1. Suppose a tabular data set with several attributes (any type) is given to you. List the sequence of preprocessing steps you will employ. Given the name of the appropriate algorithm you know for each step. Briefly explain the algorithm in less than 20 words. Make appropriate assumptions, if any.[5]

A raw data is most of the times not cleaned, unstructured and have a lot of data Discrepancies. Hence, before mining we need to preprocess this dataset. Preprocesing refers to cleaning and Transforming data in such a way that it removes all those discrepancies, and sometimes makes it optimised over space & time by aggregations as well.
It's done in 4 Stages →

Data Cleaning → Data Integration → Data Transformation → Data Reduction

1) Data Cleaning: We ~~clean the null values, outliers~~ remove/substitute the null values/outliers first as they can skew our results.
→ Removing Null Valued Rows ✓
→ Substituting ~~by some~~ that Cell Value by the Central tendency value of that attribute (Mean/Median/Mode). ✓
→ Removing outliers by InterQuartile Method.
→ Removing noise, etc ✓

3.75

2) Data Integration: When we merge two or more datasets, discrepancy like data types, high correlation with an existing attribute, etc. can occur.
→ Correlation and Covariance Analysis
↳ If we are adding an attribute but its correlation with an already existing attribute is really high then this merging won't give any new result. (Corr (X→Y): If X inc, Y inc (+ve corr), If X↑ Y↓ (-ve corr), 1 (Independent))
→ Check the data types while Merging. My ID can be Numerical in one Dataset but Nominal in the other.

3) Data Transformation: Refers to transforming data in such a way that it is more easier to applying mining over it in future. We can aggregate multiple attributes, etc.
→ ~~Converting~~ AOI (Attribute Oriented Induction)
↳ Data is transformmed into categorical values (Generalization).
→ ~~which helps in better results~~ Normalisation (Min-max method): Range = Max - Min)

4) Data Reduction: This refers to removing/reducing unwanted attributes from our dataset.
↳ Dimensionality Reduction
↳ If we have to find corr. btw age and wages, we will remove the attribute - food preference as it won't give any reasonable result, if it has a lot of NULL values then it will just skew our result. So, better to remove it from this Analysis.

2. (i) Explain the concepts of (a) Data (b) Information (c) Knowledge. [3]
(ii) Explain the concepts of data warehouse and data lake. [2]

(i)

(a) Data refers to raw facts. 0.5

(b) Information refers to the patterns we extract after analysing data using various methods.

(c) Knowledge is what ~~[illegible]~~ patterns

(b) Information refers to the processed facts which still don't have a meaning by themselves. 0.25

(c) Knowledge refers to the patterns we can extract from this processed data i.e. Information. 0.25

Rain → Data

↓

It's Raining → Information

↓

It's Raining bcz it's Monsoon → Knowledge.

(ii) Data warehouse: Data warehouse is an alternate location of storing your whole database other than the operational Database on which your company is working currently.

↳ It's non-volatile (Doesn't gets erased bcz after uploading you just access, never delete it), time-invariant (has a lot of historical data as well), & subject-oriented (Data marts of diff. subjects like Sales, Customers, etc. exist)

→ Structure

0.75

Data Lake: Data lake is similar to Warehouse, only diff. being in warehouse you put structured & Re-processed Data. But Data lake contains both structured as well as un-structured datasets.

Eg. Used by Scientists, etc. for extracting new knowledge.

→ 0.25.

2

3. Consider the following relational table as an input to compute data cube. Give the contents of (i) any two base cells (ii) Any two aggregate cells (iii) Any two ancestor-decedent cells [5]

SALES			
Model	Year	Color	Sales
Chevy	1990	red	5
Chevy	1990	white	87
Chevy	1990	blue	62
Chevy	1991	red	54
Chevy	1991	white	95
Chevy	1991	blue	49
Chevy	1992	red	31
Chevy	1992	white	54
Chevy	1992	blue	71
Ford	1990	red	64
Ford	1990	white	62
Ford	1990	blue	63
Ford	1991	red	52
Ford	1991	white	9
Ford	1991	blue	55
Ford	1992	red	27
Ford	1992	white	62
Ford	1992	blue	39

(i) Base Cells → The Cells in Base Cuboid.

Base Cuboid is the Cuboid formed at n-d level in the hierarchy.

Here, n = 4, so Base Cuboid has cells of form →

(Model, Year, Color, Sales)

eg → (Chevy, 1990, red, 5)

(Chevy, 1990, white, 87)

(ii) Aggregate Cells → Cells in aggregate cuboid.

Aggregate Cuboid is the Cuboid at any level except for 0 & n, where atleast 1 of the attributes are aggregated.

eg → (Model, Year, Color, *)

↳ (Chevy, 1990, red, ALL)

→ (Model, Year, *, Color)

↳ (Chevy, 1990, ALL, 5)

(iii) Ancestor-Decedent Cells : ~~If this is the Base Cuboid assuming this data cube is formed using BUC algorithm (Top → Down)~~ who will have aggregations first is ancestor.

~~so this level is the highest ancestor~~

~~(Model, Year, Color, Sales)~~ (Model, Year, Color, ALL) → Ancestor

↓ child parent

~~(Model, Year, Color,~~ (Model, Year, Color, Sales) → Descendant

~~as after aggregating one time~~

eg → (Chevy, 1990, red, ALL) → (Chevy, 1990, red, 5)

(Chevy, 1990, ALL) → (Chevy, 1990, white, 87)

white

↳ The ancestor is aggregate of all those possibilities where color is red or white respectively.

$\rightarrow$ min = $\frac{2}{3}$ = 0.66%.

4. Consider that the database has only three transactions: {<$a_1, a_2, \ldots, a_{100}$>, <$a_1, a_2, \ldots, a_{50}$>, <$a_1, a_2, \ldots, a_{25}$>} Suppose min_sup=2. Give the equation for all frequent itemsets. Also, list all the closed and maximal frequent itemsets. [5]

Frequent Itemsets: Itemsets that are frequent acc. to the given threshold. Here, itemsets which cross the min. support value.

Itemset	frequency
(i) <$a_1, a_2 \ldots a_{100}$>	1
(ii) <$a_1, a_2 \ldots a_{50}$>	2
(iii) <$a_1, a_2 \ldots a_{25}$>	3

(subset of both i & ii)
(" " all three i, ii, & iii)

Support$^{(A \to B)}$ = Prob (AUB) = $\frac{\text{No. of Transactions of (AUB)}}{\text{No. of Trans. (Dataset)}}$

Support (i) = $\frac{1}{3}$ = 0.33 X (0.33 < 0.66)

Support (ii) = $\frac{2}{3}$ = 0.66 ✓

Support (iii) = $\frac{3}{3}$ = 1 ✓

Eqn?

Hence, **freq. Itemsets : <$a_1, a_2, \ldots a_{50}$>; <$a_1, a_2 \ldots a_{25}$>**

Closed frequent Itemsets: Frequent itemsets whose superset (if exists) does not have the same support as that of our itemset.

freq. Itemsets: (i) <$a_1, a_2 \ldots a_{50}$> → Support = 2

(ii) <$a_1, a_2 \ldots a_{25}$> → Support = 3

(i) ⊃ (ii) i.e. <$a_1, a_2 \ldots a_{50}$> is a superset of <$a_1, a_2 \ldots a_{25}$>,
but Sup(i) < Sup(ii)
Hence (ii) accepted.

No superset of (i) exists, hence (i) also accepted

Closed freq. Itemsets : <$a_1, a_2 \ldots a_{50}$>, <$a_1, a_2 \ldots a_{25}$>

Maximal freq. itemsets : freq. itemsets whose superset does not exist in the dataset.

(i) <$a_1, a_2 \ldots a_{50}$>

(ii) <$a_1, a_2 \ldots a_{25}$>

(i) ⊃ (ii) hence (ii) not accepted.

No superset of (i) exists, & hence (i) accepted.

Maximal freq. dataset → <$a_1, a_2 \ldots a_{50}$>

5. Consider the following set of frequent 3-itemsets: {1,2,3}, {1,2,4}, {1,2,5}, {1,3,4}, {1,3,5}, {2,3,4}, {2,3,5}, {3,4,5}. Assume that there are only five items in the dataset. [5]
   (i) List all candidate 4-itemsets generated by candidate generation process in apriori.
   (ii) List all candidate 4-itemsets which survive pruning step of apriori.

(i) Joining over the ~~initial~~ two places in each set (as only 5 items can be in Dataset)

Join ({1,2,3}, {1,2,4}) → {1,2,3,4} ⇒ {1,2,3} ✓, {1,2,4} ✓, {2,3,4} ✓
Join ({1,2,3}, {1,2,5}) → {1,2,3,5} → {1,2,3} ✓, {1,2,5} ✓, {2,3,5} ✓
Join ({1,2,4}, {1,2,5}) → {1,2,4,5} ⇒ {1,2,4} ✓, {1,2,5} ✓, {2,4,5} ✗ → X
Join ({1,3,4}, {1,3,5}) → {1,3,4,5} → {1,3,4} ✓, {1,4,5} ✗, {3,4,5} ✓ → X
Join ({2,3,4}, {2,3,5}) → {2,3,4,5} ⇒ {2,3,4} ✓, {2,4,5} ✗, {3,4,5} ✓ → X

~~Join ({1,2,4}, {3,4,5}) → {1,2,3,4,5}~~

In Apriori, if the subsets are not frequent, then we prune that branch, bcz even the aggregate won't be frequent.

(i) All candidate 4-itemsets generated by candidate generation
= {1,2,3,4}, {1,2,3,5}, {1,2,4,5}, {1,3,4,5}, {2,3,4,5}

(ii) Now the subsets of these itemsets which are not frequent (i.e. weren't present in our 3-itemset collection will be pruned).
as Apriori works on principle of Monotonocity i.e. the support value of X can't be greater than Y if X is a superset of Y.

Hence, checking, we found that 4-itemset candidates who survived pruning are →

{1,2,3,4}, {1,2,3,5}

(4) can elaborate a little more → show subset checking

6. For the following table, apply FP-growth approach and generate FP-tree by considering minimum support as 2. Also, compute conditional pattern base and conditional FP-tree associated with the conditional node e. [5]

FP-tree: Frequent Pattern tree.

Min_Sup = 2

Items	freq.	Priority
a	8	1
b	7	2
c	6	3
d	5	4
e	3	5

Transactional Data Set

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}

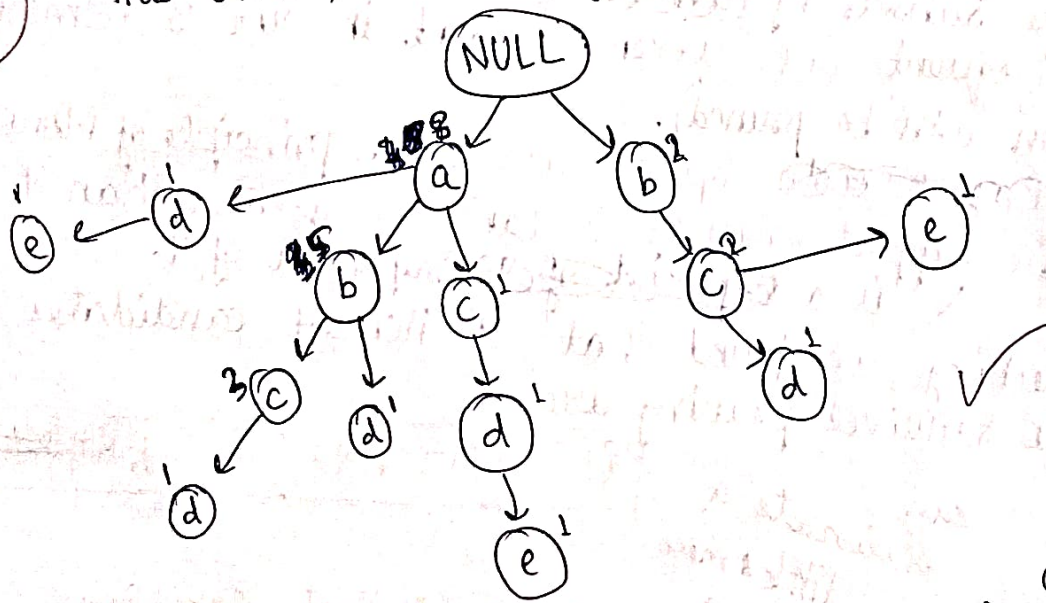
Min Sup=2, followed by each item.

↳ Priority is given acc to frequency. Higher the frequence, more priority will be Given.

Priority Order: (a > b > c > d > e)

- The Items in Transactional data set are already arranged in this order, so no changes there.

(4.5)



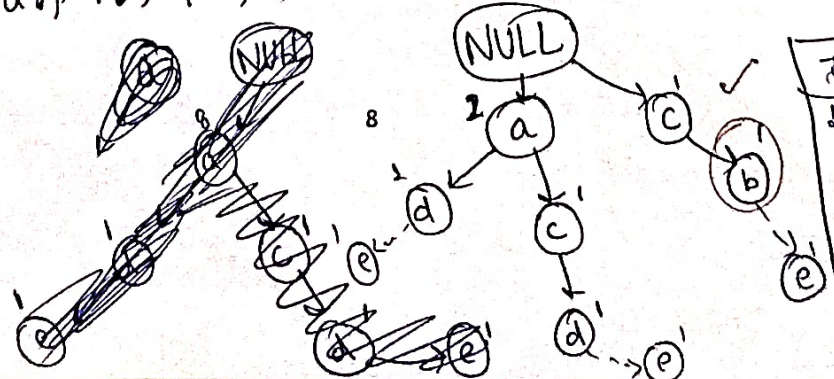
Fore, Conditional Pattern Base:

{a,d}:1, {a,c,d}:1, {b,c}:1

	freq	Priority
a	2	1
b	1	4
c	2	2
d	2	3

a > c > d > b missing ??

FP-tree →



	Items
1	{a,d}
2	{a,c,d}
3	{c,b}

7. Define the problem of extracting high utility itemsets. The so called downward closure property (or anti-monotone property) is not satisfied in *utility mining* model. Explain why? You can consider an example. [5]

Anti-Monotonic Property: Given two itemsets X & Y, if X is a superset of Y, then X can't have support greater than support of Y.

High utility itemsets are required to have generalised knowledge of dataset.

8. (i)Explain why BUC can not exploit optimizations like caching results and amortize scans. (ii)Discuss how the ordering of dimensions with respect to cardinality of attribute and skew of the attribute will impact the performance of BUC approach. [5]

BUC: (Bottom up Algorithm)
↳ Which in actual is Top-Down.
nice explanation!!
It works on the principles of Apriori and starts from the most Generalised dataset to come down in the hierarchy. In the way if there's any itemset which does not satisfies the min. support, then that branch will be pruned (Monotonicity).

(i) Caching Results: It is an optimisation method which is used to hold some aggregate value (eg. Partial Sum) in memory so that we don't have to compute it again & Again.
It's not useful in BUC bcz we are not calculating anything here, just pruning down the itemsets at each level which doesn't exceeds min_sup. In algorithms like Sort, this Cache helps from ABC to AB in pipeline, but here we deleted our branch only after pruning so no going from ABC to AB in many cases.

(ii) Amortize Scans: Scan just Once and then Compute. Eg. You scan and sort ABCD and in real time you are passing each tuple to sort ABC, ABD, AB, etc. through pipelining.
But BUC is completely recursive & multiple Scans are hence required.

As already discussed, BUC works on Apriori. So Higher the Cardinality (no. of distinct elements in an attribute), less will be their support Value. And their will be more chances of them not passing the min. support criteria.
(more no. of partitions, small).
Hence, Higher the Cardinality of attribute, earlier the pruning, is better.

Skew: Attributes shift data towards one side. So there partition size increases & no. decreases. And hence, even though they are skew they might pass the min. support criterion. Hence, no point in pruning them earlier as might not a lot of unwanted data be removed.

9. What issue with <support, confidence> framework in extracting interesting association rules? How <support, confidence, lift> framework filters uninteresting association rules from the association rules [5]

Support $(X \to Y)$: In How many transactions X & Y are bought together out of all the transactions that happened.

$$Supp(X \to Y) = \frac{P(A \cup B)}{P(Dataset)}$$

Confidence $(X \to Y)$: How many transactions if X are bought Y is also bought.

$$Conf(X \to Y) = P(Y|X) = \frac{Sup(X \to Y)}{Sup(X)} = \frac{P(X \cup Y)}{P(X)}$$

Lift $(X \to Y)$: How are the Given two attributes correlated.

$$Lift(X \to Y) = \frac{P(A \cup B)}{P(A)P(B)}$$

$$= \begin{cases} < 1 \to -\text{vely correlated} \\ > 1 \to +\text{vely correlated} \\ = 1 \to X \text{ \& } Y \text{ are Independent} \end{cases}$$

Let's take an example to understand this better: (Total Transactions = 10,000)

Let the no. of Games be 6000
" " Videos be 7500
and no. of both be 4000

[No. means 'no. of transactions' Containing Games, Videos or both here]

Then Support $(G \to V)$: i.e. Games & Videos are bought together

$$= \frac{P(G \cup V)}{P(D)} = \frac{4000}{10000} = 0.4$$

Confidence $(G \to V)$: If G is Bought then V are Bought

$$= \frac{P(G \cup V)}{P(G)} = \frac{4000}{6000} = \frac{2}{3} = 0.66$$

This Confidence shows that in 66% of the transactions, G is getting bought and then V is getting bought.

It might also happen that G is not a cause for V but it's just a coincidence, or maybe they are -vely correlated i.e. if G is increasing that's why it's decreasing V.

Let's apply Lift & check for correlation

$$Lift(X \to Y) = \frac{P(G \cup V)}{P(G)P(V)} = \frac{4000}{6000 \times 7500} = < 1$$

i.e. **Both Games & Videos are Negatively Correlated**

Hence, if by just checking supp. and confidence we would have asked that put Games & Video Aisles together, it might have negatively affected our business.

One of the reasons behind this failure is having sparse dataset or a lot of Null Values.
Measures like support & confidence don't give right answers if data are very sparse.