Data Warehousing and Data Mining

Note By; Pravin Gupta(GUPTA TUTORIAL)

College;Birendra Multiple Campus(BMC) Bharatpur , Chitwan

Data Warehousing and Data Mining Playlist:https://bit.ly/3OuZKIC

Youtube Channel;https://www.youtube.com/@guptatutorial

Website;www.pravingupta.com.np

Telegram;https://t.me/Gupta_Tutorial

Facebook id;https://m.facebook.com/prabin.gupta.92

Instagram id;https://www.instagram.com/prabin.gupta.92/

If my note really helps you, then you can support me by scanning QR for my hard work.

Unit-5
Mining Frequent Patterns

Frequent Patterns:

Frequent Patterns are patterns that appears in a data set frequently. Frequent patterns can be frequent-itemsets, frequent subsequences, or frequent substructures. for example, a set of items, such as milk and bread, that appear frequently together in a transaction data set is a frequent itemset.

- A subsequences, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a frequent subsequences.

- A substructures can refers to different structural forms, such as sub-graphs or sub-trees. If a substructure occurs frequently, it is called a frequent structured pattern or frequent substructure.

- Finding such frequent patterns plays an essential role in mining associations, correlations and many other interesting relationships among data.

Market Basket Analysis:

It is the earliest form of the frequent pattern mining for association rules. It analyzes customer buying habits by finding associations between the different items that customers place in their "shopping baskets".

The discovery of these associations can help retailers develop marketing strategies by analyzing which items are frequently purchased together by customers.

Example: For instance, if customers are buying milk, how likely are they to also buy bread on the same trip? Such information can lead to increased sales by helping retailers do selective marketing and design different stores layouts.

Some Terms:

Itemset: A set of items is referred to as an itemset. An itemset that contains K items is a K-itemset. The set {computer, antivirus-software} is a 2-itemset.

Support Count (σ/sigma): It is the frequency of occurrence of a itemset.

Frequent Itemset:- An itemset whose support is greater than or equal to minimum support threshold. i.e, an itemset that occurs more than minimum specified number.

GOOD MORNING
PAGE NO. :
DATE : __/__/____

Closed Itemset:- An itemset X is closed in a data set D if there exists no proper super-itemset Y such that Y has the same support count as X in D.

V.V.IMP

Association Rules:

Association rules are if/then statements that help uncover relationships between clearly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he/she is 60% likely to purchase milk."

An association rule has two parts, an antecedent (if part) and a consequent (then part). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent.

Depending on the following two parameters, the important relationship are observed.

Support:- It indicates how frequently the if/then relationship appears in the database.

Confidence:- It tells about the number of times these relationships have been found to be true. e.g "If a customer buys a dozen eggs, he/she is 60% likely to purchase milk." Here 60% is the confidence.

Support $(A \Rightarrow B) = P(A \cup B)$	Confidence $(A \Rightarrow B) = \frac{P(A \cup B)}{P(A)} = \frac{\text{support}}{P(A)}$
Support of association rule $A \Rightarrow B$ is the probability that the database contains both A and B.	Confidence of $A \Rightarrow B$ is conditional probability that the transactions that contains item A also contains item B.

Examples

Calculate support and confidence of rule bread ⇒ milk considering the example given below;

Transaction_ID	Milk	Bread	Butter
1	1	1	0
2	0	0	1
3	0	0	0
4	1	1	1
5	0	1	0

Here 1 denotes true
0 denotes False.

Sol

Support (bread ⇒ milk) = 2/5 = 0.4 = 40%. (milk ∩ bread दोनों को (i.e 1 दोनों को) / Total no. of transaction.)

Confidence (bread ⇒ milk) = 2/3 = 0.66 = 66%. (Total no. of bread (i.e 1))

V.V Imp

Types of Association Rules:

(i) Single Dimensional :- Association rules involving single predicate, repeated multiple times are called single dimensional rules. Consider the example below, which is single dimensional association rule.

buys(X, "digital camera") ⇒ buys(X, "Printer")

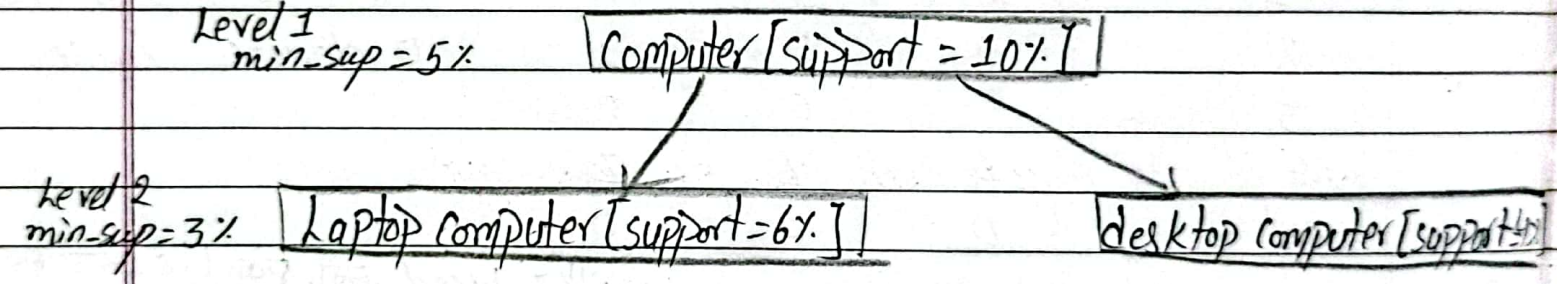
ie if X buys digital camera then X is likely to buy Printer.

GOOD MORNING
PAGE NO. :
DATE : __/__/____

(ii) **Multidimensional**:- Association rules that involves two or more predicates are referred as multidimensional association rules. Consider the rule given below that contains three predicates (age, occupation and buys) each of which occurs only once in the rule.

age(X, "20:::29") ^ occupation(X, "student") ⇒ buys(X, "laptop").

(iii) **Multilevel**:- Association rules generated from mining data at ~~multi level~~ multiple levels of abstraction are called multilevel association rules.



(iv) **Quantitative**:- Database attributes can be quantitative. These attributes have a finite number of possible values, with no ordering among the values (e.g. occupation, brand, ~~etc~~ color). Quantitative attributes are numeric and have an implicit ordering among values (e.g age, price)

Finding Frequent Itemset:

1) Apriori Algorithm: It is a classic algorithm used in data mining for learning association rules. Mining association rules basically means finding the items that are purchased together more frequently than others. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties.

→ Apriori employs an iterative approach known as a level-wise search, where frequent k-itemsets are used to explore frequent (k+1) itemsets.

→ First, the set of frequent 1-itemsets that satisfy minimum support is found by scanning the database. The resulting set is denoted by L_1.

→ Next, L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found.

→ The finding of each L_k requires one full scan of database. To improve efficiency of level-wise generation of frequent itemsets, Apriori property is used. Apriori property states that any subsets of frequent itemset must be frequent.

2ta object का bichma association find करिनाई Apriori algorithm use garxam.

GOOD MORNING
PAGE NO. :
DATE : __/__/____

Example

Find the frequent itemsets base on the following transaction dataset and then generate the association rules from the frequent itemsets by using Apriori algorithm and output only strong association rules. Use Support threshold = 50%, Confidence = 80%.

Transaction	list of items
T1	I1, I2, I3
T2	I2, I3, I4
T3	I4, I5
T4	I1, I2, I4
T5	I1, I2, I3, I5
T6	I1, I2, I3, I4

Sol

Here,

Given Support threshold = 50% = 0.5 × 6 = 3 (no. of transaction)

∴ minimum support threshold (min_sup) = 3

Iteration-1; First of all, create the itemsets of the size of 1 this itemset is called the candidate itemset, C_1 and calculate their support values.

list of items लाई arrange गर्ने

Itemset	Support Count
{I1}	4
{I2}	5
{I3}	4
{I4}	4
{I5}	2

→ I1 कति वटा छ count गर्ने अनि तेस्को like-wise sabai count गर्ने

Now generating frequent 1-itemset F_1 from the candidate 1-itemset C_1. For this we compare the support count value

of each itemset with min-sup = 3 if it is less then eliminating such itemsets. As you can see here, I5 itemsets has a support count value of 2 which is less than the min support value 3. So, it does not meet min-sup = 3. Thus it is discarded, in the upcoming iterations, only I1, I2, I3, I4 meet min-sup count. We have the frequent 1-itemset F_1 as shown below;

Itemset	Support count
{I1}	4
{I2}	5
{I3}	4
{I4}	4

Iteration-2; Candidate 2-itemset C_2 and frequent 2-itemset generation. Here, Candidate 2-itemset C_2 is generated by joining frequent 1-itemset to itself that is by finding the all possible combinations of itemset in F_1.

Itemset	Support count
{I1, I2}	4
{I1, I3}	3
{I1, I4}	2
{I2, I3}	4
{I2, I4}	3
{I3, I4}	2

Now comparing support count of each itemset with min_sup and Itemsets having Support less than 3 are eliminated again. Here, item set {I1, I4} and {I3, I4} does not meet min_sup, thus it is deleted so we obtain the following frequent itemset F_2.

Itemset	Support Count
{I1, I2}	4
{I1, I3}	3
{I2, I3}	4
{I2, I4}	3

Iteration-3; Candidate 3-itemset C_3 and Frequent 3-itemset generation. Here, Candidate 3-itemset C_3 is generated by joining Frequent 2-itemset to itself

Finding all possible combinations of itemset in F_2.
Here, all possible C_3 are: {I1, I2, I3}, {I1, I2, I4}, {I1, I3, I4}, {I2, I3, I4}.

we can see for itemset

- {I1, I2, I3} subsets {I1, I2}, {I1, I3}, {I2, I3} are occuring in F_2 thus {I1, I2, I3} is frequent.
- {I1, I2, I4} subsets {I1, I2}, {I1, I4}, {I2, I4} but {I1, I4} does not belong to F_2 (above table) so, it is not frequent so, it is deleted.
- {I1, I3, I4} subsets {I1, I3}, {I1, I4}, {I3, I4} but {I1, I4} does not belong to F_2 (above table) so, it is not frequent so, it is deleted.
- {I2, I3, I4} subsets {I2, I3}, {I2, I4}, {I3, I4} but {I3, I4} does not belong to F_2 (above table) so, it is not frequent. So, it is deleted.

Now, we find the support count for each 3-itemset C_3 as shown in below.

Itemset	Support Count
{I1, I2, I3}	3

Support of the itemset {I1, I2, I3} is 3 so, it passes the min-sup = 3 so, F_3 is

Itemset	Support Count
{I1, I2, I3}	3

From this frequent itemset F3 we cannot further generate candidate itemsets so we stop here and our final frequent itemsets is $F_3 = \{I1, I2, I3\}$

Now generating association rules from the frequent itemset {I1, I2, I3} obtained above.

From the frequent itemset F_3 discovered above the association could be:

(i) $\{I1 \wedge I2\} \rightarrow \{I3\}$

Confidence = $\frac{\text{support } \{I1, I2, I3\}}{\text{support } \{I1, I2\}} = \frac{3}{4} \times 100\% = 75\%$

(ii) $\{I1 \wedge I3\} \rightarrow \{I2\}$

Confidence = $\frac{\text{support } \{I1, I2, I3\}}{\text{support } \{I1, I3\}} = \frac{3}{3} \times 100\% = 100\%$

(iii) $\{I2 \wedge I3\} \rightarrow \{I1\}$

Confidence = $\frac{\text{support } \{I1, I2, I3\}}{\text{support } \{I2, I3\}} = \frac{3}{4} \times 100\% = 75\%$

(iv) $\{I1\} \rightarrow \{I2 \wedge I3\}$

Confidence = $\frac{\text{support } \{I1, I2, I3\}}{\text{support } \{I1\}} = \frac{3}{4} \times 100\% = 75\%$

(v) $\{I2\} \Rightarrow \{I1 \wedge I3\}$

$$\text{confidence} = \frac{\text{support}\{I1, I2, I3\}}{\text{support}\{I2\}} = \frac{3}{5} \times 100\% = 60\%.$$

(vi) $\{I3\} \Rightarrow \{I1 \wedge I2\}$

$$\text{Confidence} = \frac{\text{support}\{I1, I2, I3\}}{\text{support}\{I3\}} = \frac{3}{4} \times 100\% = 75\%.$$

Here, the association rule number (ii) is strong rule since it passes minimum confidence threshold is 80%.

So, output strong association rule is

$\{I1 \wedge I3\} \rightarrow \{I2\}$ Ans

Model Questions

Given the following data set, find the frequent itemset using Apriori algorithm with minimum support 3

T1	{A, B, C, D, E, F}
T2	{B, C, D, E, F, G}
T3	{A, D, E, H}
T4	{A, D, F, I, J}
T5	{B, D, E, K}

Sol?

Here,

minimum Support = 3

Iteration-1; First of all create the itemsets of the size of 1 this itemset is called the candidate itemset C1 and calculate their support values.

Itemset	Support count
{A}	3
{B}	3
{C}	2
{D}	5
{E}	4
{F}	3
{G}	1
{H}	1
{I}	1
{J}	1
{K}	1

Compare with support count value of each itemset with minimum support = 3 if it is less then eliminating such itemsets.

Itemset	Support count
{A}	3
{B}	3
{D}	5
{E}	4
{F}	3

Iteration-2; Here, Candidate 2-itemset C_2 is generated by Joining frequent 1-itemset to itself that is by finding the all possible combinations of itemset in F_1.

Itemset	Supportcount
{A,B}	1
{A,D}	3
{A,E}	2
{A,F}	2
{B,D}	3
{B,E}	3
{B,F}	2
{D,E}	4
{D,F}	3
{E,F}	2

Compare with minimum support=3 if it is less then eliminate such itemsets

Itemset	Supportcount
{A,D}	3
{B,D}	3
{B,E}	3
{D,E}	4
{D,F}	3

Iteration-3; Candidate 3-itemset C_3 and frequent 3-itemset generation. Here, candidate 3-itemset C_3 is generated by joining frequent 2-itemset to itself

Finding all-possible combinations of itemsets in F_2. Here, all possible C_3 are: {A,B,D}, {A,B,E}, {A,B,F}, {B,D,E}, {B,D,F}, {D,E,F} and so on.

we can see for itemset

- {A,B,D} subsets {A,B}, {A,D}, {B,D} but {A,B} doesnot belong to F2 (above table) so, it is not frequent so, it is deleted.
- {A,B,E} subsets {A,B}, {A,E}, {B,E} but {A,B} and {A,E} doesnot belong to F2 (above table) so, it is not frequent so, it is deleted.
- {B,D,E} subsets {B,D}, {B,E}, {D,E} are occuring in F_2 thus {B,D,E} is frequent.
- {B,D,F} subsets {B,D}, {B,F}, {D,F} but {B,F} doesnot belong to F2 (above table) so, it is not frequent so, it is deleted.
- {D,E,F} subsets {D,E}, {D,F}, {E,F} but {E,F} does not belong to F2 (above table) so, it is not frequent so, it is deleted.

Now, we find the Support count for each 3-itemset C_3 as shown in below

Itemset	Support Count
{B,D,E}	3

Support of the itemset {B,D,E} is 3 so, it passes the minimum support = 3 so, F_3 is

Itemset	Support Count
{B,D,E}	3

From this frequent itemset F3 we cannot further generate candidate itemset so, we stop here and our final frequent itemsets is $F_3 = \{B,D,E\}$ Ans

GOOD MORNING
PAGE NO.:
DATE: / /

Model Set-1

1. A database has five transaction. Consider the values as min_sup = 50% and min_confidence = 75%.

TID	Items
T1	Burger, chicken, Milk
T2	Burger, chicken, cheese
T3	Burger, chicken, clothes, cheese, Milk
T4	chicken, clothes, Milk
T5	chicken, Milk, clothes

(a) Find the all frequent itemsets using Apriori Algorithm
(b) List all strong association rules (with min_Support = 50% and min_Confidence = 75%.)

So] Here,

$$min\_support = 50\% = 0.5 \times 5 = 2.5$$

(a) Iteration-1; First of all create the itemsets of the size of 1 this itemset is called the candidate itemset C_1 and calculate their support values.

Itemset	SupportCount
{Burger}	3
{chicken}	5
{Milk}	4
{cheese}	2
{clothes}	3

Compare with support count value of each itemset with min_support = 2.5 if it is less then eliminating such itemsets.

Itemset	Support count
{Burger}	3
{chicken}	5
{Milk}	4
{cheese{clothes}	3

Iteration-2; Here, candidate 2-itemset C_2 is generated by joining frequent 1-itemset to itself that is by finding the all possible combinations of itemset in F1.

Itemset	Support count
{Burger, chicken}	3
{Burger, Milk}	2
{Burger, clothes}	1
{chicken, Milk}	4
{chicken, clothes}	3
{Milk, clothes}	3

Compare with minimum support = 2.5 if it is less then eliminate such itemset.

Itemset	Support count
{Burger, chicken}	3
{chicken, Milk}	4
{chicken, clothes}	3
{Milk, clothes}	3

Iteration-3; Candidate 3-itemset C_3 and frequent 3-itemset generation. Here, candidate 3-itemset C_3 is generated by joining frequent 2-itemset to itself.

Finding all-possible combinations of itemsets in F_2. Here all possible C_3 are; {Burger, chicken, Milk}, {Burger, chicken, clothes}, {chicken, Milk, clothes} and so on.

We can see for itemset

- {Burger, chicken, Milk} subsets {Burger, chicken}, {Burger, Milk}, {chicken, Milk} but {Burger, Milk} doesnot belong to F2 (above table) so, it is not frequent. So, it is deleted.

- {Burger, chicken, clothes} subsets {Burger, chicken}, {Burger, clothes}, {chicken, clothes} but {Burger, clothes} doesnot belong to F2 (above table) so, it is not frequent. So, it is deleted.

- {chicken, Milk, clothes} subsets {chicken, Milk}, {chicken, clothes}, {Milk, clothes} are occuring in F_2 thus {chicken, Milk, clothes} is frequent

Now, we find the support count for each 3-itemset C_3 as shown in below

Itemset	Support Count
{chicken, Milk, clothes}	3

Support of the itemset {chicken, Milk, clothes} is 3 so, it passes the minimum support=2.5 so F_3 is

Itemset	Support count
{chicken, Milk, clothes}	3

From this frequent itemset F_3 we cannot further generate candidate itemset so, we stop here and our final frequent itemset is

$$F_3 = \{\text{chicken, Milk, clothes}\}$$

(b) Now, generating association rules from the frequent itemset {chicken, Milk, clothes} obtained above.

From the frequent itemset F_3 discovered above the association could be

(i) $\{\text{chicken} \wedge \text{Milk}\} \rightarrow \{\text{clothes}\}$

$$\text{Confidence} = \frac{\text{support}\{\text{chicken, Milk, clothes}\}}{\text{support}\{\text{chicken, Milk}\}} = \frac{3}{4} \times 100\% = 75\%$$

(ii) $\{\text{chicken} \wedge \text{clothes}\} \rightarrow \{\text{Milk}\}$

$$\text{confidence} = \frac{\text{support}\{\text{chicken, Milk, clothes}\}}{\text{support}\{\text{chicken, clothes}\}} = \frac{3}{3} \times 100\% = 100\%$$

(iii) $\{\text{Milk} \wedge \text{clothes}\} \rightarrow \{\text{chicken}\}$

$$\text{Confidence} = \frac{\text{support}\{\text{chicken, Milk, clothes}\}}{\text{support}\{\text{Milk, clothes}\}} = \frac{3}{3} \times 100\% = 100\%$$

(iv) $\{\text{chicken}\} \rightarrow \{\text{Milk} \wedge \text{clothes}\}$

$$\text{Confidence} = \frac{\text{support}\{\text{chicken, Milk, clothes}\}}{\text{support}\{\text{chicken}\}} = \frac{3}{5} \times 100\% = 60\%$$

(v) $\{Milk\} \rightarrow \{chicken \wedge clothes\}$

$$confidence = \frac{Support\{chicken, Milk, clothes\}}{Support\{Milk\}} = \frac{3}{4} \times 100\% = 75\%$$

(vi) $\{clothes\} \rightarrow \{chicken \wedge Milk\}$

$$confidence = \frac{Support\{chicken, Milk, clothes\}}{Support\{clothes\}} = \frac{3}{3} \times 100\% = 100\%$$

Here, the association rule number (ii), (iii) & (vi) is strong rule since it passes minimum confidence threshold is 75%.

So, output strong association rule is

$\{chicken \wedge clothes\} \rightarrow \{Milk\}$,
$\{Milk \wedge clothes\} \rightarrow \{chicken\}$ &
$\{clothes\} \rightarrow \{chicken \wedge Milk\}$ } Ans

Model set-2

Q.N (3) Consider the following transaction data set 'D' shows 9 transactions and list of items - using Apriori algorithm to find frequent itemset min-support threshdd is 22% and minimum confidence required is 70%. Then, generate association rules.

Tid	list of items
T1	I_1, I_2, I_3, I_5
T2	I_2, I_4
T3	I_2, I_3
T4	I_1, I_2, I_4
T5	I_2, I_3 I_1, I_3
T6	I_2, I_3
T7	I_1, I_3
T8	I_1, I_2, I_3, I_5
T9	I_1, I_2, I_3

Sol

Here,

$$\text{min-support} = 22\% = 0.22 \times 9 = 1.98$$

Iteration-1; First of all create the itemsets of the size of 1 this itemset is called candidate itemset C_1 and calculate their support values.

Itemset	Support Count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

C_1

Compare with support count value of each itemset with min-support = 1.98 if it is less then eliminating such itemsets.

Itemset	Support Count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

L_1

Iteration-2; Here, Candidate 2-itemset C_2 is generated by Joining frequent 1-itemset to itself that is by finding the all possible combinations of itemset in L_1.

Itemset	Support Count
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_4\}$	1
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2
$\{I_3, I_4\}$	0
$\{I_3, I_5\}$	1
$\{I_4, I_5\}$	0

C_2

Compare with minimum support = 1.98 if it is less then eliminate such itemsets.

Itemset	Support count
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2

L_2

Iteration-3: Candidate 3-itemset C_3 and frequent 3-itemset generation. Here, Candidate 3-itemset C_3 is generated by Joining frequent 2-itemset to itself.

Finding all-possible combinations of itemset in F_2. Here, all possible C_3 are: $\{I_1, I_2, I_3\}$, $\{I_1, I_2, I_4\}$, $\{I_1, I_2, I_5\}$, $\{I_1, I_3, I_4\}$, $\{I_2, I_3, I_4\}$

we can see for itemset.

- $\{I_1, I_2, I_3\}$ - Subsets $\{I_1, I_2\}$, $\{I_1, I_3\}$, $\{I_2, I_3\}$ are occuring in F_2 thus $\{I_1, I_2, I_3\}$ is frequent.
- $\{I_2, I_2, I_4\}$ subsets $\{I_1, I_2\}$, $\{I_1, I_4\}$, $\{I_2, I_4\}$ but $\{I_1, I_4\}$ doesnot belong to F2 (above table) So, it is not frequent so, it is deleted.
- $\{I_1, I_2, I_5\}$ Subsets $\{I_1, I_2\}$, $\{I_1, I_5\}$, $\{I_2, I_5\}$ are occuring in F_2 thus $\{I_1, I_2, I_5\}$ is frequent.
- $\{I_1, I_3, I_4\}$ subsets $\{I_1, I_3\}$, $\{I_1, I_4\}$, $\{I_3 I_4\}$ but $\{I_1, I_4\}$ and $\{I_3, I_4\}$ doesnot belong to F2 (above table) so, it is not frequent. so, it is deleted.
- $\{I_2, I_3, I_4\}$ subsets $\{I_2, I_3\}$, $\{I_2, I_4\}$, $\{I_3, I_4\}$ but $\{I_3, I_4\}$ does not belong to F2 (above table) so, it is not frequent. So, it is deleted.

Now, we find the support_count for each 3-itemset C_3 as shown in below:

Itemset	Support Count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

C_3

Support of the compare with min_support = 1.98 if it is less then eliminate such itemsets.

Iteration-4;

itemset	Support count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

L_3.

Iteration-4; The algorithm uses L_3 to Join L_3 to generate a Candidate set of 4-itemsets C_4. Although the Join results in $\{I_1, I_2, I_3, I_5\}$. this itemset is pruned since its subsets $\{I_2, I_3, I_5\}$ is not frequent

Thus,

itemset	Support Count
$\{I_1, I_2, I_3, I_5\}$	1

C_4

Compare with min_support = 1.98 and it is found that the $C_4 = 1$ So, algorithm terminates.

Thus, frequent itemsets are : $\{I_1, I_2\}$, $\{I_1, I_3\}$, $\{I_1, I_5\}$, $\{I_2, I_3\}$, $\{I_2, I_4\}$, $\{I_2, I_5\}$, $\{I_1, I_2, I_3\}$, $\{I_1, I_2, I_5\}$

GOOD MORNING
PAGE NO. :
DATE : __/__/____

Now, generating association rules from the frequent itemset

$\{I_1, I_2\}$, $\{I_1, I_3\}$, $\{I_1, I_5\}$, $\{I_2, I_3\}$, $\{I_2, I_4\}$, $\{I_2, I_5\}$, $\{I_1, I_2, I_3\}$ and $\{I_1, I_2, I_5\}$

Let's take $\{I_1, I_2, I_5\}$ you may also take $\{I_1, I_2, I_3\}$

(i) $\{I_1 \wedge I_2\} \rightarrow \{I_5\}$

$$\text{Confidence} = \frac{\text{Support}\{I_1, I_2, I_5\}}{\text{Support}\{I_1, I_2\}} = \frac{2}{4} \times 100\% = 50\% \text{ Rejected}$$

(ii) $\{I_1 \wedge I_5\} \rightarrow \{I_2\}$

$$\text{Confidence} = \frac{\text{Support}\{I_1, I_2, I_5\}}{\text{Support}\{I_2, I_5\}} = \frac{2}{2} \times 100\% = 100\% \text{ Selected}$$

(iii) $\{I_2 \wedge I_5\} \rightarrow \{I_1\}$

$$\text{Confidence} = \frac{\text{Support}\{I_2, I_2, I_5\}}{\text{Support}\{I_2, I_5\}} = \frac{2}{2} \times 100\% = 100\% \text{ selected}$$

(iv) $\{I_1\} \rightarrow \{I_2 \wedge I_5\}$

$$\text{Confidence} = \frac{\text{Support}\{I_1, I_2, I_5\}}{\text{Support}\{I_1\}} = \frac{2}{6} \times 100\% = 33\% \text{ Rejected}$$

(v) $\{I_2\} \rightarrow \{I_1 \wedge I_5\}$

$$\text{Confidence} = \frac{\text{Support}\{I_1, I_2, I_5\}}{\text{Support}\{I_2\}} = \frac{2}{7} \times 100 = 29\% \text{ Rejected.}$$

(vi) $\{I_5\} \rightarrow \{I_1 \wedge I_2\}$

$$\text{Confidence} = \frac{\text{Support}\{I_1, I_2, I_5\}}{\text{Support}\{I_5\}} = \frac{2}{2} \times 100\% = 100\% \text{ Selected}$$

Here, in this way we have found three strong association rules.

Book Exercise

16. A database has 15 transaction contains the 9 items only. A = {A1, A2, A3, A4, A5, A6, A7, A8, A9} Let min Sup = 20% and min Conf = 60%. find all frequent itemsets using Apriori algorithm. List all strong association rules.

A1	A2	A3	A4	A5	A6	A7	A8	A9
1	0	0	0	1	1	0	1	0
0	1	0	1	0	0	0	1	0
0	0	0	1	1	0	1	0	0
0	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0
0	1	1	1	0	0	0	0	0
0	1	0	0	0	1	1	0	1
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0
0	0	1	0	1	0	1	0	0
0	0	1	0	1	0	1	0	0
0	0	0	0	1	1	0	1	0
0	1	0	1	0	1	1	0	0
1	0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0	1

1 → Presence of item in the transaction and
0 → Absence of item in the transaction

Sol Given,

minimum Support (min-Supp) = 20% = $\frac{20}{100} \times 15 = 3$.

∴ Min-Sup = 3

Iteration-1; First of all create the itemsets of the size of 1 this itemset is called candidate itemset C_1 and calculate their Support values.

Itemset	Support count
A1	2
A2	6
A3	6
A4	4
A5	8
A6	5
A7	7
A8	4
A9	2

Compare with support count value of each itemset with min_sup = 3 if it is less then eliminating such itemsets.

Itemsets	Support count
A2	6
A3	6
A4	4
A5	8
A6	5
A7	7
A8	4

Iteration-2; Here, Candidate 2-itemset C_2 is generated by Joining frequent 1-itemset to itself that is by finding all possible combinations of itemsets in F1

Itemset	Support count
{A2, A3}	3
{A2, A4}	3
{A2, A5}	0
{A2, A6}	2
{A2, A7}	2
{A2, A8}	1
{A3, A4}	1
{A3, A5}	3
{A3, A6}	0
{A3, A7}	3
{A3, A8}	0
{A4, A5}	1
{A4, A6}	0
{A4, A7}	2
{A4, A8}	1
{A5, A6}	3
{A5, A7}	5
{A5, A8}	2
{A6, A7}	3
{A6, A8}	2
{A7, A8}	0

Comparing with Support count value of each itemset with min-sup = 3 if it is less then eliminating such itemset.

Itemset	Support count
{A2, A3}	3
{A2, A4}	3
{A3, A5}	3
{A3, A7}	3
{A5, A6}	3
{A5, A7}	5
{A6, A7}	3

- Iteration-3; Candidate 3-itemset C_3 and frequent 3-itemset generation. Here, Candidate 3-itemset C_3 is generated by joining frequent 2-itemset to itself

Finding all-possible combinations of itemsets in F2. Here all possible C_3 are:
{A2, A3, A4}, {A2, A3, A5}, {A2, A3, A6}, {A2, A3, A7}, {A3, A4, A5}, {A3, A4, A6}, {A3, A4, A7}, {A3, A5, A7}, {A3, A5, A6}, and so on.
we can see. for itemset.

- {A2, A3, A4} subsets {A2, A3}, {A2, A4}, and {A3, A4} but {A3, A4} does not belong to F2 (above table) so, it is not frequent. so, it is deleted.
- {A2, A3, A5} subsets {A2, A3}, {A2, A5} and {A3, A5} but {A2, A5} does not belong to F2 (above table) so, it is not frequent. so, it is deleted.

Similarly {A2, A3, A6}, {A2, A3, A7}, {A3, A4, A5}, {A3, A4, A6}, {A3, A4, A7} & {A3, A5, A6}

- {A3, A5, A7} subsets {A3, A5}, {A3, A7} & {A5, A7} are occuring in F2 thus {A3, A5, A7} is frequent

Now, we find the support count for each 3-itemset C_3 as shown in below

Itemset	Support count
{A3, A5, A7}	3

Support count is 3 which is equal to min_support so, it passes.

Itemset	Support Count
{A3, A5, A7}	3

From this frequent itemset F_3 we cannot further generate candidate itemset so, we stop here and our final frequent itemsets is {A3, A5, A7} Ans

Now, for association rules from the frequent itemset {A3, A5, A7}

(i) {A3, A5} → {A7} Confidence = $\frac{\text{support}\{A3, A5, A7\}}{\text{supp}\{A3, A5\}} = \frac{3}{3} \times 100\% = 100\%$.

(ii) {A5, A7} → {A3} confidence = $3/5 \times 100\% = 60\%$.

(iii) {A3, A7} → {A5} Conf. = $3/3 \times 100\% = 100\%$.

(iv) {A5} → {A3, A7} Conf. = $3/8 \times 100\% = 37.5\%$.

(v) {A3} → {A5, A7} Conf = $3/6 \times 100\% = 50\%$.

(vi) {A7} → {A3, A5} Conf = $3/7 \times 100\% = 42\%$.

Now, the strong association rules are {A3, A5} → {A7}, {A5, A7} → {A3}, {A3, A7} → {A5} since it passes minimum confidence threshold 60%.

GOOD MORNING
PAGE NO. :
DATE : __/__/____

Limitations of Apriori Algorithm

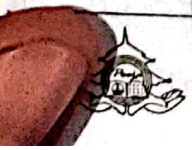
- It needs to generate a huge number of candidates sets.
- Multiple scans of transaction database so, to mine large data sets for long patterns this algorithm is not a good choice.
- When database is scanned to check C_k for creating F_k, a large number of transactions will be scanned even they do not contain any K-itemset.

Methods to Improve Apriori Efficiency

To improve the Apriori efficiency need to reduce passes of transaction database scans, shrink number of candidates, facilitate support counting of candidates. Many methods are available for improving the efficiency of the algorithm.

(i) Hash-Based Technique:- This method uses a hash-based structure called a hash table for generating the K-itemsets and its corresponding count. It uses a hash function for generating the table.

(ii) Transaction Reduction: This method reduces the number of transactions scanning in future iterations. The transactions which do not contain K-frequent items are marked or removed because such transaction cannot contain (K+1) frequent itemsets.



(iii) Partitioning: This method requires only two database scans to mine the frequent itemsets. It says that for any itemset to be potentially frequent in the database, it should be frequent in at least one of the partitions of the database. In Scan 1 partition database and find local frequent patterns and In Scan 2 consolidate global frequent patterns.

(iv) Sampling :- This method picks a random sample S from Database D and then searches for frequent itemset in S using Apriori. Scan database once to verify frequent itemsets found in Sample S, only borders of closure of frequent patterns are checked for example:- check abcd instead ab, ac, ... etc. Scan database again to find missed frequent patterns. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min-sup

(v) Dynamic Itemset Counting :- This technique can add new candidate itemsets at any marked start point of the database during the scanning of the database. Find longer frequent patterns based on shorter frequent patterns and local database partitions.

GOOD MORNING
PAGE NO.:
DATE: __/__/____

Application Areas of Apriori Algorithm

(i) In Education field: Extracting association rules in data mining of admitted students through characteristics and specialties

(ii) In the Medical field: Analysis of the patient's database

(iii) In Forestry:- Analysis of probability and intensity of Forest fire with the forest fire data.

(iv) Apriori is used by many companies like Amazon in the Recommender System and by Google for the auto-complete feature.

Advantages and Disadvantages of Apriori Algorithm

Advantages
- Easy to understand algorithm
- Join and prune steps are easy to implement on large itemsets in large databases.

Disadvantages
- It requires high computation if the itemsets are very large and the minimum support is kept very low.
- The entire database needs to be scanned.

2) FP-Growth Algorithm:

The FP-Growth Algorithm is an alternative way to find frequent itemsets without using candidate generations, thus improving performance. It uses a divide-and-conquer strategy. It uses special data structure named frequent-pattern tree (FP-Tree), which retains the itemset association information. It uses two step approach.

Step-1: Build a compact data structure called the FP-tree. It is built using two passes over the data-set.

Step-2: Extracts frequent itemsets directly from the FP-tree by traversing through FP-Tree.

Examples: Find all frequent itemsets or frequent patterns in the following database by using FP-growth algorithm. Take minimum support (min sup) = 2

TID	List of item IDs
1	I1, I2, I5
2	I2, I4
3	I2, I3
4	I1, I2, I4
5	I1, I3
6	I2, I3
7	I1, I3
8	I1, I2, I3, I5
9	I1, I2, I3

GOOD MORNING
PAGE NO. :
DATE : __/__/____

Sol Now, building a FP tree of given transaction database.

Constructing 1-itemsets and counting support count for each item set.

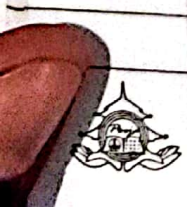
Itemset	Support Count
I1	6
I2	7
I3	6
I4	2
I5	2

Compare with support count value of each itemset with min-support = 2 if it is less then eliminating such itemsets.

Itemset	Support count
I1	6
I2	7
I3	6
I4	2
I5	2

Sorting frequent 1-itemsets in decreasing order of their support count. (give high Priority to the high value)

Itemset	Support count
I2	7
I1	6
I3	6
I4	2
I5	2

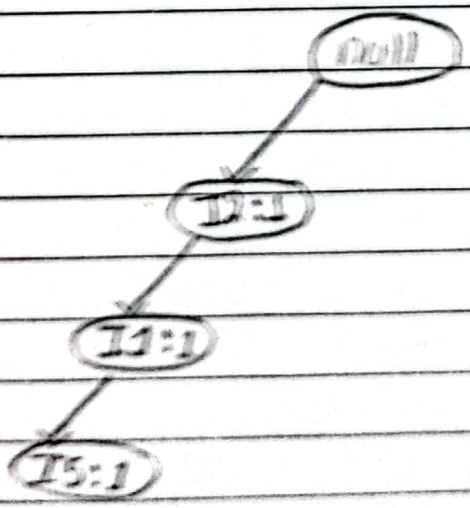


Now, ordering each itemsets in based on frequent 1 - itemsets above

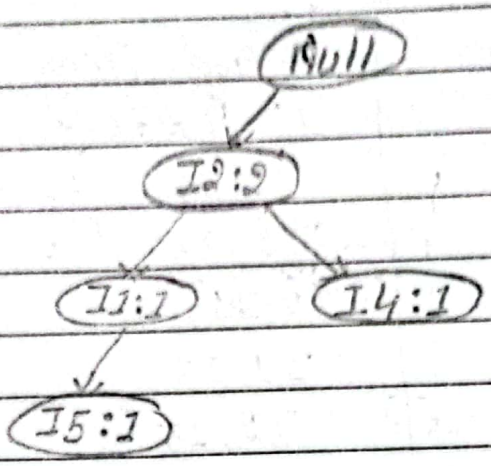
TID	List of items	Ordered items
1	I1, I2, I5	I2, I1, I5
2	I2, I4	I2, I4
3	I2, I3	I2, I3
4	I1, I2, I4	I2, I1, I4
5	I1, I3	I1, I3
6	I2, I3	I2, I3
7	I1, I3	I1, I3
8	I1, I2, I3, I5	I2, I1, I3, I5
9	I1, I2, I3	I2, I1, I3

Now, drawing FP-Tree by using ordered itemsets one by one:

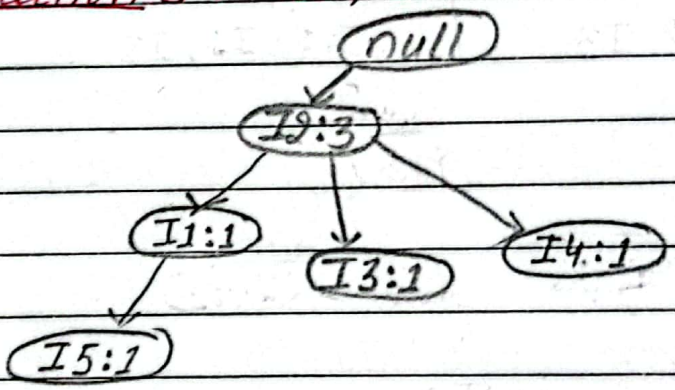
1. For Transaction 1 : I2, I1, I5



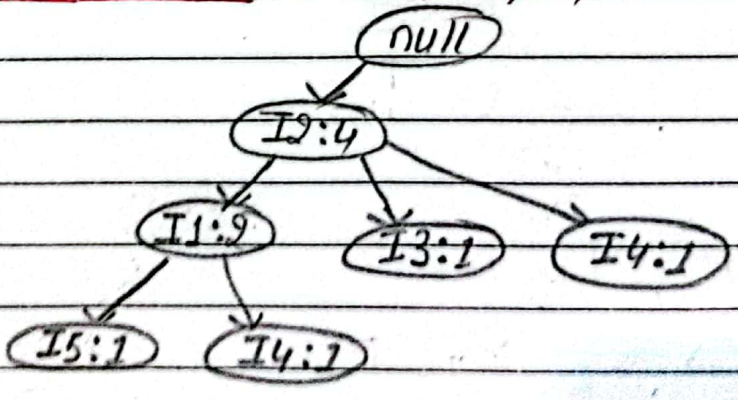
2. For Transaction 2 : I2, I4



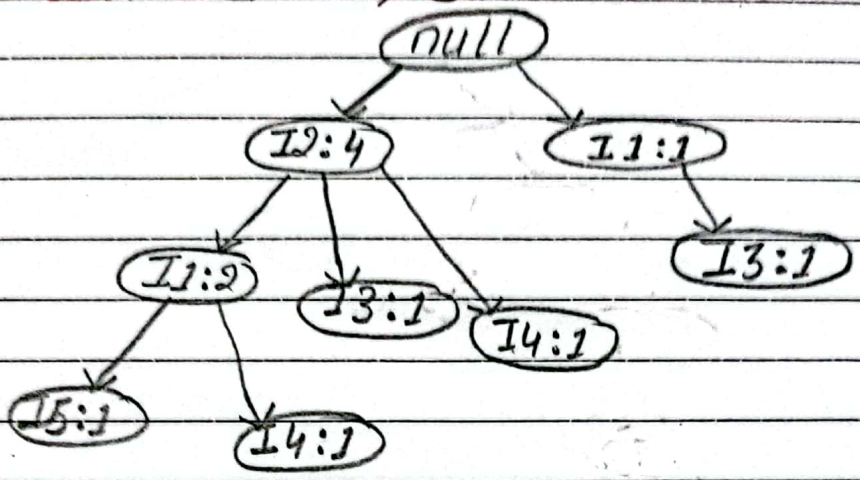
3: For Transaction 3 : I2, I3



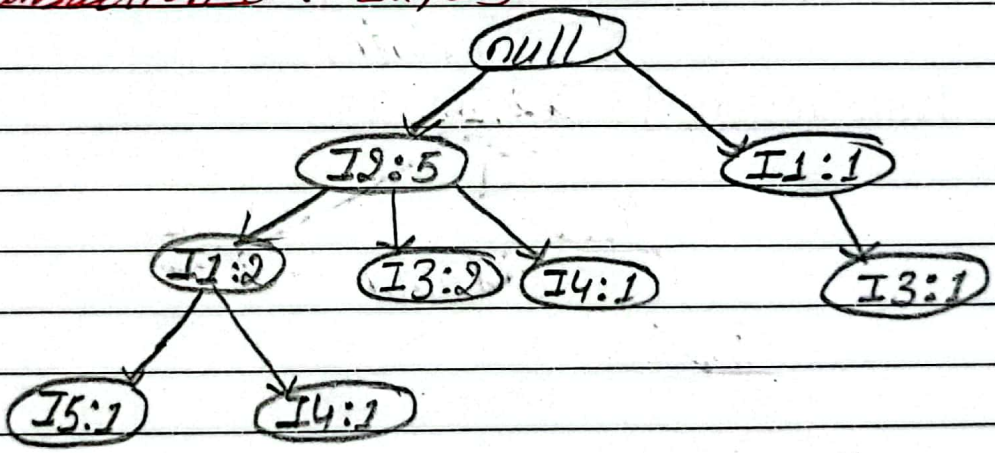
4: For Transaction 4 : I2, I1, I4



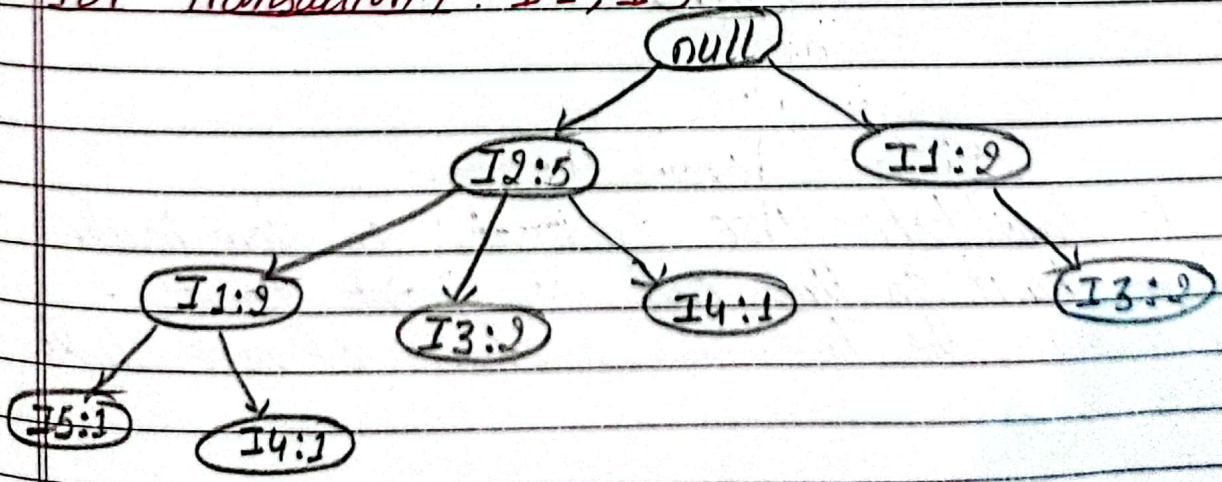
5. For Transaction 5 : I1, I3



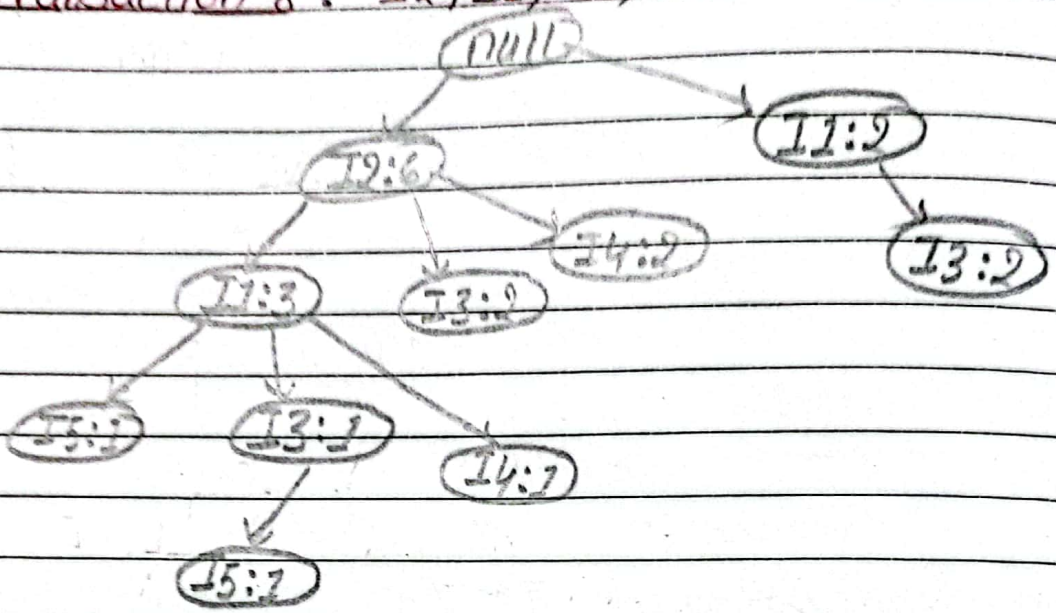
6. For Transaction 6 : I2, I3



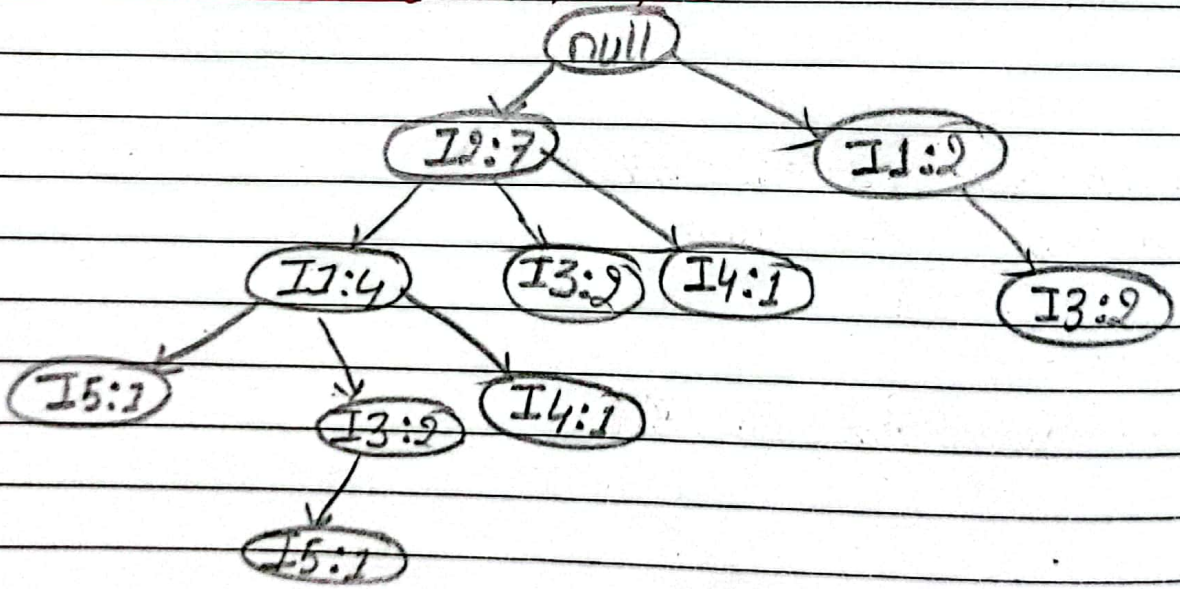
7. For Transaction 7 : I1, I3



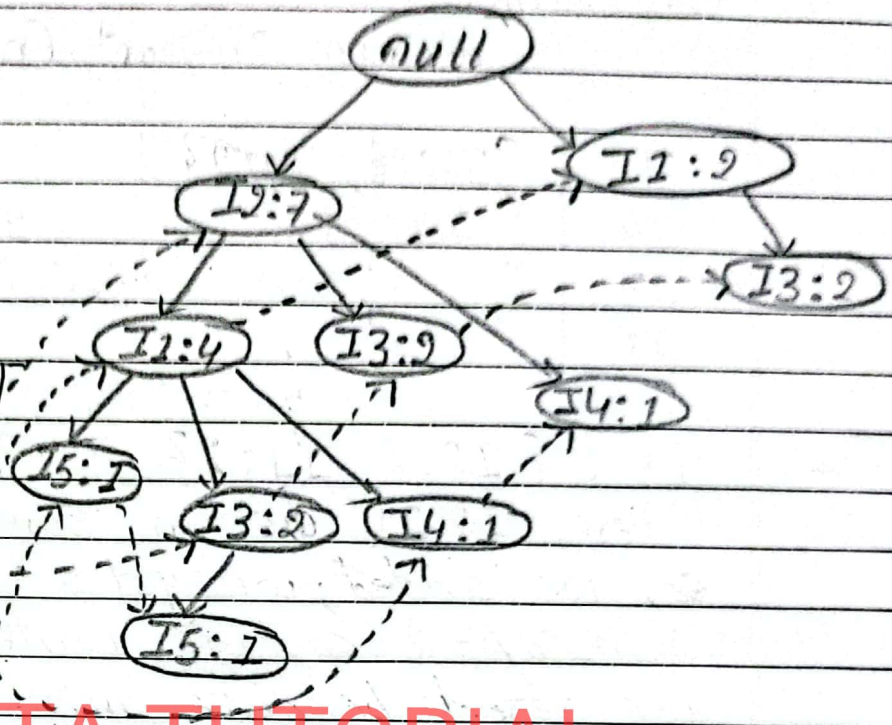
8. For Transaction 8 : I2, I1, I3, I5



9. For Transaction 9 : I2, I1, I3



Now, to facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links.



Item ID	Support Count	Node Link
I2	7	-
I1	6	.....
I3	6	- - - - -
I4	2	- - - - -
I5	2	- - - - -

Now, we need to find

(i) Conditional Pattern Base
(ii) Conditional FP Tree
(iii) Frequent Pattern Generated

(i) Conditional Pattern Base

For this start with last item in sorted order that is ~~I5, I4, I~~ I5, I4, I3, I1, I2

	Conditional Pattern Base
I5	{I2, I1, I3:1}, {I2, I1:1}
I4	{I2, I1:1}, {I2:1}
I3	{I2, I1:2}, {I2:2}, {I1:2}
I1	{I2:4}

Here, I1 is in two size of tree but we don't take {I1:2} because it is directly link with null So, we have to take other i.e {I1:4}.

** Here we don't take I2 because it is directly link with null if we found other part on the tree we can take it.

GOOD MORNING
PAGE NO.:
DATE: __/__/____

(11) Conditional FP Tree

minimum Support (min-supp) = 2

I5	{I2:2, I1:2}
I4	{I2:2}
I3	{
I1	

- I5 : {I2:2, I1:2}
  Here, I2, I1 are in same ~~size~~ side of branch so, we added; we got 2 which is equal to min_sup=2 so, we write it and I3:1 so, we reject because min-sup=2 so,

- I4 : {I2:2}
  but we can't write {I1:1} because min-sup=2 which is smaller than min-sup so we reject

- I3 : {I2:4, I1:2}, {I1:2}
  Here, {I2,I1:2} & {I2:2} lies on same side of branch so, we merged it and we get.
  {I2:4, I1:2} and {I1:2} lies on other branch so, we write it seperatly if it meet min-sup=2 then we write otherwise no need to write.

- I1 : {I2:4}
  which is min-sup=2

GOOD MORNING
PAGE NO. :
DATE : __/__/____

(iii) Frequent Pattern Generated

Now, generate the frequent pattern from the conditional FP-Tree.

	Frequent Pattern Generated
I5	{I2, I5 : 2}, {I1, I5 : 2}, {I2, I1, I5 : 2}
I4	{I2, I4 : 2}
I3	{I2, I3 : 4}, {I1, I3 : 4}, {I2, I1, I3 : 2}
I1	{I2, I1 : 4}

{I2 : 4, I1 : 2}, {I1 : 2}

combine and add I3 ; {I1, I3 : 4}

{I1 : 2} combine I2, I1 and add I3

So, {I2, I1, I3 : 2}

These are the frequent patterns generated by FP-growth algorithm for the given transaction database.

2078 Questions

2. Generate the frequent pattern from the following data set using FP growth, where minimum support = 3

T_ID	Items
1	A, B, C, D, F, H
2	A, B, D, E, F
3	C, D, F
4	B, H
5	A, C, F, G, H
6	C, D, E, G
7	A, C, D, I

Sol?

Given,

minimum support = 3

Constructing 1-itemsets and Counting Support count for each itemset

Itemsets	Support count
A	4
B	2
C	5
D	5
E	2
F	4
G	2
H	3
I	1

Compare with support count value of each itemset with minimum support = 3 if it is less then eliminating such itemsets.

Itemsets	Support count
A	4
C	5
D	5
F	4
H	3

Sorting frequent 1-itemsets in decreasing order of their support count (give high priority to the high value)

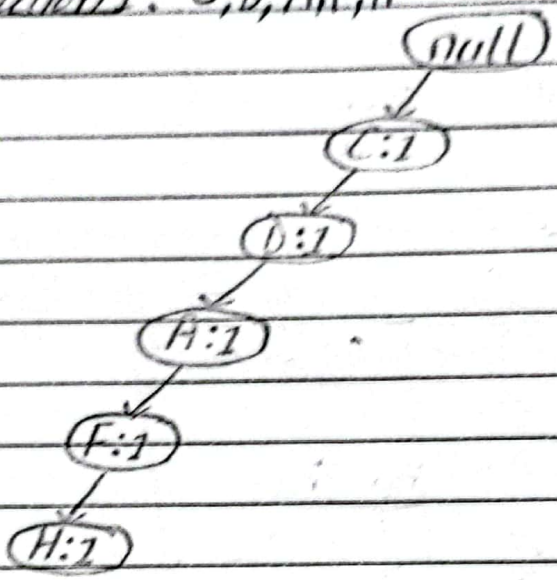
Itemsets	Support count
C	5
D	5
A	4
F	4
H	3

Now, ordering each itemsets in based on frequent 1-itemset

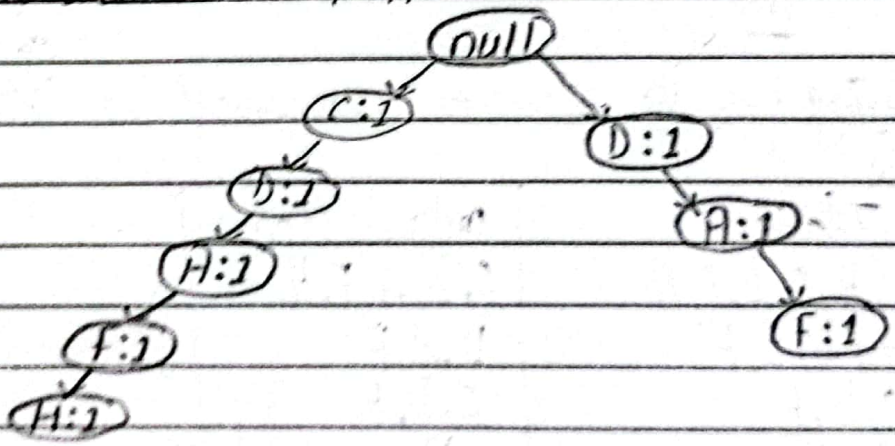
T_ID	Items	ordered items
1	A, B, C, D, F, H	C, D, A, F, H
2	A, D, E, F	D, A, F
3	C, D, F	C, D, F
4	B, H	H
5	A, C, F, G, H	C, A, F, H
6	C, D, E, G	C, D
7	A, C, D, I	C, D, A

Now, drawing FP-Tree by using ordered itemsets one by one.

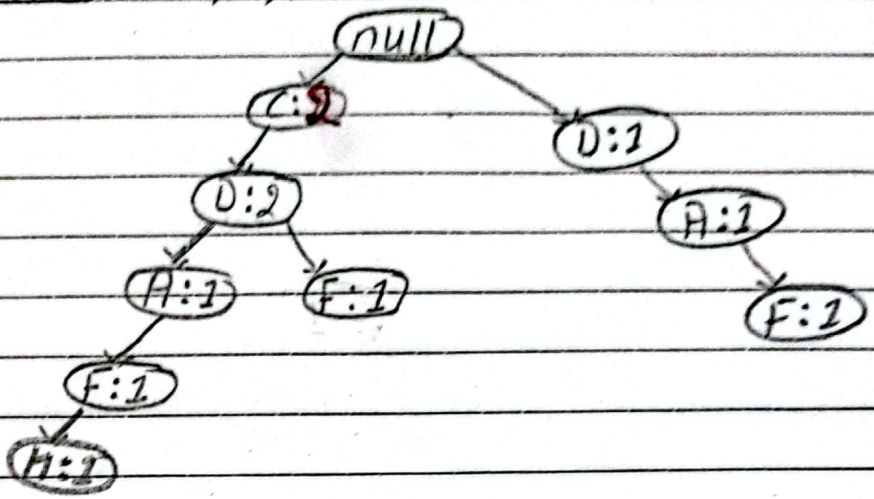
1. For Transaction 1: C, D, A, F, H



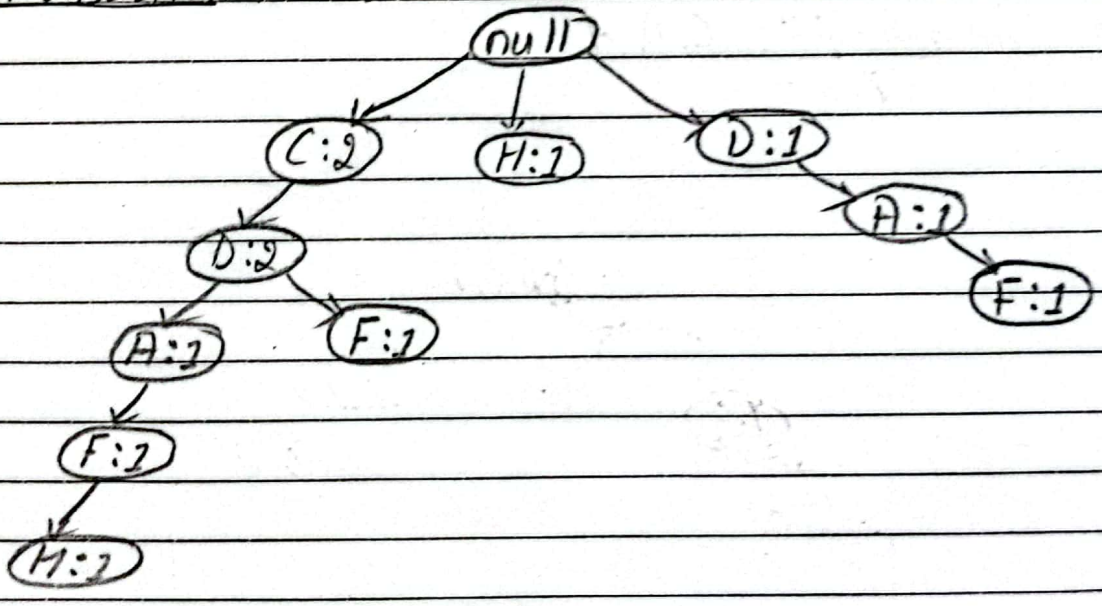
2: For Transaction 2: D, A, F



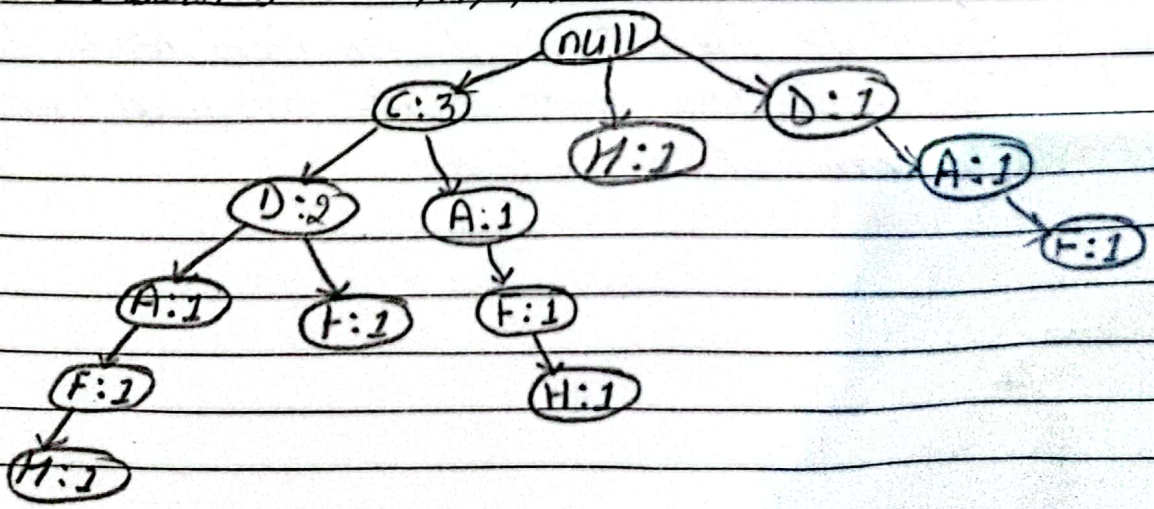
3. For Transaction 3: C, D, F



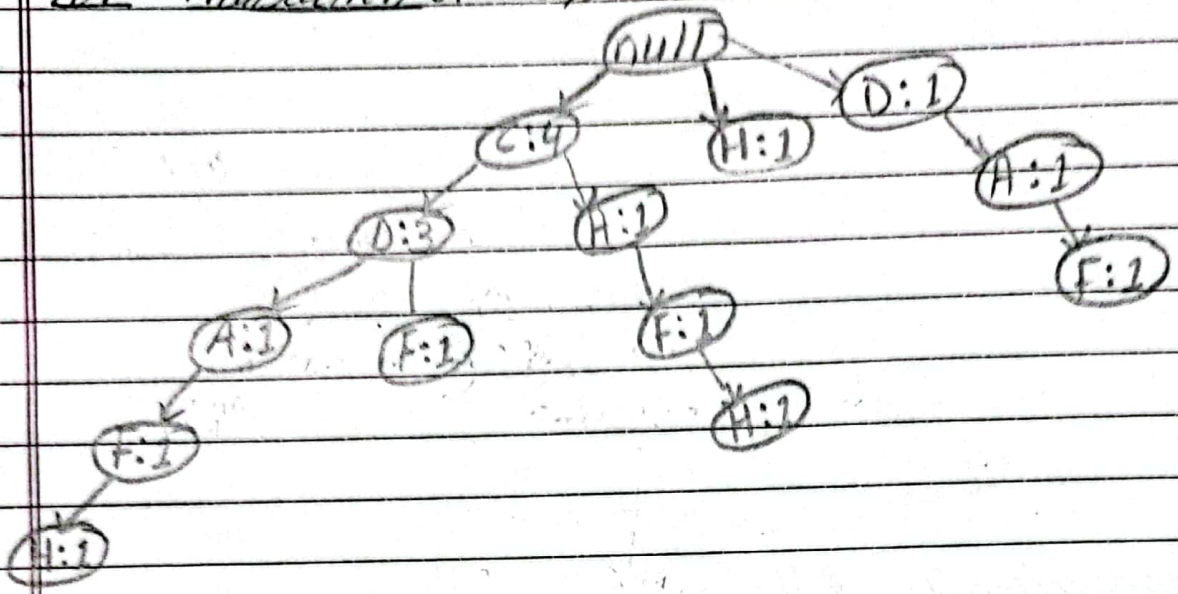
4. For Transaction 4: B H



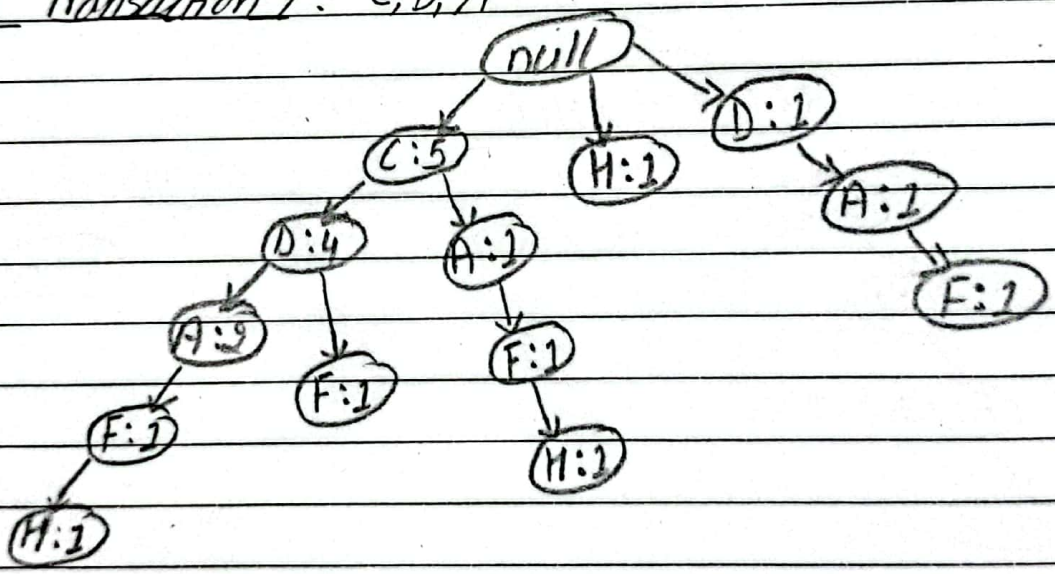
5. For Transaction 5: C, A, F, H



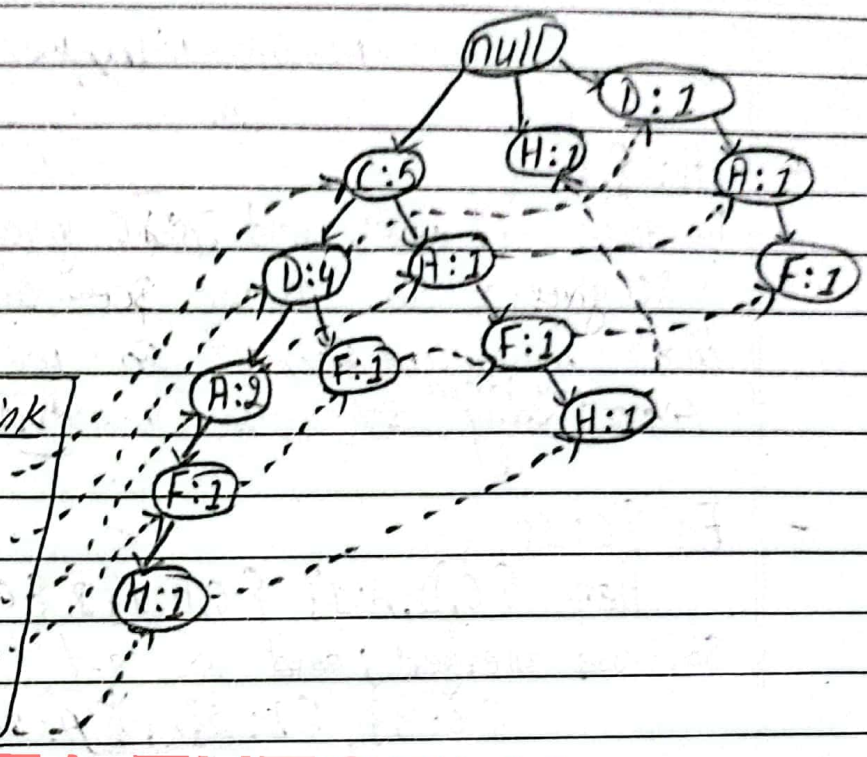
6. For Transaction 6: C, D



7. For Transaction 7: C, D, A



Now, to facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of nodes-link.



Item ID	Support count	Node link
C	5	
D	5	
A	4	
F	4	
H	3	

Now, we need to find

(i) Conditional Patter Base
(ii) Conditional FP-Tree
(iii) Frequent Pattern Generated

(i) Conditional Patten Base

For this start with last item in sorted order.

Itemset	Conditional Pattern Base
H	{C,D,A,F:1} , {C,A,F:1}
F	{C,D,A:1} , {C,D:1} , {C,A:1} , {D,A:1}
A	{C,D:2} , {C:1} , {D:1}
D	{C:4}

GOOD MORNING
PAGE NO. :
DATE : __/__/____

(11) Conditional FP-Tree

minimum support : 3

- H: { ∅

because if we add C&C gives 2, A&A gives 2, F&F gives 2 D&D gives 1 so, as we see all the support values is smaller than minimum support so we reset them and we can't find any item set so, we write ~~empty~~ ∅.

- F: {C:3}

Here {C,D,A:1}, {C,D:1} & {C,A:1} lies on same branch So, we merged ; ~~and~~ we get

{C:3}, {D:2}, {A:2}

but these two ~~doesn't~~ is smaller than minimum support=3 so, we reset and ~~{D,A:~~ {D,A:1} lies on other branch of tree we write it separately in set for that minimum support=3 should be greater or equal but {D,A:1} is less so we reject.

- A: {C:3}

Here {C,D:2} & {C:1} lies on same branch so, we merged and we get.

{C:3}, {D:2} but {D:2} is smaller than minimum support=3 So we reset them.

{D:1} lies on other branch of tree and support value is smaller than minimum support=3 so, we reset.

- D: {C:4}

greater than minimum support = 3

(iii) Frequent Pattern Generated

Now, generate the frequent pattern from the conditional FP-Tree.

Itemset	Frequent Pattern Generated
H	{ }
F	{C, F : 3}
A	{C, A : 3}
D	{C, D : 4}

In this way the frequent patterns are generated by FP-growth algorithm for the given transaction database.

GOOD MORNING
PAGE NO.:
DATE: __/__/____

2079 Questions

1. Find frequent itemsets and association rules from the transaction database given below using FP-growth algorithm. Assume minimum support is 50% and minimum confidence is 60%.

Transaction ID	Items Purchased
1	Sausage, Peanut, Beer
2	Peanut, Beer, Apple
3	Apple, Milk
4	Sausage, Peanut, Apple
5	Sausage, Peanut, Beer, Milk
6	Sausage, Peanut, Beer, Apple

Sol?

Given,

minimum support = 50% = 0.5 × 6 = 3

Constructing 1-itemsets and counting support count for each itemset

Itemsets	Support count
Sausage	4
Peanut	5
Beer	4
Apple	4
Milk	2

Compare with support count value of each itemset with minimum support = 3 if it is less then eliminating such itemsets.

Itemsets	Support count
Sausage	4
Peanut	5
Beer	4
Apple	4

Sorting frequent 1-itemsets in decreasing order of their support count (give high priority to the high value)

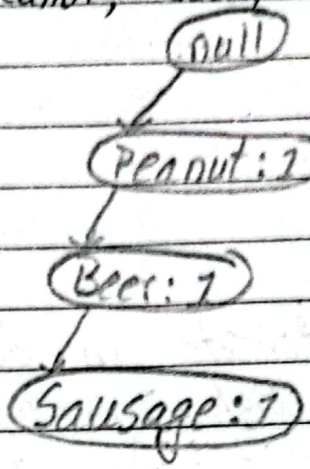
Itemsets	Support count
Peanut	5
Apple	4
Beer	4
Sausage	4

Now, ordering each itemsets in based on frequent 1-itemsets

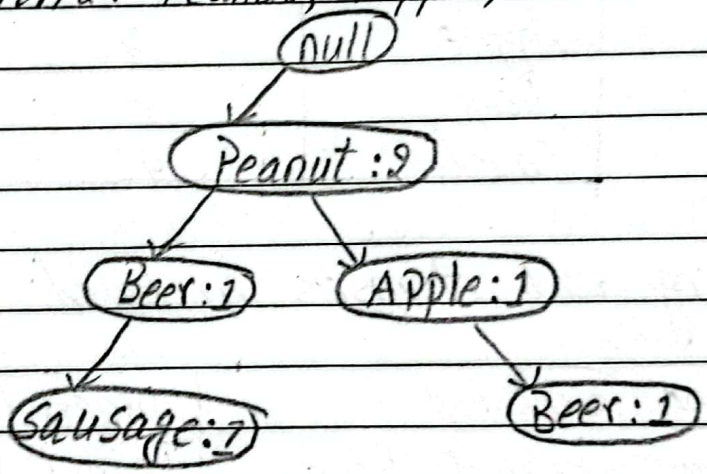
T_ID	Items Purchased	Ordered items
1	Sausage, Peanut, Beer	Peanut, Beer, Sausage
2	Peanut, Beer, Apple	Peanut, ~~Beer~~ Apple, Beer
3	Apple, Milk	Apple
4	Sausage, Peanut, Apple	Peanut, Apple, Sausage
5	Sausage, Peanut, Beer, Milk	Peanut, Beer, Sausage
6	Sausage, Peanut, Beer, Apple	Peanut, ~~Be~~Apple, Beer, Sausage

Now, drawing FP-Tree by using ordered items one by one.

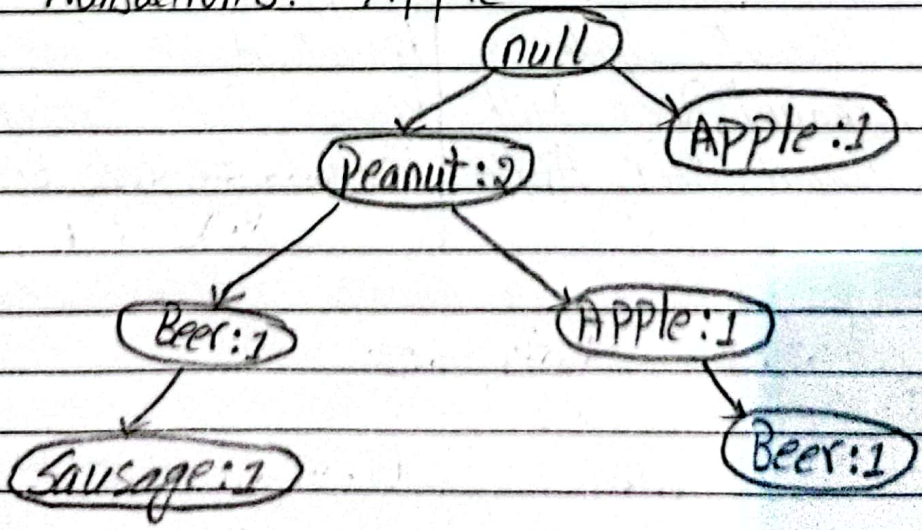
1. For Transaction 1: Peanut, Beer, Sausage



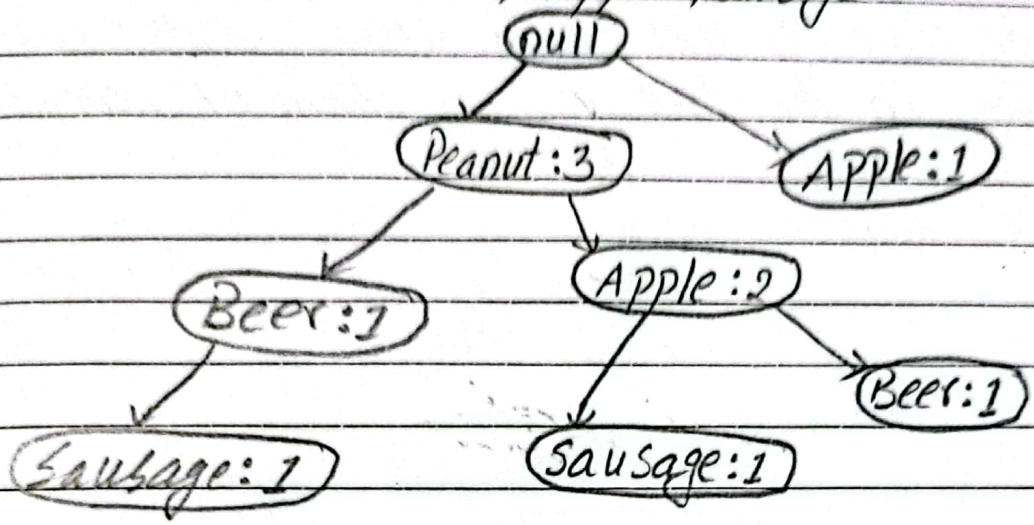
2. For Transaction 2: Peanut, Apple, Beer



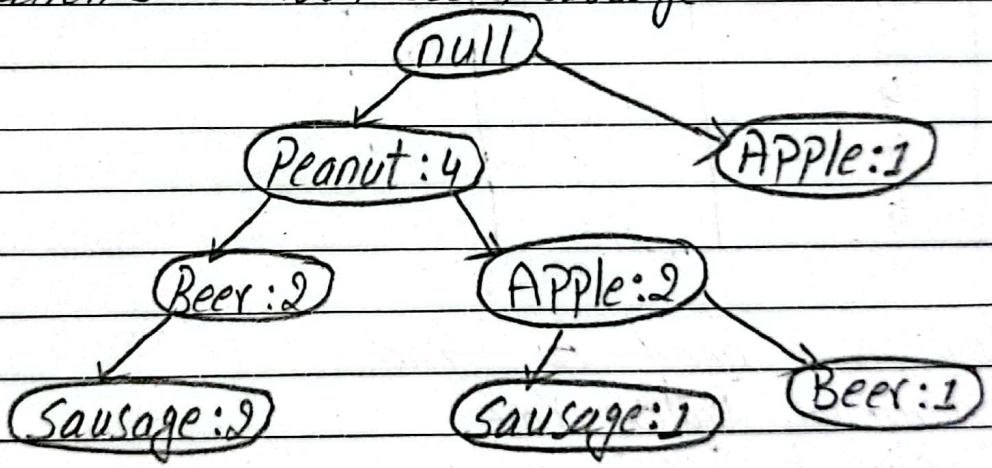
3. For Transaction 3: Apple



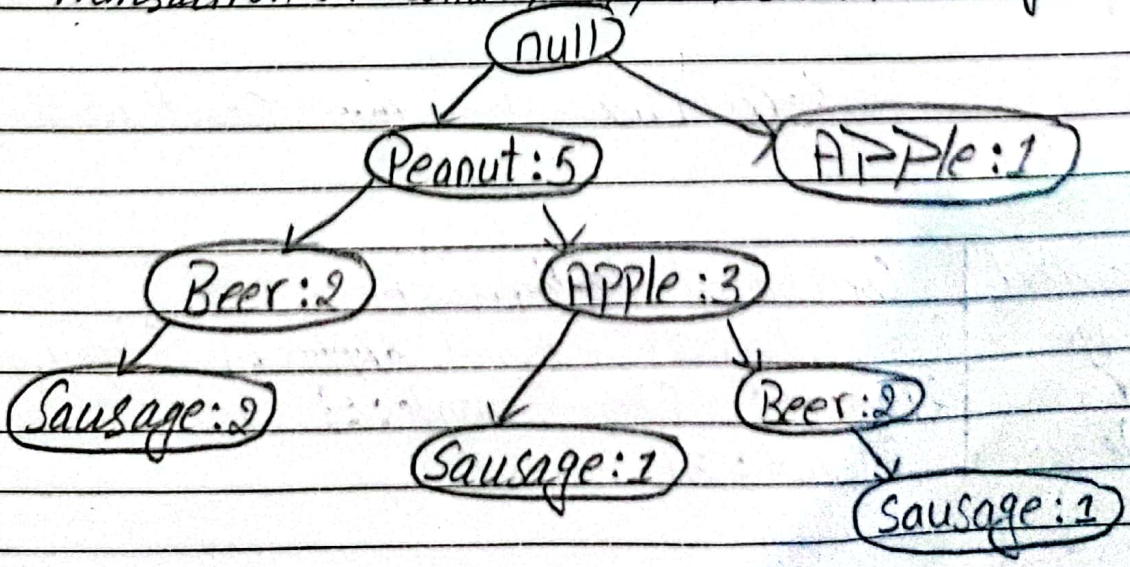
4. For Transaction 4: Peanut, Apple, Sausage



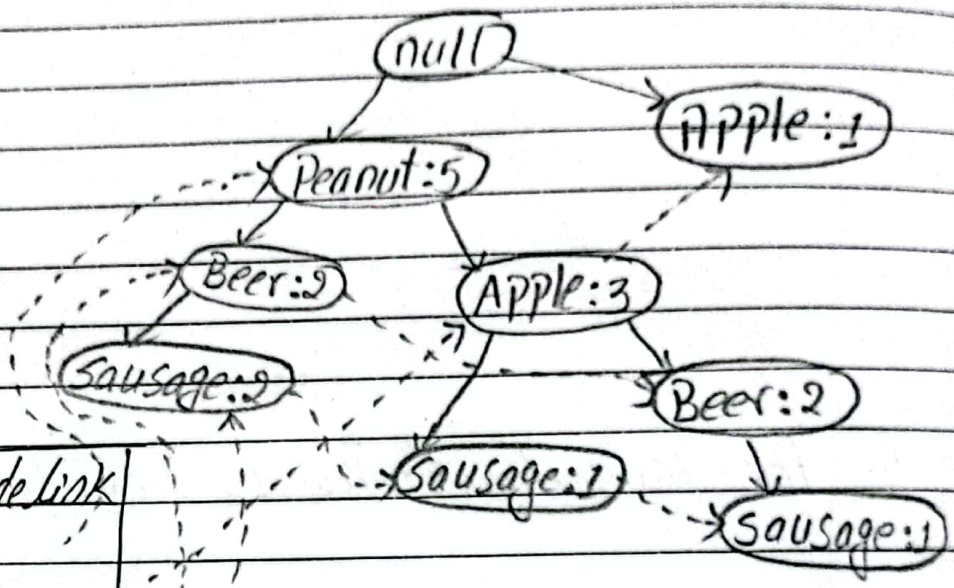
5. For Transaction 5: Peanut, Beer, Sausage



6. For Transaction 6: Peanut, Apple, Beer, Sausage



Now to facilitate tree traversal, an item header table is built so that each item points to its occurrence in the tree via a chain of node-links.



Itemset	Supportcount	Node link
Peanut	5	
Apple	4	
Beer	4	
Sausage	4	

Now, we need to find

(i) Conditional Pattern Base
(ii) Conditional FP-Tree
(iii) Frequent Pattern Generated.

(i) Conditional Pattern Base:- For this start with last item in sorted list

Itemset	Conditional Pattern Base
Sausage	{Peanut, Beer:2}, {Peanut, Apple:1}, {Peanut, Apple, Beer:1}
Beer	{Peanut:2}, {Peanut, Apple:2}
Apple	{Peanut:3}

(II) Conditional FP-Tree

minimum support = 3

- Sausage {Peanut: 4, Beer: 3}
  and we don't write Apple: 2 because the minimum support = 3 &,

- Beer {Peanut: 4}
  same reasons.

- Apple {Peanut: 3}

Hence,

Itemsets	Conditional FP-Tree
Sausage	{Peanut: 4, Beer: 3}
Beer	{Peanut: 4}
Apple	{Peanut: 3}

(III) Frequent Pattern Generated: Now, Generated the frequent pattern from the conditional FP-Tree

Itemsets	Frequent Pattern Generated
Sausage	{Peanut, Sausage: 4}, {Beer, Sausage: 3}, {Peanut, Beer, Sausage: 3}
Beer	{Peanut, Beer: 4}
Apple	{Peanut, Apple: 3}

For finding association rules;

Itemset	Support count
{Sausage, Peanut}	4
{Sausage, Beer}	3
{Sausage, Apple}	2
{Peanut, Beer}	4
{Peanut, Apple}	3
{Beer, Apple}	2

Compare with support count value of each itemset with minimum support = 3 if it is less then eliminating such itemsets.

Itemset	Support Count
{Sausage, Peanut}	4
{Sausage, Beer}	3
{Peanut, Beer}	4
{Peanut, Apple}	3

Now, generating association rules from the frequent pattern Generated

Let's take {Peanut, Beer, Sausage : 3}

(1) {Peanut ^ Beer} → {Sausage}

$$\text{Confidence} = \frac{\text{Support}\{\text{Peanut, Beer, Sausage}\}}{\text{Support}\{\text{Peanut, Beer}\}}$$

$$= \frac{3}{4} \times 100\%$$

$$= 75\%$$

(II) {Peanut, ∧ Sausage} → {Beer}

confidence = $\frac{3}{4} \times 100\% = 75\%$

(III) {Beer ∧ Sausage} → {Peanut}

confidence = $\frac{3}{3} \times 100\% = 100\%$

(IV) {Peanut} → {Beer ∧ Sausage}

confidence = $\frac{3}{5} \times 100\% = 60\%$

(V) {Beer} → {Peanut ∧ Sausage}

confidence = $\frac{3}{4} \times 100\% = 75\%$

(VI) {Sausage} → {Peanut ∧ Beer}

confidence = $\frac{3}{4} \times 100\% = 75\%$

Here, the strong association rule are;

{Peanut ∧ Beer} → {Sausage}, {Peanut ∧ Sausage} → {Beer}, {Beer ∧ Sausage} → {Peanut}, {Beer} → {Peanut ∧ Sausage}, {Sausage} → {Peanut ∧ Beer} and {Bpeanut} → {Beer ∧ Sausage}

Ans

GOOD MORNING
PAGE NO. :
DATE : __/__/____

Advantages of FP-Growth

Frequent Pattern Growth Faster than Apriori due to following reasons;

- No candidate generation, no candidate test
- Uses compact data structure called FP-~~Test~~ Tree
- Eliminates repeated database scan
- Basic operation is counting and FP-tree building

Disadvantages of FP-Growth

- FP-Tree is more cumbersome and difficult to build than Apriori.
- It may be expensive.
- When the database is large, the algorithm may not fit in the shared memory.

Difference between Apriori and FP-Growth

	Apriori	FP-Growth
(i)	It is an array-based algorithm	It is a tree-based algorithm
(ii)	It uses Join and Prune technique.	It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy minimum support
(iii)	Apriori uses a Breadth-first Search	FP Growth uses a depth-first Search
(iv)	Apriori utilizes a level-wise approach	FP Growth utilizes a pattern-growth approach

(v)	Candidate generation is very parallelizable	Data are very interdependent; each node needs the root.
(vi)	It requires large memory space due to large number of candidate generation	It requires less memory space due to compact structure and no candidate generation.
(vii)	It scans the database multiple times for generating candidate sets.	It scans the database only twice for constructing frequent pattern tree.

From Association Mining to Correlation Analysis:

- Most association rule mining algorithms employ a support confidence framework.
- Although minimum support and confidence thresholds helps to exclude uninteresting rules, many rules so generated are not still interesting to the users.
- This is especially true when mining at low support thresholds.

- Support-Confidence framework can be supplemented with additional interstingness measures based on statistical significance and correlation analysis.
- Some association rules $\{A \Rightarrow B\}$ that satisfy minimum support and threshold may be uninteresting if 'A' and 'B' are negatively correlated with each other. This type of rules can be excluded by using the measure correlation analysis, Lift is the measure that measures correlation.

Lift:- The lift between the occurrence of A and B can be measured as below:

$$\text{Lift}(A,B) = \frac{P(A \cup B)}{P(A)P(B)} = \frac{\text{confidence}(A \Rightarrow B)}{\text{support}(B)}$$

If the resulting value of above equation is less than 1, then the occurrence of A is negatively correlated with the occurrence of B. 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.