptoms [20 pts]

A patient goes to the doctor for a medical condition, the doctor suspects three diseases as the cause of the condition. The three diseases are D_1, D_2, D_3, which are marginally independent from each other. There are four symptoms S_1, S_2, S_3, S_4 which the doctor wants to check for presence in order to find the most probable cause of the condition. The symptoms are conditionally dependent to the three diseases as follows: S_1 depends only on D_1, S_2 depends on D_1 and D_2. S_3 is depends on D_1 and D_3, whereas S_4 depends only on D_3. Assume all random variables are Boolean, they are either 'true' or 'false'.

1. Draw the Bayesian network for this problem. [2 pts]

★ **SOLUTION:** The Bayesian network is shown in Figure 2.

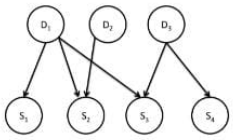


Figure 2: The Bayesian network for disease symptom problem.

2. Write down the expression for the joint probability distribution as a product of conditional probabilities. [2 pts]

★ **SOLUTION:**

$$\mathbf{P}(D_1, D_2, D_3, S_1, S_2, S_3, S_4) = \mathbf{P}(D_1)\,\mathbf{P}(D_2)\,\mathbf{P}(D_3)\,\mathbf{P}(S_1 \mid D_1)\,\mathbf{P}(S_2 \mid D_1, D_2)\,\mathbf{P}(S_3 \mid D_1, D_3)\,\mathbf{P}(S_4 \mid D_3)$$

3. What is the number of independent parameters that is required to describe this joint distribution? [3 pts]

★ **SOLUTION:** The number of independent parameters is 15. The number of independent parameters needed to describe each conditional probability distribution that is part of the joint are listed in Table 1.

4. Assume there were no conditional independence between the variables, how many independent parameters would be required then? [3 pts]

CPT	Number of independent parameters
$\mathbf{P}(D_1)$	1
$\mathbf{P}(D_2)$	1
$\mathbf{P}(D_3)$	1
$\mathbf{P}(S_1 \mid D_1)$	2
$\mathbf{P}(S_2 \mid D_1, D_2)$	4
$\mathbf{P}(S_3 \mid D_1, D_3)$	4
$\mathbf{P}(S_4 \mid D_3)$	2
Total number of independent parameters	15

Table 1: Number of independent parameters for each conditional probability distribution.

★ **SOLUTION:** Without conditional independence assumptions, the number of parameters required to specify the joint would be 127. There are 7 random variables, and each of them take 2 values, therefore $2^7 - 1 = 127$

5. What is the Markov Blanket of variable S_2? [2 pts]

★ **SOLUTION:** The Markov Blanket of S_2 is D_1 and D_2.

6. What is an example of the 'explaining away' phenomenon in the graph? [2 pts]

★ **SOLUTION:** S_3 depends on D_1 and D_3. When we observe S_3 is true and if we observe that D_1 is true, D_3 would be a less likely cause for S_3. This is an example 'explaining away' phenomenon.

7. If we observe the fourth symptom,(S_4 = true), for which diseases do we gain information? [3 pts]

★ **SOLUTION:** By observing S_4 = true we gain information about only disease D_3. Notice D_1 and D_2 are independent of S_4.

8. Suppose we observed second symptom is present in the patient (S_2 = true), what does observing the fourth symptom (S_4 = true) tell us now? [3 pts]

★ **SOLUTION:** With $S_2 = true$, observing $S_4 = true$ still gives us information only about D3. If we had observed $S_3 = true$ instead of S2, then observing $S_4 = true$ would have given us information about D_1 and D_3 due to explaining away.

▼ Q) How does k-Nearest Neighbour Classifier computes the distance of the attributes which are not numeric and how a good value of k is determined?

In the k-nearest neighbor (k-NN) classification algorithm, the distance between two data points is typically computed using a distance metric such as Euclidean distance. However, this assumes that the attributes are numeric. If the attributes are not numeric, you will need to use a different distance metric that is appropriate for the type of data you are working with.

One option is to use a distance metric specifically designed for categorical data, such as the Hamming distance. The Hamming distance between two categorical data points is the number of attributes that are different between the two points. For example, if two data points have the same values for all attributes except for one, the Hamming distance between them would be 1.

Another option is to convert the categorical attributes to numeric attributes using a technique such as one-hot encoding, and then use a distance metric such as Euclidean distance. One-hot encoding converts each categorical attribute into a set of binary attributes, with one attribute for each possible value of the categorical attribute. For example, if a categorical attribute has three possible values (A, B, and C), one-hot encoding would convert it into three binary attributes, with a value of 1 indicating the presence of a particular value and a value of 0 indicating its absence.

As for determining a good value for k, this is typically done through experimentation and cross-validation. One approach is to try a range of values for k and evaluate the performance of the model using a metric such as accuracy or F1 score. The value of k that results in the best performance is then chosen as the final model. It is also common to use a technique called k-fold cross-validation, which involves dividing the data into k folds, training the model on k-1 folds, and evaluating it on the remaining fold. This process is repeated k times, with a different fold used for evaluation each time. The final model is then trained on the full dataset using the value of k that resulted in the best performance during cross-validation.

▼ Q) Consider a sequence dataset containing at least 4 sequences which are made using a combination of 6 different characters. The sequences should be of reasonable length. Use any Sequence Mining algorithm to extract frequent sequences having a minimum support =2. The dataset should be your own and not chosen from any example given anywhere in the books.

Sequence 1: ABCABC
Sequence 2: ABCDEF
Sequence 3: ABCDEE
Sequence 4: ABCDEE

To extract the frequent sequences with a minimum support of 2 using a sequence mining algorithm, we can use the Apriori algorithm.

First, we need to define the minimum support threshold. In this case, the minimum support is 2, so we will consider any sequence that appears in at least 2 of the 4 sequences as a frequent sequence.

Next, we can use the Apriori algorithm to generate all possible sequences and count their occurrences in the dataset. For example, the sequence "ABC" appears in 3 of the 4 sequences, so it is a frequent sequence with a support of 3. The sequence "ABCD" appears in 2 of the 4 sequences, so it is also a frequent sequence with a support of 2.

Finally, we can prune the list of frequent sequences to remove any sequences that are not frequent according to the minimum support threshold. In this case, we would keep the sequences "ABC" and "ABCD" because they both have a support of at least 2, but we would discard the other sequences because they do not meet the minimum support threshold.

Overall, the Apriori algorithm is a useful tool for extracting frequent sequences from a dataset and can be used to identify patterns and trends in the data.

▼ Q) A credit card company receives thousands of applications for issue of new cards. The application contains several attributes like Name, Age and many more. The problem is to categorize the rules {a} $-\!\!\rightarrow$ {b} and {b} $-\!\!\rightarrow$ {a} have the same applications into two classes “Good Credit” or “Bad credit” based on past experience of the company. Think of a suitable, manageable and sufficient dataset for the above problem and using a decision tree classifier find the root of the tree.

A suitable, manageable, and sufficient dataset for the above problem might include the following attributes:

1. Age: The age of the credit card applicant.
2. Income: The annual income of the credit card applicant.
3. Credit score: The credit score of the credit card applicant.

4. Debt-to-income ratio: The ratio of the applicant's monthly debt payments to their monthly income.
5. Number of credit cards: The number of credit cards the applicant currently holds.
6. Payment history: A record of the applicant's payment history on their current credit cards.
7. Credit utilization: The percentage of the applicant's credit limit that they are currently using.

Using this dataset, we can train a decision tree classifier to predict whether an applicant is a good credit risk or a bad credit risk.

To find the root of the tree, we would first need to choose an attribute to split on. This could be done using a measure such as information gain, which measures how much uncertainty is reduced by making a particular split. For example, if the attribute with the highest information gain is "credit score," then the root of the tree would be a node that splits on credit score.

From there, the tree would continue to grow by making additional splits on other attributes until it reaches a point where it can accurately classify applicants as either "good credit" or "bad credit." The root of the tree would be the first node in the tree, which represents the attribute that was used to make the initial split.