Odd Sem., AY 2024-25

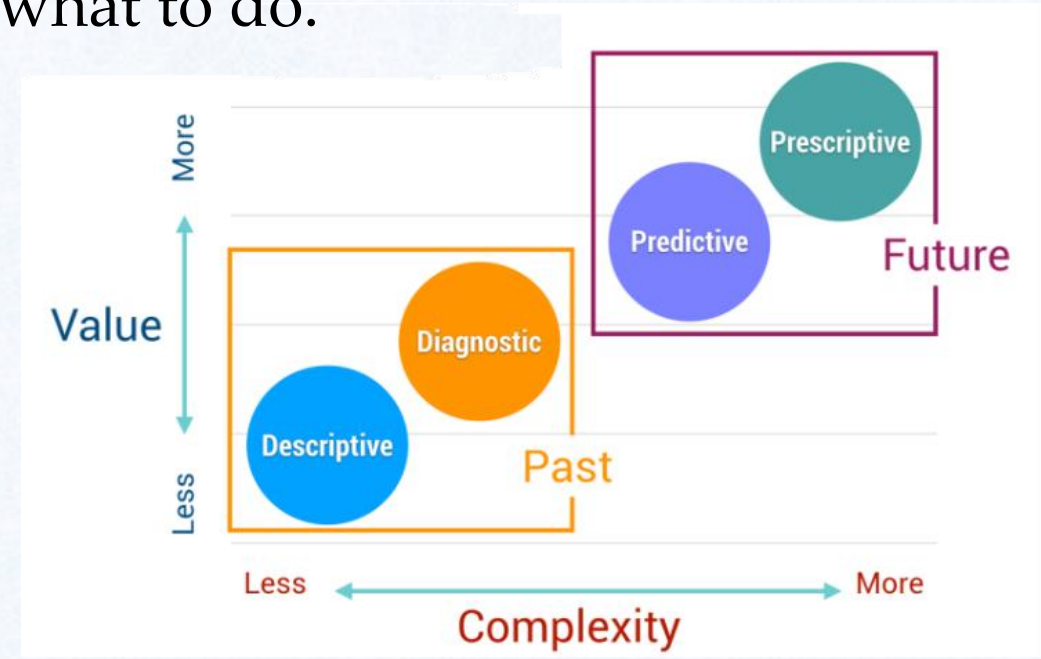
Lecture: 16, 17, & 18

Fond. Of DS (DSPC201)

Course Instructor: Dr Vikram Singh

Type of Analytics

- Descriptive: what's already happened.
- Diagnostic : why did it happen.
- Predictive :forecasts what might happen in the future.
- Prescriptive: recommends what to do.



•Source : https://datarundown.com/4-types-of-analytics/#google_vignette

Descriptive Analytics: what's already happened'

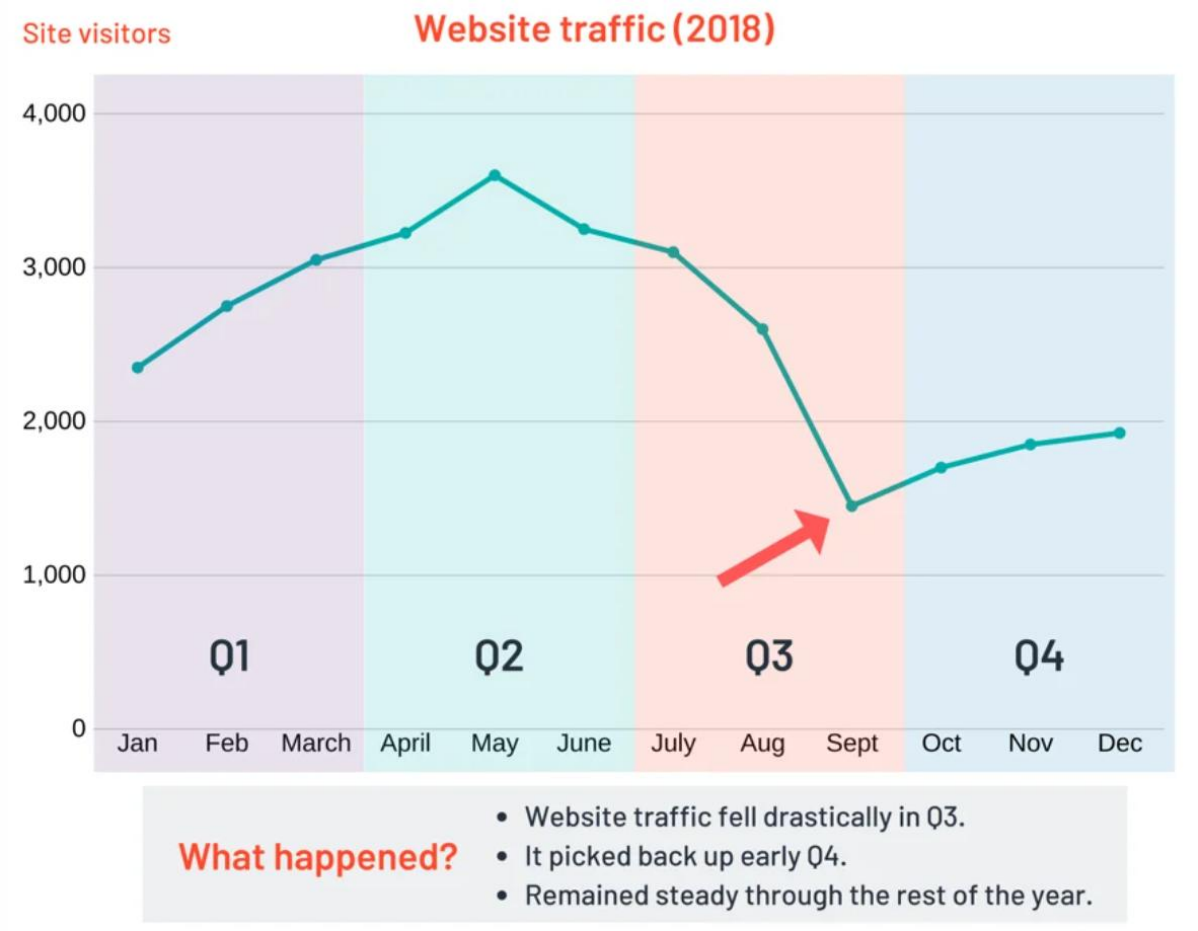
- **Purpose:** *identify anomalies & errors, and reveal relationships between variables/aspects/dimensions/features.*
- **Important to do:** Keep data in *an accessible format (involves organizing/sorting/manipulating)* it to derive meaningful insights.
- Methods :
  - Visualization,…*graphs & plots*
  - Central Tendency… *Mean, Median, Mode,...* & Dispersion
  - Variability… *Range, SD, Variance, Variance*
  - Correlations...*Co-Variance, Correlation Analysis, Chi-sqared, etc., Frequent Pattern Mining, Association Rules Mining.*

More read: https://www.jaroeducation.com/blog/descriptive-analysis-types-and-advantages/

Descriptive Analytics: what's already happened '

Scenario: *Let's say website traffic numbers fell just short of its goal in 2018.*

That's enough reason to run a descriptive analysis to see what went wrong.



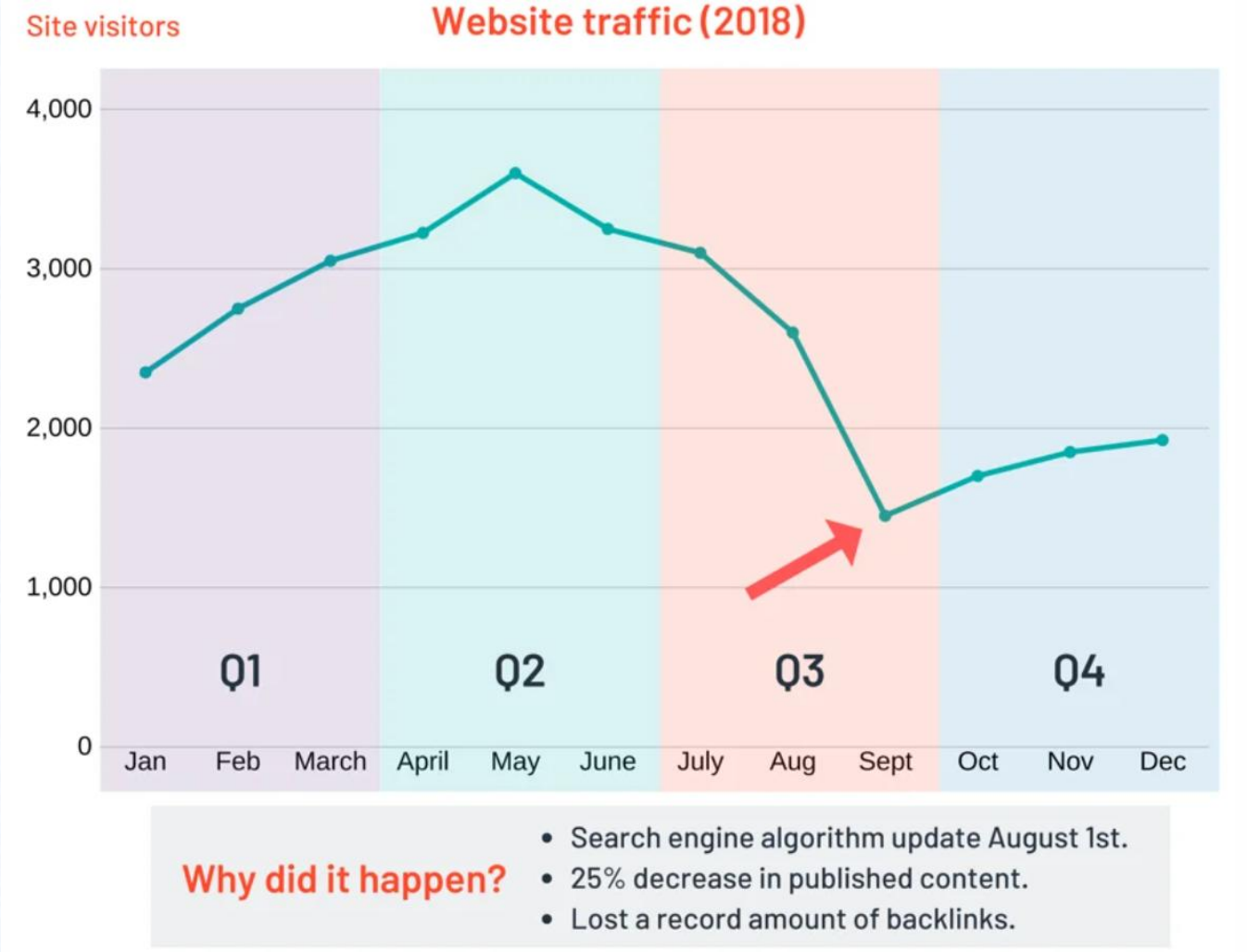
More read: https://www.g2.com/articles/types-of-data-analytics /

Diagnostic Analytics: 'why did it happen'

- **Purpose:** determine the *causes of trends & correlations between variables*
- **Important to do:** *design a 'Hypothesis'* and *approve/disaprove.*
  - Hypotheses are historically-oriented, *e.g., Why are people cancelling their subscriptions?, how areas like company culture and benefits could be improved?*
- Methods :
  - Root-Cause Analysis
  - Co-relations & Hypothesis testing…*Chi-squared Testing, Wilcoxon rank-sum test*
  - Regression Analysis… *Regression Analysis*
  - Fishbone Analysis and Pareto Analysis

Diagnostic Analytics: 'why did it happen'

Scenario:
Exactly why did website traffic plummet so sharply?



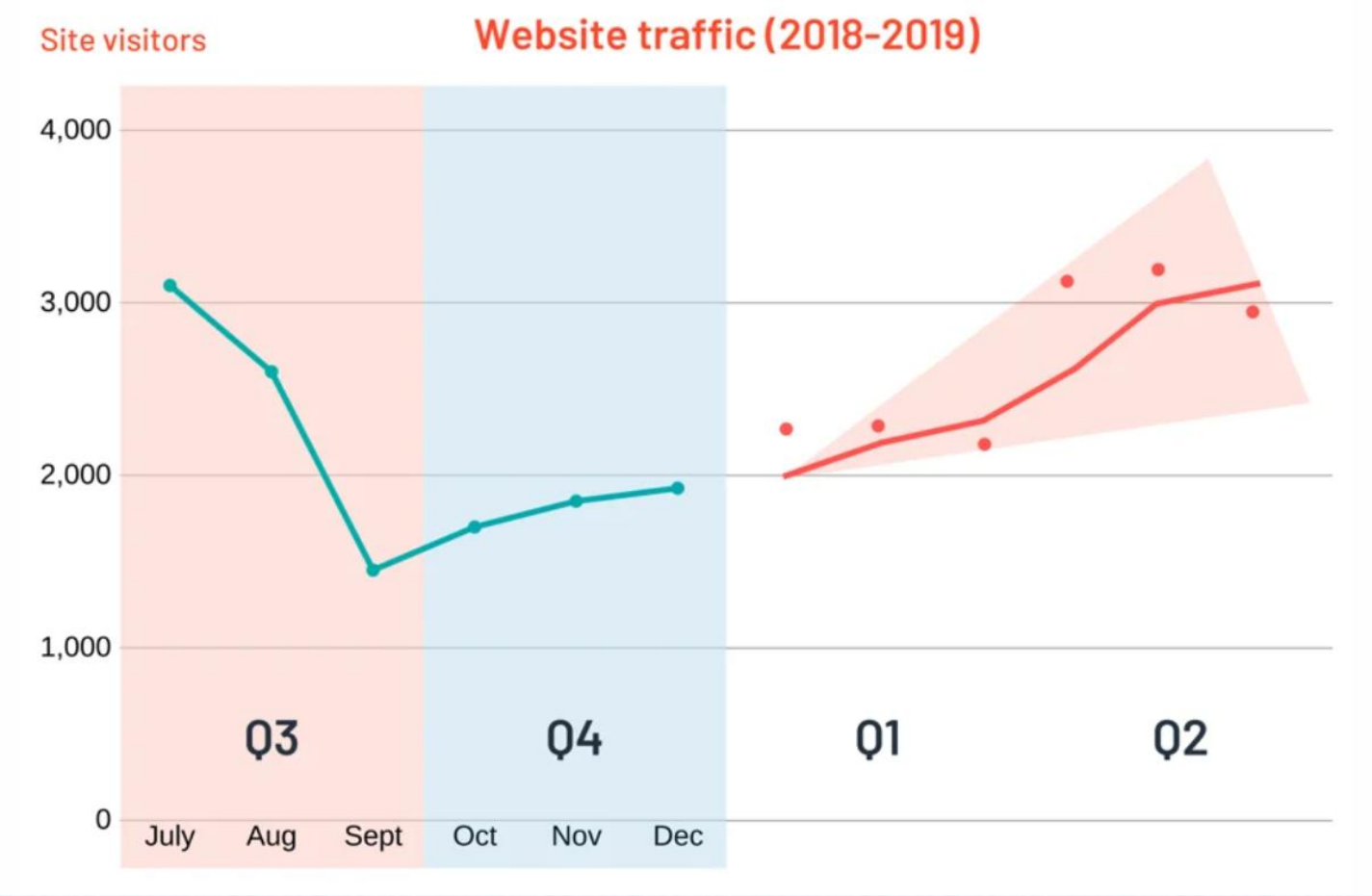
More read: https://www.g2.com/articles/types-of-data-analytics /

Predictive Analytics: *'forecasts what might happen in the future'*

- **Purpose:** determine the *causes of trends & correlations between variables*
- **Important to do:** *design a 'Hypothesis'* and *approve/disaprove.*
  - Hypotheses are future-oriented.
  - Produces Proactive findings .
- Methods :
  - Regression Analysis… *Regression Analysis*
  - Classification …
  - Clustering ….
  - Neural Networks…

Predictive Analytics: *'forecasts what might happen in the future'*

Scenario: *An accurate website traffic number can be generated for the next few quarters.*



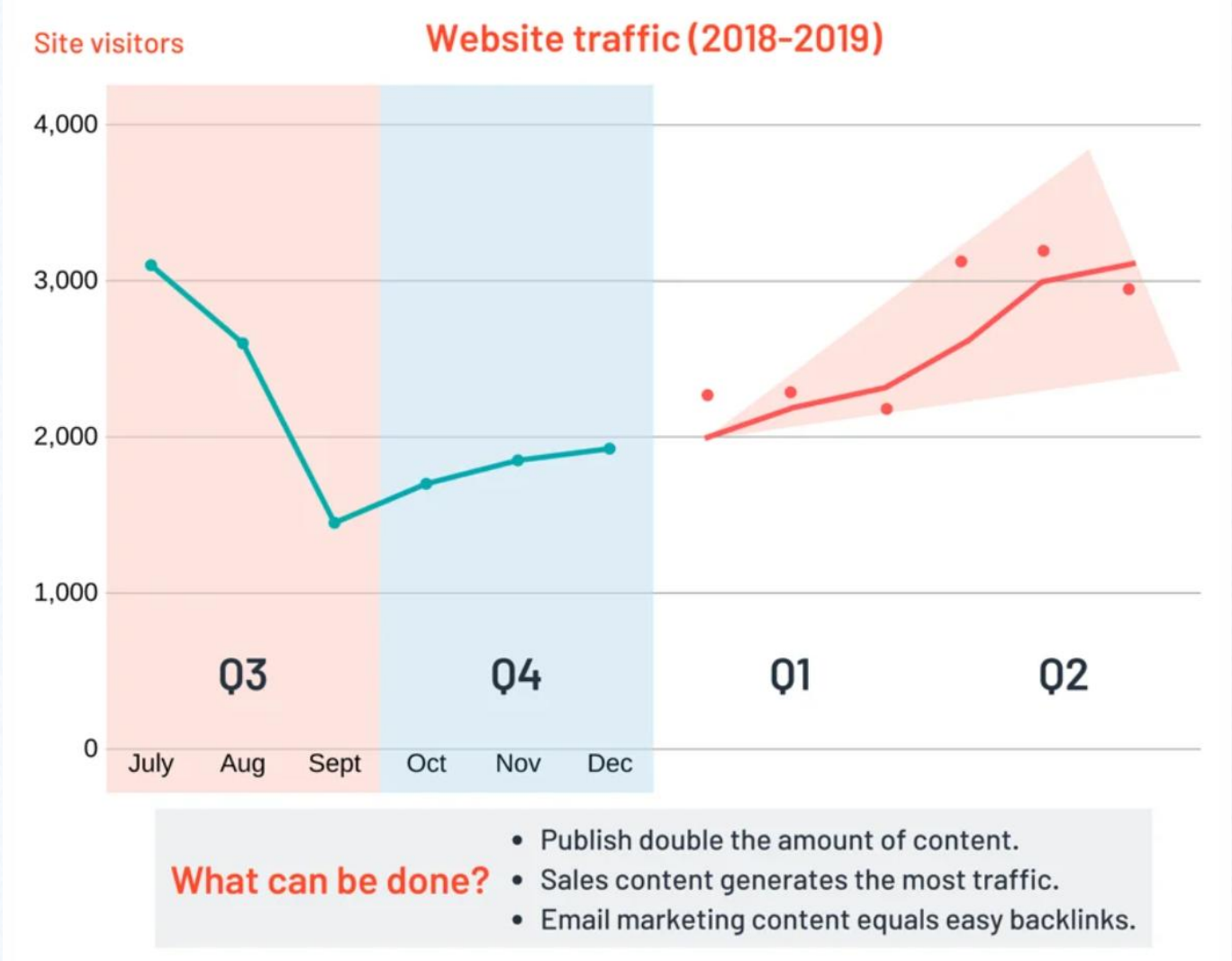
More read: https://www.g2.com/articles/types-of-data-analytics /

Prescriptive analytics: *'recommends what to do'*

- **Purpose: Verify** both *historical data* and *external information.*
- **Important to do:** *design a 'Hypothesis'* and *approve/disaprove.*
  - Hypotheses are future-oriented.
  - Requires Data scientist or scientists with prior knowledge.
- Methods :
  - Optimization…*Linear/Non-Linear Programming.*
  - Simulation
  - Game Theory & Decision Support System

Prescriptive analytics: *'recommends what to do'*

Scenario: With Predictive Analysis reveals where website traffic should be headed, *what are some actionable items to get it there?*



More read: https://www.g2.com/articles/types-of-data-analytics /

Comparison

Descriptive	Diagnostic	Predictive	Prescriptive
Uses historical data	Uses historical data	Uses historical data	Uses historical data
Reconfigures data into easy-to-read formats	Identifies data anomalies	Fills in gaps in available data	Estimates outcomes based on variables
Describes the state of your business operations	Highlights data trends	Creates data models	Offers suggestions about outcomes
Learns from the past	Investigates under-lying issues	Forecasts potential future outcomes	Uses algorithms, AI and machine leanring
Answer "What" Questions	Answers "Why" Questions	Answers "What Might Happen?"	Answers "If, Then" Questions

Descriptive Analysis:

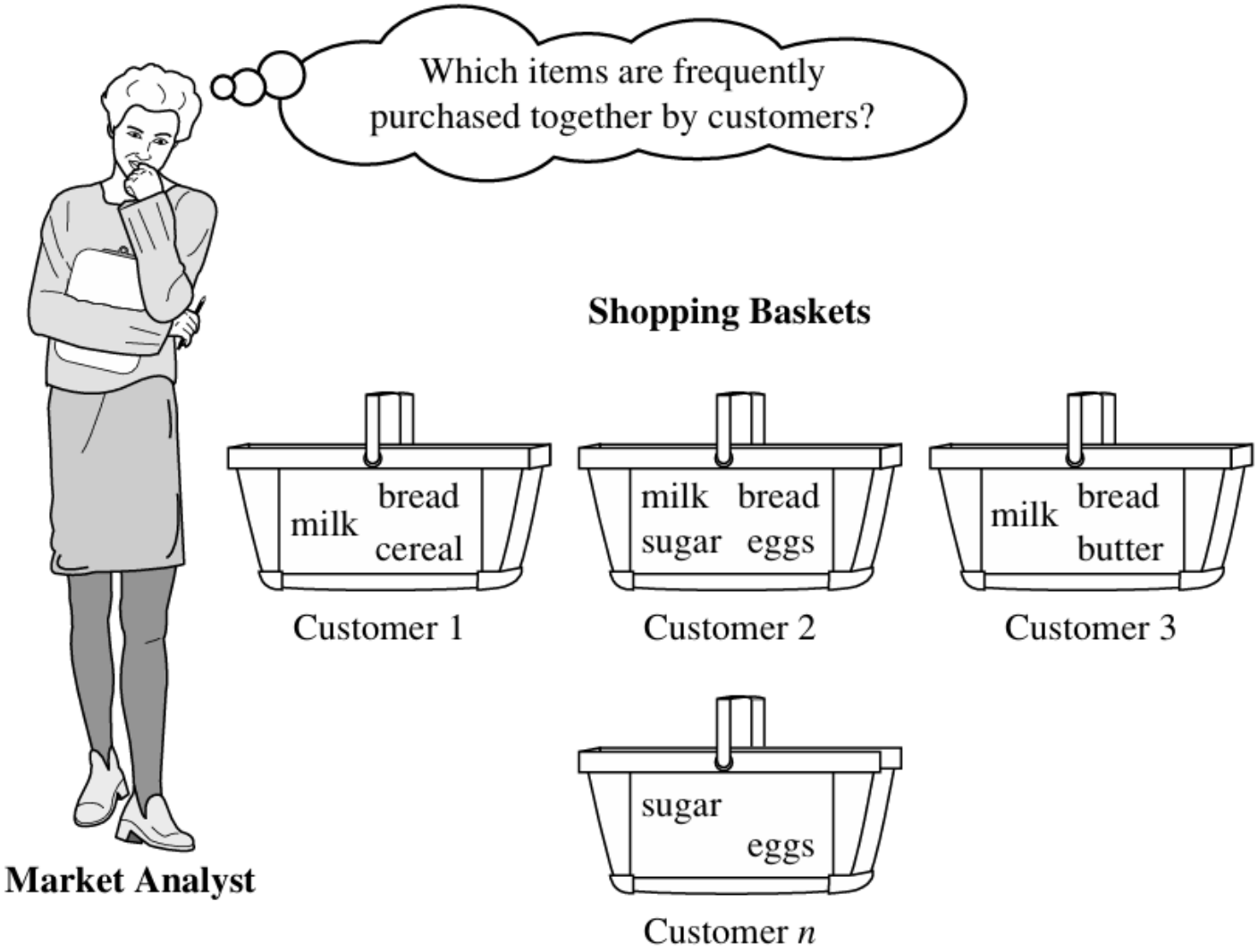
- Easy to use.
- Need Data to be prepared in Simple forms.
- No need of Data Model. *(Why Data Model not required?)*
- Hypothesis is not required .
- Use-Case: *Frequent Pattern Analysis is Important* **??**

Why Is Freq. Pattern Mining Important?

- *Freq. pattern:* An intrinsic and important property of datasets
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: discriminative, frequent pattern analysis
  - Cluster analysis: frequent pattern-based clustering
  - Data warehousing: iceberg cube and cube-gradient
  - Semantic data compression: fascicles
  - Broad applications

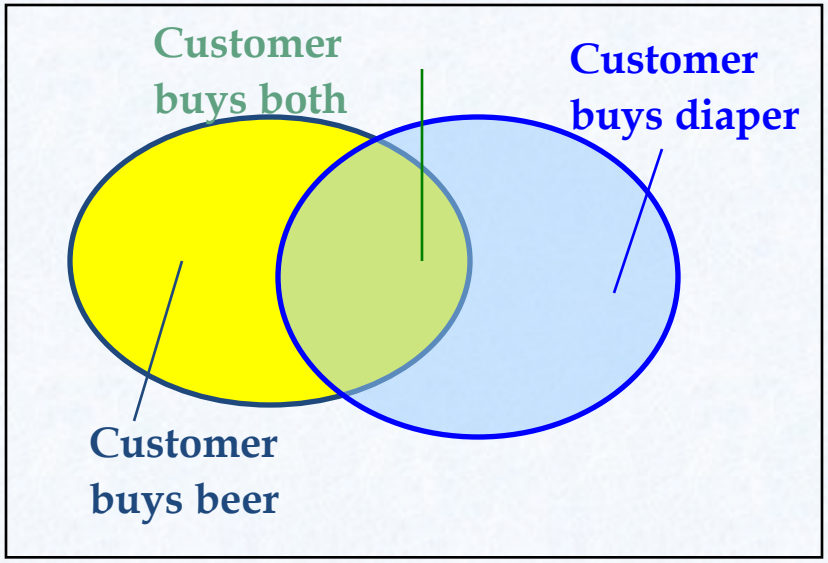
What Is Frequent Pattern Analysis?

- Frequent pattern: a pattern (*a set of items, subsequences, substructures, etc.*) that **occurs frequently in a data set.**
- Motivation: Finding **inherent regularities** in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?
- **Applications**
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.


Which items are frequently purchased together by customers?
Shopping Baskets
milk
bread
cereal
Customer 1
milk
bread
sugar
eggs
Customer 2
milk
bread
butter
Customer 3
sugar
eggs
Customer n
Market Analyst

Frequent Patterns?

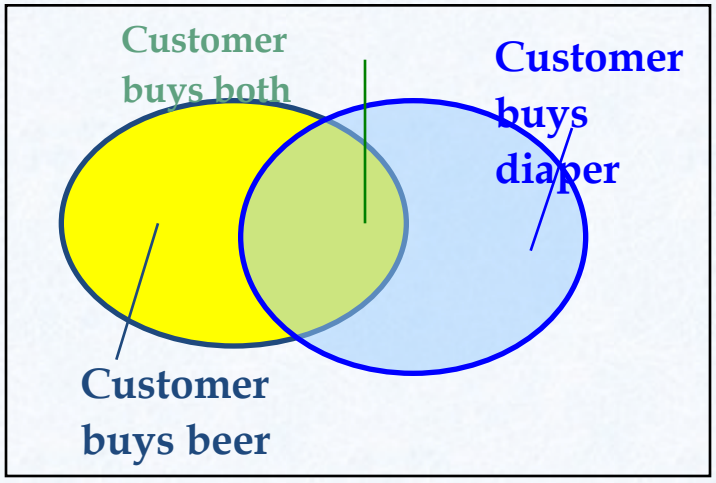
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$
- *(absolute) support*, or, *support count* of X: Frequency or occurrence of an itemset X
- *(relative) support*, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is *frequent* if X's support is no less than a *minsup* threshold

Association Rules?

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

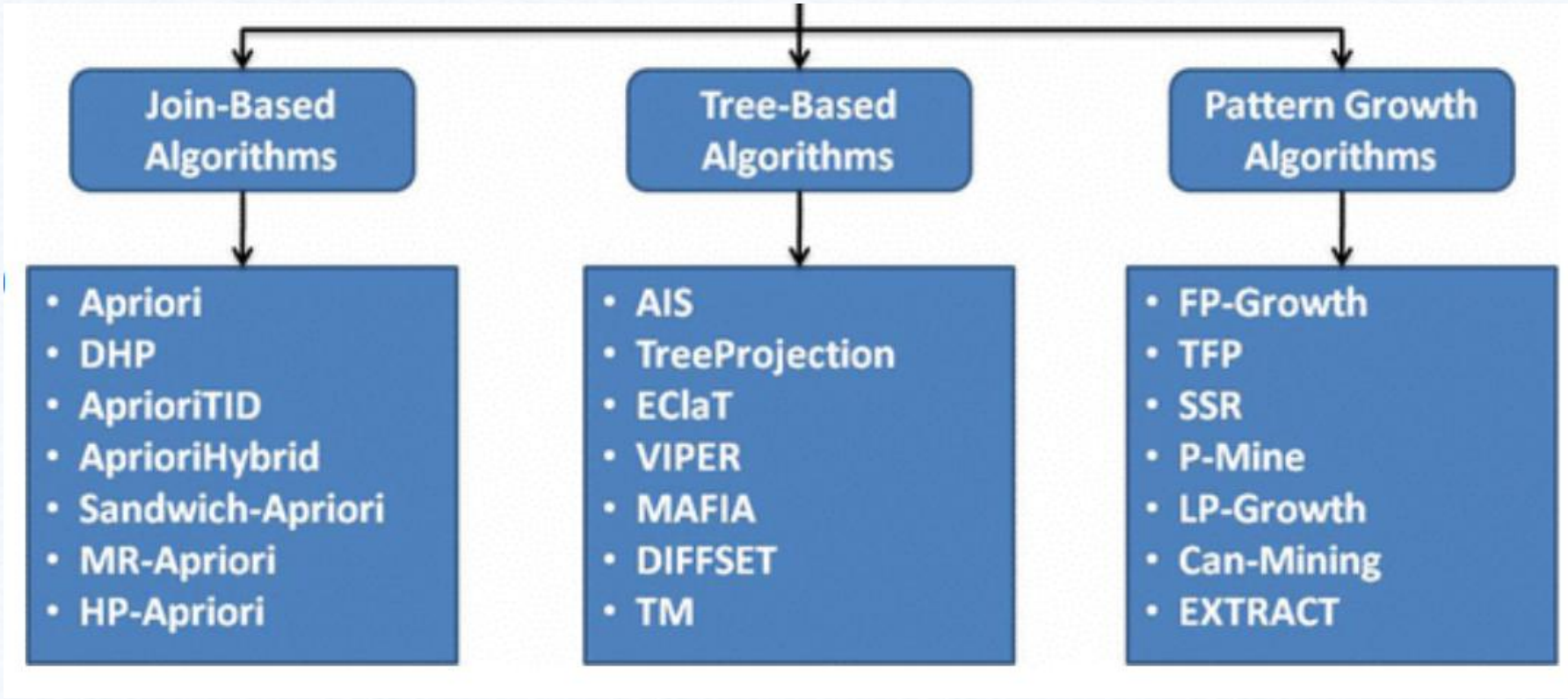
- Association rules: (many more!)
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

Frequent Patterns - Association Rule Support & Confidence

- The patterns can be represented in the form of association rules.
  - **For example,** the information that **customers who purchase computers also tend to buy antivirus software**
  - **Association Rule : -** computer >> antivirus software

- [Support 2%, Confidence 60%].
  - Support 2% means that 2% of all the transactions under analysis show that computer and antivirus SW are purchased together.
  - Confidence 60% means that 60% of the customers who purchased a computer also bought the Antivirus SW.

Approaches: FPM-ARM



Apriori Algorithm

- Based on the principle:

If an itemset is frequent, then all its subsets must also be frequent.

- Find **Frequent Itemsets**
- based on *Confined Candidate Generation*
- Working examples: click here & click here

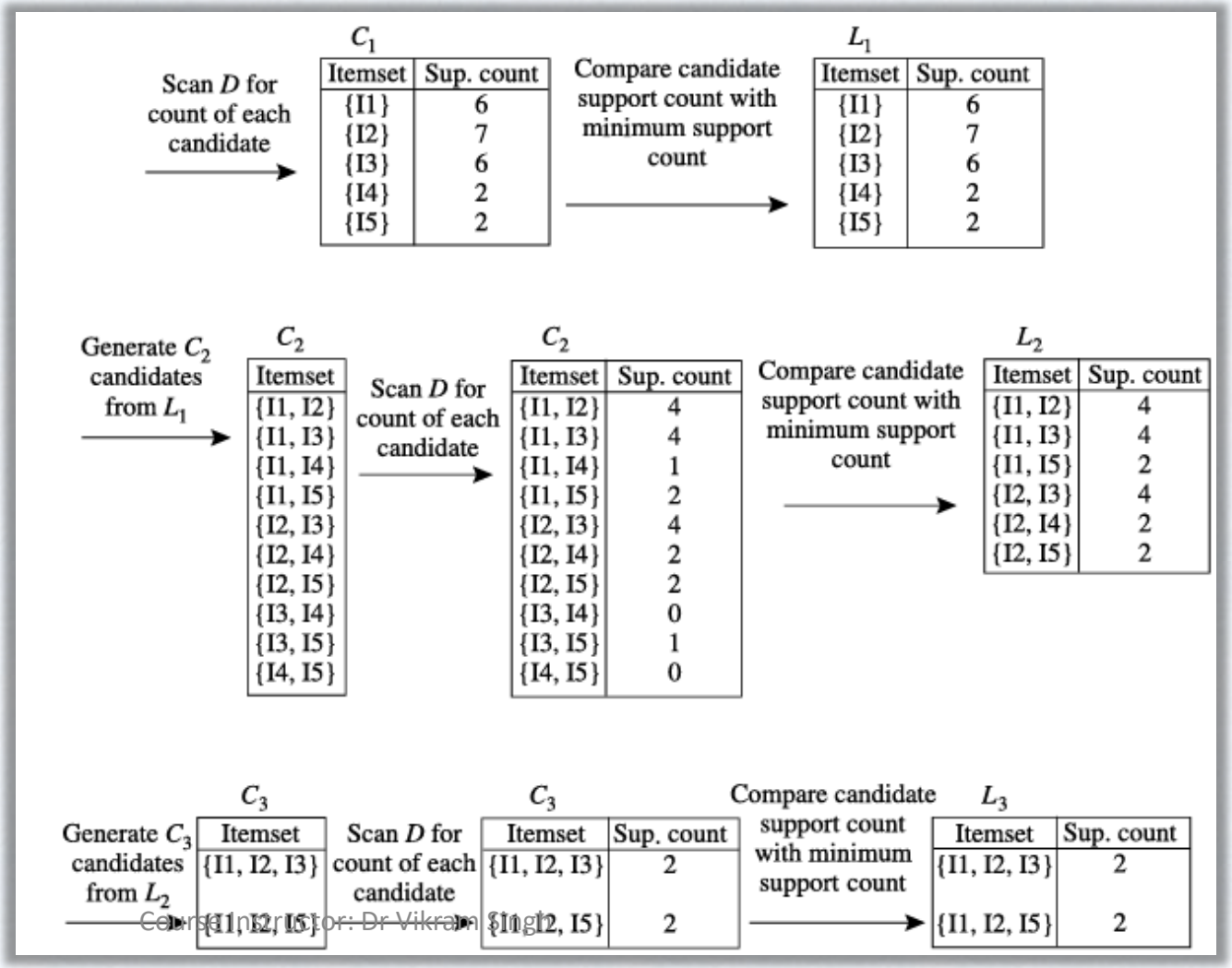
Working Example: "Generating FP using Apriori Algorithm "

Phase -I: Generating Frequent Patterns

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

C1: candidate of 1-itemset
L1: Frequent 1-itemset

For **Minimum Support is 2.**



Working Example: "Generating FP using Apriori Algorithm "

Phase-II : Generating Association Rules

Frequent Itemset:
{I1, I2, I3}
{I1, I2, I5}

Original Transactions DB

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Generated Association Rules & Apply Confidence score

$$
\begin{aligned}
\{I1, I2\} &\Rightarrow I5, \quad confidence = 2/4 = 50\% \\
\{I1, I5\} &\Rightarrow I2, \quad confidence = 2/2 = 100\% \\
\{I2, I5\} &\Rightarrow I1, \quad confidence = 2/2 = 100\% \\
I1 &\Rightarrow \{I2, I5\}, \quad confidence = 2/6 = 33\% \\
I2 &\Rightarrow \{I1, I5\}, \quad confidence = 2/7 = 29\% \\
I5 &\Rightarrow \{I1, I2\}, \quad confidence = 2/2 = 100\%
\end{aligned}
$$

If you apply Confidence % as =50%??

Only 4 Rules are of interest

The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
    C_{k+1} = candidates generated from L_k;
    for each transaction t in database do
        increment the count of all candidates in C_{k+1} that are
        contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
    end
return $\cup_k L_k$;

Further Improvement of the Apriori Method

- Major computational challenges
  - Multiple scans of transaction database
  - Huge number of candidates
  - Tedious workload of support counting for candidates
- Improving Apriori: general ideas
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

FP Growth Algorithm

- Based on the principle:

'Grow long patterns from short ones using local frequent items only'

- A ***Frequent Pattern Growth*** Approach
- based on ***FP Tree approach*** {How FP-Tree as a DS?}
- uses ***Conditional Base*** {as a Sub Database} ***of each item***
- More working examples: click here & click here

Pattern-Growth Approach: *Mining Frequent Patterns Without Candidate Generation*

- Bottlenecks of the Apriori approach
  - *Breadth-first (i.e., level-wise) search*
  - *Candidate generation* and *test*
    - Often generates a huge number of candidates
- The FP Growth Approach [*J. Han, J. Pei, and Y. Yin, SIGMOD' 00*]
  - *Depth-first search*
  - *Avoid explicit candidate generation*
- Major philosophy: Grow long patterns from short ones using local frequent items only
  - "abc" is a frequent pattern
  - Get all transactions having "abc", i.e., project DB on abc: DB|abc
  - "d" is a local frequent item in DB|abc → abcd is a frequent pattern

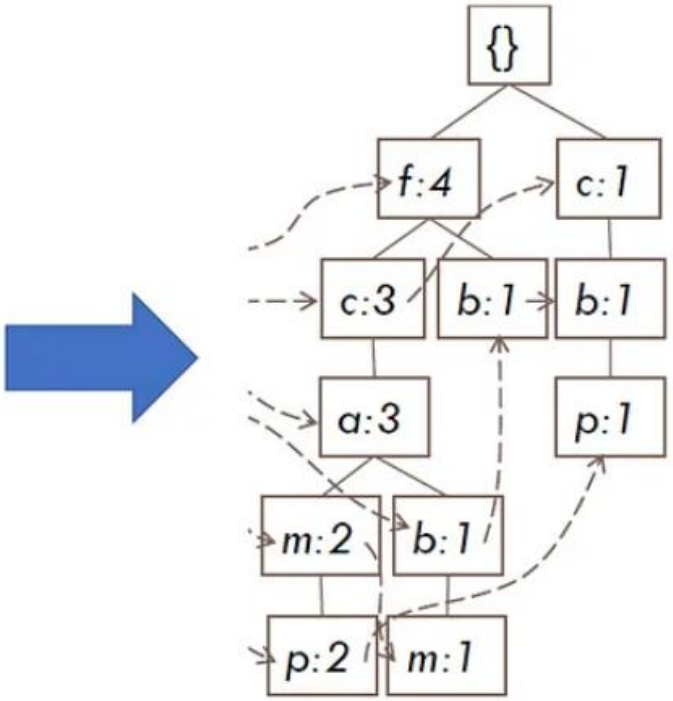
Frequent Pattern Tree (FP Tree)??

- FP Tree is a Data Structure (Trie Data Structure) .
  - It represents the *frequent itemsets* in the input dataset compactly and efficiently.
- *Major components:*
  - **Root Node:** an empty set. It has no associated item but a pointer to the first node of each item in the tree.
  - **Item Node:** Each item node in the FP-tree represents a unique item in the dataset. It stores the *item name and the frequency count of the item* in the dataset.
  - **Header Table:** The header table lists all the unique items in the dataset, along with their frequency count. It is used to track each item's location in the FP tree.
  - **Child Node:** Each child node of an item node represents an item that co-occurs with the item the parent node represents in at least one transaction in the dataset.
  - **Node Link:** The node-link is a pointer that connects each item in the header table to the first node of that item in the FP-tree. It is used to traverse the conditional pattern base of each item during the mining process.

Frequent Pattern Tree (FP Tree)??

- FP Tree is Data Structure.

TID	Items bought
100	{a, c, d, f, g, i, m, p}
200	{a, b, c, f, i, m, o}
300	{b, f, h, j, o}
400	{b, c, k, s, p}
500	{a, c, e, f, l, m, n, p}



Working Example: "FP Growth Algorithm "

Phase I: Ordered-Item Set Preparation

Transaction ID	Items
T1	{E, K, M, N, O, Y}
T2	{D, E, K, N, **O**, Y}
T3	{A, E, K, M}
T4	{C, K, M, U, Y}
T5	{C, E, I, K, O, O}

Prepare 1-Itemset

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

Apply Minimum support. e.g. 3.

Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}

Prepare *Ordered Item set* for each Transaction

Prepare Frequent Pattern Set (**L**) = **{K : 5, E : 4, M : 3, O : 4, Y : 3}**

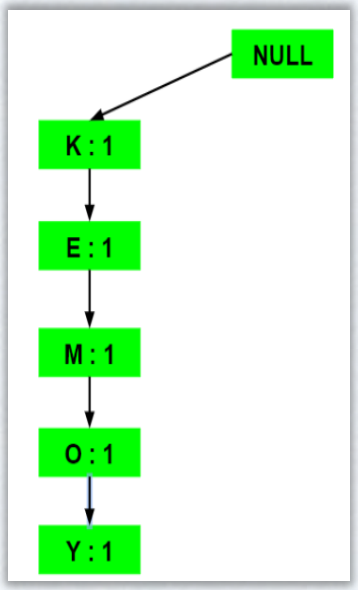
Source: https://www.geeksforgeeks.org/frequent-pattern-growth-algorithm/

Working Example: "FP Growth Algorithm "

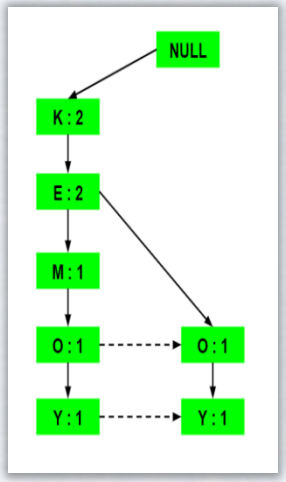
Phase II: FP Tree Construction

All the *Ordered-Item sets* are inserted into a *Trie Data Structure*, **e.g. FP Tree**

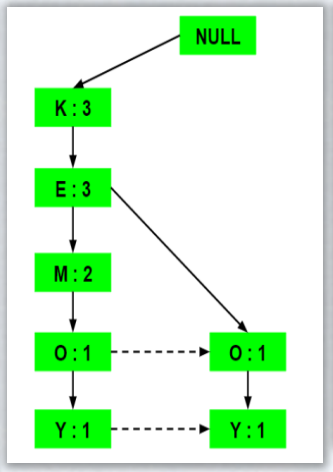
Inserting the set
{K, E, M, O, Y}



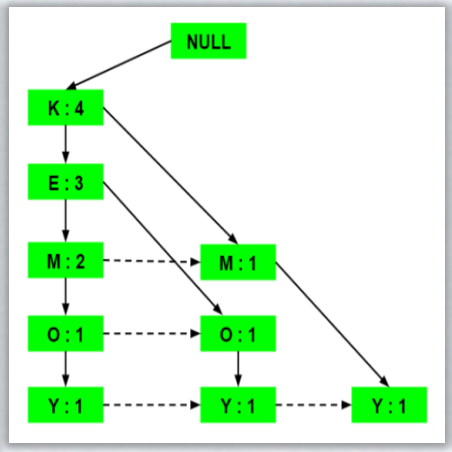
Inserting the set
{K, E, O, Y}



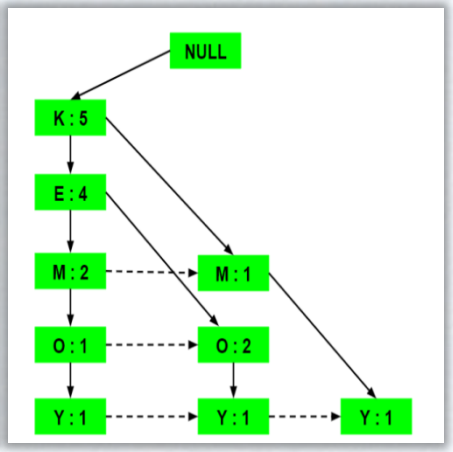
Inserting the set
{K, E, M}



Inserting the set
{K, M, Y}



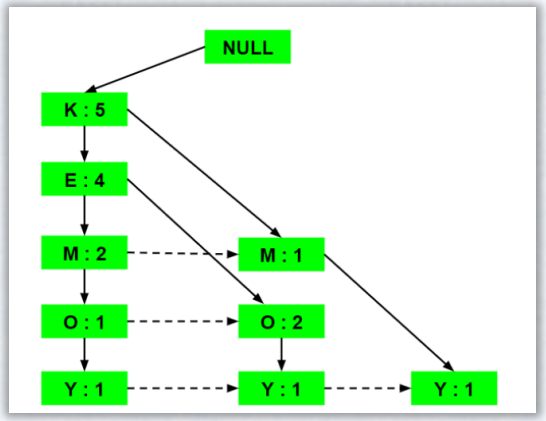
Inserting the set
{K, E, O}



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

Working Example: "FP Growth Algorithm "

Phase III: Computing Conditional Pattern Base & FP



For *each* item, the **Conditional Pattern Base** is computed which is "*path labels of all the paths which lead to any node of the given item in the **FP tree***".

Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	

Conditional Frequent Pattern Tree
{K : 3}
{K,E : 3}
{K : 3}
{K : 4}

Association Rules

K-> Y, Y-> K
K->E O, O->KE, OE->K …
K->M, M->K
E->K, K->E

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 4>}
K	

Apply Confidence Values/ Threshold to get Final ARs.

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

Working Example: "FP Growth Algorithm "

Phase IV: Generating Association Rules

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 4>}
K	

Transaction ID	Items
T1	{E, K, M, N, O, Y}
T2	{D, E, K, N, O, Y}
T3	{A, E, K, M}
T4	{C, K, M, U, Y}
T5	{C, E, I, K, O, O}

Association Rules

K-> Y Confidence=
Y-> K Confidence=
K->EO Confidence=
O->KE Confidence=
E-> OK Confidence=
KO->E Confidence=
OE->K Confidence=
EK-> O Confidence=
K->M Confidence=
M->K Confidence=
E->K Confidence=
K->E Confidence=

Apply Confidence Values/ Threshold to get Final ARs., *e.g. , 50%*

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

Advantages of the Pattern Growth Approach

- *Divide-and-conquer*:
  - Decompose both the mining task and DB according to the frequent patterns obtained so far
  - Lead to focused search of smaller databases
- Other factors
  - No candidate generation, no candidate test
  - Compressed database: FP-tree structure
  - No repeated scan of entire database
  - Basic ops: counting local freq items and building sub FP-tree, no pattern search and matching
- A good open-source implementation and refinement of FPGrowth
  - FPGrowth+ (Grahne and J. Zhu, FIMI'03)

Factor	FP Growth Algorithm	Apriori Algorithm
Working	FP Growth uses FP-tree to mine frequent itemsets.	Apriori algorithm mines frequent items in an iterative manner - 1-itemsets, 2-itemsets, 3-itemsets, etc.
Candidate Generation	Generates frequent itemsets by constructing the FP-Tree and recursively generating conditional pattern bases.	Generates candidate itemsets by joining and pruning.
Data Scanning	Scans the database only twice to construct the FP-Tree and generate conditional pattern bases.	Scans the database multiple times for frequent itemsets.
Memory Usage	Requires less memory than Apriori as it constructs the FP-Tree, which compresses the database	Requires a large amount of memory to store candidate itemsets.
Speed	Faster due to efficient data compression and generation of frequent itemsets.	Slower due to multiple database scans and candidate generation.
Scalability	Performs well on large datasets due to efficient data compression and generation of frequent itemsets.	Performs poorly on large datasets due to a large number of candidate itemsets.