A COURSE MATERIAL ON

SCAJA61 – DATA MINING

BY

R.SUDHANESH, M.Sc., MCA., M.Phil.,

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE

PARVATHYS ARTS AND SCIENCE COLLEGE, DINDIGUL

2021-2022

SYLLABUS

DATA MINING (III BCA)

UNIT-I

Introduction: Data mining application-data mining techniques-data mining case studies the future of data mining-data mining software

Association rules mining: Introduction-Basics-task and a Naïve algorithm-Apriori algorithm-improve the efficiency of the Apriori algorithm-mining frequent pattern without candidate generation(FP-growth)-performance evaluation of algorithms

UNIT-II

Data warehousing: Introduction-Operational data sources-data warehousing-data warehousing design- guidelines for data warehousing implementation- data warehousing-metadata

Online analytical processing(OLAP): Introduction-OLAP characteristics of OLAP system-multidimensional view and data cube- Data cube Implementation-Data Cube Operations OLAP implementation guidelines

UNIT-III

Classification: Introduction-decision tree-over fitting and pruning-DT rules- Naive Bayes method-estimation predictive accuracy of classification methods-other evaluation criteria for classification method-classification software

UNIT-IV

Cluster analysis: cluster analysis-types of data-computing distances-types of cluster analysis methods-partitioned methods-hierarchical methods-density based methods-dealing with large databases-quality and validity of cluster analysis methods-cluster analysis software

UNIT-V

Web data mining: Introduction-web terminology and characteristics- locality and hierarchy in the web-web content mining-web usage mining-web structure mining-web mining software

Search engines: search engines functionality- search engines architecture- ranking of web pages.

UNIT – I

INTRODUCTION:

Data Mining is a collection of techniques for effective automated discovery of previously unknown, valid, useful and understandable patterns in large data warehouses. Data Mining is all about discovering unsuspected or previously unknown relationships amongst the data. **"Data Mining"** refers to the extraction of useful information from a bulk of data or data warehouses. Data Mining is also known as **Knowledge Discovery or Knowledge Extraction.**

DATA MINING APPLICATION:

i) Prediction and Description:

Predictor will be constructed that predicts a continuous valued function or ordered value. For example, Manager predict customer spend sale future revenue for a company.

Describes concepts or task relevant data sets in summarative(characterization and comparison), informative, discriminative forms.

ii) Relationship Marketing:

Customer Relationship Management is all about acquiring and retaining customers, also improving customers' loyalty and implementing customer focused strategies. To maintain a proper relationship with a customer a business need to collect data and analyse the information. This is where data mining plays its part. With data mining technologies the collected data can be used for analysis.

iii) Customer Profiling:

It is the process of using relevant and available information to describe characteristics of group of customers tends to identify their discriminators from other customers. For example Insurance.

iv) Customer Segmentation:

Data mining aids in aligning the customers into a distinct segment and can tailor the needs according to the customers. Market is always about retaining the customers. Data mining allows to find a segment of customers based on vulnerability and the business could offer them with special offers and enhance satisfaction.

v) Outliers Identification and detecting fraud:

In **data mining**, **anomaly detection** (also **outlier detection**) is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the **data.** A supervised method includes collection of sample records. These records are classified fraudulent or non-fraudulent. A model is built using this data and the algorithm is made to identify whether the record is fraudulent or not.

vi) Website Design and promotion:

Web mining helps to improve the power of web search engine by classifying the web documents identifying the web pages. It is used for Web Searching e.g., Google, Yahoo etc and Vertical Searching e.g., FatLens, Become etc. Web mining is used to predict user behavior. Web mining is very useful of a particular Website and e-service e.g., landing page optimization.

DATA MINING TECHNIQUES:

i) Association Rule Mining

Association Rule Mining is used when you want to find an association between different objects in a set, find frequent patterns in a transaction database, relational databases or any other information repository. The applications of Association Rule Mining are found in Marketing, Basket Data Analysis (or Market Basket Analysis) in retailing, clustering and classification. The most common approach to find these patterns is Market Basket Analysis, which is a key technique used by large retailers like Amazon, Flipkart, etc to analyze customer buying habits by finding associations between the different items that customers place in their “shopping baskets”.

ii) Supervised Classification:

Supervised methods are methods that attempt to discover the relationship between input attributes (sometimes called independent variables) and a target attribute (sometimes referred to as a dependent variable). The relationship discovered is represented in a structure referred to as a model. It is useful to distinguish between two main supervised models: classification models (classifiers) and Regression Models. Regression models map the input space into a real-value domain.

iii) Cluster Analysis:

Cluster Analysis in Data Mining means that to find out the group of objects which are similar to each other in the group but are different from the object in other groups. In clustering, a group of different data objects is classified as similar objects. Data sets are divided into different groups in the cluster analysis, which is based on the similarity of the data. After the classification of data into various groups, a label is assigned to the group. It helps in adapting to the changes by doing the classification.

iv) Web Data Mining:

Web Mining is the process of Data Mining techniques to automatically discover and extract information from Web documents and services. Web mining can be broadly divided into three different types of techniques of mining: **Web Content Mining, Web Structure Mining, and Web Usage Mining.** These are explained as following below.

1. **Web Content Mining:**

   Web content mining is the application of extracting useful information from the content of the web documents. Web content consist of several types of data – text, image, audio, video etc. Content data is the group of facts that a web page is designed. This type of mining performs scanning and mining of the text, images and groups of web pages according to the content of the input.

2. **Web Structure Mining:**

   Web structure mining is the application of discovering structure information from the web. The structure of the web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages.

3. **Web Usage Mining:**

   Web usage mining is the application of identifying or discovering interesting usage patterns from large data sets. And these patterns enable you to understand the user behaviors or something like that. In web usage mining, user access data on the web and collect data in form of logs. So, Web usage mining is also called log mining.

v) Search Engines:

Search Engine refers to a huge database of internet resources such as web pages, newsgroups, programs, images etc. It helps to locate information on World Wide Web. User can **search** for any information by passing query in form of keywords or phrase.

vi) Data Warehousing and OLAP:

A **Data Warehousing** (DW) is process for collecting and managing data from varied sources to provide meaningful business insights. A Data warehouse is typically used to connect and analyze business data from heterogeneous sources. The data warehouse is the core of the BI system which is built for data analysis and reporting.

OLAP stands for "Online Analytical Processing." **OLAP** allows users to analyze database information from multiple database systems at one time. Because of its powerful **data** analysis capabilities, **OLAP** processing is often used for **data mining**, which aims to discover new relationships between different sets of **data**.

DATA MINING CASE STUDIES THE FUTURE OF DATA MINING:

1. **Multimedia Data Mining**

   This is one of the latest methods which is catching up because of the growing ability to capture useful data accurately. It involves the extraction of data from different kinds of multimedia sources such as audio, text, hypertext, video, images, etc. and the data is converted into a numerical representation in different formats. This method can be used in clustering and classifications, performing similarity checks, and also to identify associations.

2. **Ubiquitous Data Mining**

   This method involves the mining of data from mobile devices to get information about individuals. In spite of having several challenges in this type such as complexity, privacy, cost, etc. this method has a lot of opportunities to be enormous in various industries especially in studying human-computer interactions.

3. **Distributed Data Mining**

   This type of data mining is gaining popularity as it involves the mining of huge amount of information stored in different company locations or at different organizations. Highly sophisticated algorithms are used to extract data from different locations and provide proper insights and reports based upon them.

4. **Spatial and Geographic Data Mining**

   This is new trending type of data mining which includes extracting information from environmental, astronomical, and geographical data which also includes images taken from outer space. This type of data mining can reveal various aspects such as distance and topology which is mainly used in geographic information systems and other navigation applications.

5. **Time Series and Sequence Data Mining**

   The primary application of this type of data mining is study of cyclical and seasonal trends. This practice is also helpful in analyzing even random events which occur outside the normal series of events. This method is mainly being use by retail companies to access customer's buying patterns and their behaviors.

DATA MINING SOFTWARE:

i) Sisense

Basically, it allows companies of any size and industry to mash up data sets from various sources. Also, build a repository of rich reports that are shared across departments. If you want to test the software the vendor offers a great free trial plan.

ii) Oracle Data Mining

It is a representative of the Company's Advanced Analytics Database. Also, a market leader companies use to maximize the potential of their data. Hence, make accurate predictions. The system works with a powerful data algorithm to target best customers. Also, it identifies both anomalies and cross-selling opportunities. The user can also apply a different predictive model to need. Further, it customizes customer profiles in the desired way.

iii) RapidMiner

It is an integrated environment dedicated to **machine learning** and text mining. Also, it's one of the best predictive analysis systems available on the market. We can use tools for business intelligence, research, and application development. Created on an open source model, RapidMiner is offered both on-premise and in private cloud infrastructures. The RapidMiner suite consists of three modules:

> i. **RapidMiner Studio** dedicated to workflow and prediction design, prototyping and validation;
>
> ii. The **RapidMiner Server** used to share. Also, to operationalize the predictive models created within the Studio;
>
> iii. The **RapidMiner Radoop** that simplified predictive analysis.

iv) Microsoft SharePoint

It is everyone's first association when speaking of data management, and there is a good reason for that. The application is web-based and fully integrated with all Microsoft Office products. While it may not be the most powerful analyzer you can find. Although, SharePoint certainly is the simplest information tool beginners should consider.

v) IBM Cognos

Cognos is IBM's business intelligence suite use for reporting and data analytics. As it includes several customizable components. And that makes it applicable to all niches and industries. Such as Cognos Connection, the Query Studio, the Report Studio the Analysis Studio.

vi) KNIME

It’s an open data analysis platform. In this, you can deploy, scale and familiarize within less than no time. In the BI world, KNIME is known as the app. Also, it made predictive intelligence accessible to inexperienced users. Moreover, the data-driven innovation system helps uncover data potential. Also, it includes more than 1000 modules and ready-to-use examples and an array of integrated tools and algorithms.

vii) Dundas BI

Hence, it is known for its superb integrations and fast insights. The system brings together several analytic tools. Also, it allows unlimited transformation of industry data. Further, it enriches standard reporting with appealing tables, graphs, and charts.

viii) Board

It is an intelligence management toolkit recommend by our experts to all companies. Generally, for companies that are looking to improve decision making. The platform combines business intelligence and performance management in a single package. Also, it collects data from literally any available source. And streamlines reporting letting you extract documents in all preferred formats.

ASSOCIATION RULES MINING- INTRODUCTION:

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 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.

Before we start defining the rule, let us first see the basic definitions.

Support Count(   ) – Frequency of occurrence of a itemset.

Here   ({Milk, Bread, Diaper})=2

Frequent Itemset – An itemset whose support is greater than or equal to minsup threshold.

Association Rule – An implication expression of the form X -> Y, where X and Y are any 2 itemsets.

Example: {Milk, Diaper}->{Beer}

TID	ITEMS
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Rule Evaluation Metrics –

- **Support(s) –**

  The number of transactions that include items in the {X} and {Y} parts of the rule as a percentage of the total number of transaction.It is a measure of how frequently the collection of items occur together as a percentage of all transactions.

- **Support = (X+Y) total –**

  It is interpreted as fraction of transactions that contain both X and Y.

- **Confidence(c) –**

  It is the ratio of the no of transactions that includes all items in {B} as well as the no of transactions that includes all items in {A} to the no of transactions that includes all items in {A}.

- **Conf(X=>Y) = Supp(X Y) Supp(X) –**

  It measures how often each item in Y appears in transactions that contains items in X also.

- **Lift(l) –**

  The lift of the rule X=>Y is the confidence of the rule divided by the expected confidence, assuming that the itemsets X and Y are independent of each other.The expected confidence is the confidence divided by the frequency of {Y}.

- **Lift(X=>Y) = Conf(X=>Y) Supp(Y) –**

  Lift value near 1 indicates X and Y almost often appear together as expected, greater than 1 means they appear together more than expected and less than 1 means they appear less than expected. Greater lift values indicate stronger association.

Example – From the above table, {Milk, Diaper}=>{Beer}

s= ({Milk, Diaper, Beer}) |T|

= 2/5

= 0.4

c= (Milk, Diaper, Beer) (Milk, Diaper)

= 2/3

= 0.67

l= Supp({Milk, Diaper, Beer}) Supp({Milk, Diaper})*Supp({Beer})

= 0.4/(0.6*0.6)

= 1.11

The Association rule is very useful in analyzing datasets. The data is collected using bar-code scanners in supermarkets. Such databases consists of a large number of transaction records which list all items bought by a customer on a single purchase. So the manager could know if certain groups of items are consistently purchased together and use this data for adjusting store layouts, cross-selling, promotions based on statistics.

A NAÏVE ALGORITHM:

Naive Bayes is a probabilistic machine learning algorithm that can be used in a wide variety of classification tasks. Typical applications include filtering spam, classifying documents, sentiment prediction etc. The name naive is used because it assumes the features that go into the model is independent of each other. That is changing the value of one feature, does not directly influence or change the value of any of the other features used in the algorithm.

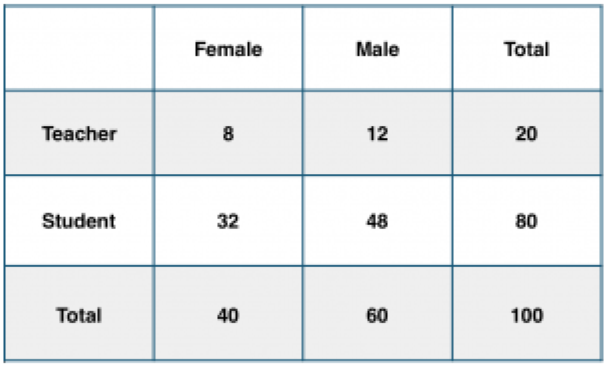
Conditional Probability:

Coin Toss and Fair Dice Example

When you flip a fair coin, there is an equal chance of getting either heads or tails. So you can say the probability of getting heads is 50%.Similarly what would be the probability of getting a 1 when you roll a dice with 6 faces? Assuming the dice is fair, the probability of 1/6 = 0.166.

School Example

Let’s see a slightly complicated example. Consider a school with a total population of 100 persons. These 100 persons can be seen either as ‘Students’ and ‘Teachers’ or as a population of ‘Males’ and ‘Females’. With below tabulation of the 100 people, what is the conditional probability that a certain member of the school is a ‘Teacher’ given that he is a ‘Man’?



	Female	Male	Total
Teacher	8	12	20
Student	32	48	80
Total	40	60	100

To calculate this, you may intuitively filter the sub-population of 60 males and focus on the 12 (male) teachers.So the required conditional probability P(Teacher | Male) = 12 / 60 = 0.2.

$$P(Teacher \mid Male) = \frac{P(Teacher \bigcap Male)}{P(Male)} = 12/60 = 0.2$$

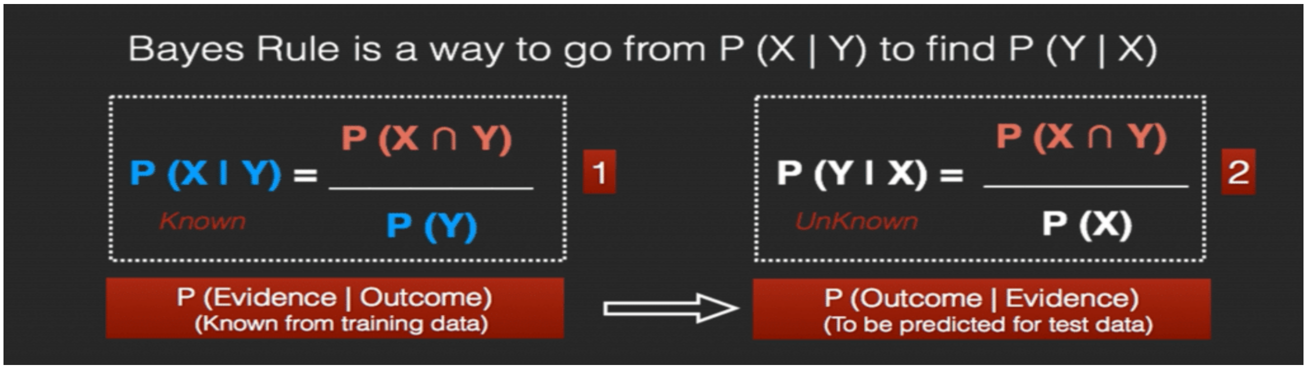
This can be represented as the intersection of Teacher (A) and Male (B) divided by Male (B). Likewise, the conditional probability of B given A can be computed. The Bayes Rule that we use for Naive Bayes, can be derived from these two notations.

$$P(A \mid B) = \frac{P(A \bigcap B)}{P(B)} \qquad (1)$$

$$P(B \mid A) = \frac{P(A \bigcap B)}{P(A)} \qquad (2)$$

THE BAYES RULE

The Bayes Rule is a way of going from P(X|Y), known from the training dataset, to find P(Y|X). To do this, we replace A and B in the above formula, with the feature X and response Y. For observations in test or scoring data, the X would be known while Y is unknown. And for each row of the test dataset, you want to compute the probability of Y given the X has already happened. What happens if Y has more than 2 categories? we compute the probability of each class of Y and let the highest win.



Bayes Rule is a way to go from P (X | Y) to find P (Y | X)

$$P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)} \quad 1$$

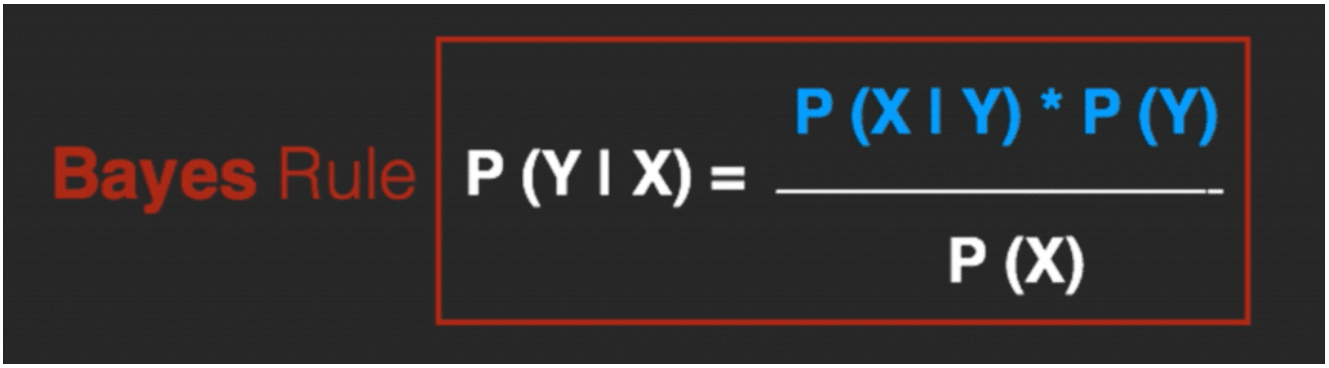
Known

$$P(Y \mid X) = \frac{P(X \cap Y)}{P(X)} \quad 2$$

UnKnown

P (Evidence | Outcome)
(Known from training data)

P (Outcome | Evidence)
(To be predicted for test data)

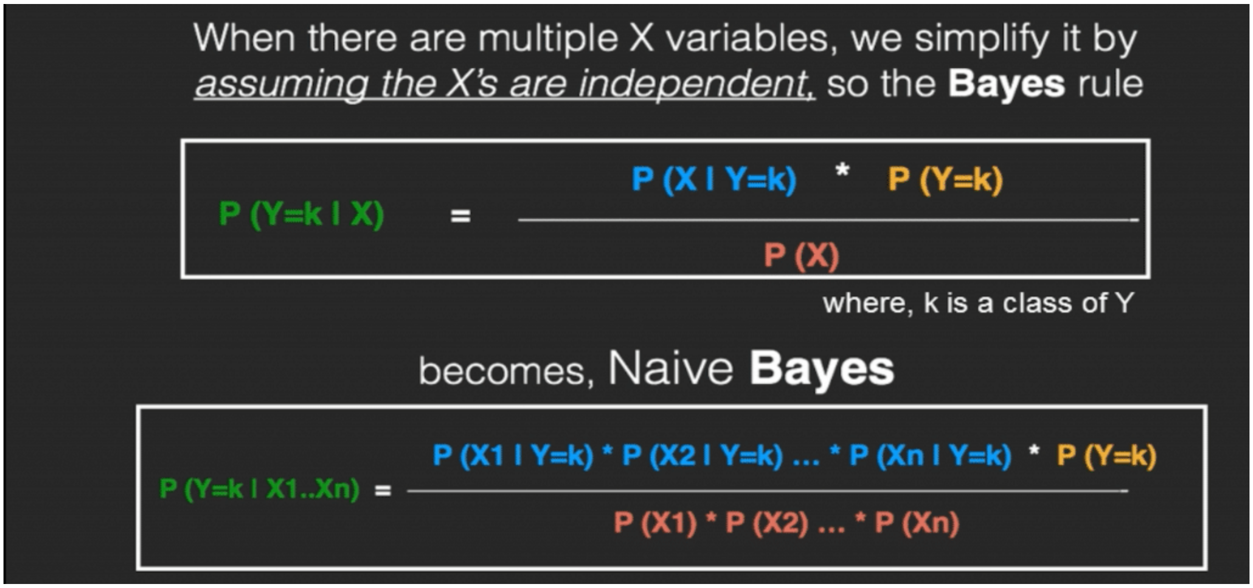


Bayes Rule

$$P(Y \mid X) = \frac{P(X \mid Y) * P(Y)}{P(X)}$$

THE NAIVE BAYES

The Bayes Rule provides the formula for the probability of Y given X. But, in real-world problems, you typically have multiple X variables.When the features are independent, we can extend the Bayes Rule to what is called Naive Bayes. It is called ‘Naive’ because of the naive assumption that the X’s are independent of each other. Regardless of its name, it’s a powerful formula.



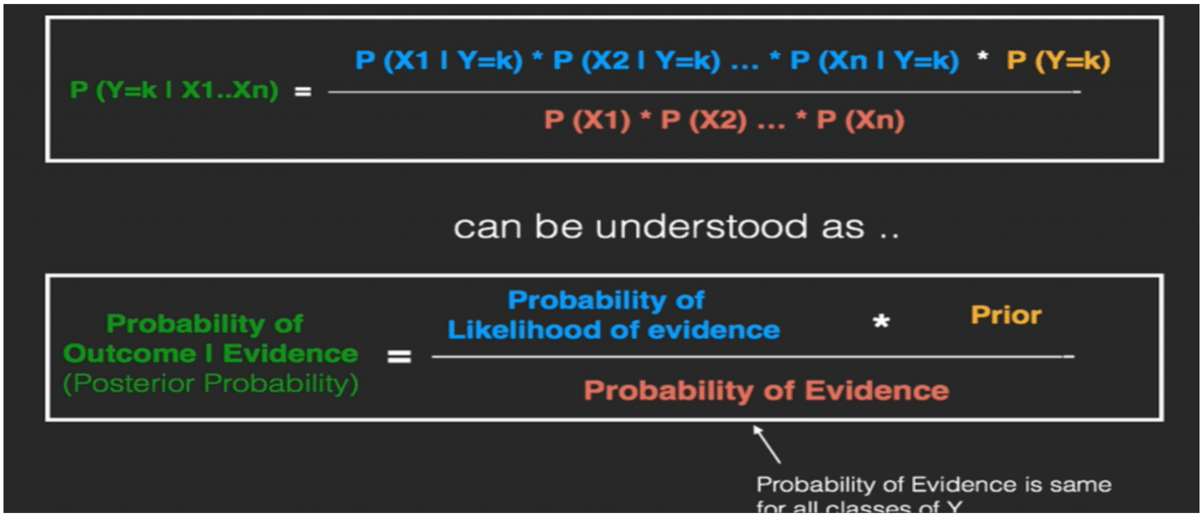
When there are multiple X variables, we simplify it by *assuming the X’s are independent*, so the **Bayes** rule

$$P(Y=k \mid X) = \frac{P(X \mid Y=k) \; * \; P(Y=k)}{P(X)}$$

where, k is a class of Y

becomes, Naive **Bayes**

$$P(Y=k \mid X1..Xn) = \frac{P(X1 \mid Y=k) * P(X2 \mid Y=k) \ldots * P(Xn \mid Y=k) \; * \; P(Y=k)}{P(X1) * P(X2) \ldots * P(Xn)}$$



$$P(Y=k \mid X1..Xn) = \frac{P(X1 \mid Y=k) * P(X2 \mid Y=k) \ldots * P(Xn \mid Y=k) \; * \; P(Y=k)}{P(X1) * P(X2) \ldots * P(Xn)}$$

can be understood as ..

$$\begin{array}{c}\text{Probability of}\\ \text{Outcome | Evidence}\\ \text{(Posterior Probability)}\end{array} = \frac{\begin{array}{c}\text{Probability of}\\ \text{Likelihood of evidence}\end{array} \; * \; \text{Prior}}{\text{Probability of Evidence}}$$

Probability of Evidence is same for all classes of Y

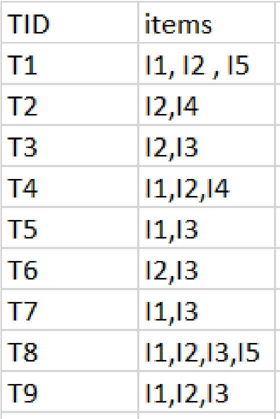
In technical jargon, the left-hand-side (LHS) of the equation is understood as the posterior probability or simply the posterior.The RHS has 2 terms in the numerator.

The first term is called the **‘Likelihood of Evidence’**. It is nothing but the conditional probability of each X’s given Y is of particular class ‘c’. Since all the X’s are assumed to be independent of each other, you can just multiply the ‘likelihoods’ of all the X’s and called it the ‘Probability of likelihood of evidence’. This is known from the training dataset by filtering records where Y=c. The second term is called the prior which is the overall probability of Y=c, where c is a class of Y. In simpler terms, Prior = count(Y=c) / n_Records.

APRIORI ALGORITHM:

Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets. To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called *Apriori property* which helps by reducing the search space.

Apriori Property –All non-empty subset of frequent itemset must be frequent. The key concept of Apriori algorithm is its anti-monotonicity of support measure. Apriori assumes that Before we start understanding the algorithm, go through some definitions which are explained in my previous post. Consider the following dataset and we will find frequent itemsets and generate association rules for them.



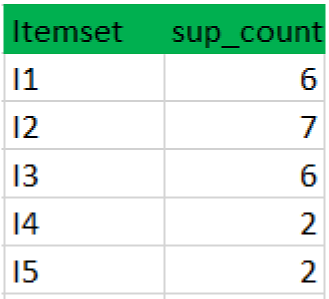
TID	items
T1	I1, I2 , I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

minimum support count is 2

minimum confidence is 60%

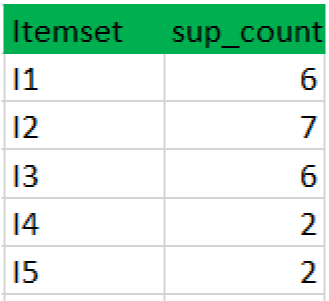
Step-1: K=1

(I) Create a table containing support count of each item present in dataset – Called **C1(candidate set)**



Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

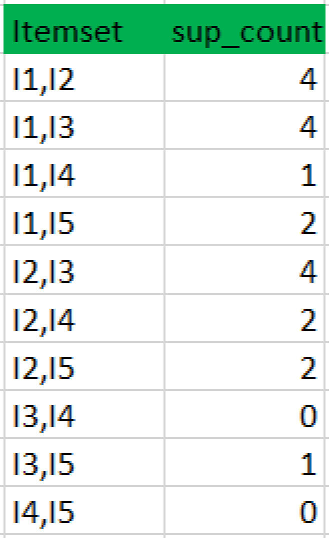
(II) compare candidate set item's support count with minimum support count(here min_support=2 if support_count of candidate set items is less than min_support then remove those items). This gives us itemset L1.



Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

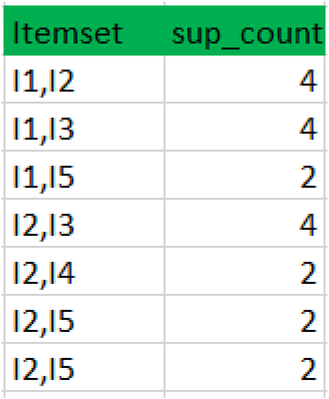
Step-2: K=2

- Generate candidate set C2 using L1 (this is called join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common.
- Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)
- Now find support count of these itemsets by searching in dataset.



Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

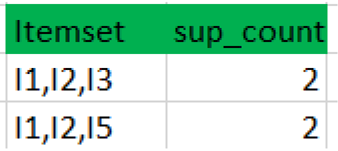
(II) compare candidate (C2) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L2.



Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I2,I5	2

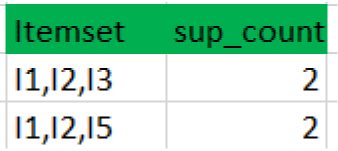
Step-3:

- Generate candidate set C3 using L2 (join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common. So here, for L2, first element should match. So itemset generated by joining L2 is {I1, I2, I3}{I1, I2, I5}{I1, I3, i5}{I2, I3, I4}{I2, I4, I5}{I2, I3, I5}
- Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
- find support count of these remaining itemset by searching in dataset.



Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

(II) Compare candidate (C3) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L3.



Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

Step-4:

- Generate candidate set C4 using L3 (join step). Condition of joining L_{k-1} and L_{k-1} (K=4) is that, they should have (K-2) elements in common. So here, for L3, first 2 elements (items) should match.
- Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So no itemset in C4
- We stop here because no frequent itemsets are found further

Thus, we have discovered all the frequent item-sets. Now generation of strong association rule comes into picture. For that we need to calculate confidence of each rule.

Confidence –

A confidence of 60% means that 60% of the customers, who purchased milk and bread also bought butter.

Confidence(A->B)=Support_count(A∪B)/Support_count(A)

So here, by taking an example of any frequent itemset, we will show the rule generation.

Itemset {I1, I2, I3} //from L3 SO rules can be

[I1^I2]=>[I3] //confidence = sup(I1^I2^I3)/sup(I1^I2) = 2/4*100=50%

[I1^I3]=>[I2] //confidence = sup(I1^I2^I3)/sup(I1^I3) = 2/4*100=50%

[I2^I3]=>[I1] //confidence = sup(I1^I2^I3)/sup(I2^I3) = 2/4*100=50%

[I1]=>[I2^I3] //confidence = sup(I1^I2^I3)/sup(I1) = 2/6*100=33%

[I2]=>[I1^I3] //confidence = sup(I1^I2^I3)/sup(I2) = 2/7*100=28%

[I3]=>[I1^I2] //confidence = sup(I1^I2^I3)/sup(I3) = 2/6*100=33%

So if minimum confidence is 50%, then first 3 rules can be considered as strong association rules.

Limitations of Apriori Algorithm

Apriori Algorithm can be slow. The main limitation is time required to hold a vast number of candidate sets with much frequent itemsets, low minimum support or large itemsets i.e. it is not an efficient approach for large number of datasets. For example, if there are 10^4 from frequent 1-itemsets, it need to generate more than 10^7 candidates into 2-length which in turn they will be tested and accumulate. Furthermore, to detect frequent pattern in size 100 i.e. v1, v2… v100, it have to generate 2^100 candidate itemsets that yield on costly and wasting of time of candidate generation. So, it will check for many sets from candidate itemsets, also it will scan database many times repeatedly for finding candidate itemsets. Apriori will be very low and inefficiency when memory capacity is limited with large number of transactions.

IMPROVE THE EFFICIENCY OF THE APRIORI ALGORITHM:

Many methods are available for improving the efficiency of the algorithm.

1. **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.

2. **Transaction Reduction:** This method reduces the number of transactions scanning in iterations. The transactions which do not contain frequent items are marked or removed.
3. **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.
4. **Sampling:** This method picks a random sample S from Database D and then searches for frequent itemset in S. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min_sup.
5. **Dynamic Itemset Counting:** This technique can add new candidate itemsets at any marked start point of the database during the scanning of the database

MINING FREQUENT PATTERN WITHOUT CANDIDATE GENERATION (FP-GROWTH):

This algorithm is an improvement to the Apriori method. A frequent pattern is generated without the need for candidate generation. FP growth algorithm represents the database in the form of a tree called a frequent pattern tree or FP tree. This tree structure will maintain the association between the itemsets. The database is fragmented using one frequent item. This fragmented part is called “pattern fragment”. The itemsets of these fragmented patterns are analyzed. Thus with this method, the search for frequent itemsets is reduced comparatively.

FP Tree

Frequent Pattern Tree is a tree-like structure that is made with the initial itemsets of the database. The purpose of the FP tree is to mine the most frequent pattern. Each node of the FP tree represents an item of the itemset. The root node represents null while the lower nodes represent the itemsets. The association of the nodes with the lower nodes that is the itemsets with the other itemsets are maintained while forming the tree.

Frequent Pattern Algorithm Steps

The frequent pattern growth method lets us find the frequent pattern without candidate generation.

Let us see the steps followed to mine the frequent pattern using frequent pattern growth algorithm:

#1) The first step is to scan the database to find the occurrences of the itemsets in the database. This step is the same as the first step of Apriori. The count of 1-itemsets in the database is called support count or frequency of 1-itemset.

#2) The second step is to construct the FP tree. For this, create the root of the tree. The root is represented by null.

#3) The next step is to scan the database again and examine the transactions. Examine the first transaction and find out the itemset in it. The itemset with the max count is taken at the top, the next itemset with lower count and so on. It means that the branch of the tree is constructed with transaction itemsets in descending order of count.

#4) The next transaction in the database is examined. The itemsets are ordered in descending order of count. If any itemset of this transaction is already present in another branch (for example in the 1st transaction), then this transaction branch would share a common prefix to the root.

This means that the common itemset is linked to the new node of another itemset in this transaction.

#5) Also, the count of the itemset is incremented as it occurs in the transactions. Both the common node and new node count is increased by 1 as they are created and linked according to transactions.

#6) The next step is to mine the created FP Tree. For this, the lowest node is examined first along with the links of the lowest nodes. The lowest node represents the frequency pattern length 1. From this, traverse the path in the FP Tree. This path or paths are called a conditional pattern base.

Conditional pattern base is a sub-database consisting of prefix paths in the FP tree occurring with the lowest node (suffix).

#7) Construct a Conditional FP Tree, which is formed by a count of itemsets in the path. The itemsets meeting the threshold support are considered in the Conditional FP Tree.

#8) Frequent Patterns are generated from the Conditional FP Tree.

Example Of FP-Growth Algorithm

Support threshold=50%, Confidence= 60%

Table 1

Transaction	**List of items**
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5

Transaction	List of items
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Solution:

Support threshold=50% => 0.5*6= 3 => min_sup=3

1. Count of each item

Table 2

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

2. Sort the itemset in descending order.

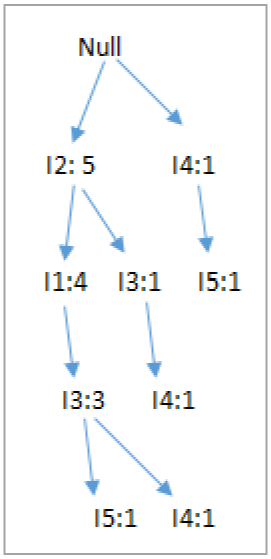
Table 3

Item	Count
I2	5
I1	4
I3	4
I4	4

3. Build FP Tree

1. Considering the root node null.

2. The first scan of Transaction T1: I1, I2, I3 contains three items {I1:1}, {I2:1}, {I3:1}, where I2 is linked as a child to root, I1 is linked to I2 and I3 is linked to I1.
3. T2: I2, I3, I4 contains I2, I3, and I4, where I2 is linked to root, I3 is linked to I2 and I4 is linked to I3. But this branch would share I2 node as common as it is already used in T1.
4. Increment the count of I2 by 1 and I3 is linked as a child to I2, I4 is linked as a child to I3. The count is {I2:2}, {I3:1}, {I4:1}.
5. T3: I4, I5. Similarly, a new branch with I5 is linked to I4 as a child is created.
6. T4: I1, I2, I4. The sequence will be I2, I1, and I4. I2 is already linked to the root node, hence it will be incremented by 1. Similarly I1 will be incremented by 1 as it is already linked with I2 in T1, thus {I2:3}, {I1:2}, {I4:1}.
7. T5:I1, I2, I3, I5. The sequence will be I2, I1, I3, and I5. Thus {I2:4}, {I1:3}, {I3:2}, {I5:1}.
8. T6: I1, I2, I3, I4. The sequence will be I2, I1, I3, and I4. Thus {I2:5}, {I1:4}, {I3:3}, {I4 1}.



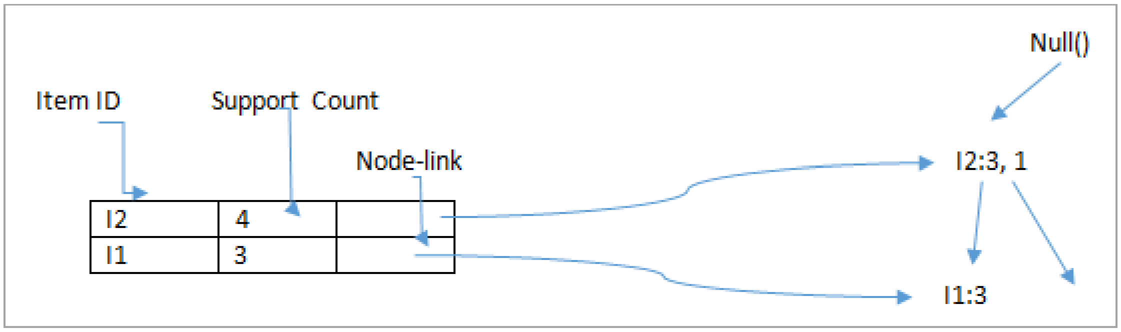
4. Mining of FP-tree is summarized below:

1. The lowest node item I5 is not considered as it does not have a min support count, hence it is deleted.
2. The next lower node is I4. I4 occurs in 2 branches , {I2,I1,I3:,I41},{I2,I3,I4:1}. Therefore considering I4 as suffix the prefix paths will be {I2, I1, I3:1}, {I2, I3: 1}. This forms the conditional pattern base.
3. The conditional pattern base is considered a transaction database, an FP-tree is constructed. This will contain {I2:2, I3:2}, I1 is not considered as it does not meet the min support count.
4. This path will generate all combinations of frequent patterns : {I2,I4:2},{I3,I4:2},{I2,I3,I4:2}

5. For I3, the prefix path would be: {I2,I1:3},{I2:1}, this will generate a 2 node FP-tree : {I2:4, I1:3} and frequent patterns are generated: {I2,I3:4}, {I1:I3:3}, {I2,I1,I3:3}.
6. For I1, the prefix path would be: {I2:4} this will generate a single node FP-tree: {I2:4} and frequent patterns are generated: {I2, I1:4}.

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I4	{I2,I1,I3:1},{I2,I3:1}	{I2:2, I3:2}	{I2,I4:2},{I3,I4:2},{I2,I3,I4:2}
I3	{I2,I1:3},{I2:1}	{I2:4, I1:3}	{I2,I3:4}, {I1:I3:3}, {I2,I1,I3:3}
I1	{I2:4}	{I2:4}	{I2,I1:4}

The diagram given below depicts the conditional FP tree associated with the conditional node I3.



Advantages Of FP Growth Algorithm

1. This algorithm needs to scan the database only twice when compared to Apriori which scans the transactions for each iteration.
2. The pairing of items is not done in this algorithm and this makes it faster.
3. The database is stored in a compact version in memory.
4. It is efficient and scalable for mining both long and short frequent patterns.

Disadvantages Of FP-Growth Algorithm

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

UNIT – II

DATA WAREHOUSING: INTRODUCTION:

A **Data Warehouse** refers to a place where data can be stored for useful mining. It is like a quick computer system with exceptionally huge data storage capacity. Data warehouse combines data from numerous sources which ensure the data quality, accuracy, and consistency. Data warehouse boosts system execution by separating analytics processing from transnational databases. Data flows into a data warehouse from different databases. A data warehouse works by sorting out data into a pattern that depicts the format and types of data. Query tools examine the data tables using patterns. **Data warehouses** and **databases** both are relative data systems, but both are made to serve different purposes. A data warehouse is built to store a huge amount of historical data and empowers fast requests over all the data, typically using **Online Analytical Processing** (OLAP). A database is made to store current transactions and allow quick access to specific transactions for ongoing business processes, commonly known as **Online Transaction Processing** (OLTP).

OPERATIONAL DATA STORES:

An **ODS** has been described by **Inmon** and **Imhoff** (1996) as a subject-oriented, integrated, volatile, current valued data store, containing only detailed corporate data. A data warehouse is a documenting database that includes associatively recent as well as historical information and may also include aggregate data.

The **ODS** is a **subject-oriented**. It is organized around the significant information subject of an enterprise. In a university, the subjects may be students, lecturers and courses while in the company the subjects might be users, salespersons and products.

The **ODS** is an **integrated**. That is, it is a group of subject-oriented record from a variety of systems to provides an enterprise-wide view of the information.

The **ODS** is a **current-valued**. That is, an **ODS** is up-to-date and follow the current status of the data. An ODS does not contain historical information. Since the OLTP system data is changing all the time, data from underlying sources refresh the ODS as generally and frequently as possible.

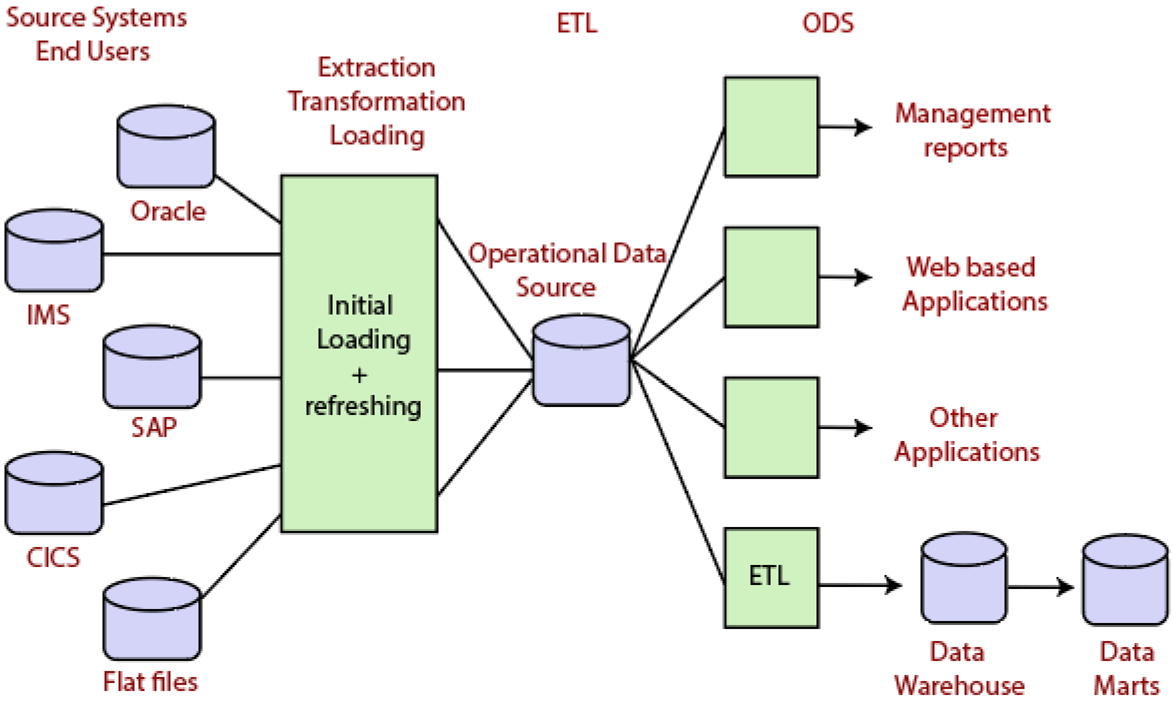
The **ODS** is **volatile**. That is, the data in the ODS frequently changes as new data refreshes the ODS.

The ODS is a **detailed**. That is, ODS is detailed enough to serve the need of the operational management staff in the enterprise. The granularity of the information in the ODS does not have to be precisely the same as in the source **OLTP** system.

ODS Design and Implementation

The extraction of data from source databases needs to be efficient, and the quality of records needs to be maintained. Since the data is refreshed generally and frequently, suitable checks are required to ensure the quality of data after each refresh. An **ODS** is a read-only database other than regular refreshing by the OLTP systems. Customer should not be allowed to update ODS information.

Populating an ODS contains an acquisition phase of extracting, transforming and loading information from OLTP source systems. This procedure is **ETL**. Completing populating the database, analyze for anomalies and testing for performance are essential before an ODS system can go online.



Operational Data Store Structure

DATA WAREHOUSING

A data warehouse is built to support management functions whereas data mining is used to extract useful information and patterns from data. Data warehousing is the process of compiling information into a data warehouse. It is a technology that aggregates structured data from one or more sources so that it can be compared and analyzed rather than transaction processing. A **data warehouse** is designed to support management decision-making process by providing a platform for data cleaning, data integration and data consolidation. A data warehouse contains subject-oriented, integrated, time-variant and non-volatile data.Data warehouse consolidates data from many sources while ensuring data quality, consistency and accuracy. Data warehouse improves system performance by separating analytics processing from transnational databases. Data flows into a data warehouse from the various databases. A data warehouse works by organizing data into a schema which describes the layout and type of data. Query tools analyze the data tables using schema.

Comparison between Data warehouse and OLTP

	Data warehouse	**OLTP**
System orientation	It is a market-oriented database used by management for the decision making process.	It is a customer-oriented database used by employees.
Data	Data warehouse contains historical data which is aggregated and summarized.	The data present in the transactional database is current data and too detailed.
Database Design	Data warehouse follows star and snow flake schema model for designing the database.	The database follows the entity-relationship(ER) database, model
Data Access	The data warehouse has read-only access with complex queries. It does not support data access for day-to-day operations	The OLTP database is accessible via simple queries. It is used for short transactions. OLTP requires mechanisms such as concurrency and recovery control.

Data View	The information contained in the data warehouse comes from multiple data sources. It spans multiple database schemas. It requires multidimensional data view.	The OTLP focuses mainly on current data.
Data Operations	Data warehouse requires a lot of scans	OLTP follows indexing and hashing on the primary key
Database size	The size of DW is more than terabytes of data	The size of OLTP ranges from few gigabytes to hundreds of gigabytes
Database records accessed	About millions of records can be accessed at one time.	About tens or hundreds of records can be accessed at one time.
Database Function	DW is used for information processing and analysis	OLTP is used for operational processing and transaction processing.
Data Query	DW information can be processed by complex queries	OLTP database operations can be accessed by simple queries

DIFFERENCE BETWEEN ODS AND DATA WAREHOUSE

Operational Database	**Data Warehouse**
Operational systems are designed to support high-volume transaction processing.	Data warehousing systems are typically designed to support high-volume analytical processing (i.e., OLAP).
Operational systems are usually concerned with current data.	Data warehousing systems are usually concerned with historical data.
Data within operational systems are mainly	Non-volatile, new data may be added

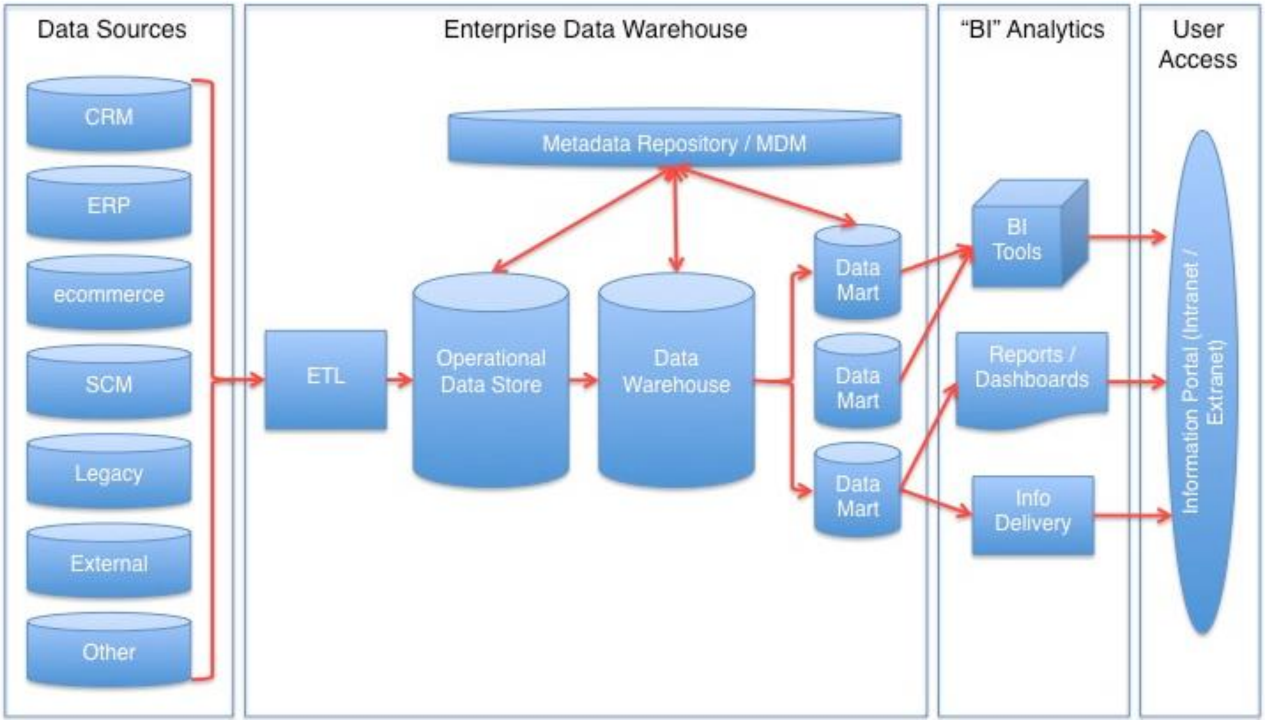
updated regularly according to need.	regularly. Once Added rarely changed.
It is designed for real-time business dealing and processes.	It is designed for analysis of business measures by subject area, categories, and attributes.
It is optimized for a simple set of transactions, generally adding or retrieving a single row at a time per table.	It is optimized for extent loads and high, complex, unpredictable queries that access many rows per table.
It is optimized for validation of incoming information during transactions, uses validation data tables.	Loaded with consistent, valid information, requires no real-time validation.
It supports thousands of concurrent clients.	It supports a few concurrent clients relative to OLTP.
Operational systems are widely process-oriented.	Data warehousing systems are widely subject-oriented
Operational systems are usually optimized to perform fast inserts and updates of associatively small volumes of data.	Data warehousing systems are usually optimized to perform fast retrievals of relatively high volumes of data.
Data In	Data Out
Less Number of data accessed.	Large Number of data accessed.
Relational databases are created for on-line transactional Processing (OLTP)	Data Warehouse designed for on-line Analytical Processing (OLAP)

Data Mart

Data mart contains a subset of organization-wide data. This subset of data is valuable to specific groups of an organization. For example, the marketing data mart may contain data related to items, customers, and sales. Data marts are confined to subjects.

Points to remember about data marts −

- Window-based or Unix/Linux-based servers are used to implement data marts. They are implemented on low-cost servers.
- The implementation data mart cycles is measured in short periods of time, i.e., in weeks rather than months or years.
- The life cycle of a data mart may be complex in long run, if its planning and design are not organization-wide.
- Data marts are small in size.
- Data marts are customized by department.
- The source of a data mart is departmentally structured data warehouse.
- Data mart are flexible.



DATA WAREHOUSING DESIGN

To design an effective and efficient data warehouse, we need to understand and analyze the business needs and construct a **business analysis framework**. Each person has different views regarding the design of a data warehouse. These views are as follows −

- **The top-down view** − This view allows the selection of relevant information needed for a data warehouse.

- **The data source view** − This view presents the information being captured, stored, and managed by the operational system.

- **The data warehouse view** − This view includes the fact tables and dimension tables. It represents the information stored inside the data warehouse.

- **The business query view** − It is the view of the data from the viewpoint of the end-user.

Three-Tier Data Warehouse Architecture

Generally a data warehouses adopts a three-tier architecture. Following are the three tiers of the data warehouse architecture.

- **Bottom Tier** − The bottom tier of the architecture is the data warehouse database server. It is the relational database system. We use the back end tools and utilities to feed data into the bottom tier. These back end tools and utilities perform the Extract, Clean, Load, and refresh functions.

- **Middle Tier** − In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.

    - By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.

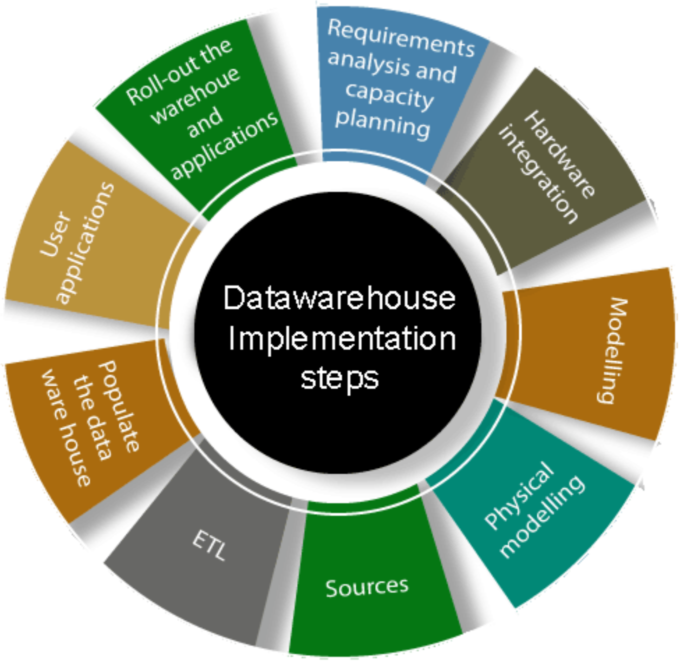
    - By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.

- **Top-Tier** − This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

GUIDELINES FOR DATA WAREHOUSING IMPLEMENTATION

1. Requirements analysis and capacity planning: The first process in data warehousing involves defining enterprise needs, defining architectures, carrying out capacity planning, and selecting the hardware and software tools. This step will contain be consulting senior management as well as the different stakeholder.

2. Hardware integration: Once the hardware and software has been selected, they require to be put by integrating the servers, the storage methods, and the user software tools.


Datawarehouse Implementation steps
Requirements analysis and capacity planning
Hardware integration
Modelling
Physical modelling
Sources
ETL
Populate the data ware house
User applications
Roll-out the warehoue and applications

3. Modeling: Modelling is a significant stage that involves designing the warehouse schema and views. This may contain using a modeling tool if the data warehouses are sophisticated.

4. Physical modeling: For the data warehouses to perform efficiently, physical modeling is needed. This contains designing the physical data warehouse organization, data placement, data partitioning, deciding on access techniques, and indexing.

5. Sources: The information for the data warehouse is likely to come from several data sources. This step contains identifying and connecting the sources using the gateway, ODBC drives, or another wrapper.

6. ETL: The data from the source system will require to go through an ETL phase. The process of designing and implementing the ETL phase may contain defining a suitable ETL tool vendors and purchasing and implementing the tools. This may contains customize the tool to suit the need of the enterprises.

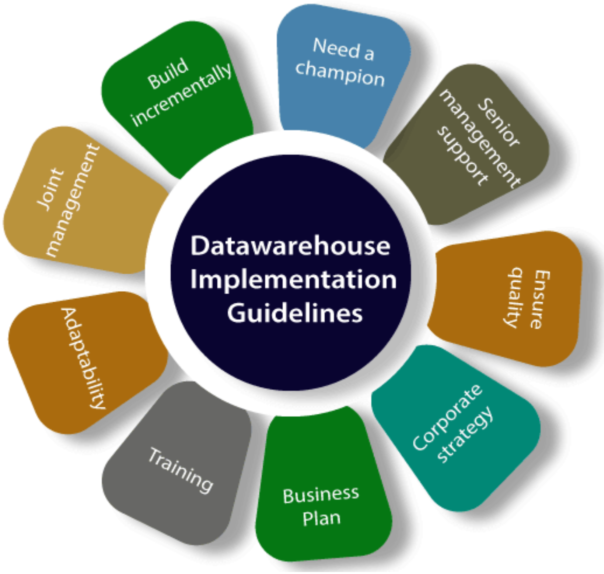
7. Populate the data warehouses: Once the ETL tools have been agreed upon, testing the tools will be needed, perhaps using a staging area. Once everything is working adequately, the ETL tools may be used in populating the warehouses given the schema and view definition.

8. User applications: For the data warehouses to be helpful, there must be end-user applications. This step contains designing and implementing applications required by the end-users.

9. Roll-out the warehouses and applications: Once the data warehouse has been populated and the end-client applications tested, the warehouse system and the operations may be rolled out for the user's community to use.

Implementation Guidelines

1. Build incrementally: Data warehouses must be built incrementally. Generally, it is recommended that a data marts may be created with one particular project in mind, and once it is implemented, several other sections of the enterprise may also want to implement similar systems. An enterprise data warehouses can then be implemented in an iterative manner allowing all data marts to extract information from the data warehouse.



2. Need a champion: A data warehouses project must have a champion who is active to carry out considerable researches into expected price and benefit of the project. Data warehousing projects requires inputs from many units in an enterprise and therefore needs to be driven by someone who is needed for interacting with people in the enterprises and can actively persuade colleagues.

3. Senior management support: A data warehouses project must be fully supported by senior management. Given the resource-intensive feature of such project and the time they can take to implement, a warehouse project signal for a sustained commitment from senior management.

4. Ensure quality: The only record that has been cleaned and is of a quality that is implicit by the organizations should be loaded in the data warehouses.

5. Corporate strategy: A data warehouse project must be suitable for corporate strategies and business goals. The purpose of the project must be defined before the beginning of the projects.

6. Business plan: The financial costs (hardware, software, and peopleware), expected advantage, and a project plan for a data warehouses project must be clearly outlined and understood by all stakeholders. Without such understanding, rumors about expenditure and benefits can become the only sources of data, subversion the projects.

7. Training: Data warehouses projects must not overlook data warehouses training requirements. For a data warehouses project to be successful, the customers must be trained to use the warehouses and to understand its capabilities.

8. Adaptability: The project should build in flexibility so that changes may be made to the data warehouses if and when required. Like any system, a data warehouse will require to change, as the needs of an enterprise change.

9. Joint management: The project must be handled by both IT and business professionals in the enterprise. To ensure that proper communication with the stakeholder and which the project is the target for assisting the enterprise's business, the business professional must be involved in the project along with technical professionals.

DATA WAREHOUSING-METADATA

Metadata is simply defined as data about data. The data that is used to represent other data is known as metadata. For example, the index of a book serves as a metadata for the contents in the book. In other words, we can say that metadata is the summarized data that leads us to detailed data. In terms of data warehouse, we can define metadata as follows.

- Metadata is the road-map to a data warehouse.
- Metadata in a data warehouse defines the warehouse objects.

- Metadata acts as a directory. This directory helps the decision support system to locate the contents of a data warehouse.

ONLINE ANALYTICAL PROCESSING (OLAP): INTRODUCTION

OLAP is a category of software that allows users to analyze information from multiple database systems at the same time. It is a technology that enables analysts to extract and view business data from different points of view. OLAP stands for Online Analytical Processing. OLAP system is an OLAP cube (also called a 'multidimensional cube' or a hypercube). It consists of numeric facts called *measures* that are categorized by *dimensions*. The measures are placed at the intersections of the hypercube, which is spanned by the dimensions as a vector space. The usual interface to manipulate an OLAP cube is a matrix interface, like Pivot tables in a spreadsheet program, which performs projection operations along the dimensions, such as aggregation or averaging. The cube metadata is typically created from a star schema or snowflake schema or fact constellation of tables in a relational database. Measures are derived from the records in the fact table and dimensions are derived from the dimension tables. Each *measure* can be thought of as having a set of *labels*, or meta-data associated with it. A *dimension* is what describes these *labels*; it provides information about the *measure*.

CHARACTERISTICS OF OLAP SYSTEM

BASIS FOR COMPARISON	OLTP	OLAP
Basic	It is an online transactional system and manages database modification.	It is an online data retrieving and data analysis system.
Focus	Insert, Update, Delete information from the database.	Extract data for analyzing that helps in decision making.
Data	OLTP and its transactions are	Different OLTPs database becomes

BASIS FOR COMPARISON	OLTP	OLAP
	the original source of data.	the source of data for OLAP.
Transaction	OLTP has short transactions.	OLAP has long transactions.
Time	The processing time of a transaction is comparatively less in OLTP.	The processing time of a transaction is comparatively more in OLAP.
Queries	Simpler queries.	Complex queries.
Normalization	Tables in OLTP database are normalized (3NF).	Tables in OLAP database are not normalized.
Integrity	OLTP database must maintain data integrity constraint.	OLAP database does not get frequently modified. Hence, data integrity is not affected.

FASMI characteristics of OLAP methods

i) Fast

It defines which the system targeted to deliver the most feedback to the client within about five seconds, with the elementary analysis taking no more than one second and very few taking more than 20 seconds.

ii) Analysis

It defines which the method can cope with any business logic and statistical analysis that is relevant for the function and the user, keep it easy enough for the target client. Although some preprogramming may be needed we do not think it acceptable if all application definitions have to be allow the user to define new Adhoc calculations as part of the analysis and to document on the data in any desired method, without having to program so we excludes products (like Oracle

Discoverer) that do not allow the user to define new Adhoc calculation as part of the analysis and to document on the data in any desired product that do not allow adequate end user-oriented calculation flexibility.

iii) Share

It defines which the system tools all the security requirements for understanding and, if multiple write connection is needed, concurrent update location at an appropriated level, not all functions need customer to write data back, but for the increasing number which does, the system should be able to manage multiple updates in a timely, secure manner.

iv) Multidimensional

This is the basic requirement. OLAP system must provide a multidimensional conceptual view of the data, including full support for hierarchies, as this is certainly the most logical method to analyze business and organizations.

v) Information

The system should be able to hold all the data needed by the applications. Data sparsity should be handled in an efficient manner.

CODD'S OLAP characteristics:

OLAP was introduces by **Dr.E.F.Codd** in 1993 and he presented 12 rules regarding OLAP:

1. **Multidimensional Conceptual View:** Multidimensional data model is provided that is intuitively analytical and easy to use. A multidimensional data model decides how the users perceive business problems.
2. **Transparency:** It makes the technology, underlying data repository, computing architecture and the diverse nature of source data totally transparent to users.
3. **Accessibility:** Access should provided only to the data that is actually needed to perform the specific analysis, presenting a single, coherent and consistent view to the users.
4. **Consistent Reporting Performance:** Users should not experience any significant degradation in reporting performance as the number of dimensions or the size of the database increases. It also ensures users must perceive consistent run time, response time or machine utilization every time a given query is run.
5. **Client/Server Architecture:** It conforms the system to the principles of client/server architecture for optimum performance, flexibility, adaptability and interoperability.
6. **Generic Dimensionality:** It should be ensured that very data dimension is equivalent in both structure and operational capabilities. Have one logical structure for all dimensions.

7. **Dynamic Sparse Matrix Handling:** Adaption should be of the physical schema to the specific analytical model being created and loaded that optimizes sparse matrix handling.
8. **Multi-user Support:** Support should be provided for end users to work concurrently with either the same analytical model or to create different models from the same data.
9. **Unrestricted Cross-dimensional Operations:** System should have abilities to recognize dimensional and automatically perform roll-up and drill-down operations within a dimension or across dimensions.
10. **Intuitive Data Manipulation:** Consolidation path reorientation, drill-down and roll-up and other manipulations to be accomplished intuitively should be enabled and directly via point and click actions.
11. **Flexible Reporting:** Business user is provided capabilities to arrange columns, rows and cells in manner that gives the facility of easy manipulation, analysis and synthesis of information.
12. **Unlimited Dimensions and Aggregation Levels:** There should be at least fifteen or twenty data dimensions within a common analytical model.

MULTIDIMENSIONAL VIEW AND DATA CUBE

When data is grouped or combined in multidimensional matrices called Data Cubes. The data cube method has a few alternative names or a few variants, such as "Multidimensional databases," "materialized views," and "OLAP (On-Line Analytical Processing)." Data cubes could be sparse in many cases because not every cell in each dimension may have corresponding data in the database. Techniques should be developed to handle sparse cubes efficiently. If a query contains constants at even lower levels than those provided in a data cube, it is not clear how to make the best use of the precomputed results stored in the data cube. The model view data in the form of a data cube. OLAP tools are based on the multidimensional data model. Data cubes usually model n-dimensional data. A data cube enables data to be modeled and viewed in multiple dimensions. A multidimensional data model is organized around a central theme, like sales and transactions. A fact table represents this theme. Facts are numerical measures. Thus, the fact table contains measure (such as Rs_sold) and keys to each of the related dimensional tables. Dimensions are a fact that defines a data cube. Facts are generally quantities, which are used for analyzing the relationship between dimensions.

DATA CUBE IMPLEMENTATION

The goal is to retrieve the decision support information from the data cube in the most efficient way possible. Three possible solutions are:

1. **Pre-compute all cells in the cube**
2. **Pre-compute no cells**
3. **Pre-compute some of the cells**

If the whole cube is pre-computed, then queries run on the cube will be very fast. The disadvantage is that the pre-computed cube requires a lot of memory. The size of a cube for *n* attributes $A_1,...,A_n$ with cardinalities $|A_1|,...,|A_n|$ is $\pi|A_i|$. This size increases exponentially with the number of attributes and linearly with the cardinalities of those attributes. To minimize memory requirements, we can pre-compute none of the cells in the cube. The disadvantage here is that queries on the cube will run more slowly because the cube will need to be rebuilt for each query. As a compromise between these two, we can pre-compute only those cells in the cube which will most likely be used for decision support queries. The trade-off between memory space and computing time is called the ***space-time trade-off***, and it often exists in data mining and computer science in general.

Data cubes are mainly categorized into two categories:

- **Multidimensional Data Cube:** Most OLAP products are developed based on a structure where the cube is patterned as a multidimensional array. These multidimensional OLAP (MOLAP) products usually offers improved performance when compared to other approaches mainly because they can be indexed directly into the structure of the data cube to gather subsets of data. When the number of dimensions is greater, the cube becomes sparser. That means that several cells that represent particular attribute combinations will not contain any aggregated data. This in turn boosts the storage requirements, which may reach undesirable levels at times, making the MOLAP solution untenable for huge data sets with many dimensions. Compression techniques might help; however, their use can damage the natural indexing of MOLAP.

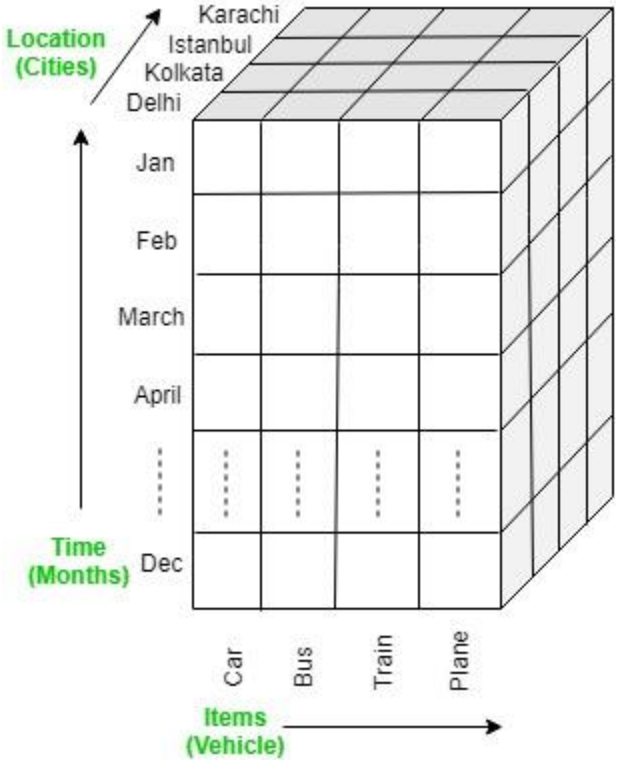
- **Relational OLAP**: Relational OLAP make use of the relational database model. The ROLAP data cube is employed as a bunch of relational tables (approximately twice as many as the quantity of dimensions) compared to a multidimensional array. Each one of these tables, known as a cuboid, signifies a specific view.

DATA CUBE OPERATIONS

There are five basic analytical operations that can be performed on an OLAP cube:

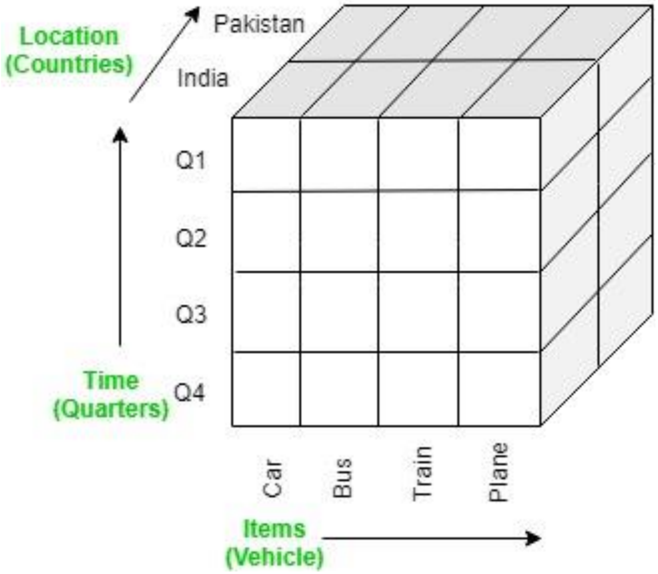
1. **Drill down:** In drill-down operation, the less detailed data is converted into highly detailed data. It can be done by:
   - Moving down in the concept hierarchy
   - Adding a new dimension

   In the cube given in overview section, the drill down operation is performed by moving down in the concept hierarchy of *Time* dimension (Quarter -> Month).

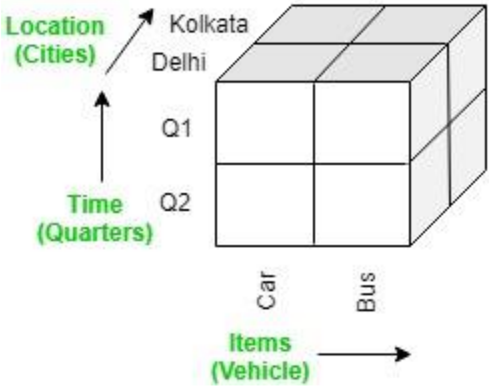


2. **Roll up:** It is just opposite of the drill-down operation. It performs aggregation on the OLAP cube. It can be done by:
   - Climbing up in the concept hierarchy
   - Reducing the dimensions

In the cube given in the overview section, the roll-up operation is performed by climbing up in the concept hierarchy of *Location* dimension (City -> Country).

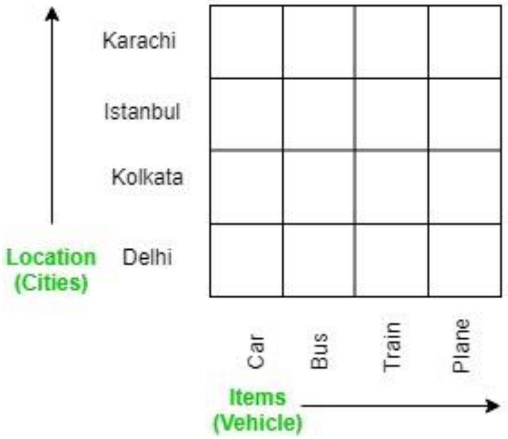


3. **Dice:** It selects a sub-cube from the OLAP cube by selecting two or more dimensions. In the cube given in the overview section, a sub-cube is selected by selecting following dimensions with criteria:
   - Location = “Delhi” or “Kolkata”
   - Time = “Q1” or “Q2”
   - Item = “Car” or “Bus”

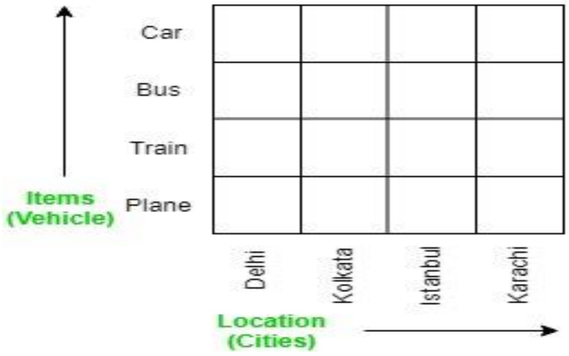


4. **Slice:** It selects a single dimension from the OLAP cube which results in a new sub-cube creation. In the cube given in the overview section, Slice is performed on the dimension

Time = “Q1”.



5. **Pivot:** It is also known as *rotation* operation as it rotates the current view to get a new view of the representation. In the sub-cube obtained after the slice operation, performing pivot operation gives a new view of it.



GUIDELINES FOR OLAP IMPLEMENTATION

i) Vision: the users develop a clear vision for the OLAP system

ii) Senior Management Support: OLAP project should be support senior managers.

iii) Selecting an OLAP Tool: OLAP team should familiar themselves with the ROLAP and MOLAP tools

iv) Corporate Strategy: OLAP strategy should fit with the enterprise strategy

v) Focus on the users: OLAP project should be focused on the users

vi) Joint Management: OLAP must managed by both IT and business professionals

vii) Review and adapt: reviews of project must be require to ensure the project.

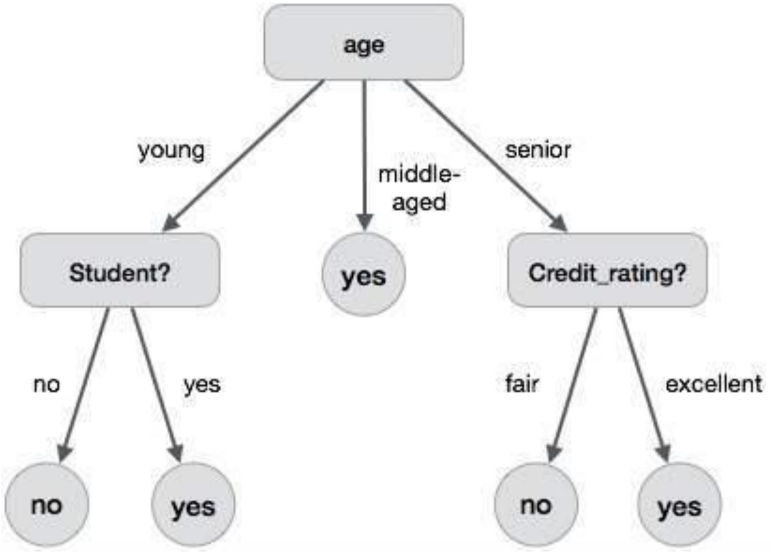
UNIT – III

CLASSIFICATION: INTRODUCTION:

Classification is a data mining technique that predicts categorical class labels while prediction models continuous-valued functions. Many classification and prediction methods have been proposed by researchers in machine learning, expert systems, statistics, and neurobiology. Each tuple or record is assumed to belong to a predefined class as determined by one of the attributes, called the class label attribute. The data records or tuples analyzed to build the model collectively form the training data set. The individual tuples or records making up the training set are referred to as training samples and are randomly selected from the sample population. Since the class label(categorical attribute) of each training sample is provided, this step is also known as **supervised learning** (i.e., the learning of the model is "supervised" in that it is told to which each training sample belongs). It contrasts with **unsupervised learning** (or clustering), in which the class label of each training sample is unknown, and the number or set of classes to be learned may be known in advance. Typically, the learned model is represented in the form of classification rules, decision trees, or statistical or mathematical formulae.

DECISION TREE:

ecision tree induction is the method of learning the decision trees from the training set. The training set consists of attributes and class labels. Applications of decision tree induction include astronomy, financial analysis, medical diagnosis, manufacturing, and production. A decision tree is a flowchart tree-like structure that is made from training set tuples. The dataset is broken down into smaller subsets and is present in the form of nodes of a tree. The tree structure has a root node, internal nodes or decision nodes, leaf node, and branches. The root node is the topmost node. It represents the best attribute selected for classification. Internal nodes of the decision nodes represent a test of an attribute of the dataset leaf node or terminal node which represents the classification or decision label. The branches show the outcome of the test performed. Some decision trees only have *binary nodes*, that means exactly two branches of a node, while some decision trees are non-binary.



OVER FITTING AND PRUNING:

In decision trees, over-fitting occurs when the tree is designed so as to perfectly fit all samples in the training data set. Thus it ends up with branches with strict rules of sparse data. Thus this effects the accuracy when predicting samples that are not part of the training set.

One of the methods used to address over-fitting in decision tree is called **pruning** which is done after the initial training is complete. In pruning, you trim off the branches of the tree, i.e., remove the decision nodes starting from the leaf node such that the overall accuracy is not disturbed. This is done by segregating the actual training set into two sets: training data set, D and validation data set, V. Prepare the decision tree using the segregated training data set, D. Then continue trimming the tree accordingly to optimize the accuracy of the validation data set, V.

Pruning is a technique in machine learning and search algorithms that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.

DT RULES:

Rules provide **model transparency**, a window on the inner workings of the model. Rules show the basis for the model's predictions. Oracle Data Mining supports a high level of model transparency. While some algorithms provide rules, *all* algorithms provide **model details**. Confidence and support are properties of rules. These statistical measures can be used to rank the rules and hence the predictions.

Support: The number of records in the training data set that satisfy the rule.

Confidence: The likelihood of the predicted outcome, given that the rule has been satisfied.

NAIVE BAYES METHOD:

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

$$\underset{\text{Posterior Probability}}{P(c\mid x)} = \frac{\overset{\text{Likelihood}}{P(x\mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c\mid \mathrm{X}) = P(x_1\mid c)\times P(x_2\mid c)\times\cdots\times P(x_n\mid c)\times P(c)$$

Above,

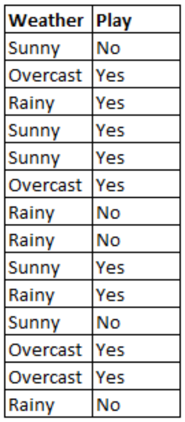
- *P*(*c*|*x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- *P*(*c*) is the prior probability of *class.*
- *P*(*x*|*c*) is the likelihood which is the probability of *predictor* given *class.*
- *P*(*x*) is the prior probability of *predictor*.

Naive Bayes algorithm

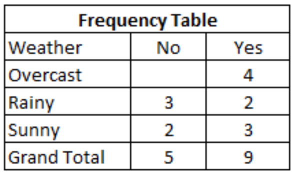
Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let’s follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

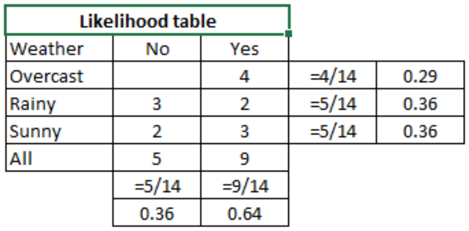
Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.



Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No



Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9



Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

Pros and Cons of Naive Bayes

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Applications of Naive Bayes Algorithms

- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

ESTIMATION PREDICTIVE ACCURACY OF CLASSIFICATION METHODS:

The accuracy of the classification method is the ability of the method to correctly determine the class of randomly selected data instance. Accuracy measured using a number of metrics these include **sensitivity, specificity, precision and accuracy.** The methods for estimating errors include **hold-out, random subsampling, k-fold cross validation and leave one out.** A confusion matrix tell us how many got misclassified but also what misclassified occurred

Let us assume, Total of object T, correctly classified C,

$$\textbf{ERROR RATE}=C/T$$

. For example,

Predicted Class	True Class		
	I	II	III
I	8	1	1
II	2	9	2
III	0	0	7

False Positive (FP), ClassI=2, ClassII=4 and ClassIII=0

False Negative (FN), ClassI=2, ClassII=1 and ClassIII=3

$$\textbf{Sensitivity}=TP/TP+FN$$

$$\textbf{Specificity}= TN/TN+FP$$

i) Hold-out Method (Test Sample Method):

The **holdout method** is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The function approximator fits a function using the training set only. Then the function approximator is asked to predict the output values for the data in the testing set (it has never seen these output values before). The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model.

The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

ii) Random sub-sampling method:

Random subsampling, which is also known as Monte Carlo crossvalidation, as multiple holdout or as repeated evaluation set , is based on randomly splitting the data into subsets, whereby the size of the subsets is defined by the user. The random partitioning of the data can be repeated arbitrarily often. In contrast to a full crossvalidation procedure, random subsampling has been shown to be asymptotically consistent resulting in more pessimistic predictions of the test data compared with crossvalidation. The predictions of the test data give a realistic estimation of the predictions of external validation data

iii) k-Fold cross validation method

K-fold cross validation is one way to improve over the holdout method. The data set is divided into *k* subsets, and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then the average error across all *k* trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set *k-1* times. The variance of the resulting estimate is reduced as *k* is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch *k* times, which means it takes *k* times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set *k* different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

iv) Leave one out method

Leave-one-out cross validation is K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set. That means that N separate times, the function approximator is trained on all the data except for one point and a prediction is made for that point. As before the average error is computed and used to evaluate the model. The evaluation given by leave-one-out cross validation error (LOO-XVE) is good, but at first pass it seems very expensive to compute. Fortunately, locally weighted learners can make LOO predictions just as easily as they make regular predictions. That means computing the LOO-XVE takes no more time than computing the residual error and it is a much better way to evaluate models. We will see shortly that Vizier relies heavily on LOO-XVE to choose its metacodes.

v) bootstrap method

Bootstrapping is any test or metric that relies on random sampling with replacement. Bootstrapping allows assigning measures of accuracy (defined in terms of bias, variance, confidence intervals, prediction error or some other such measure) to sample estimates. This technique allows estimation of the sampling distribution of almost any statistic using random sampling methods. Generally, it falls in the broader class of resampling methods. Bootstrapping is the practice of estimating properties of an estimator (such as its variance) by measuring those properties when sampling from an approximating distribution. One standard choice for an approximating distribution is the empirical distribution function of the observed data.

OTHER EVALUATION CRITERIA FOR CLASSIFICATION METHOD:

Estimating predictive accuracy of classification methods and some techniques for improving the accuracy.

1) Speed:

Speed not just the time or computation cost of constructing a model, it also includes time required to learn to use the model.

2) Robustness

Data errors are common, when the data is being collected from number of sources and errors may remain even after data cleaning.

3) Scalability

Many Data mining methods were designed for small data sets. Many have been modified to deal with large problems.

4) Interpretability

The end user is able to understand and gain insight from the results produced by the method.

5) Goodness of the Model

For model to be effective, it needs to fit the problem that is being solved.

CLASSIFICATION SOFTWARE:

- 11Ants Model Builder, upgrades Microsoft Excel into a powerful, simple to use data mining / predictive analytics tool, with regression, classification and rank likelihood.
- ADAPA® from Zementis, a framework for deployment, integration, and execution of various predictive algorithms, including neural networks, support vector machines, regression models, and decision trees.
- Affinium Model Suite, includes linear regression, logistic regression, CHAID, neural networks, and genetic algorithms.
- Dataladder ProductMatch, uses best in class Semantic Technology to recognize and transform unstructured and unpredictable data.
- KINOsuite PR, extracts rules from trained neural networks.
- Knowledge Studio, featuring multiple data mining models in a visual, easy-to-use interface.
- MLF: machine learning framework for Mathematica, the multi-method system for creating understandable computational models from data.
- Oracle Data Mining, embeds data-mining functionality into the Oracle database, for making classifications, predictions, and associations.
- Polyanalyst, features multiple classification algorithms: Decision Trees, Fuzzy Logic, and Memory Based reasoning.
- Predictive Dynamix Data Mining Suite integrates statistical, graphical, and ROC analysis with neural network, clustering, and fuzzy models.
- PredictionWorks, includes decision tree (gini, entropy, C4.5), logistic regression, k nearest-neighbor, naive bayes and linear regression. Free test over the web!
- Previa Classpad, provides an interactive environment for classification using neural networks, decision trees, and bayesian networks.
- perClass, easy-to-use Matlab toolbox for training pattern recognition algorithms and C library for real-time execution in custom applications (formerly PRSD Studio).
- prudsys DISCOVERER: non-linear decision trees (NDTs) and sparse grid methods for classification

UNIT – IV

CLUSTER ANALYSIS:

Clustering is an unsupervised Machine Learning-based Algorithm that comprises a group of data points into clusters so that the objects belong to the same group. Cluster is a group of objects that belongs to the same class. Clustering is the process of making a group of abstract objects into classes of similar objects. A cluster of data objects can be treated as one group. While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups. The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.

CLUSTERING METHODS

Clustering methods can be classified into the following categories −

- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

Partitioning Method

Suppose we are given a database of ‘n’ objects and the partitioning method constructs ‘k’ partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements −

- Each group contains at least one object.
- Each object must belong to exactly one group.
- For a given number of partitions (say k), the partitioning method will create an initial partitioning.

- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

Hierarchical Methods

This method creates a hierarchical decomposition of the given set of data objects. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here −

- Agglomerative Approach
- Divisive Approach

i) Agglomerative Approach

This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.

ii) Divisive Approach

This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid, i.e., once a merging or splitting is done, it can never be undone.

Approaches to Improve Quality of Hierarchical Clustering

Here are the two approaches that are used to improve the quality of hierarchical clustering −

- Perform careful analysis of object linkages at each hierarchical partitioning.
- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

Density-based Method

This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

Grid-based Method

In this, the objects together form a grid. The object space is quantized into finite number of cells that form a grid structure.

Advantages

- The major advantage of this method is fast processing time.
- It is dependent only on the number of cells in each dimension in the quantized space.

Model-based methods

In this method, a model is hypothesized for each cluster to find the best fit of data for a given model. This method locates the clusters by clustering the density function. It reflects spatial distribution of the data points.

This method also provides a way to automatically determine the number of clusters based on standard statistics, taking outlier or noise into account. It therefore yields robust clustering methods.

Constraint-based Method

In this method, the clustering is performed by the incorporation of user or application-oriented constraints. A constraint refers to the user expectation or the properties of desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. Constraints can be specified by the user or the application requirement.

PARTITIONING METHOD (K-MEAN) IN DATA MINING

Partitioning Method:

This clustering method classifies the information into multiple groups based on the characteristics and similarity of the data. Its the data analysts to specify the number of clusters that has to be generated for the clustering methods.

In the partitioning method when database(D) that contains multiple(N) objects then the partitioning method constructs user-specified(K) partitions of the data in which each partition represents a cluster and a particular region. There are many algorithms that come under partitioning method some of the popular ones are K-Mean, PAM(K-Mediods), CLARA algorithm (Clustering Large Applications) etc.

K-Mean (A centroid based Technique):

The K means algorithm takes the input parameter K from the user and partitions the dataset containing N objects into K clusters so that resulting similarity among the data objects inside the group (intracluster) is high but the similarity of data objects with the data objects from outside the cluster is low (intercluster). The similarity of the cluster is determined with respect to the mean value of the cluster.

It is a type of square error algorithm. At the start randomly k objects from the dataset are chosen in which each of the objects represents a cluster mean(centre). For the rest of the data objects, they are assigned to the nearest cluster based on their distance from the cluster mean. The new mean of each of the cluster is then calculated with the added data objects.

Algorithm: K mean:

Input:

K: The number of clusters in which the dataset has to be divided

D: A dataset containing N number of objects

Output:

A dataset of K clusters

Method:

1. Randomly assign K objects from the dataset(D) as cluster centres(C)
2. (Re) Assign each object to which object is most similar based upon mean values.
3. Update Cluster means, i.e., Recalculate the mean of each cluster with the updated values.
4. Repeat Step 4 until no change occurs.

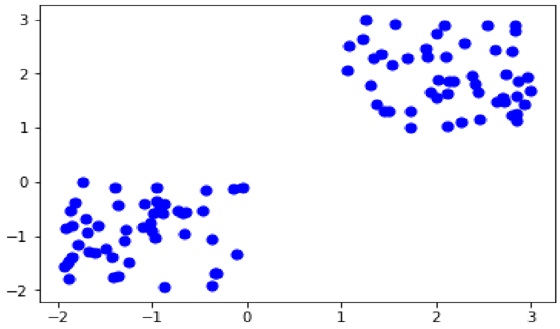


Figure – K-mean Clustering

Example: Suppose we want to group the visitors to a website using just their age as follows:

16, 16, 17, 20, 20, 21, 21, 22, 23, 29, 36, 41, 42, 43, 44, 45, 61, 62, 66

Initial Cluster:

K=2

Centroid(C1) = 16 [16]

Centroid(C2) = 22 [22]

Iteration-1:

C1 = 16.33 [16, 16, 17]

C2 = 37.25 [20, 20, 21, 21, 22, 23, 29, 36, 41, 42, 43, 44, 45, 61, 62, 66]

Iteration-2:

C1 = 19.55 [16, 16, 17, 20, 20, 21, 21, 22, 23]

C2 = 46.90 [29, 36, 41, 42, 43, 44, 45, 61, 62, 66]

Iteration-3:

C1 = 20.50 [16, 16, 17, 20, 20, 21, 21, 22, 23, 29]

C2 = 48.89 [36, 41, 42, 43, 44, 45, 61, 62, 66]

Iteration-4:

C1 = 20.50 [16, 16, 17, 20, 20, 21, 21, 22, 23, 29]

C2 = 48.89 [36, 41, 42, 43, 44, 45, 61, 62, 66]

No change Between Iteration 3 and 4, so we stop. Therefore we get the clusters **(16-29)** and **(36-66)** as 2 clusters we get using K Mean Algorithm.

HIERARCHICAL CLUSTERING (AGGLOMERATIVE AND DIVISIVE CLUSTERING)

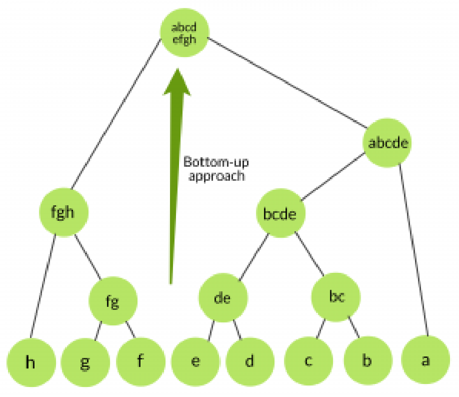
In data mining and statistics, hierarchical clustering analysis is a method of cluster analysis which seeks to build a hierarchy of clusters i.e. tree type structure based on the hierarchy.

Basically, there are two types of hierarchical cluster analysis strategies –

1. **Agglomerative Clustering:** Also known as bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering. This clustering algorithm does not require us to prespecify the number of clusters. Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

Algorithm :

```
given a dataset ($d_1$, $d_2$, $d_3$, ....$d_N$) of size N
# compute the distance matrix
for i=1 to N:
    # as the distance matrix is symmetric about
    # the primary diagonal so we compute only lower
    # part of the primary diagonal
    for j=1 to i:
        dis_mat[i][j] = distance[$d_i$, $d_j$]
each data point is a singleton cluster
repeat
    merge the two cluster having minimum distance
    update the distance matrix
untill only a single cluster remains
```



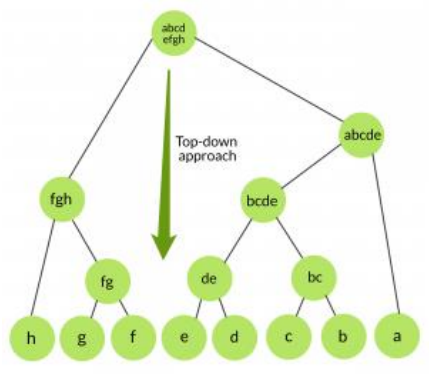
Hierarchical agglomerative clustering

2. **Divisive clustering :** Also known as top-down approach. This algorithm also does not require to prespecify the number of clusters. Top-down clustering requires a method for

splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been splitted into singleton cluster.

Algorithm :

```
given a dataset ($d_1$, $d_2$, $d_3$, ....$d_N$) of size N
at the top we have all data in one cluster
the cluster is split using a flat clustering method eg. K-Means etc
repeat
choose the best cluster among all the clusters to split
split that cluster by the flat clustering algorithm
untill each data is in its own singleton cluster
```



Hierarchical divisive clustering

Hierarchical Agglomerative *vs* Divisive clustering –

- Divisive clustering is more *complex* as compared to agglomerative clustering, as in case of divisive clustering we need a flat clustering method as “subroutine” to split each cluster until we have each data having its own singleton cluster.
- Divisive clustering is more *efficient* if we do not generate a complete hierarchy all the way down to individual data leaves. Time complexity of a naive agglomerative clustering is **O(n^3)** because we exhaustively scan the N x N matrix dist_mat for the lowest distance in each of N-1 iterations. Using priority queue data structure we can reduce this complexity to **O(n^2logn)**. By using some more optimizations it can be brought down to **O(n^2)**. Whereas for divisive clustering given a fixed number of top levels, using an efficient flat algorithm like K-Means, divisive algorithms are linear in the number of patterns and clusters.
- Divisive algorithm is also more *accurate*. Agglomerative clustering makes decisions by considering the local patterns or neighbor points without initially taking into account the global distribution of data. These early decisions cannot be undone. whereas divisive clustering takes into consideration the global distribution of data when making top-level partitioning decisions.

DENSITY BASED CLUSTERING:

Clustering analysis or simply Clustering is basically an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense. It comprises of many different methods based on different evolution. Fundamentally, all clustering methods use the same approach i.e. first we calculate similarities and then we use it to cluster the data points into groups or batches. Here we will focus on **Density-based spatial clustering of applications with noise** (DBSCAN) clustering method.

Clusters are dense regions in the data space, separated by regions of the lower density of points. The ***DBSCAN algorithm*** is based on this intuitive notion of "clusters" and "noise". The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

Why DBSCAN ?

Partitioning methods (K-means, PAM clustering) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.

Real life data may contain irregularities, like –

i) Clusters can be of arbitrary shape such as those shown in the figure below.

ii) Data may contain noise.

DBSCAN algorithm requires two parameters –

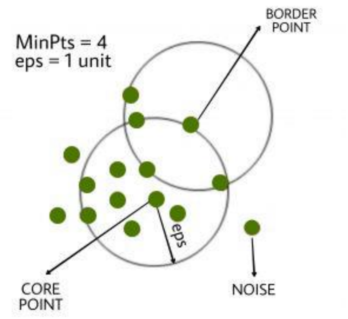
1. **eps** : It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered as neighbors. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the ***k-distance graph***.
2. **MinPts**: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

In this algorithm, we have 3 types of data points.

***Core Point**: A point is a core point if it has more than MinPts points within eps.*

***Border Point**: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.*

***Noise or outlier**: A point which is not a core point or border point.*

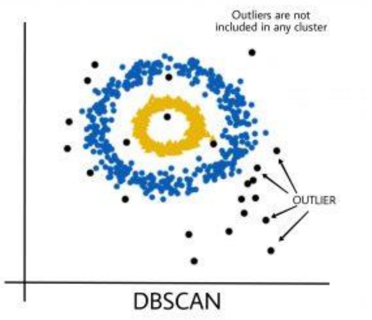


DBSCAN algorithm can be abstracted in the following steps –

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density connected points and assign them to the same cluster as the core point.
4. A point *a* and *b* are said to be density connected if there exist a point *c* which has a sufficient number of points in its neighbors and both the points *a* and *b* are within the *eps distance*. This is a chaining process. So, if *b* is neighbor of *c*, *c* is neighbor of *d*, *d* is neighbor of *e*, which in turn is neighbor of *a* implies that *b* is neighbor of *a*.
5. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

Disadvantage Of K-MEANS:

1. K-Means forms spherical clusters only. This algorithm fails when data is not spherical (i.e. same variance in all directions).
2. K-Means algorithm is sensitive towards outlier. Outliers can skew the clusters in K-Means in very large extent.



3. K-Means algorithm requires one to specify the number of clusters a priory etc.

QUALITY AND VALIDITY OF CLUSTER ANALYSIS METHODS

1. **RELATIVE CLUSTERING VALIDATION**, which evaluates the clustering structure by varying different parameter values for the same algorithm (e.g.,: varying the number of clusters k). It's generally used for determining the optimal number of clusters.
2. **EXTERNAL CLUSTERING VALIDATION**, which consists in comparing the results of a cluster analysis to an externally known result, such as externally provided class labels. Since we know the "true" cluster number in advance, this approach is mainly used for selecting the right clustering algorithm for a specific dataset.
3. **INTERNAL CLUSTERING VALIDATION**, which use the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information. It can be also used for estimating the number of clusters and the appropriate clustering algorithm without any external data. Internal validation measures reflect often **the compactness, the connectedness and the separation** of the cluster partitions.

i) **Compactness or cluster cohesion:** Measures how close are the objects within the same cluster. A lower within-cluster variation is an indicator of a good compactness (i.e., a good clustering). The different indices for evaluating the compactness of clusters are base on distance measures such as the cluster-wise within average/median distances between observations.

ii) **Separation:** Measures how well-separated a cluster is from other clusters. The indices used as separation measures include:

- distances between cluster centers
- the pairwise minimum distances between objects in different clusters

iii) **Connectivity:** corresponds to what extent items are placed in the same cluster as their nearest neighbors in the data space. The connectivity has a value between 0 and infinity and should be minimized.

4. **CLUSTERING STABILITY VALIDATION**, which is a special version of internal validation. It evaluates the consistency of a clustering result by comparing it with the clusters obtained after each column is removed, one at a time.

CLUSTER ANALYSIS SOFTWARE

- **BayesiaLab,** includes Bayesian classification algorithms for data segmentation and uses Bayesian networks to automatically cluster the variables.
- **ClustanGraphics3**, hierarchical cluster analysis from the top, with powerful graphics
- **CMSR Data Miner,** built for business data with database focus, incorporating rule-engine, neural network, neural clustering (SOM), decision tree, hotspot drill-down, cross table deviation analysis, cross-sell analysis, visualization/charts, and more.
- **CViz Cluster Visualization**, for analyzing large high-dimensional datasets; provides full-motion cluster visualization.
- **IBM SPSS Modeler**, includes Kohonen, Two Step, K-Means clustering algorithms
- **Autoclass C,** an unsupervised Bayesian classification system from NASA, available for Unix and Windows
- **CLUTO**, provides a set of partitional clustering algorithms that treat the clustering problem as an optimization process.
- **Databionic ESOM Tools**, a suite of programs for clustering, visualization, and classification with Emergent Self-Organizing Maps (ESOM).
- **David Dowe** Mixture Modeling page for modeling statistical distribution by a mixture (or weighted sum) of other distributions

UNIT- V

WEB DATA MINING:

Web Mining is the process of Data Mining techniques to automatically discover and extract information from Web documents and services. The main purpose of web mining is discovering useful information from the World-Wide Web and its usage patterns.

Applications of Web Mining:

1. Web mining helps to improve the power of web search engine by classifying the web documents and identifying the web pages.
2. It is used for Web Searching e.g., Google, Yahoo etc and Vertical Searching e.g., FatLens, Become etc.
3. Web mining is used to predict user behavior.
4. Web mining is very useful of a particular Website and e-service e.g., landing page optimization.

Web mining can be broadly divided into three different types of techniques of mining: Web Content Mining, Web Structure Mining, and Web Usage Mining.

1. **Web Content Mining:**

   Web content mining is the application of extracting useful information from the content of the web documents. Web content consist of several types of data – text, image, audio, video etc. Content data is the group of facts that a web page is designed. It can provide effective and interesting patterns about user needs. Text documents are related to text mining, machine learning and natural language processing. This mining is also known as text mining. This type of mining performs scanning and mining of the text, images and groups of web pages according to the content of the input.
2. **Web Structure Mining:**

   Web structure mining is the application of discovering structure information from the web. The structure of the web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages. Structure mining basically shows the structured summary of a particular website. It identifies relationship between web pages linked by information or direct link connection. To determine the connection between two commercial websites, Web structure mining can be very useful.
3. **Web Usage Mining:**

Web usage mining is the application of identifying or discovering interesting usage patterns from large data sets. And these patterns enable you to understand the user behaviors or something like that. In web usage mining, user access data on the web and collect data in form of logs. So, Web usage mining is also called log mining.

WEB TERMINOLOGY AND CHARACTERISTICS

The web is seen as having two-tier architecture. The first tier is a Web server that serves the information to the client machine and the second tier is the client that displays that information to the user. This architecture is supported by 3 web standards, namely **HTML** (Hyper-Text Markup Language) for defining the web document content, **URL** (Uniform Resource Locator) for naming and identifying remote information resources in the global web world, **HTTP** (Hyper-Text Transfer Protocol) for managing the transfer of information from the server to the client.

Web terminology based on *World Wide Web* (**WWW**), is the set of all the nodes which are interconnected by hypertext links. A *links* expresses one or more relationships between two or more resources. A *Web page* is a collection of information, consisting of one or more web resources, intended to be rendered simultaneously, and identified by single URL. A *browser* is a program which allows a person to view the contents of web pages.

1.Graph Terminology

A *directed graph* as a set of nodes denoted by V and Edges denoted by E.

LOCALITY AND HIERARCHY IN THE WEB

A website of any enterprise usually has the homepage as the homepage as the root of the tree likes in any hierarchical structure and as go down the tree user find more and more detailed information about some aspects of the enterprise. Web structure has strong locality feature to the extent that almost two thirds of all links are to sites within the enterprise domain. The one third links to sites outside the enterprise domain tend to have a higher percentage of broken links. About half of the local links have been found to be within the directory of the Web page and quarter of the local links refer to pages higher up or lower down the tree hierarchy. It possible to classify web pages into several types,

- **Homepage or Head page-** These page representing an entry point for an enterprise website or section within enterprise or an individual web page.
- **Index page –** These pages that assist the user to navigate through the enterprise's websites.

- **Reference page-** These page that provide some basic information that is used by a number of other pages.
- **Content Page-** These pages that only provide content and have little role in assisting a user's navigation.

A number of simple principles have been developed to help design the structure and content of a website. Three basic principles are,

1) Relevant linkage principle: It is assumed that links from a page point to other relevant resources. Links are often assumed to reflect the judgment of the page creator. By providing a link to another page it is assumed that the creator is making recommendation for the other relevant page.

2) Topical unity principle: It is assumed that web pages that are co-cited are related.

3) Lexical affinity principle: It is assumed that the text and the links within a page are relevant to each other. It is assumed that the text on page on a page has been chosen carefully by the creator to be related to a theme.

WEB CONTENT MINING

Web content mining is related but different from data mining and text mining. It is related to data mining because many data mining techniques can be applied in Web content mining. It is related to text mining because much of the web contents are texts. However, it is also quite different from data mining because Web data are mainly semi-structured and/or unstructured, while data mining deals primarily with structured data. Web content mining is also different from text mining because of the semi-structure nature of the Web, while text mining focuses on unstructured texts. Web content mining thus requires creative applications of data mining and/or text mining techniques and also its own unique approaches.

- **Data/information extraction**: Our focus will be on extraction of structured data from Web pages, such as products and search results. Extracting such data allows one to provide services. Two main types of techniques, machine learning and automatic extraction are covered.
- **Web information integration and schema matching**: Although the Web contains a huge amount of data, each web site (or even page) represents similar information differently. How to identify or match semantically similar data is a very important problem with many practical applications. Some existing techniques and problems are examined.

- **Opinion extraction from online sources**: There are many online opinion sources, e.g., customer reviews of products, forums, blogs and chat rooms. Mining opinions (especially consumer opinions) is of great importance for marketing intelligence and product benchmarking. We will introduce a few tasks and techniques to mine such sources.
- **Knowledge synthesis**: Concept hierarchies or ontology are useful in many applications. However, generating them manually is very time consuming. A few existing methods that explores the information redundancy of the Web will be presented. The main application is to synthesize and organize the pieces of information on the Web to give the user a coherent picture of the topic domain..
- **Segmenting Web pages and detecting noise**: In many Web applications, one only wants the main content of the Web page without advertisements, navigation links, copyright notices. Automatically segmenting Web page to extract the main content of the pages is interesting problem. A number of interesting techniques have been proposed in the past few years.

i) Web Document Clustering

Many document clustering algorithms rely on off-line clustering of the entire document collection (e.g., Cutting et. al., 93; Silverstein and Pedersen, 97), but the Web search engines' collections are too large and fluid to allow off-line clustering. Therefore clustering has to be applied to the much smaller set of documents returned in response to a query. Because the search engines service millions of queries per day, free of charge, the CPU cycles and memory dedicated to each individual query are severely curtailed. Thus, clustering has to be performed on a separate machine, which receives search engine results as input, creates clusters and presents them to the user.clustering methods:

1. Relevance: The method ought to produce clusters that group documents relevant to the user's query separately from irrelevant ones.

2. Browsable Summaries: The user needs to determine at a glance whether a cluster's contents are of interest. We do not want to replace sifting through ranked lists with sifting through clusters. Therefore the method has to provide concise and accurate descriptions of the clusters.

3. Overlap: Since documents have multiple topics, it is important to avoid confining each document to only one cluster (Hearst, 98).

4. Snippet-tolerance: The method ought to produce high quality clusters even when it only has access to the snippets returned by the search engines, as most users are unwilling to wait while the system downloads the original documents off the Web.

5. Speed: A very patient user might sift through 100 documents in a ranked list presentation. We want clustering to allow the user to browse through at least an order of magnitude more documents. Therefore the clustering method ought to be able to cluster up to one thousand snippets in a few seconds. For the impatient user, each second counts.

6. Incrementality: To save time, the method should start to process each snippet as soon as it is received over the Web.

Suffix Tree Clustering (STC) - a novel, incremental, O(n)1 time algorithm designed to meet these requirements. STC does not treat a document as a set of words but rather as a string, making use of proximity information between words. STC relies on a suffix tree to efficiently identify sets of documents that share common phrases and uses this information to create clusters and to succinctly summarize their contents for users. Figure 1 shows the output of such a search using our demonstration system. We provide preliminary experimental evidence that STC satisfies the speed, snippet tolerance, and relevance requirements, and that it benefits from creating overlapping clusters. We believe the shared phrases of a cluster provide an informative way of summarizing its contents, but a user study to validate this belief is an area for future work.

ii) FingerPrinting:

An approach for computing a large number of documents is based on the idea of fingerprinting documents. A document may be divided in all possible substrings of length. These substrings are called *Shingles.* Based on the shingles user can define resemblance(X,Y) and containment C(X,Y) between two documents X and Y follows. We assume S(X) and S(Y) to be a set of shingles for document X and Y respectively.

$$\mathbf{R(X,Y)=\{S(X) \cap S(Y)\} /\{S(X) \cup S(Y)\}}$$

$$\mathbf{C(X,Y)=\{S(X) \cap S(Y)\} /\{S(X)\}}$$

An algorithm like the following may now be used to find similar documents,

1.Collect all the documents that one wishes to compare

2.Choose a suitable shingle width and compute the shingles for each document

3.Compare the shingles for each pair of documents

4.Identify those documents that are similar

The Web is very large and this algorithm requires enormous storage to store the shingles and very long processing time to finish pairwise comparison for say even 100 million documents. This approach is sometimes called ***full fingerprinting.***

WEB USAGE MINING

Web Usage Mining focuses on techniques that could predict the behavior of users while they are interacting with the WWW. Web usage mining, discover user navigation patterns from web data, tries to discovery the useful information from the secondary data derived from the interactions of the users while surfing on the Web. Web usage mining collects the data from Web log records to discover user access patterns of web pages. There are several available research projects and commercial tools that analyze those patterns for different purposes. The insight knowledge could be utilized in personalization, system improvement, site modification, business intelligence and usage characterization.

The only information left behind by many users visiting a Web site is the path through the pages they have accessed. Most of the Web information retrieval tools only use the textual information, while they ignore the link information that could be very valuable. In general, there are mainly four kinds of data mining techniques applied to the web mining domain to discover the user navigation pattern:

- Association Rule mining
- Sequential pattern
- Clustering
- Classification

Association rules

Association rule is the most basic rule of data mining methods which is used more than other methods in the web usage mining. This method enables web site for more efficient organization of content or provide recommendations for effective cross-selling product. These rules are statements in the form X => Y where (X) and (Y) are the set of available items in a series of transactions. The rule of X => Y states that, transactions that contain items in X, may also include items in Y. association rules in the web usage mining are used to find relationships between pages that frequently appear next to one another in user sessions. For example, a rule can be obtained in the following format:

A.html, B.html => C.html

This rule shows, if user observes A and B pages, most likely will observe page C at the same meeting. A common algorithm to extract association rules is Apriori algorithm. Some criteria are presented to assess the rules extracted from the web usage data .Also, a method is presented by using association rules and fuzzy logic to extract data using the web fuzzy association rules.

Sequential patterns

Sequential patterns are used to discover the subsequence in the large volume of sequential data .In web usage mining , sequential patterns are used to find user navigation patterns which appear frequently at meetings. A sequential pattern is often as follows: 70% of users who have first observed the page A.html and then page B.html, have observed page C,html in the same session , too. The sequential patterns may seem to association rules. Actually, algorithms that are used to extract association rules, can also be used to generate sequential patterns. But the sequential patterns are included the time, it means that the sequence of events occurred is defined in sequential patterns.

Clustering

Clustering techniques diagnose groups of similar items among high volumes of data. This is done based on distance functions which measures the degree of similarity between different items. Clustering in web usage mining is used for grouping similar meetings. What is important in this type of search, is contrast of the user group and individual group. Two types of interesting clustering can be found in this area: 1- user clustering, 2- page clustering. Clustering of user records is usually used to analyze the tasks in web mining and web analytics.

WEB STRUCTURE MINING

Web structure mining is to generate structural summary about the Web site and Web page. Technically, Web content mining mainly focuses on the structure of inner-document, while Web structure mining tries to discover the link structure of the hyperlinks at the inter-document level. Based on the topology of the hyperlinks, Web structure mining will categorize the Web pages and generate the information, such as the similarity and relationship between different Web sites. Web structure mining can also have another direction -- discovering the structure of Web document itself. This type of structure mining can be used to reveal the structure (schema) of Web pages, this would be good for navigation purpose and make it possible to compare/integrate Web page schemes. This type of structure mining will facilitate introducing database techniques for accessing information in Web pages by providing a reference schema.

i) The HITS Algorithm

Hyperlink Induced Topic Search (HITS) Algorithm is a Link Analysis Algorithm that rates webpages, developed by Jon Kleinberg. This algorithm is used to the web link-structures to discover and rank the webpages relevant for a particular search. HITS uses hubs and authorities to define a recursive relationship between webpages. Before understanding the HITS Algorithm, we first need to know about Hubs and Authorities.

- Given a query to a Search Engine, the set of highly relevant web pages are called **Roots**. They are potential **Authorities**.
- Pages which are not very relevant but point to pages in the Root are called **Hubs**. Thus, an Authority is a page that many hubs link to whereas a Hub is a page that links to many authorities.

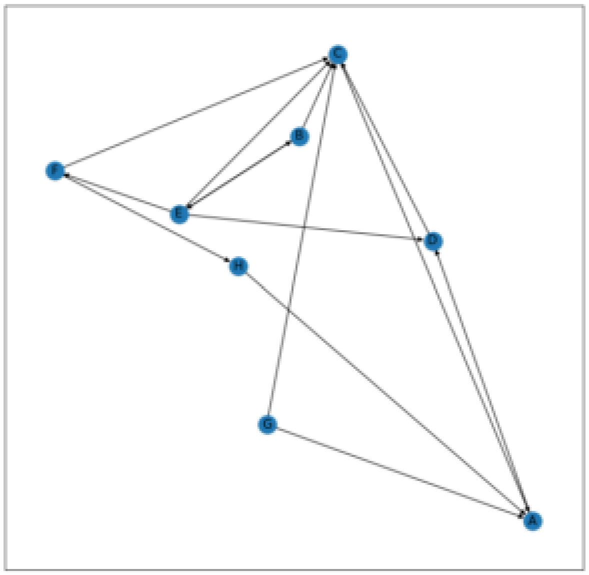
Algorithm –

Let number of iterations be k.

-> *Each node is assigned a Hub score = 1 and an Authority score = 1.*

-> *Repeat k times:*

- ***Hub update :** Each node's Hub score = (Authority score of each node it points to).*
- ***Authority update :** Each node's Authority score = (Hub score of each node pointing to it).*
- *Normalize the scores by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores. (optional)*
- Now, let's see how to implement this algorithm using Networxx Module.



On running HITS Algorithm with (without Normalization),

Initially,

Hub Scores:	Authority Scores:
A -> 1	A -> 1
B -> 1	B -> 1
C -> 1	C -> 1
D -> 1	D -> 1
E -> 1	E -> 1
F -> 1	F -> 1
G -> 1	G -> 1
H -> 1	H -> 1

After 1st iteration,

Hub Scores:	Authority Scores:
A -> 1	A -> 3
B -> 2	B -> 2
C -> 1	C -> 4
D -> 2	D -> 2
E -> 4	E -> 1
F -> 1	F -> 1
G -> 2	G -> 0
H -> 1	H -> 1

After 2nd iteration,

Hub Scores:	Authority Scores:
A -> 2	A -> 4
B -> 5	B -> 6
C -> 3	C -> 7
D -> 6	D -> 5
E -> 9	E -> 2
F -> 1	F -> 4
G -> 7	G -> 0
H -> 3	H -> 1

After 3rd iteration,

Hub Scores:	Authority Scores:
A -> 5	A -> 13
B -> 9	B -> 15
C -> 4	C -> 27
D -> 13	D -> 11
E -> 22	E -> 5
F -> 1	F -> 9
G -> 11	G -> 0
H -> 4	H -> 3

ii) The Problems with the HITS Algorithm

1. Hubs and authorities: A clear-cut distinction between hubs and authorities may not be appropriate since may sites are hubs as well as authorities.

2. Topic drift: Certain arrangements of tightly connected documents, perhaps due to mutually reinforcing relationships between hosts, can dominate the HITS computation.

3. Automatically generated links: Some of the links are computer generate every page in my school has link to school homepage and to copyright page.

4. Non-relevant documents: Some quires can return non-relevant documents in the highly ranked quires and this can then lead to erroneous results from the HITS algorithm.

5. Efficiency: The real time performance of the algorithm is not good given steps that involve finding sites that are pointed to by pages in the root pages.

WEB MINING SOFTWARE

1. R

R is a language or a free environment for statistical computing and graphics. It has been made accessible from scripting languages like Python, Ruby, Perl, etc.

2. Octoparse

Octoparse is a simple but powerful web data mining tool that automates web data extraction. It allows you to create highly accurate extraction rules. (You know I will definitely mention our tool.) Crawlers run in Octoparse are determined by the configured rule. The extraction rule would tell Octoparse: which website is to go to; where the data is you plan to crawl; what kind of data you want, etc.

3. Oracle Data Mining (ODM)

Oracle Data Mining is a data mining software by Oracle. Oracle Data Mining is implemented in the Oracle Database kernel, and mining models are first-class database objects. Oracle Data Mining processes use built-in features of Oracle Database to maximize scalability and make efficient use of system resources.

4. Tableau

Tableau offers a family of interactive data visualization products focused on business intelligence. Tableau allows instantaneous insight by transforming data into visually appealing, interactive visualizations called dashboards. This process takes only seconds or minutes rather than months or years and is achieved through the use of an easy-to-use, drag-and-drop interface.

5. Scrapy

Scrapy is an open-source framework for collecting data from websites. It is written in Python and you can write the rules to extract web data.

SEARCH ENGINES:

A **web search engine** or **Internet search engine** is a software system that is designed to carry out **web search** (**Internet search**), which means to search the World Wide Web in a systematic way for particular information specified in a textual web search query. The search results are generally presented in a line of results, often referred to as search engine results pages (SERPs). The information may be a mix of links to web pages, images, videos, infographics, articles, research papers, and other types of files. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained only by human editors, search engines also maintain real-time information by running an algorithm on a web crawler. Internet content that is not capable of being searched by a web search engine is generally described as the deep web.

SEARCH ENGINES FUNCTIONALITY

A search engine carries out a variety of tasks. Some of these tasks are similar to those described for CRISP, these include,

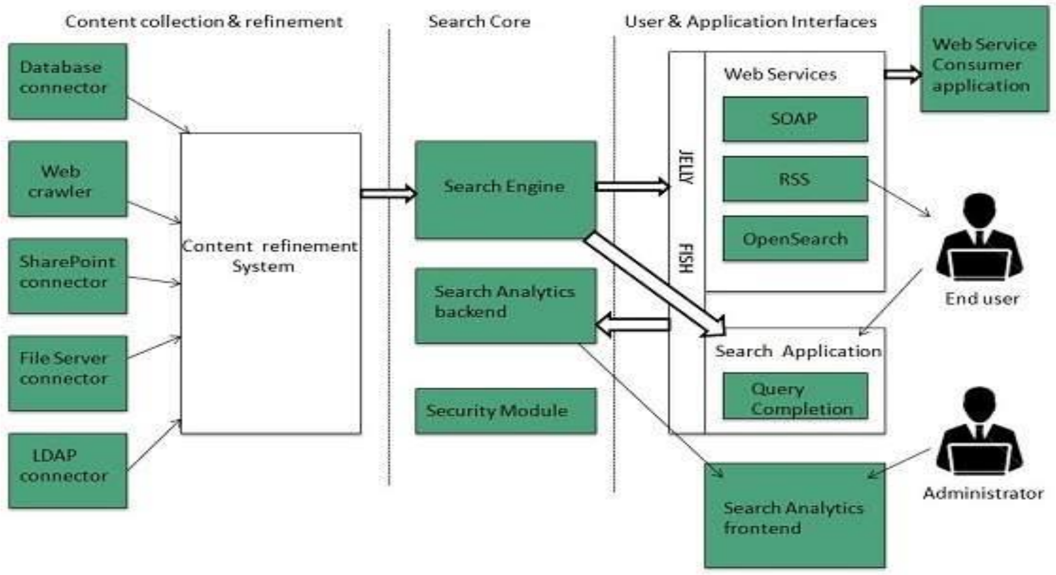
1. **Collecting Information**: A search engine would normally collected web page or information about them by web crawling
2. **Evaluating and categorizing information:** It may be necessary to categories information based on some ontology used by the search engine.
3. **Creating a database and creating indexes:** The information collected needs to be stored either in a database or some kind of file system.

4. **Computing ranks of the Web documents:** A variety of methods are being used to determine the rank of each page retrieved in response to user query.
5. **Checking quires and executing them:** Queries posed by users need to be checked.
6. **Presenting results:** The search engine must determine what results to present and how to display.
7. **Profiling the users**: To improve search performance to search engines carry out user profiling that deals with the way users use search engines.

SEARCH ENGINES ARCHITECTURE

Web crawler, database and the search interface are the major component of a search engine that actually makes search engine to work. Search engines make use of Boolean expression AND, OR, NOT to restrict and widen the results of a search. Following are the steps that are performed by the search engine:

- The search engine looks for the keyword in the index for predefined database instead of going directly to the web to search for the keyword.
- It then uses software to search for the information in the database. This software component is known as web crawler.
- Once web crawler finds the pages, the search engine then shows the relevant web pages as a result. These retrieved web pages generally include title of page, size of text portion, first several sentences etc.



1. Crawling – discovering information on web pages

Every search engine has this vital software component called the crawler, or spider or bots that go through (or “crawls”) each of your web pages storing its contents in the search engine database. A crawler can either scan up the new contents on the web page or also locate older data. The bots crawl all the contents on web pages; often several websites at a time, following each and every hyperlink that links to both internal and external pages until it can no longer find any more information.

2. Indexing – creation of index database

After contents of the web pages are crawled, they now have to be indexed based on the occurrences various keywords. This would increase the efficiency of the search engines in accurately fetching information corresponding to a particular query in a short span of time. Every query given by the user will consist of a few phrases or keywords. While indexing the contents on a web page, common articles such as “a”, “an” and “the” are avoided and the indexed information is stored in an organized manner.

Search engine designers develop search algorithms that search for contents by looking the match between the keywords entered by the user with those found within the web page content using the index. If there is a good match between these then, search engines consider it as one of the results.

3. Results – fetching of the relevant data

The different hyperlinks displayed after you search for a particular key phrase are the results. Every search engine has its own algorithm that sort and displays the most relevant data as results. So, you may not get the same website rankings for a single keyword on various search engines. As mentioned earlier, comparison for equality of keywords is done by the algorithms using the index.

The whole working of search engines is a complex process that depends on the algorithms developed. And, with each of the search engines not entirely revealing their algorithms, it is not seemingly possible to understand how things work. But, we now know for sure that the crawlers or bots have a huge role to play!

4. Indexing Process

Indexing process comprises of the following three tasks:

Text acquisition-It identifies and stores documents for indexing. Text Transformation-It transforms document into index terms or features. Index Creation-It takes index terms created by text transformations and create data structures to support fast searching.

5.Query Process

Query process comprises of the following three tasks:

User interaction-It supporst creation and refinement of user query and displays the results.

Ranking-It uses query and indexes to create ranked list of documents.

Evaluation-It monitors and measures the effectiveness and efficiency. It is done offline.

RANKING OF WEB PAGES.

PageRank (**PR**) is an algorithm used by Google Search to rank web pages in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages. PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. The numerical weight that it assigns to any given element *E* is referred to as the *PageRank of E* and denoted by **PR(E).**A PageRank results from a mathematical algorithm based on the webgraph, created by all World Wide Web pages as nodes and hyperlinks as edges, taking into consideration authority hubs such as cnn.com or usa.gov. The rank value indicates an importance of a particular page.

A hyperlink to a page counts as a vote of support. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself.Numerous academic papers concerning PageRank have been published since Page and Brin's original paper. In practice, the PageRank concept may be vulnerable to manipulation. Research has been conducted into identifying falsely influenced PageRank rankings. The goal is to find an effective means of ignoring links from documents with falsely influenced PageRank.

Other link-based ranking algorithms for Web pages include the HITS algorithm invented by Jon Kleinberg (used by Teoma and now Ask.com),the IBM CLEVER project, the TrustRank algorithm and the Hummingbird algorithm

Page Rank Algorithm

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called “iterations”, through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

Simplified algorithm

Assume a small universe of four web pages: A, B, C and D. Links from a page to itself, or multiple outbound links from one single page to another single page, are ignored. PageRank is initialized to the same value for all pages. In the original form of PageRank, the sum of PageRank over all pages was the total number of pages on the web at that time, so each page in this example would have an initial value of 1. However, later versions of PageRank, and the remainder of this section, assume a probability distribution between 0 and 1. Hence the initial value for each page in this example is 0.25.

The PageRank transferred from a given page to the targets of its outbound links upon the next iteration is divided equally among all outbound links.

If the only links in the system were from pages B, C, and D to A, each link would transfer 0.25 PageRank to A upon the next iteration, for a total of 0.75.

$$\mathbf{PR(A)=PR(B)+PR(C)+PR(D)}$$

Suppose instead that page B had a link to pages C and A, page C had a link to page A, and page D had links to all three pages. Thus, upon the first iteration, page B would transfer half of its existing value, or 0.125, to page A and the other half, or 0.125, to page C. Page C would transfer all of its existing value, 0.25, to the only page it links to, A. Since D had three outbound links, it would transfer one third of its existing value, or approximately 0.083, to A. At the completion of this iteration, page A will have a PageRank of approximately 0.458.

$$\mathbf{PR(A)=PR(B)/2+PR(C)/1+PR(D)/3}$$

In other words, the PageRank conferred by an outbound link is equal to the document’s own PageRank score divided by the number of outbound links L( ).

$$\mathbf{PR(A)=(PR(B)/L(B))+(PR(C)/L(C))+(PR(D)/L(D))}$$

The algorithm involves a damping factor for the calculation of the pagerank. It is like the income tax which the govt extracts from one despite paying him itself.