DATA MINING

B.Tech. IV Year I Semester								
Course Code	Category	Hours / Week			Credits	Maxumum Marks		
		L	T	P	C	CIA	SEE	Total
CS4102PC	Elective	2	0	0	2	25	75	100
Contact classes: 36	Tutorial Classes : NIL	Practical classes : NIL				Total Classes :36		
Prerequisites: • A course on "Database Management Systems" • Knowledge of probability and statistics								

Course Objectives:

- It presents methods for mining frequent patterns, associations,

and correlations.

- It then describes methods for data classification and

prediction, and data–clustering approaches.

- It covers mining various types of data stores such as spatial,

textual, multimedia, streams.

Course Outcomes:

- Ability to understand the types of the data to be mined and

  present a general classification of tasks and primitives tointegrate a data mining system.
- Apply pre processing methods for any given raw data.
- Extract interesting patterns from large amounts of data.
- Discover the role played by datamining in various fields.
- Choose and employ suitable datamining algorithms to buildanalytical applications
- Evaluate the accuracy of supervised and unsupervised modelsand algorithms.

COURSE SYLLABUS

MODULE- I

Data Mining: Data–Types of Data–, Data Mining Functionalities–Interestingness Patterns–Classification of Data Mining systems–Datamining Task primitives–Integration of Datamining system with aData warehouse–Major issues in Data Mining–Data Preprocessing.

MODULE- II

Association Rule Mining: Mining Frequent Patterns–Associationsand correlations –Mining Methods–MiningVarious kinds ofAssociation Rules–Correlation Analysis –Constraint basedAssociation mining. Graph Pattern Mining, SPM.

MODULE- III

Classification: Classification and Prediction– Basic concepts–Decision tree induction–Bayesian classification, Rule–basedclassification, Lazylearner.

MODULE- IV

Clustering and Applications: Cluster analysis–Types of Data inCluster Analysis–Categorization of Major Clustering Methods–Partitioning Methods, Hierarchical Methods– Density–BasedMethods, Grid–Based Methods, Outlier Analysis.

MODULE- V

Advanced Concepts: Basic concepts in Mining data streams–MiningTime–series data—Miningsequence patterns in Transactionaldatabases– Mining Object– Spatial– Multimedia–Text and Webdata –Spatial Datamining–Multimedia Datamining–TextMining–Mining theWorld Wide Web.

TEXTBOOKS:

1. Data Mining– Concepts and Techniques –Jiawei Han & MichelineKamber, 3rd Edition Elsevier.

2. Data Mining Introductory and Advanced topics – Margaret HDunham, PEA.

REFERENCE BOOK:

1. Ian H.Witten and Eibe Frank, Data Mining: Practical MachineLearning Toolsand Techniques (SecondEdition), MorganKaufmann, 2005.

MODULE- I

Data Mining: Data–Types of Data–, Data Mining Functionalities–Interestingness Patterns–Classification of Data Mining systems–Datamining Task primitives–Integration of Datamining system with aData warehouse–Major issues in Data Mining–Data Preprocessing.

1.1 Introduction:

Data mining is the process of extracting useful information from large sets of data. It involves using various techniques from statistics, machine learning, and database systems to identify patterns, relationships, and trends in the data. This information can then be used to make data-driven decisions, solve business problems, and uncover hidden insights. Applications of data mining include customer profiling and segmentation, market basket analysis, anomaly detection, and predictive modeling. Data mining tools and technologies are widely used in various industries, including finance, healthcare, retail, and telecommunications.

1.2Types of Data:

Qualitative Data Type

Qualitative or Categorical Data describes the object under consideration using a finite set of discrete classes. It means that this type of data can't be counted or measured easily using numbers and therefore divided into categories. The gender of a person (male, female, or others) is a good example of this data type.

These are usually extracted from audio, images, or text medium. Another example can be of a smartphone brand that provides information about the current rating, the color of the phone, category of the phone, and so on. All this information can be categorized as Qualitative data. There are two subcategories under this

Nominal

These are the set of values that don't possess a natural ordering. Let's understand this with some examples. The color of a smartphone can be considered as a nominal data type as we can't compare one color with others.

It is not possible to state that 'Red' is greater than 'Blue'. The gender of a person is another one where we can't differentiate between male, female, or others. Mobile phone categories whether it is midrange, budget segment, or premium smartphone is also nominal data type.

Nominal data types in statistics are not quantifiable and cannot be measured through numerical units. Nominal types of statistical data are valuable while conducting qualitative research as it extends freedom of opinion to subjects.

Ordinal

These types of values have a natural ordering while maintaining their class of values. If we consider the size of a clothing brand then we can easily sort them according to their name tag in the order of small < medium < large. The grading system while marking candidates in a test can also be considered as an ordinal data type where A+ is definitely better than B grade.

These categories help us deciding which encoding strategy can be applied to which type of data. Data encoding for Qualitative data is important because machine learning models can't handle

these values directly and needed to be converted to numerical types as the models are mathematical in nature.

For nominal data type where there is no comparison among the categories, one-hot encoding can be applied which is similar to binary coding considering there are in less number and for the ordinal data type, label encoding can be applied which is a form of integer encoding.

Quantitative Data Type

This data type tries to quantify things and it does by considering numerical values that make it countable in nature. The price of a smartphone, discount offered, number of ratings on a product, the frequency of processor of a smartphone, or ram of that particular phone, all these things fall under the category of Quantitative data types.

The key thing is that there can be an infinite number of values a feature can take. For instance, the price of a smartphone can vary from x amount to any value and it can be further broken down based on fractional values. The two subcategories which describe them clearly are:

Discrete

The numerical values which fall under are integers or whole numbers are placed under this category. The number of speakers in the phone, cameras, cores in the processor, the number of sims supported all these are some of the examples of the discrete data type.

Discrete data types in statistics cannot be measured – it can only be counted as the objects included in discrete data have a fixed value. The value can be represented in decimal, but it has to be whole. Discrete data is often identified through charts, including bar charts, pie charts, and tally charts.

1.3 Data Mining Functionalities:

The process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining.In other words, we can say that Data Mining is the process of investigating hidden patterns of information to various perspectives for categorization into useful data, which is collected and assembled in particular areas such as data warehouses, efficient analysis, data mining algorithm, helping decision making and other data requirement to eventually cost-cutting and generating revenue.

Data mining is the act of automatically searching for large stores of information to find trends and patterns that go beyond simple analysis procedures. Data mining utilizes complex mathematical algorithms for data segments and evaluates the probability of future events.

Data Mining is also called Knowledge Discovery of Data (KDD).Data Mining is a process used by organizations to extract specific data from huge databases to solve business problems. It primarily turns raw data into useful information.

1.4 Interestingness Patterns:

KDD Process:

KDD (Knowledge Discovery in Databases) is a process that involves the extraction of useful, previously unknown, and potentially valuable information from large datasets. The KDD process is an iterative process and it requires multiple iterations of the above steps to extract accurate knowledge from the data.The following steps are included in KDD process:

Data Cleaning

Data cleaning is defined as removal of noisy and irrelevant data from collection.

1. Cleaning in case of **Missing values**.
2. Cleaning **noisy** data, where noise is a random or variance error.
3. Cleaning with **Data discrepancy detection** and **Data transformation tools**.

Data Integration

Data integration is defined as heterogeneous data from multiple sources combined in a common source(DataWarehouse). Data integration using **Data Migration tools, Data Synchronization tools and ETL**(Extract-Load-Transformation) process.

Data Selection

Data selection is defined as the process where data relevant to the analysis is decided and retrieved from the data collection. For this we can use **Neural network, Decision Trees, Naive bayes, Clustering**, and **Regression** methods.

Data Transformation

Data Transformation is defined as the process of transforming data into appropriate form required by mining procedure. Data Transformation is a two step process:

1. **Data Mapping**: Assigning elements from source base to destination to capture transformations.
2. **Code generation**: Creation of the actual transformation program.

Data Mining

Data mining is defined as techniques that are applied to extract patterns potentially useful. It transforms task relevant data into **patterns, and d**ecides purpose of model using **classification** or **characterization**.

Pattern Evaluation

Pattern Evaluation is defined as identifying strictly increasing patterns representing knowledge based on given measures. It find **interestingness score** of each pattern, and uses **summarization** and **Visualization** to make data understandable by user.

Knowledge Representation

This involves presenting the results in a way that is meaningful and can be used to make decisions.

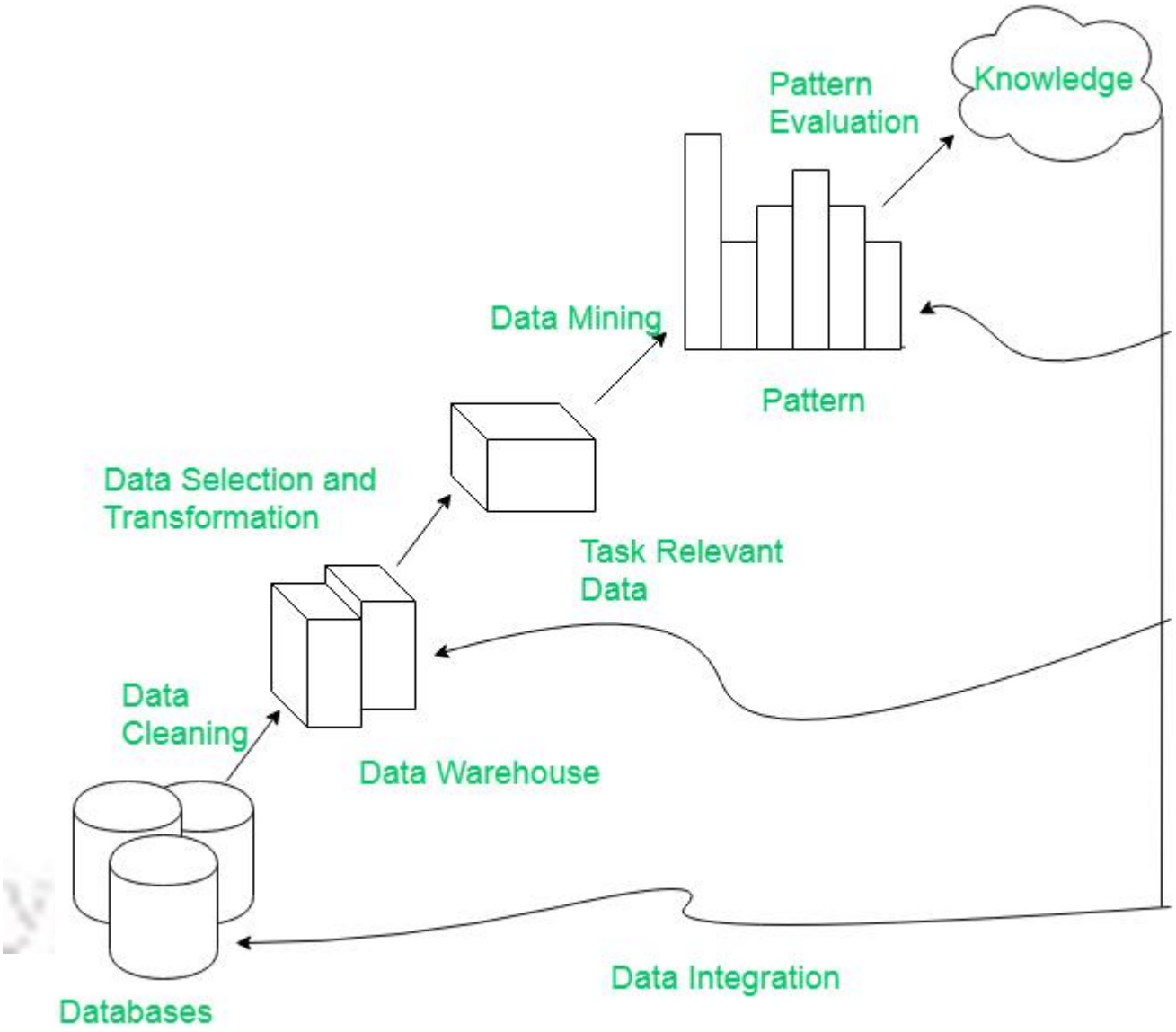


Fig:KDD Process

1.5 Major issues in Data Mining :

Data mining, the process of extracting knowledge from data, has become increasingly important as the amount of data generated by individuals, organizations, and machines has grown exponentially. However, data mining is not without its challenges. In this article, we will explore some of the main challenges of data mining.

1]Data Quality

The quality of data used in data mining is one of the most significant challenges. The accuracy, completeness, and consistency of the data affect the accuracy of the results obtained. The data may contain errors, omissions, duplications, or inconsistencies, which may lead to inaccurate results. Moreover, the data may be incomplete, meaning that some attributes or values are missing, making it challenging to obtain a complete understanding of the data.

Data quality issues can arise due to a variety of reasons, including data entry errors, data storage issues, data integration problems, and data transmission errors. To address these challenges, data mining practitioners must apply data cleaning and data preprocessing techniques to improve the quality of the data. Data cleaning involves detecting and correcting errors, while data preprocessing involves transforming the data to make it suitable for data mining.

2]DataComplexity

Data complexity refers to the vast amounts of data generated by various sources, such as sensors, social media, and the internet of things (IoT). The complexity of the data may make it challenging to process, analyze, and understand. In addition, the data may be in different formats, making it challenging to integrate into a single dataset.

To address this challenge, data mining practitioners use advanced techniques such as clustering, classification, and association rule mining. These techniques help to identify patterns and relationships in the data, which can then be used to gain insights and make predictions.

3]DataPrivacyandSecurity

Data privacy and security is another significant challenge in data mining. As more data is collected, stored, and analyzed, the risk of data breaches and cyber-attacks increases. The data may contain personal, sensitive, or confidential information that must be protected. Moreover, data privacy regulations such as GDPR, CCPA, and HIPAA impose strict rules on how data can be collected, used, and shared.

To address this challenge, data mining practitioners must apply data anonymization and data encryption techniques to protect the privacy and security of the data. Data anonymization involves removing personally identifiable information (PII) from the data, while data encryption involves using algorithms to encode the data to make it unreadable to unauthorized users.

4]Scalability

Data mining algorithms must be scalable to handle large datasets efficiently. As the size of the dataset increases, the time and computational resources required to perform data mining operations also increase. Moreover, the algorithms must be able to handle streaming data, which is generated continuously and must be processed in real-time.

To address this challenge, data mining practitioners use distributed computing frameworks such as Hadoop and Spark. These frameworks distribute the data and processing across multiple nodes, making it possible to process large datasets quickly and efficiently.

4]interpretability

Data mining algorithms can produce complex models that are difficult to interpret. This is because the algorithms use a combination of statistical and mathematical techniques to identify patterns

and relationships in the data. Moreover, the models may not be intuitive, making it challenging to understand how the model arrived at a particular conclusion. To address this challenge, data mining practitioners use visualization techniques to represent the data and the models visually. Visualization makes it easier to understand the patterns and relationships in the data and to identify the most important variables.

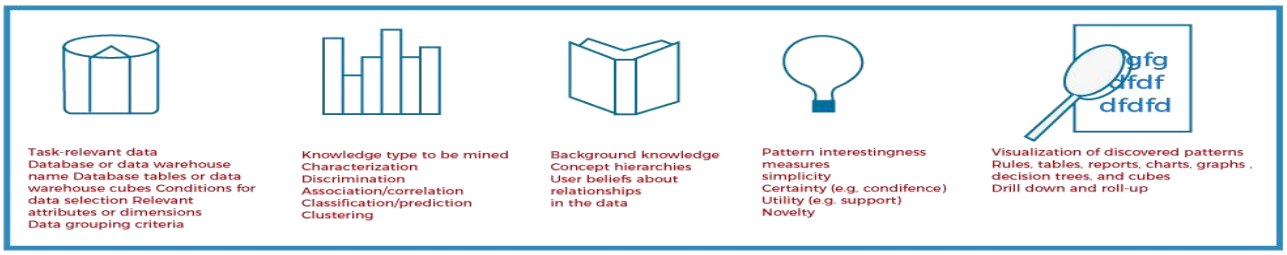
5]Ethics

Data mining raises ethical concerns related to the collection, use, and dissemination of data. The data may be used to discriminate against certain groups, violate privacy rights, or perpetuate existing biases. Moreover, data mining algorithms may not be transparent, making it challenging to detect biases or discrimination.

1.5 DataMining Tasks premitives:

A data mining task can be specified in the form of a data mining query, which is input to the data mining system. A data mining query is defined in terms of data mining task primitives. These primitives allow the user to interactively communicate with the data mining system during discovery to direct the mining process or examine the findings from different angles or depths. The data mining primitives specify the following,

1. Set of task-relevant data to be mined.
2. Kind of knowledge to be mined.
3. Background knowledge to be used in the discovery process.
4. Interestingness measures and thresholds for pattern evaluation.
5. Representation for visualizing the discovered patterns.



A data mining query language can be designed to incorporate these primitives, allowing users to interact with data mining systems flexibly. Having a data mining query language provides a foundation on which user-friendly graphical interfaces can be built.

Designing a comprehensive data mining language is challenging because data mining covers a wide spectrum of tasks, from data characterization to evolution analysis. Each task has different requirements. The design of an effective data mining query language requires a deep understanding of the power, limitation, and underlying mechanisms of the various kinds of data mining tasks. This facilitates a data mining system's communication with other information systems and integrates with the overall information processing environment.

1.6 Integration of Datamining system with aData warehouse:

With the exponential growth of data, data mining systems should be efficient and highly performative to build complex machine learning models, it is expected that a good variety of data mining systems will be designed and developed.

Comprehensive information processing and data analysis will be continuously and systematically surrounded by data warehouse and databases.

Data Mining System Architecture

A critical question in design is whether we should integrate data mining systems with database systems.

Integrating Data Mining systems with Databases and Data Warehouses with these methods

- **No Coupling**
- **Loose Coupling**
- **Semi-Tight Coupling**
- **Tight Coupling**

No Coupling

No coupling means that a DM system will not utilize any function of a DB or DW system.

It may fetch data from a particular source (such as a file system), process data using some data mining algorithms, and then store the mining results in another file.

Drawbacks:

First, a Database/Data Warehouse system provides a great deal of flexibility and efficiency at storing, organizing, accessing, and processing data.

Without using a Database/Data Warehouse system, a Data Mining system may spend a substantial amount of time finding, collecting, cleaning, and transforming data.

Second, there are many tested, scalable algorithms and data structures implemented in Database and Data Warehouse systems.

Loose Coupling

Loose coupling means that a Data Mining system will use some facilities of a Database or Data warehouse system, fetching data from a data repository managed by these systems, performing data mining, and then storing the mining results either in a file or in a designated place in a Database or Data Warehouse.

Loose coupling is better than no coupling because it can fetch any portion of data stored in Databases or Data Warehouses by using query processing, indexing, and other system facilities.

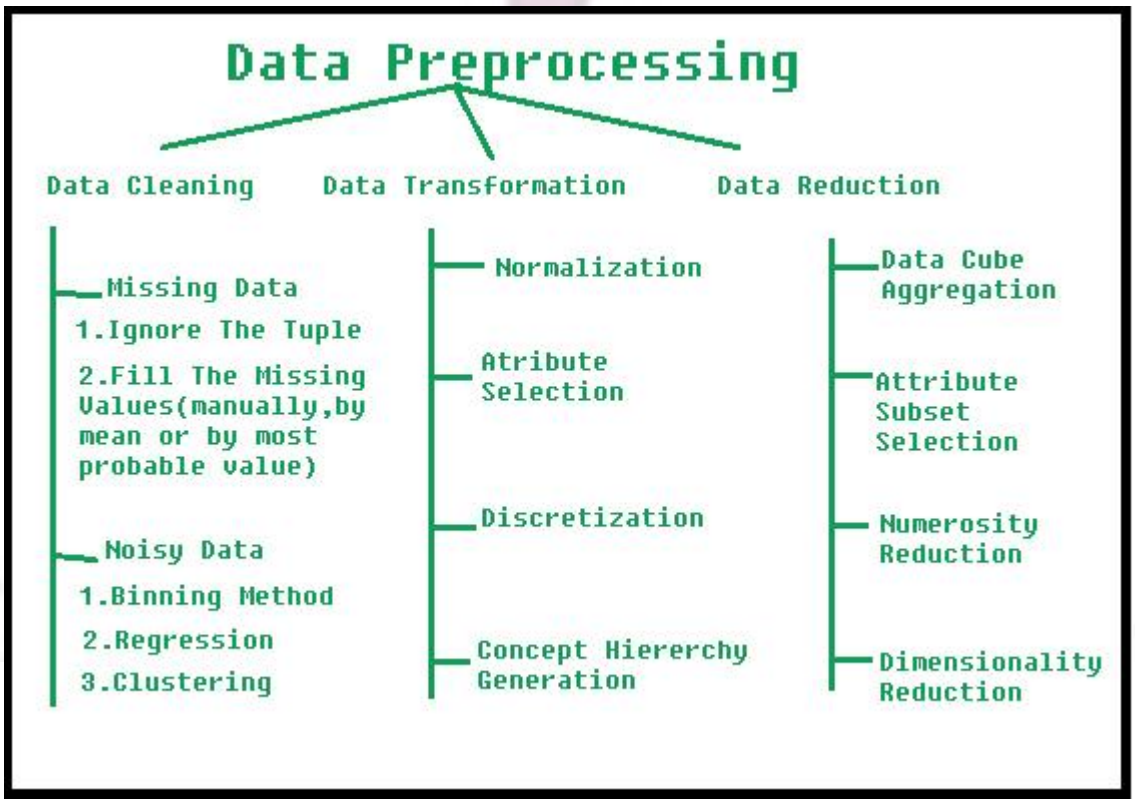
Drawbacks

It's difficult for loose coupling to achieve high scalability and good performance with large data sets.

Semi-Tight Coupling - Enhanced Data Mining Performance
The semi-tight coupling means that besides linking a Data Mining system to a Database/Data Warehouse system, efficient implementations of a few essential data mining primitives (identified by the analysis of frequently encountered data mining functions) can be provided in the Database/Data Warehouse system.
These primitives can include sorting, indexing, aggregation, histogram analysis, multi-way join, and pre-computation of some essential statistical measures, such as sum, count, max, min, standard deviation.
This design will enhance the performance of Data Mining systems.
Tight Coupling - A Uniform Information Processing Environment
Tight coupling means that a Data Mining system is smoothly integrated into the Database/Data Warehouse system.
The data mining subsystem is treated as one functional component of the information system.
Data mining queries and functions are optimized based on mining query analysis, data structures, indexing schemes, and query processing methods of a Database or Data Warehouse system.

1.7 Data Preprocessing:

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.



Steps Involved in Data Preprocessing:

1.DataCleaning:
The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

- **(a).MissingData:**
  This situation arises when some data is missing in the data. It can be handled in various ways. Some of them are:

1. **Ignorethetuples:**

This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

2. **FilltheMissingvalues:**

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

- **(b).NoisyData:**
  Noisy data is a meaningless data that can't be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :
  1. **BinningMethod:**
     This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

  2. **Regression:**
     Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).

  3. **Clustering:**
     This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2.DataTransformation:
This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. **Normalization:**
   It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

2. **AttributeSelection:**
   In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

3. **Discretization:**
   This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

4. **ConceptHierarchyGeneration:**
   Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute "city" can be converted to "country".

3.DataReduction:
Data reduction is a crucial step in the data mining process that involves reducing the size of the dataset while preserving the important information. This is done to improve the efficiency of data analysis and to avoid overfitting of the model. Some common steps involved in data reduction are:
Feature Selection: This involves selecting a subset of relevant features from the dataset. Feature selection is often performed to remove irrelevant or redundant features from the dataset. It can be done using various techniques such as correlation analysis, mutual information, and principal component analysis (PCA).
Feature Extraction: This involves transforming the data into a lower-dimensional space while preserving the important information. Feature extraction is often used when the original features are high-dimensional and complex. It can be done using techniques such as PCA, linear discriminant analysis (LDA), and non-negative matrix factorization (NMF).
Sampling: This involves selecting a subset of data points from the dataset. Sampling is often used to reduce the size of the dataset while preserving the important information. It can be done using techniques such as random sampling, stratified sampling, and systematic sampling.
Clustering: This involves grouping similar data points together into clusters. Clustering is often used to reduce the size of the dataset by replacing similar data points with a representative centroid. It can be done using techniques such as k-means, hierarchical clustering, and density-based clustering.
Compression: This involves compressing the dataset while preserving the important information. Compression is often used to reduce the size of the dataset for storage and transmission purposes. It can be done using techniques such as wavelet compression, JPEG compression, and gzip compression.

MODULE II
Association Rules

Association Rule Mining: Mining Frequent Patterns–Associationsand correlations –Mining Methods–MiningVarious kinds ofAssociation Rules–Correlation Analysis –Constraint basedAssociation mining. Graph Pattern Mining, SPM.

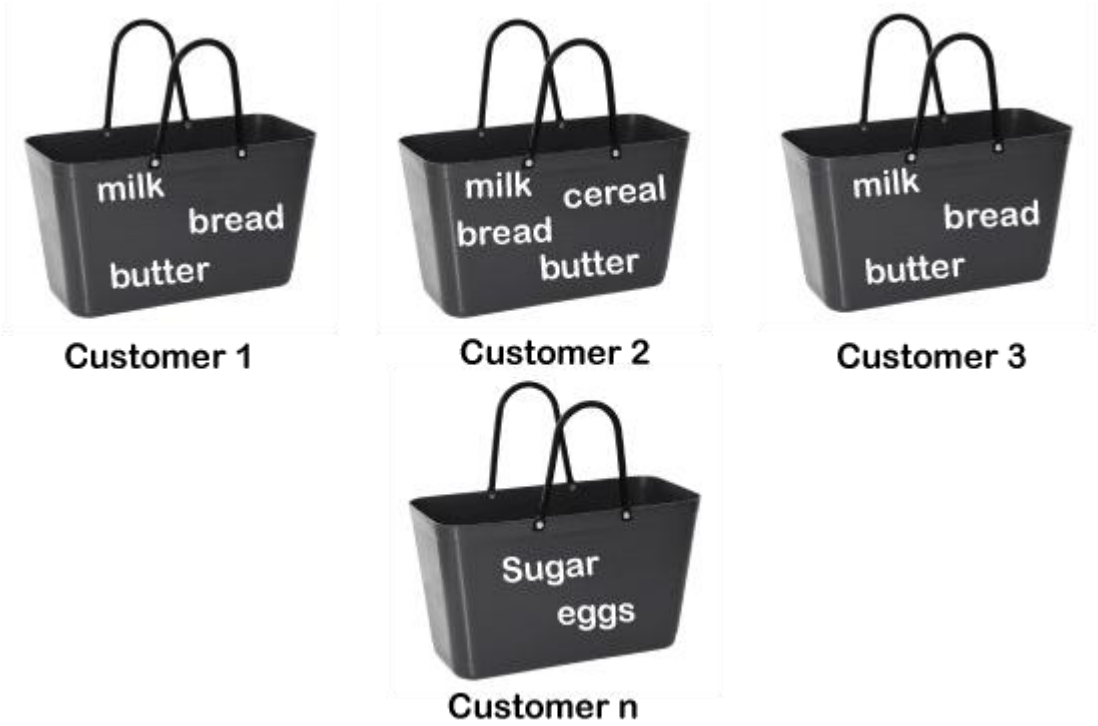
2.1 Association Rule Mining:

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

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

The association rule learning is one of the very important concepts of machine learning, and it is employed in **Market Basket analysis, Web usage mining, continuous production, etc.** Here market basket analysis is a technique used by the various big retailer to discover the associations between items. We can understand it by taking an example of a supermarket, as in a supermarket, all products that are purchased together are put together.

For example, if a customer buys bread, he most likely can also buy butter, eggs, or milk, so these products are stored within a shelf or mostly nearby. Consider the below diagram:



1. **Apriori**
2. **Eclat**
3. **F-P Growth Algorithm**

Association rule learning works on the concept of If and Else Statement, such as if A then B.

4. 

Here the If element is called **antecedent**, and then statement is called as **Consequent**. These types of relationships where we can find out some association or relation between two items is known *as single cardinality*. It is all about creating rules, and if the number of items increases, then cardinality also increases accordingly. So, to measure the associations between thousands of data items, there are several metrics. These metrics are given below:

2.3 Frequent Item Set Generation:

1. **Support**
2. **Confidence**
3. **Lift**

Let's understand each of them:

Support

Support is the frequency of A or how frequently an item appears in the dataset. It is defined as the fraction of the transaction T that contains the itemset X. If there are X datasets, then for transactions T, it can be written as:

$$\text{Supp(X)} = \frac{Freq(X)}{T}$$

Confidence

Confidence indicates how often the rule has been found to be true. Or how often the items X and Y occur together in the dataset when the occurrence of X is already given. It is the ratio of the transaction that contains X and Y to the number of records that contain X.

$$\text{Confidence} = \frac{Freq(X,Y)}{Freq(X)}$$

Example On finding Frequent Itemsets – Consider the given dataset with given

TransactionId	Items
1	{A,C,D}
2	{B,C,D}
3	{A,B,C,D}
4	{B,D}
5	{A,B,C,D}

- Lets say minimum support count is 3
- Relation hold is maximal frequent => closed => frequent

1-frequent: {A} = 3; // not closed due to {A, C} and not maximal {B} = 4; // not closed due to {B, D} and no maximal {C} = 4; // not closed due to {C, D} not maximal {D} = 5; // closed item-set since not immediate super-set has same count. Not maximal

2-frequent: {A, B} = 2 // not frequent because support count < minimum support count so ignore {A, C} = 3 // not closed due to {A, C, D} {A, D} = 3 // not closed due to {A, C, D} {B, C} = 3 // not closed due to {B, C, D} {B, D} = 4 // closed but not maximal due to {B, C, D} {C, D} = 4 // closed but not maximal due to {B, C, D}

3-frequent: {A, B, C} = 2 // ignore not frequent because support count < minimum support count {A, B, D} = 2 // ignore not frequent because support count < minimum support count {A, C, D} = 3 // maximal frequent {B, C, D} = 3 // maximal frequent

4-frequent: {A, B, C, D} = 2 //ignore not frequent </

2.4 Mining Methods

APRIOIRI Algorithm:

Apriori algorithm refers to the algorithm which is used to calculate the association rules between objects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule leaning that analyzes that people who bought product A also bought product B.

The primary objective of the apriori algorithm is to create the association rule between different objects. The association rule describes how two or more objects are related to one another. Apriori algorithm is also called frequent pattern mining. Generally, you operate the Apriori algorithm on a database that consists of a huge number of transactions. Let's understand the apriori algorithm with the help of an example; suppose you go to Big Bazar and buy different products. It helps the customers buy their products with ease and increases the sales performance of the Big Bazar. In this tutorial, we will discuss the apriori algorithm with examples.

Introduction

We take an example to understand the concept better. You must have noticed that the Pizza shop seller makes a pizza, soft drink, and breadstick combo together. He also offers a discount to their customers who buy these combos. Do you ever think why does he do so? He thinks that customers who buy pizza also buy soft drinks and breadsticks. However, by making combos, he makes it easy for the customers. At the same time, he also increases his sales performance.

Similarly, you go to Big Bazar, and you will find biscuits, chips, and Chocolate bundled together. It shows that the shopkeeper makes it comfortable for the customers to buy these products in the same place.

We will understand this algorithm with the help of an example

Consider a Big Bazar scenario where the product set is P = {Rice, Pulse, Oil, Milk, Apple}. The database comprises six transactions where 1 represents the presence of the product and 0 represents the absence of the product.

Transaction ID	Rice	Pulse	Oil Milk	Apple	
t1	1	1	1	0	0
t2	0	1	1	1	0
t3	0	0	0	1	1

t4	1	1	0	1	0
t5	1	1	1	0	1
t6	1	1	1	1	1

The Apriori Algorithm makes the given assumptions

- All subsets of a frequent itemset must be frequent.
- The subsets of an infrequent item set must be infrequent.
- Fix a threshold support level. In our case, we have fixed it at 50 percent.

Step 1

Make a frequency table of all the products that appear in all the transactions. Now, short the frequency table to add only those products with a threshold support level of over 50 percent. We find the given frequency table.

Product	Frequency (Number of transactions)
Rice (R)	4
Pulse(P)	5
Oil(O)	4
Milk(M)	4

The above table indicated the products frequently bought by the customers.

Step 2

Create pairs of products such as RP, RO, RM, PO, PM, OM. You will get the given frequency table.

Itemset	Frequency (Number of transactions)
RP	4
RO	3
RM	2

PO	4
PM	3
OM	2

Step 3

Implementing the same threshold support of 50 percent and consider the products that are more than 50 percent. In our case, it is more than 3

Thus, we get RP, RO, PO, and PM

Step 4

Now, look for a set of three products that the customers buy together. We get the given combination.

1. RP and RO give RPO
2. PO and PM give POM

Step 5

Calculate the frequency of the two itemsets, and you will get the given frequency table.

Itemset	Frequency (Number of transactions)
RPO	4
POM	3

If you implement the threshold assumption, you can figure out that the customers' set of three products is RPO.

We have considered an easy example to discuss the apriori algorithm in data mining. In reality, you find thousands of such combinations.

Advantages of Apriori Algorithm

- It is used to calculate large itemsets.
- Simple to understand and apply.

Disadvantages of Apriori Algorithms

- Apriori algorithm is an expensive method to find support since the calculation has to pass through the whole database.
- Sometimes, you need a huge number of candidate rules, so it becomes computationally more expensive.

The Partition Algorithms:

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-Medoids), CLARA algorithm (Clustering Large Applications) etc. In this article, we will be seeing the working of K Mean algorithm in detail. **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

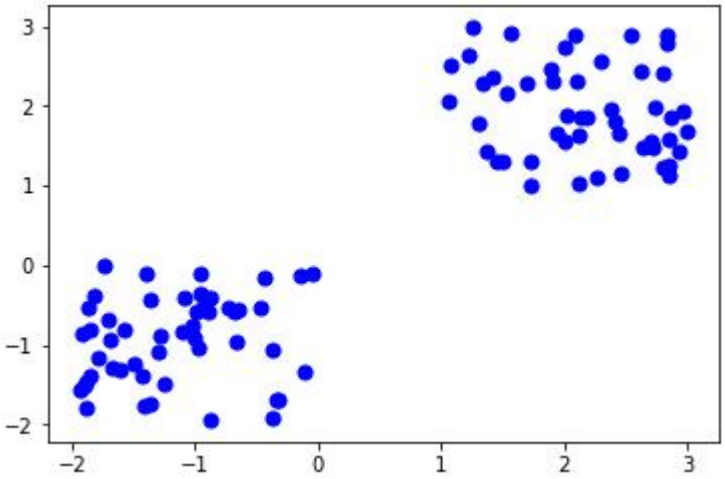
D: A dataset containing N number of objects

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 2 until no change occurs.



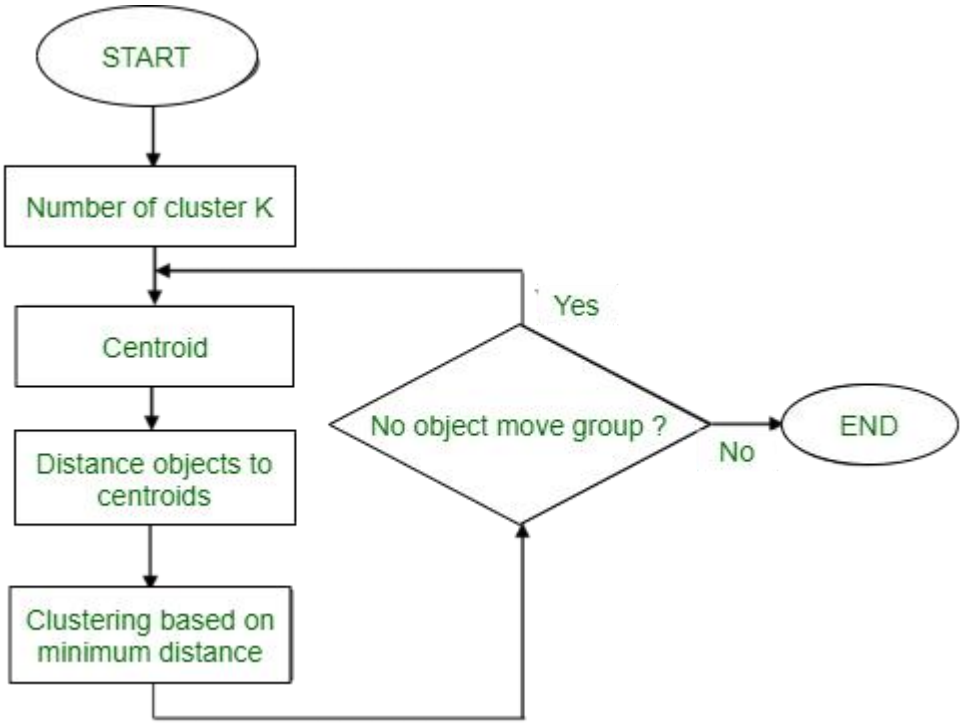


Figure – K-mean Clustering**Example:** Suppose we want to group the visitors to a website using just their age as follows:
16, 16, 17, 20, 20, 21, 21, 22, 23, 29, 36, 41, 42, 43, 44, 45, 61, 62, 66

Initial Cluster:
K=2

Centroid(C1) = 16 [16]

Centroid(C2) = 22 [22]

Note: These two points are chosen randomly from the dataset. **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.

FP- Growth Algorithms:

In Data Mining, finding frequent patterns in large databases is very important and has been studied on a large scale in the past few years. Unfortunately, this task is computationally expensive, especially when many patterns exist.

The FP-Growth Algorithm proposed by ***Han in***. This is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree). In his study, Han proved that his method outperforms other popular methods for mining frequent patterns, e.g. the Apriori Algorithm and the TreeProjection. In some later works, it was proved that FP-Growth performs better than other methods, including ***Eclat*** and ***Relim***. The popularity and efficiency of the FP-Growth Algorithm contribute to many studies that propose variations to improve its performance.

This algorithm works as follows:

- First, it compresses the input database creating an FP-tree instance to represent frequent items.
- After this first step, it divides the compressed database into a set of conditional databases, each associated with one frequent pattern.
- Finally, each such database is mined separately.

Using this strategy, the FP-Growth reduces the search costs by recursively looking for short patterns and then concatenating them into the long frequent patterns.

In large databases, holding the FP tree in the main memory is impossible. A strategy to cope with this problem is to partition the database into a set of smaller databases (called projected databases) and then construct an FP-tree from each of these smaller databases.

FP-Tree

The frequent-pattern tree (FP-tree) is a compact data structure that stores quantitative information about frequent patterns in a database. Each transaction is read and then mapped onto a path in the FP-tree. This is done until all transactions have been read. Different transactions with common subsets allow the tree to remain compact because their paths overlap.

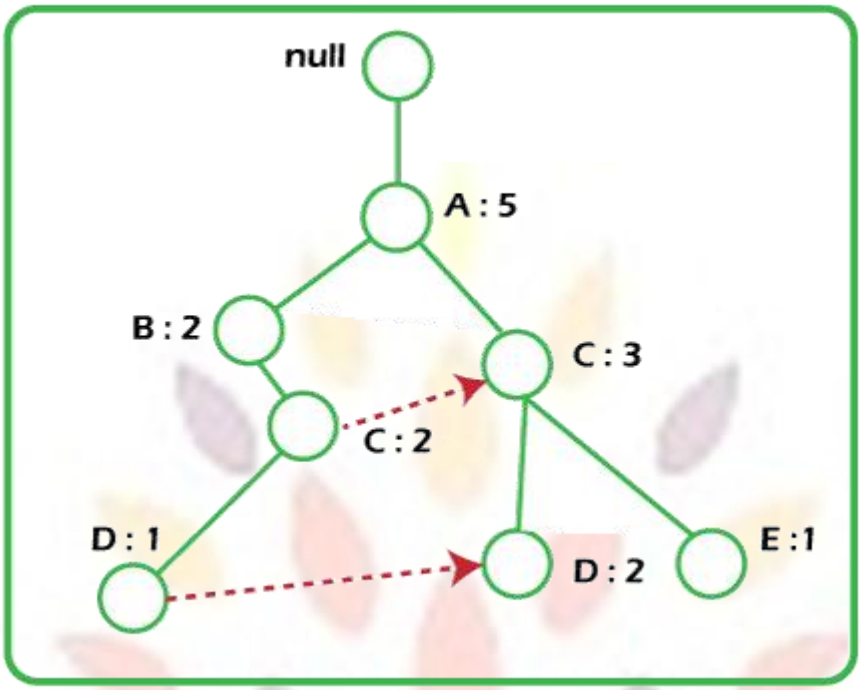
A frequent Pattern Tree is made with the initial item sets of the database. The purpose of the FP tree is to mine the most frequent pattern. Each node of the FP tree represents an item of the item set.

The root node represents null, while the lower nodes represent the item sets. The associations of the nodes with the lower nodes, that is, the item sets with the other item sets, are maintained while forming the tree.

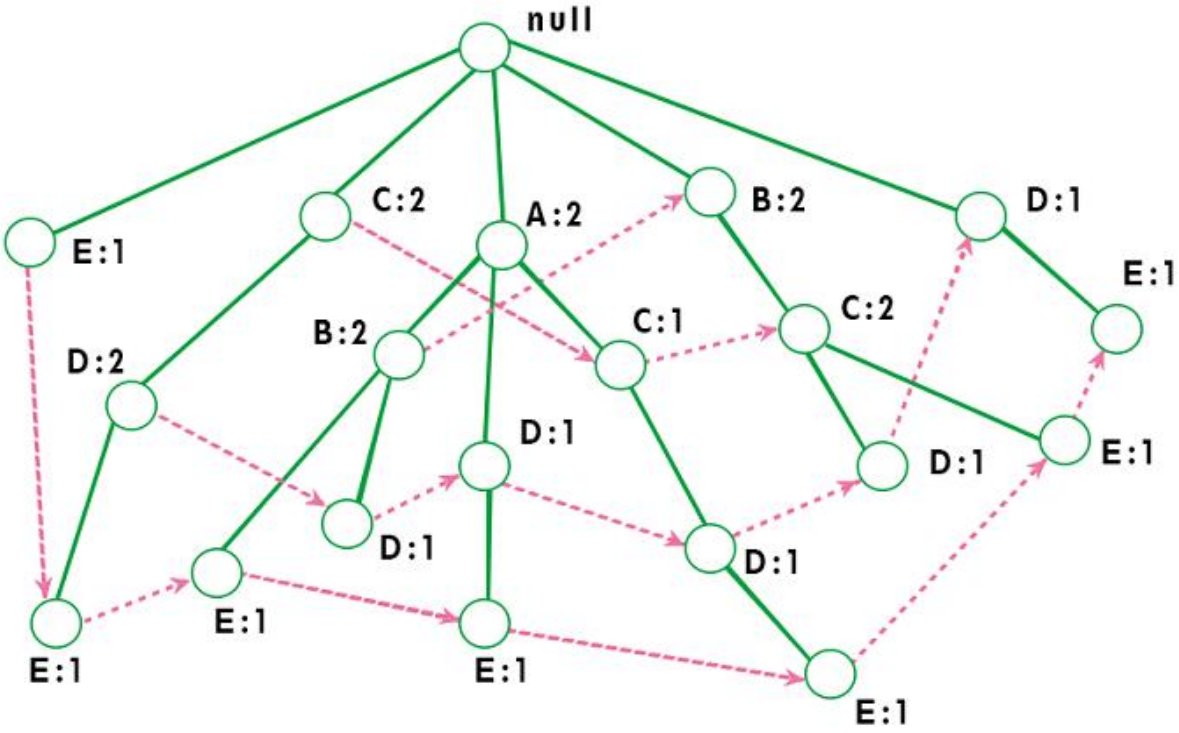
Han defines the FP-tree as the tree structure given below:

1. One root is labelled as "null" with a set of item-prefix subtrees as children and a frequent-item-header table.
2. Each node in the item-prefix subtree consists of three fields:
    - Item-name: registers which item is represented by the node;
    - Count: the number of transactions represented by the portion of the path reaching the node;
    - Node-link: links to the next node in the FP-tree carrying the same item name or null if there is none.
3. Each entry in the frequent-item-header table consists of two fields:
    - Item-name: as the same to the node;
    - Head of node-link: a pointer to the first node in the FP-tree carrying the item name.

Additionally, the frequent-item-header table can have the count support for an item. The below diagram is an example of a best-case scenario that occurs when all transactions have the same itemset; the size of the FP-tree will be only a single branch of nodes.



The worst-case scenario occurs when every transaction has a unique item set. So the space needed to store the tree is greater than the space used to store the original data set because the FP-tree requires additional space to store pointers between nodes and the counters for each item. The diagram below shows how a worst-case scenario FP-tree might appear. As you can see, the tree's complexity grows with each transaction's uniqueness.



Algorithm by Han

The original algorithm to construct the FP-Tree defined by Han is given below:

Algorithm 1: FP-tree construction

***Input*:** A transaction database DB and a minimum support threshold?

***Output*:** FP-tree, the frequent-pattern tree of DB.

***Method*:** The FP-tree is constructed as follows.

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, and then the next itemset with the lower count. 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, 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. The common node and new node count are 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 is called a conditional pattern base.
   A 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, 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.

Using this algorithm, the FP-tree is constructed in two database scans. The first scan collects and sorts the set of frequent items, and the second constructs the FP-Tree.

Example

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

Support threshold=50%, Confidence= 60%

Table 1:

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

Table 2: Count of each item

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

Table 3: Sort the itemset in descending order.

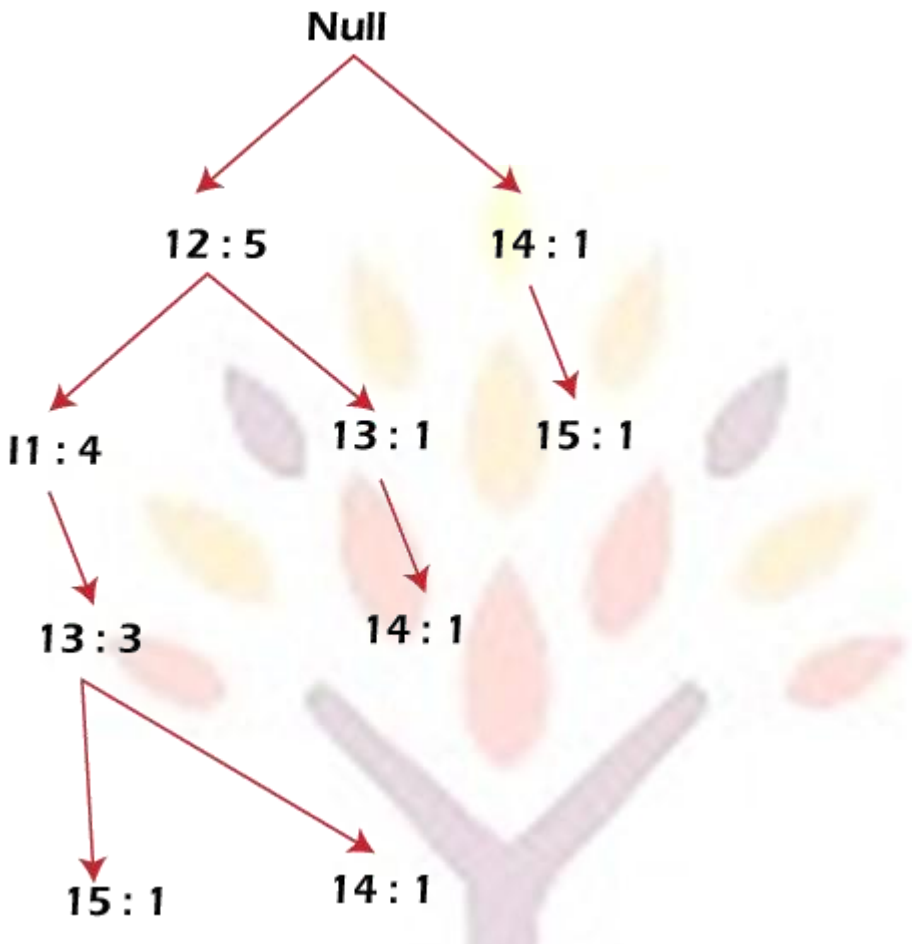
Item	Count

I2	5
I1	4
I3	4
I4	4

Build FP Tree

Let's build the FP tree in the following steps, such as:

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, I1 is linked to I2 and I3 is linked to I1.
3. T2: I2, I3, and I4 contain 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 the 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, and 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}.

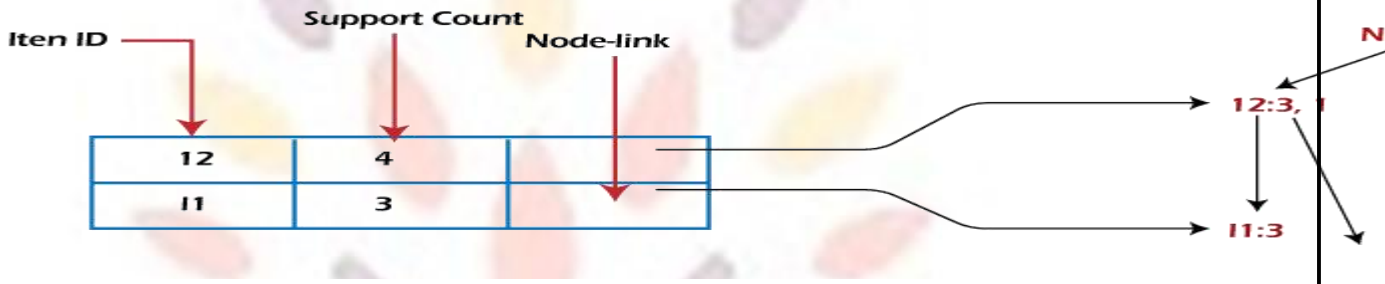


Mining of FP-tree is summarized below:

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, and 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.



dvantages of FP Growth Algorithm

Here are the following advantages of the FP growth algorithm, such as:

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

Disadvantages of FP-Growth Algorithm

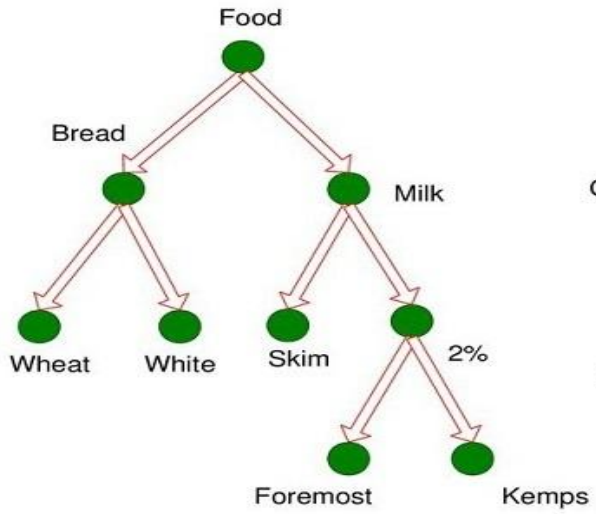
This algorithm also has some disadvantages, such as:

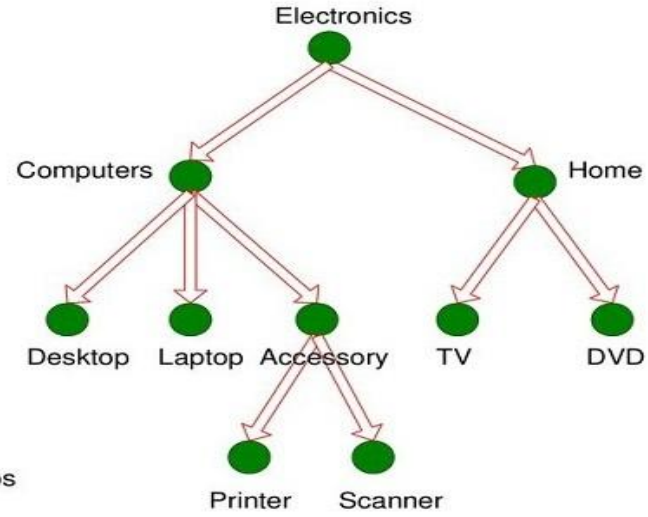
- FP Tree is more cumbersome and difficult to build than Apriori.
- It may be expensive.
- The algorithm may not fit in the shared memory when the database is large.

2.5 MiningVarious kinds ofAssociation Rules:

Association rule mining is used to discover relationships between items in a dataset. An association rule is a statement of the form "If A, then B," where A and B are sets of items. The strength of an association rule is measured using two measures: support and confidence. Support measures the

frequency of the occurrence of the items in the rule, and confidence measures the reliability of the rule.





Apriori algorithm is a popular algorithm for mining association rules. It is an iterative algorithm that works by generating candidate itemsets and pruning those that do not meet the support and confidence thresholds.

Multilevel Association Rule in data mining

Multilevel Association Rule mining is a technique that extends Association Rule mining to discover relationships between items at different levels of granularity. Multilevel Association Rule mining can be classified into two types: multi-dimensional Association Rule and multi-level Association Rule.

Multi-dimensional Association Rule mining

This is used to find relationships between items in different dimensions of a dataset. For example, in a sales dataset, multi-dimensional Association Rule mining can be used to find relationships between products, regions, and time.

Multi-level Association Rule mining

This is used to find relationships between items at different levels of granularity. For example, in a retail dataset, multi-level Association Rule mining can be used to find relationships between individual items and categories of items.

2.6 Constraint basedAssociation mining:

A data mining procedure can uncover thousands of rules from a given set of information, most of which end up being independent or tedious to the users. Users have a best sense of which "direction" of mining can lead to interesting patterns and the "form" of the patterns or rules they can like to discover.

Therefore, a good heuristic is to have the users defines such intuition or expectations as constraints to constraint the search space. This strategy is called constraint-based mining.

Constraint-based algorithms need constraints to decrease the search area in the frequent itemset generation step (the association rule generating step is exact to that of exhaustive algorithms).

The general constraint is the support minimum threshold. If a constraint is uncontrolled, its inclusion in the mining phase can support significant reduction of the exploration space because of the definition of a boundary inside the search space lattice, following which exploration is not needed.

The important of constraints is well-defined − they create only association rules that are appealing to users. The method is quite trivial and the rules space is decreased whereby remaining methods satisfy the constraints.

Constraint-based clustering discover clusters that satisfy user-defined preferences or constraints. It depends on the characteristics of the constraints, constraint-based clustering can adopt rather than different approaches.

The constraints can include the following which are as follows −

Knowledge type constraints − These define the type of knowledge to be mined, including association or correlation.

Data constraints − These define the set of task-relevant information such as Dimension/level constraints − These defines the desired dimensions (or attributes) of the information, or methods of the concept hierarchies, to be utilized in mining.

Interestingness constraints − These defines thresholds on numerical measures of rule interestingness, including support, confidence, and correlation.

Rule constraints − These defines the form of rules to be mined. Such constraints can be defined as metarules (rule templates), as the maximum or minimum number of predicates that can appear in the

rule antecedent or consequent, or as relationships between attributes, attribute values, and/or aggregates.

The following constraints can be described using a high-level declarative data mining query language and user interface. This form of constraint-based mining enables users to define the rules that they can like to uncover, thus by creating the data mining process more efficient.

Furthermore, a sophisticated mining query optimizer can be used to deed the constraints defined by the user, thereby creating the mining process more effective. Constraint-based mining boost interactive exploratory mining and analysis.

2.8 Graph Pattern Mining:

Graph mining is a process in which the mining techniques are used in finding a pattern or relationship in the given real-world collection of graphs. By mining the graph, frequent substructures and relationships can be identified which helps in clustering the graph sets, finding a relationship between graph sets, or discriminating or characterizing graphs. Predicting these patterning trends can help in building models for the enhancement of any application that is used in real-time. To implement the process of graph mining, one must learn to mine frequent subgraphs.

Frequent Subgraph Mining

Let us consider a graph h with an edge set E(h) and a vertex set V(h). Let us consider the existence of subgraph isomorphism from h to h' in such a way that h is a subgraph of h'. A label function is a function that plots either the edges or vertices to a label. Let us consider a labeled graph dataset, Let us consider s(h) as the support which means the percentage of graphs in F where h is a subgraph. A frequent graph has support that will be no less than the minimum support threshold. Let us denote it as min_support.

Steps in finding frequent subgraphs:
There are two steps in finding frequent subgraphs.

- The first step is to create frequent substructure candidates.
- The second step is to find the support of each and every candidate. We must optimize and enhance the first step because the second step is an NP-completed set where the computational complexity is accurate and high.

There are two methods for frequent substructure mining.

The Apriori-based approach: The approach to find the frequent graphs begin from the graph with a small size. The approach advances in a bottom-up way by creating candidates with extra vertex or edge. This algorithm is called an **Apriori Graph**. Let us consider Q_k as the frequent sub-structure set with a size of k. This approach acquires a level-wise mining technique. Before the Apriori Graph technique, the generation of candidates must be done. This is done by combining two same but slightly varied frequent subgraphs. After the formation of new substructures, the frequency of the graph is checked. Out of that, the graphs found frequently are used to create the next candidate. This step to generate frequent substructure candidates is a complex step. But, when

it comes to generating candidates in itemset, it is easy and effortless. Let's consider an example of having two itemsets of size three such that and So, the itemset derived using join would be pqrs. But when it comes to substructures, there is more than one method to join two substructures.

2.9 SPM:

Sequential pattern mining is the mining of frequently appearing series events or subsequences as patterns. An instance of a sequential pattern is users who purchase a Canon digital camera are to purchase an HP color printer within a month.

For retail information, sequential patterns are beneficial for shelf placement and promotions. This industry, and telecommunications and different businesses, can also use sequential patterns for targeted marketing, user retention, and several tasks.

There are several areas in which sequential patterns can be used such as Web access pattern analysis, weather prediction, production processes, and web intrusion detection.

Given a set of sequences, where each sequence includes a file of events (or elements) and each event includes a group of items, and given a user-specified minimum provide threshold of min sup, sequential pattern mining discover all frequent subsequences, i.e., the subsequences whose occurrence frequency in the group of sequences is no less than min_sup.

Let $I = \{I_1, I_2,..., I_p\}$ be the set of all items. An itemset is a nonempty set of items. A sequence is an ordered series of events. A sequence s is indicated $\{e_1, e_2, e_3 \dots e_l\}$ where event e_1 appears before e_2, which appears before e_3, etc. Event e_j is also known as element of s.

In the case of user purchase information, an event defines a shopping trip in which a customer purchase items at a specific store. The event is an itemset, i.e., an unordered list of items that the customer purchased during the trip. The itemset (or event) is indicated $(x_1x_2 \cdots x_q)$, where x_k is an item.

An item can appear just once in an event of a sequence, but can appear several times in different events of a sequence. The multiple instances of items in a sequence is known as the length of the sequence. A sequence with length l is known as l-sequence.

A sequence database, S, is a group of tuples, (SID, s), where SID is a sequence_ID and s is a sequence. For instance, S includes sequences for all users of the store. A tuple (SID, s) is include a sequence α, if α is a subsequence of s.

This phase of sequential pattern mining is an abstraction of user-shopping sequence analysis. Scalable techniques for sequential pattern mining on such records are as follows −

There are several sequential pattern mining applications cannot be covered by this phase. For instance, when analyzing Web clickstream series, gaps among clicks become essential if one required to predict what the next click can be.

In DNA sequence analysis, approximate patterns become helpful because DNA sequences can include (symbol) insertions, deletions, and mutations. Such diverse requirements can be considered as constraint relaxation or application.

2.10 Maximal Frequent Item Set:

A maximal frequent itemset is represented as a frequent itemset for which none of its direct supersets are frequent. The itemsets in the lattice are broken into two groups such as those that are frequent and those that are infrequent. A frequent itemset border, which is defined by a dashed line.

2.9 Closed frequent item set:

For example, the itemset {2, 3, 5} has a support of 3 because it appears in transactions t2, t3 and t5. The itemset {2, 3, 5} is a frequent itemset because its support is higher or equal to the minsup parameter. Furthermore, it is a closed itemsets because it has no proper superset having exactly the same support.

MODULE- III

Classification

Classification: Classification and Prediction– Basic concepts–Decision tree induction–Bayesian classification, Rule–basedclassification, Lazylearner.

3.1 Classification and Prediction:

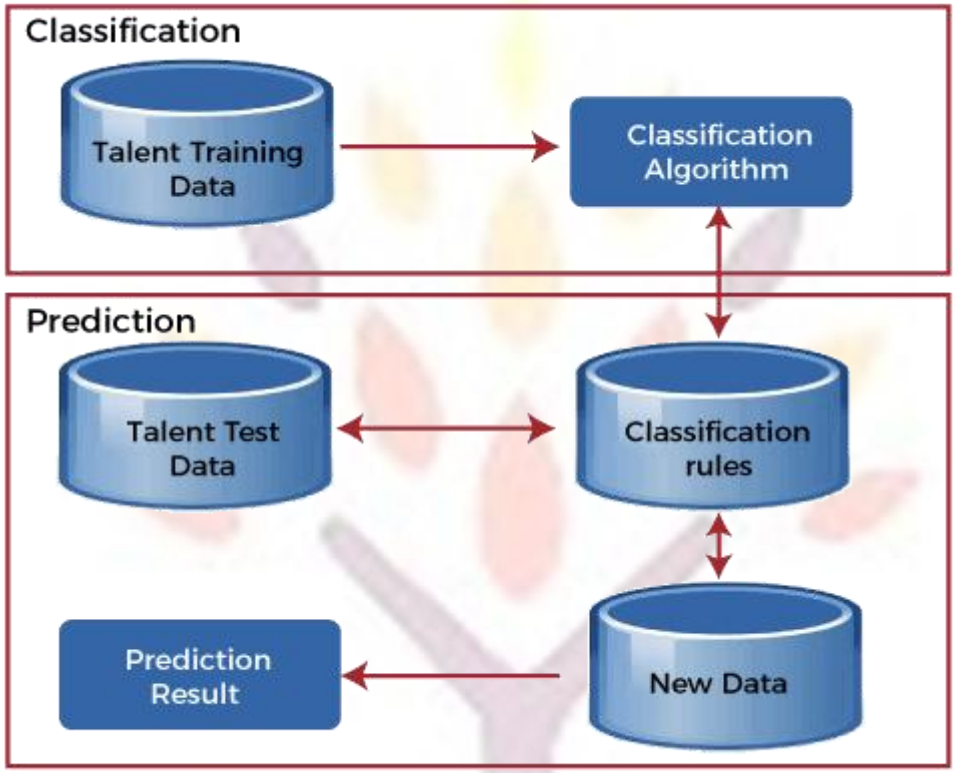
There are two forms of data analysis that can be used to extract models describing important classes or predict future data trends. These two forms are as follows:

1. Classification
2. Prediction

We use classification and prediction to extract a model, representing the data classes to predict future data trends. Classification predicts the categorical labels of data with the prediction models. This analysis provides us with the best understanding of the data at a large scale.

Classification models predict categorical class labels, and prediction models predict continuous-valued functions. For example, we can build a classification model to categorize bank loan

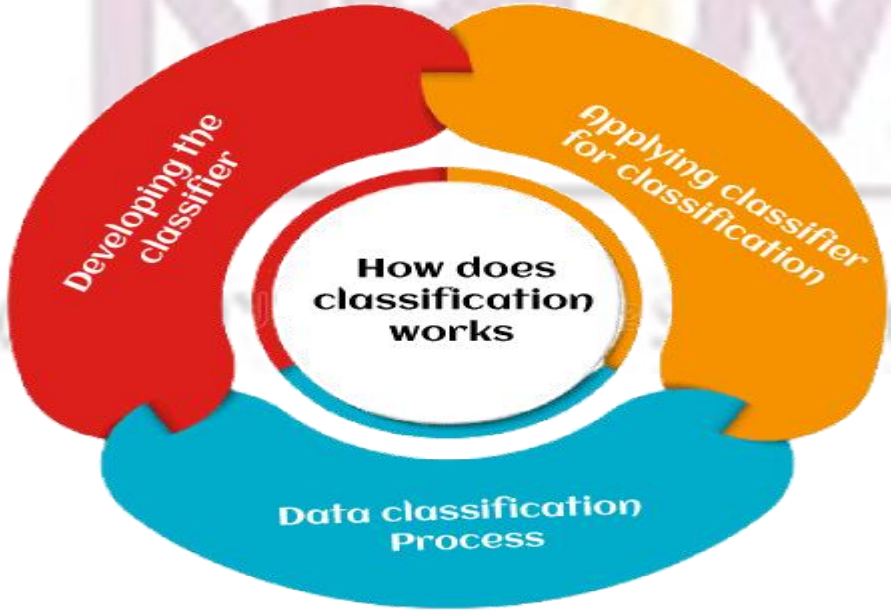
applications as either safe or risky or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.



Classification and Prediction Process

3.2 General Approaches to solving a classification problem:

The functioning of classification with the assistance of the bank loan application has been mentioned above. There are two stages in the data classification system: classifier or model creation and classification classifier.



1. **Developing the Classifier or model creation:** This level is the learning stage or the learning process. The classification algorithms construct the classifier in this stage. A classifier is constructed from a training set composed of the records of databases and their corresponding class names. Each category that makes up the training set is referred to as a category or class. We may also refer to these records as samples, objects, or data points.
2. **Applying classifier for classification:** The classifier is used for classification at this level. The test data are used here to estimate the accuracy of the classification algorithm. If the consistency is deemed sufficient, the classification rules can be expanded to cover new data records. It includes:
    - **Sentiment Analysis:** Sentiment analysis is highly helpful in social media monitoring. We can use it to extract social media insights. We can build sentiment analysis models to read and analyze misspelled words with advanced machine learning algorithms. The accurate trained models provide consistently accurate outcomes and result in a fraction of the time.
    - **Document Classification:** We can use document classification to organize the documents into sections according to the content. Document classification refers to text classification; we can classify the words in the entire document. And with the help of machine learning classification algorithms, we can execute it automatically.
    - **Image Classification:** Image classification is used for the trained categories of an image. These could be the caption of the image, a statistical value, a theme. You can tag images to train your model for relevant categories by applying supervised learning algorithms.
    - **Machine Learning Classification:** It uses the statistically demonstrable algorithm rules to execute analytical tasks that would take humans hundreds of more hours to perform.
3. **Data Classification Process:** The data classification process can be categorized into five steps:
    - Create the goals of data classification, strategy, workflows, and architecture of data classification.
    - Classify confidential details that we store.
    - Using marks by data labelling.
    - To improve protection and obedience, use effects.
    - Data is complex, and a continuous method is a classification.

3.3 Evaluation of Classifiers:

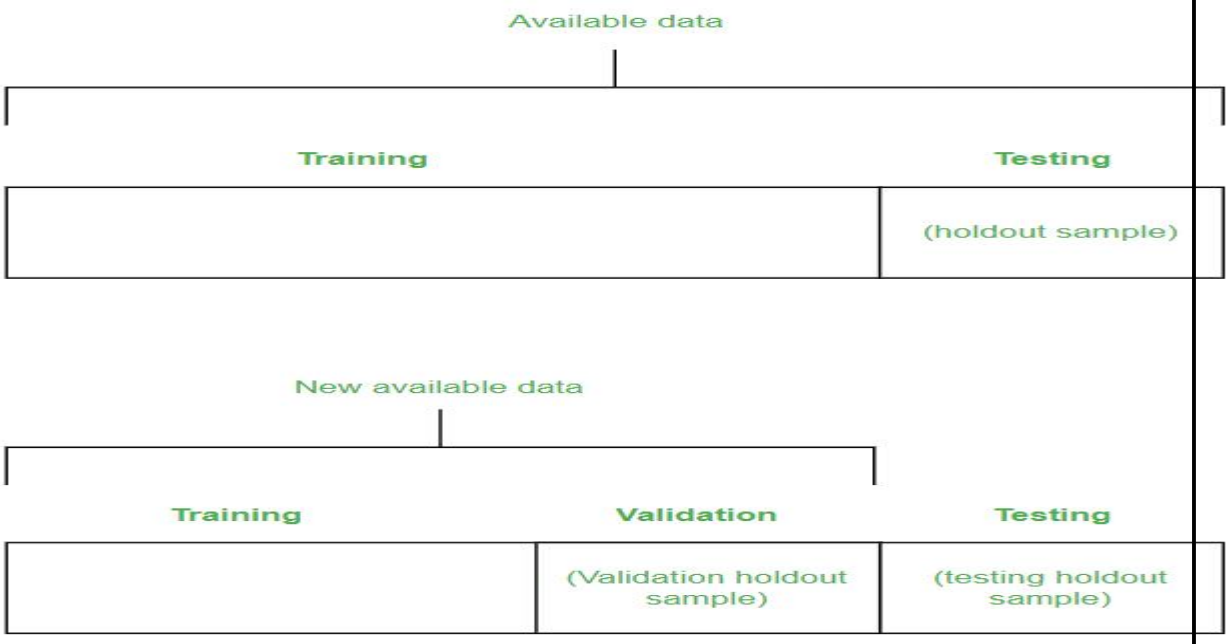
Data Mining can be referred to as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging. In this article, we will see techniques to evaluate the accuracy of classifiers.

HoldOut

In the holdout method, the largest dataset is randomly divided into three subsets:

- **A training set** is a subset of the dataset which are been used to build predictive models.
- **The validation set** is a subset of the dataset which is been used to assess the performance of the model built in the training phase. It provides a test platform for fine-tuning of the model's parameters and selecting the best-performing model. It is not necessary for all modeling algorithms to need a validation set.
- **Test sets or unseen examples** are the subset of the dataset to assess the likely future performance of the model. If a model is fitting into the training set much better than it fits into the test set, then overfitting is probably the cause that occurred here.

Basically, two-thirds of the data are been allocated to the training set and the remaining one-third is been allocated to the test set.



Random Subsampling

- Random subsampling is a variation of the holdout method. The holdout method is been repeated K times.
- The holdout subsampling involves randomly splitting the data into a training set and a test set.

- On the training set the data is been trained and the mean square error (MSE) is been obtained from the predictions on the test set.
- As MSE is dependent on the split, this method is not recommended. So a new split can give you a new MSE.
- The overall accuracy is been calculated as E = 1/K \sum_{k}^{i=1} E_{i}

Cross-Validation

- K-fold cross-validation is been used when there is only a limited amount of data available, to achieve an unbiased estimation of the performance of the model.
- Here, we divide the data into K subsets of equal sizes.
- We build models K times, each time leaving out one of the subsets from the training, and use it as the test set.
- If K equals the sample size, then this is called a "Leave-One-Out"

Bootstrapping

- Bootstrapping is one of the techniques which is used to make the estimations from the data by taking an average of the estimates from smaller data samples.
- The bootstrapping method involves the iterative resampling of a dataset with replacement.
- On resampling instead of only estimating the statistics once on complete data, we can do it many times.
- Repeating this multiple times helps to obtain a vector of estimates.
- Bootstrapping can compute variance, expected value, and other relevant statistics of these estimates.

3.4 Classification techniques:

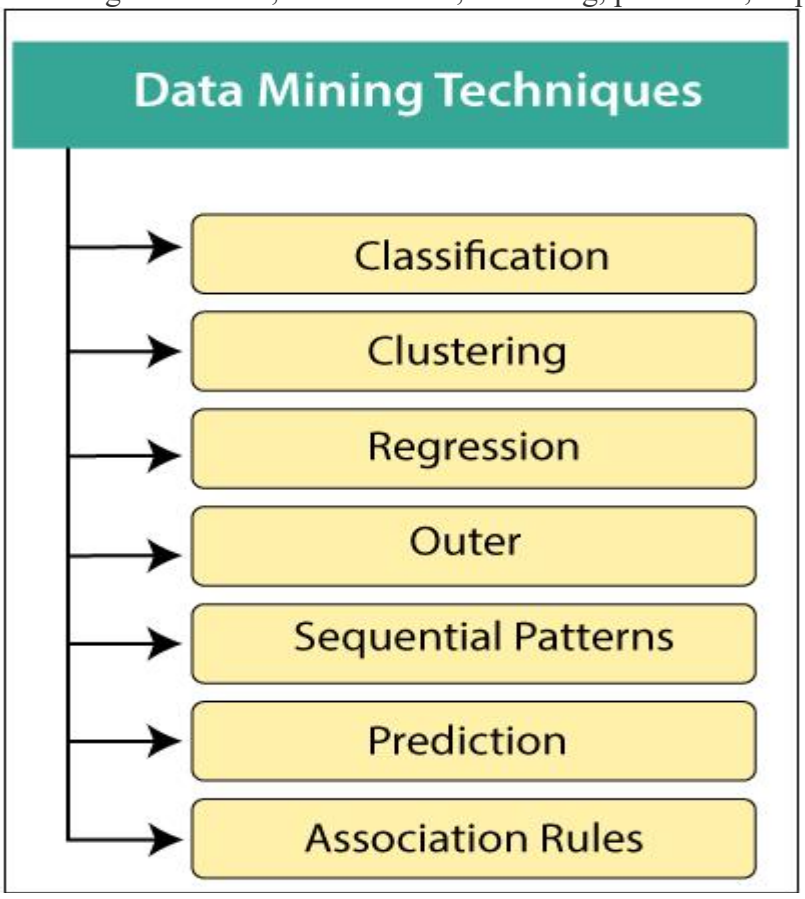
Classification is a task in data mining that involves assigning a class label to each instance in a dataset based on its features. The goal of classification is to build a model that accurately predicts the class labels of new instances based on their features.

- There are two main types of classification: binary classification and multi-class classification. Binary classification involves classifying instances into two classes, such as "spam" or "not spam", while multi-class classification involves classifying instances into more than two classes.

- The process of building a classification model typically involves the following steps

- **Data Collection:**
  The first step in building a classification model is data collection. In this step, the data relevant to the problem at hand is collected. The data should be representative of the problem and should contain all the necessary attributes and labels needed for classification. The data can be collected from various sources, such as surveys, questionnaires, websites, and databases.

In recent data mining projects, various major data mining techniques have been developed and used, including association, classification, clustering, prediction, sequential patterns, and regression.



1. Classification:

This technique is used to obtain important and relevant information about data and metadata. This data mining technique helps to classify data in different classes.

Data mining techniques can be classified by different criteria, as follows:

i. **Classification of Data mining frameworks as per the type of data sources mined:** This classification is as per the type of data handled. For example, multimedia, spatial data, text data, time-series data, World Wide Web, and so on..

ii. **Classification of data mining frameworks as per the database involved:** This classification based on the data model involved. For example. Object-oriented database, transactional database, relational database, and so on..

iii. **Classification of data mining frameworks as per the kind of knowledge discovered:** This classification depends on the types of knowledge discovered or data mining functionalities. For example, discrimination, classification, clustering, characterization, etc. some frameworks tend to be extensive frameworks offering a few data mining functionalities together..

iv. **Classification of data mining frameworks according to data mining techniques used:** This classification is as per the data analysis approach utilized, such as neural networks, machine learning, genetic algorithms, visualization, statistics, data warehouse-oriented or database-oriented, etc.
The classification can also take into account, the level of user interaction involved in the data mining procedure, such as query-driven systems, autonomous systems, or interactive exploratory systems.

2. Clustering:

Clustering is a division of information into groups of connected objects. Describing the data by a few clusters mainly loses certain confine details, but accomplishes improvement. It models data by its clusters. Data modeling puts clustering from a historical point of view rooted in statistics, mathematics, and numerical analysis. From a machine learning point of view, clusters relate to hidden patterns, the search for clusters is unsupervised learning, and the subsequent framework represents a data concept. From a practical point of view, clustering plays an extraordinary job in data mining applications. For example, scientific data exploration, text mining, information retrieval, spatial database applications, CRM, Web analysis, computational biology, medical diagnostics, and much more.

In other words, we can say that Clustering analysis is a data mining technique to identify similar data. This technique helps to recognize the differences and similarities between the data. Clustering is very similar to the classification, but it involves grouping chunks of data together based on their similarities.

3. Regression:

Regression analysis is the data mining process is used to identify and analyze the relationship between variables because of the presence of the other factor. It is used to define the probability of the specific variable. Regression, primarily a form of planning and modeling. For example, we might use it to project certain costs, depending on other factors such as availability, consumer demand, and

competition. Primarily it gives the exact relationship between two or more variables in the given data set.

4. Association Rules:

This data mining technique helps to discover a link between two or more items. It finds a hidden pattern in the data set.

Association rules are if-then statements that support to show the probability of interactions between data items within large data sets in different types of databases. Association rule mining has several applications and is commonly used to help sales correlations in data or medical data sets.

The way the algorithm works is that you have various data, For example, a list of grocery items that you have been buying for the last six months. It calculates a percentage of items being purchased together.

These are three major measurements technique:

- **Lift:**
  This measurement technique measures the accuracy of the confidence over how often item B is purchased.
  (Confidence) / (item B)/ (Entire dataset)
- **Support:**
  This measurement technique measures how often multiple items are purchased and compared it to the overall dataset.
  (Item A + Item B) / (Entire dataset)
- **Confidence:**
  This measurement technique measures how often item B is purchased when item A is purchased as well.
  (Item A + Item B)/ (Item A)

5. Outer detection:

This type of data mining technique relates to the observation of data items in the data set, which do not match an expected pattern or expected behavior. This technique may be used in various domains like intrusion, detection, fraud detection, etc. It is also known as Outlier Analysis or Outilier mining. The outlier is a data point that diverges too much from the rest of the dataset. The majority of the real-world datasets have an outlier. Outlier detection plays a significant role in the data mining field. Outlier detection is valuable in numerous fields like network interruption identification, credit or debit card fraud detection, detecting outlying in wireless sensor network data, etc.

6. Sequential Patterns:

The sequential pattern is a data mining technique specialized for **evaluating sequential data** to discover sequential patterns. It comprises of finding interesting subsequences in a set of sequences, where the stake of a sequence can be measured in terms of different criteria like length, occurrence frequency, etc.

In other words, this technique of data mining helps to discover or recognize similar patterns in transaction data over some time.

7. Prediction:

- Prediction used a combination of other data mining techniques such as trends, clustering, classification, etc. It analyzes past events or instances in the right sequence to predict a future event.

3.5 Decision Trees-Decision tree Construction:

A decision tree is a flowchart-like tree structure where each internal node denotes the feature, branches denote the rules and the leaf nodes denote the result of the algorithm. It is a versatile supervised machine-learning algorithm, which is used for both classification and regression problems. It is one of the very powerful algorithms. And it is also used in Random Forest to train on different subsets of training data, which makes random forest one of the most powerful algorithms in machine learning.

Decision Tree Terminologies

Some of the common Terminologies used in Decision Trees are as follows:

- **Root Node:** It is the topmost node in the tree, which represents the complete dataset. It is the starting point of the decision-making process.
- Decision/Internal Node: A node that symbolizes a choice regarding an input feature. Branching off of internal nodes connects them to leaf nodes or other internal nodes.
- **Leaf/Terminal Node:** A node without any child nodes that indicates a class label or a numerical value.
- **Splitting:** The process of splitting a node into two or more sub-nodes using a split criterion and a selected feature.
- **Branch/Sub-Tree:** A subsection of the decision tree starts at an internal node and ends at the leaf nodes.
- **Parent Node:** The node that divides into one or more child nodes.
- **Child Node:** The nodes that emerge when a parent node is split.
- **Impurity**: A measurement of the target variable's homogeneity in a subset of data. It refers to the degree of randomness or uncertainty in a set of examples. The **Gini index** and **entropy** are two commonly used impurity measurements in decision trees for classifications task
- **Variance**: Variance measures how much the predicted and the target variables vary in different samples of a dataset. It is used for regression problems in decision trees. **Mean squared error, Mean Absolute Error, friedman_mse, or Half Poisson deviance** are used to measure the variance for the regression tasks in the decision tree.
- **Information Gain:** Information gain is a measure of the reduction in impurity achieved by splitting a dataset on a particular feature in a decision tree. The splitting criterion is determined by the feature that offers the greatest information gain, It is used to determine the most informative feature to split on at each node of the tree, with the goal of creating pure subsets

- **Pruning**: The process of removing branches from the tree that do not provide any additional information or lead to overfitting.

3.6 Methods for Expressing attribute test conditions:

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a sequential diagram-like tree structure, where every internal node (non-leaf node) indicates a test on an attribute, each branch defines a result of the test, and each leaf node (or terminal node) influences a class label. The largest node in a tree is the root node.

Decision tree induction generates a flowchart-like structure where each internal (non-leaf) node indicates a test on an attribute, each branch corresponds to a result of the test, and each external (leaf) node indicates a class prediction.

At each node, the algorithm selects the "best" attribute to divide the data into single classes. When decision tree induction is used for attribute subset selection, a tree is generated from the given data.

Some attributes that do not occur in the tree are considered to be irrelevant. The set of attributes occurring in the tree forms the decreased subset of attributes. Decision tree induction algorithms support an approach for defining an attribute test condition and its correlating results for multiple attribute types.

Binary Attributes − A binary attribute is a nominal attribute with only two elements or states including 0 or 1, where 0 frequently represents that the attribute is absent, and 1 represents that it is present. Binary attributes are defined as Boolean if the two states are equivalent to true and false.

A binary attribute is symmetric if both of its states are equal valuable and make an equal weight. There is no preference on which results must be coded as 0 or 1. An example can be the attribute gender having the states male and female.

A binary attribute is asymmetric if the outcomes of the states are not equally essential, such as the positive and negative outcomes of a medical check for HIV. By convention, it can code the most essential result, which is generally the nearest one, by 1 (e.g., HIV positive) and the different by 0 (e.g., HIV negative).

Nominal Attributes − Nominal defines associating with names. The values of a nominal attribute are symbols or names of things. Each value defines some type of category, code, or state, etc. Nominal attributes are defined as categorical. The values do not have any significant order. In computer science, the values are also called enumerations.

Ordinal Attributes − An ordinal attribute is an attribute with applicable values that have an essential series or ranking among them, but the magnitude between successive values is unknown.

Ordinal attributes can make binary or multiway splits. Ordinal attribute values can be combined considering the grouping does not violate the order nature of the attribute values.

Numeric Attributes − A numeric attribute is quantitative. It is a computable quantity, represented in numerical or real values. It can be interval-scaled or ratio-scaled.

3.7 Algorithm for Decision tree Induction:

The decision tree algorithm may appear long, but it is quite simply the basis algorithm techniques is as follows:

The **algorithm** is based on three parameters**: D, attribute_list, and Attribute _selection_method.**

Generally, we refer to **D** as a **data partition.**

Initially, **D** is the entire set of **training tuples** and their related **class levels** (input training data).

The parameter **attribute_list** is a set of **attributes** defining the tuples.

Attribute_selection_method specifies a **heuristic process** for choosing the attribute that "best" discriminates the given tuples according to **class**.

Attribute_selection_method process applies an **attribute selection measure**.

Advantages of using decision trees:

A decision tree does not need scaling of information.

Missing values in data also do not influence the process of building a choice tree to any considerable extent.

A decision tree model is automatic and simple to explain to the technical team as well as stakeholders.

Compared to other algorithms, decision trees need less exertion for data preparation during pre-processing.

A decision tree does not require a standardization of data.

3.8 Naive-Bayes Classifier:

This article discusses the theory behind the Naive Bayes classifiers and their implementation.

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.
To start with, let us consider a dataset.

Consider a fictional dataset that describes the weather conditions for playing a game of golf. Given the weather conditions, each tuple classifies the conditions as fit("Yes") or unfit("No") for playing golf.

Here is a tabular representation of our dataset.

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

The dataset is divided into two parts, namely, **feature matrix** and the **response vector**.

- Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of **dependent features**. In above dataset, features are 'Outlook', 'Temperature', 'Humidity' and 'Windy'.
- Response vector contains the value of **class variable**(prediction or output) for each row of feature matrix. In above dataset, the class variable name is 'Play golf'.

Assumption:

The fundamental Naive Bayes assumption is that each feature makes an:

- independent
- equal

contribution to the outcome.

With relation to our dataset, this concept can be understood as:

- We assume that no pair of features are dependent. For example, the temperature being 'Hot' has nothing to do with the humidity or the outlook being 'Rainy' has no effect on the winds. Hence, the features are assumed to be **independent**.
- Secondly, each feature is given the same weight(or importance). For example, knowing only temperature and humidity alone can't predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing **equally** to the outcome.

Note: The assumptions made by Naive Bayes are not generally correct in real-world situations. In-fact, the independence assumption is never correct but often works well in practice.
Now, before moving to the formula for Naive Bayes, it is important to know about Bayes' theorem.

Bayes' Theorem

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

where A and B are events and $P(B) \neq 0$.

- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as **evidence**.
- P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence isseen.

Now, with regards to our dataset, we can apply Bayes' theorem in following way:

where, y is class variable and X is a dependent feature vector (of size *n*) where:
Just to clear, an example of a feature vector and corresponding class variable can be: (refer 1st row of dataset)

X = (Rainy, Hot, High, False)
y = No
So basically, P(y|X) here means, the probability of "Not playing golf" given that the weather conditions are "Rainy outlook", "Temperature is hot", "high humidity" and "no wind".

Naive assumption

Now, its time to put a naive assumption to the Bayes' theorem, which is, **independence** among the features. So now, we split **evidence** into the independent parts.
Now, if any two events A and B are independent, then,

P(A,B) = P(A)P(B)
Hence, we reach to the result:

which can be expressed as:

Now, as the denominator remains constant for a given input, we can remove that term:

Now, we need to create a classifier model. For this, we find the probability of given set of inputs for all possible values of the class variable *y* and pick up the output with maximum probability. This can be expressed mathematically as:

So, finally, we are left with the task of calculating P(y) and P(x_i | y).
Please note that P(y) is also called **class probability** and P(x_i | y) is called **conditional probability**.
The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of P(x_i | y).
Let us try to apply the above formula manually on our weather dataset. For this, we need to do some precomputations on our dataset.

We need to find P(x_i | y_j) for each x_i in X and y_j in y. All these calculations have been demonstrated in the tables below:

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

So, in the figure above, we have calculated P(x_i | y_j) for each x_i in X and y_j in y manually in the tables 1-4. For example, probability of playing golf given that the temperature is cool, i.e P(temp. = cool | play golf = Yes) = 3/9.
Also, we need to find class probabilities (P(y)) which has been calculated in the table 5. For example, P(play golf = Yes) = 9/14.

So now, we are done with our pre-computations and the classifier is ready!

Let us test it on a new set of features (let us call it today):

today = (Sunny, Hot, Normal, False)
So, probability of playing golf is given by:

and probability to not play golf is given by:

Since, P(today) is common in both probabilities, we can ignore P(today) and find proportional probabilities as:

and

Now, since

These numbers can be converted into a probability by making the sum equal to 1 (normalization):

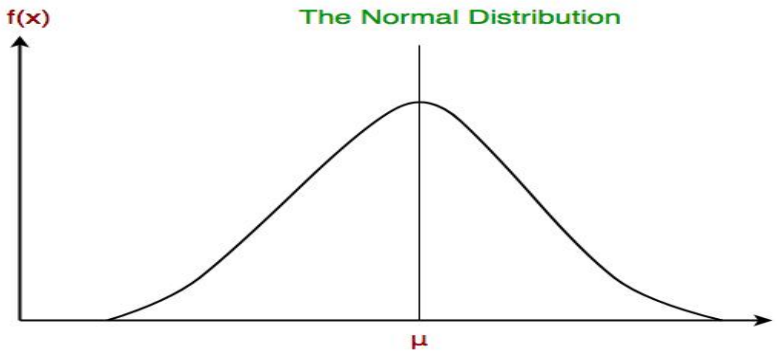
And Since

So, prediction that golf would be played is 'Yes'.

The method that we discussed above is applicable for discrete data. In case of continuous data, we need to make some assumptions regarding the distribution of values of each feature. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i \mid y)$.
Now, we discuss one of such classifiers here.

Gaussian Naive Bayes classifier

In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution**. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values as shown below:



Updated table of prior probabilities for outlook feature is as following:

The likelihood of the features is assumed to be Gaussian, hence, conditional probability is given by:

Now, we look at an implementation of Gaussian Naive Bayes classifier using scikit-learn.

3.9 Bayesian BeliefNetworks:

Bayesian classification is based on Bayes' Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Baye's Theorem

Bayes' Theorem is named after Thomas Bayes. There are two types of probabilities −

- Posterior Probability [P(H/X)]
- Prior Probability [P(H)]

where X is data tuple and H is some hypothesis.

According to Bayes' Theorem,

$$P(H/X)= P(X/H)P(H) / P(X)$$

Bayesian Belief Network

Bayesian Belief Networks specify joint conditional probability distributions. They are also known as Belief Networks, Bayesian Networks, or Probabilistic Networks.

- A Belief Network allows class conditional independencies to be defined between subsets of variables.
- It provides a graphical model of causal relationship on which learning can be performed.
- We can use a trained Bayesian Network for classification.

There are two components that define a Bayesian Belief Network −

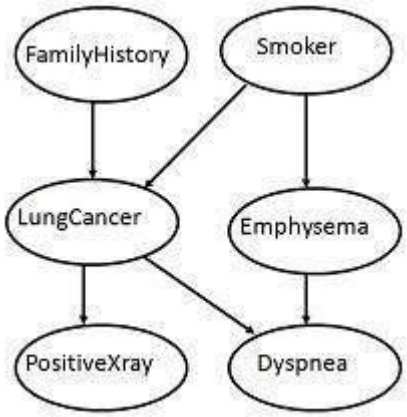
- Directed acyclic graph
- A set of conditional probability tables

Directed Acyclic Graph

- Each node in a directed acyclic graph represents a random variable.
- These variable may be discrete or continuous valued.
- These variables may correspond to the actual attribute given in the data.

Directed Acyclic Graph Representation

The following diagram shows a directed acyclic graph for six Boolean variables.



The arc in the diagram allows representation of causal knowledge. For example, lung cancer is influenced by a person's family history of lung cancer, as well as whether or not the person is a

smoker. It is worth noting that the variable PositiveXray is independent of whether the patient has a family history of lung cancer or that the patient is a smoker, given that we know the patient has lung cancer.

Conditional Probability Table

The conditional probability table for the values of the variable LungCancer (LC) showing each possible combination of the values of its parent nodes, FamilyHistory (FH), and Smoker (S) is as follows −

	FH,S	FH,-S	-FH,S	-FH,S
LC	0.8	0.5	0.7	0.1
-LC	0.2	0.5	0.3	0.9

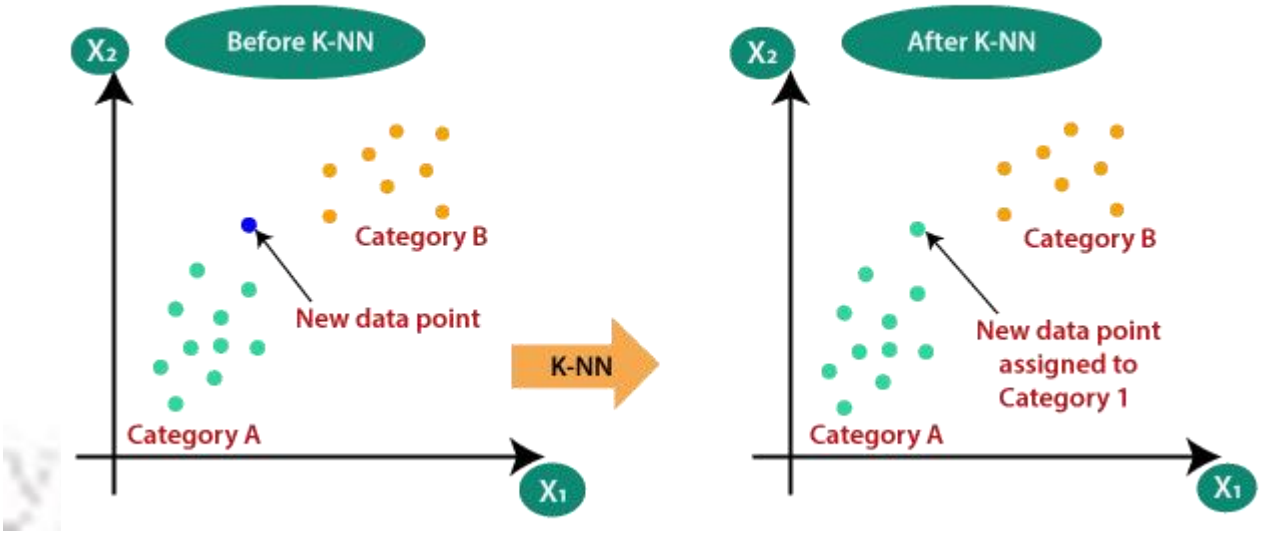
3.10 K-Nearest neighbor classification – Algorithm:

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features

of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

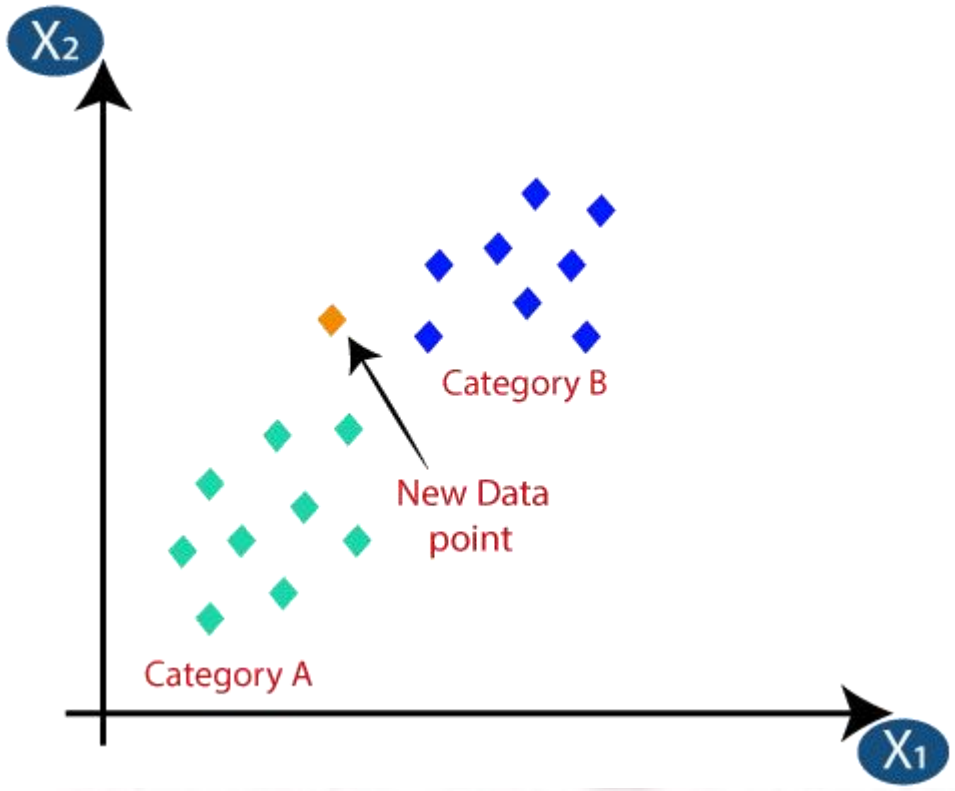


How does K-NN work?

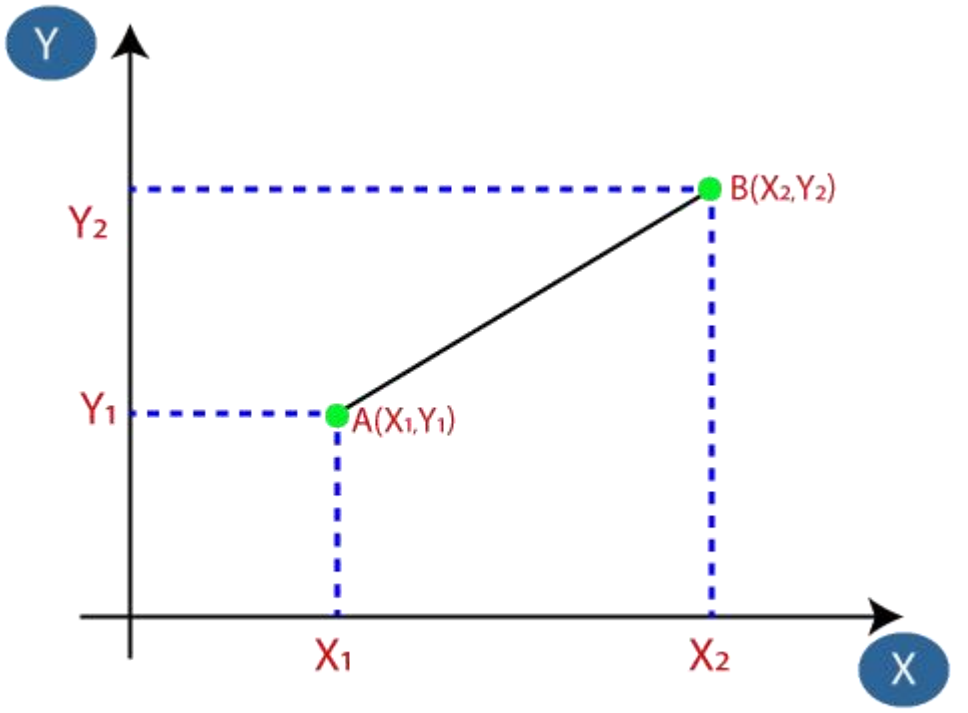
The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

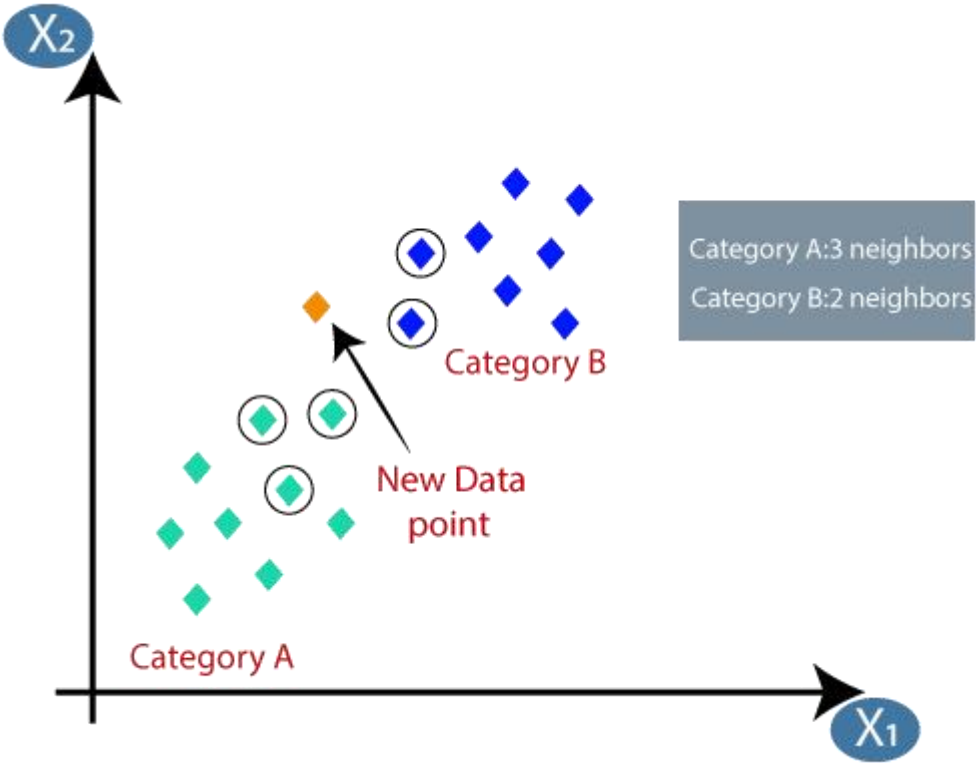


- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

3.11 Lazylearner:

Lazy learning algorithms work by memorizing the training data rather than constructing a general model.

Lazy learning is a type of machine learning that doesn't process training data until it needs to make a prediction. Instead of building models during training, lazy learning algorithms wait until they encounter a new query. This method stores and compares training examples when making predictions. It's also called instance-based or memory-based learning.

Lazy Learning Explained

Lazy learning algorithms work by memorizing the training data rather than constructing a general model. When a new query is received, lazy learning retrieves similar instances from the training set and uses them to generate a prediction. The similarity between instances is usually calculated using distance metrics, such as Euclidean distance or cosine similarity.

One of the most popular lazy learning algorithms is the **k-nearest neighbors** (k-NN) algorithm. In k-NN, the k closest training instances to the query point are considered, and their class labels are used to determine the class of the query. Lazy learning methods excel in situations where the underlying data distribution is complex or where the training data is noisy.

Examples of Real-World Lazy Learning Applications

Lazy learning has found applications in various domains. Here are a few examples:

Recommendation systems. Lazy learning is widely used in recommender systems to provide personalized recommendations. By comparing user preferences to similar users in the training set, lazy learning algorithms can suggest items or products of interest, such as movies, books, or products.

Medical diagnosis. Lazy learning can be employed in medical diagnosis systems. By comparing patient symptoms and medical histories to similar cases in the training data, lazy learning algorithms can assist in diagnosing diseases or suggesting appropriate treatments.

Anomaly detection. Lazy learning algorithms are useful for detecting anomalies or outliers in datasets. For example, an algorithm can detect credit card fraud by comparing a transaction to nearby transactions based on factors like location and history. If the transaction is unusual, such as being made in a faraway location for a large amount, it may be flagged as fraudulent.

MODULE- IV

Clustering

Clustering and Applications: Cluster analysis–Types of Data inCluster Analysis–Categorization of Major Clustering Methods–Partitioning Methods, Hierarchical Methods– Density–BasedMethods, Grid–Based Methods, Outlier Analysis.

4.1 Cluster analysis:

The process of making a group of abstract objects into classes of similar objects is known as clustering.

Points to Remember:

One group is treated as a cluster of data objects

- In the process of cluster analysis, the first step is to partition the set of data into groups with the help of data similarity, and then groups are assigned to their respective labels.
- The biggest advantage of clustering over-classification is it can adapt to the changes made and helps single out useful features that differentiate different groups.

Applications of cluster analysis :

- It is widely used in many applications such as image processing, data analysis, and pattern recognition.
- It helps marketers to find the distinct groups in their customer base and they can characterize their customer groups by using purchasing patterns.
- It can be used in the field of biology, by deriving animal and plant taxonomies and identifying genes with the same capabilities.
- It also helps in information discovery by classifying documents on the web.

4.2Types of Data inCluster Analysis :

It can be classified based on the following categories.

1. Model-Based Method
2. Hierarchical Method
3. Constraint-Based Method
4. Grid-Based Method
5. Partitioning Method
6. Density-Based Method

Requirements of clustering in data mining:

The following are some points why clustering is important in data mining.

- **Scalability –** we require highly scalable clustering algorithms to work with large databases.
- **Ability to deal with different kinds of attributes –** Algorithms should be able to work with the type of data such as categorical, numerical, and binary data.
- **Discovery of clusters with attribute shape –** The algorithm should be able to detect clusters in arbitrary shapes and it should not be bounded to distance measures.
- **Interpretability –** The results should be comprehensive, usable, and interpretable.
- **High dimensionality –** The algorithm should be able to handle high dimensional space instead of only handling low dimensional data.

4.3 Categorization of Major 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.

Points to remember −

- 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

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.

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.

4.4 Evaluation of Clustering Algorithms:

There are no precise labels or established ground truths that can be utilized to assess the clustering findings, making the assessment of clustering models difficult. According to their attributes and goals, many measures have therefore been developed to assess the effectiveness of clustering methods. Many often employed metrics include −

Silhouette Score

Based on its closeness to other data points in that cluster as well as to data points in other clusters, each data point's silhouette score evaluates how well it fits into the cluster to which it has been allocated. A score of 1 means the data point is well-clustered, whereas a value of -1 means it has been misclassified. The silhouette score goes from -1 to 1.

Calinski-Harabasz Index

A higher index value implies greater clustering performance. The Calinski-Harabasz index evaluates the ratio of between-cluster variation to within-cluster variance.

Davies-Bouldin index

A lower Davies-Bouldin index suggests greater clustering performance since it gauges the average similarity between each cluster and its most comparable cluster.

Rand Index

A higher Rand index denotes better clustering performance. It quantifies the similarity between the anticipated grouping and the ground truth clustering.

Adjusted Mutual Information (AMI)

A higher index implies greater clustering performance. The AMI evaluates the mutual information between the expected clustering and the ground truth clustering, corrected for the chance.

Choosing the Right Evaluation Metric

The nature and objectives of a clustering problem will dictate the most appropriate assessment measure to employ. If the goal of clustering is to group similar data points together, the Calinski-Harabasz index or the silhouette score can be beneficial. If the clustering results need to be compared to ground truth clustering, however, the Rand index or AMI would be more appropriate. So, it is important to consider the objectives and constraints of the clustering issue while selecting the evaluation metric.

Evaluating the Stability of Clustering Results

Clustering has certain challenges since the parameters of the algorithm and the initial conditions may affect the results. It is essential to execute the clustering technique repeatedly using multiple random initializations or settings in order to judge the sustainability of the clustering findings. One can

evaluate the stability of the clustering results using metrics such as the Jaccard index or the variance of information.

Visualizing the Clustering Results

An understanding of the data's structure and patterns can be gained by visualizing the clustering findings. Using scatter plots or heat maps, where each data point is depicted as a point or a cell with a color-coded depending on its cluster assignment, is one approach to see the clustering findings. In order to project the high-dimensional data into a lower-dimensional space and show the clusters, dimensionality reduction techniques like principal component analysis (PCA) or t-SNE can be used. In addition, visualization tools like dendrograms or silhouette plots are frequently included in cluster analysis software packages allowing users to explore the clustering outcomes.

4.3 Partitioning Clustering-K-Means Algorithm:

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-Medoids), CLARA algorithm (Clustering Large Applications) etc. In this article, we will be seeing the working of K Mean algorithm in detail. **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 2 until no change occurs.

5. 16, 16, 17, 20, 20, 21, 21, 22, 23, 29, 36, 41, 42, 43, 44, 45, 61, 62, 66
6. **Initial Cluster:**
7. K=2
8. Centroid(C1) = 16 [16]
9. Centroid(C2) = 22 [22]
10. **Note:** These two points are chosen randomly from the dataset. **Iteration-1:**
11. C1 = 16.33 [16, 16, 17]
12. C2 = 37.25 [20, 20, 21, 21, 22, 23, 29, 36, 41, 42, 43, 44, 45, 61, 62, 66]
13. **Iteration-2:**
14. C1 = 19.55 [16, 16, 17, 20, 20, 21, 21, 22, 23]
15. C2 = 46.90 [29, 36, 41, 42, 43, 44, 45, 61, 62, 66]
16. **Iteration-3:**
17. C1 = 20.50 [16, 16, 17, 20, 20, 21, 21, 22, 23, 29]
18. C2 = 48.89 [36, 41, 42, 43, 44, 45, 61, 62, 66]
19. **Iteration-4:**
20. C1 = 20.50 [16, 16, 17, 20, 20, 21, 21, 22, 23, 29]
21. C2 = 48.89 [36, 41, 42, 43, 44, 45, 61, 62, 66]
22. 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.

4.4 K- Means Additional issues:

There are various issues of the K-Means Algorithm which are as follows −

Handling Empty Clusters − The first issue with the basic K-means algorithm given prior is that null clusters can be acquired if no points are allocated to a cluster during the assignment phase. If this occurs, then a method is needed to choose a replacement centroid, because the squared error will be larger than necessary.

One method is to select the point that is farthest away from some recent centroid. If this removes the point that currently contributes some total squared error. Another method is to select the replacement centroid from the cluster that has the largest SSE. This will generally divide the cluster and decrease the complete SSE of the clustering. If there are multiple null clusters, then this process can be repeated multiple times.

Outliers − When the squared error method is used, outliers can unduly tend to the clusters that are discovered. In specific, when outliers are present, the resulting cluster centroids (prototypes) cannot be as representative as they can be, and thus, the SSE will be higher as well.

It is beneficial to find outliers and remove them beforehand. It is essential to appreciate that there are specific clustering applications for which outliers should not be removed. When clustering is used for data compression, each point should be clustered, and in some cases, including financial analysis, probable outliers, e.g.,unusually profitable users, can be the interesting points.

Reducing the SSE with Postprocessing − The method to reduce the SSE is to find more clusters, i.e., to need a larger K. In such cases, it is likely to improve the SSE, but don't require to increase the number of clusters. This is possible because Kmeans generally converge to a local minimum.

Various methods are used to "fix-up" the resulting clusters to make a clustering that has lower SSE. The method is to target on individual clusters because the complete SSE is easily the total of the SSE contributed by every cluster. It can change the total SSE by implementing several operations on the clusters, including splitting or merging clusters.

One method is to use an alternate cluster splitting and merging procedure. During a splitting procedure, clusters are divided, while during a merging procedure, clusters are combined. In this method, it is accessible to withdrawal local SSE minima and create a clustering solution with the seized number of clusters. The following are some methods used in the splitting and merging phases which are as follows −

4.5 PAM Algorithm:

There are three types of algorithms for K-Medoids Clustering:

1. **PAM (Partitioning Around Clustering)**
2. **CLARA (Clustering Large Applications)**
3. **CLARANS (Randomized Clustering Large Applications)**

PAM is the most powerful algorithm of the three algorithms but has the disadvantage of time complexity. The following K-Medoids are performed using PAM. In the further parts, we'll see what CLARA and CLARANS are.

Algorithm:

Given the value of k and unlabelled data:

1. Choose k number of random points from the data and assign these k points to k number of clusters. These are the initial medoids.

2. For all the remaining data points, calculate the distance from each medoid and assign it to the cluster with the nearest medoid.

3. Calculate the total cost (Sum of all the distances from all the data points to the medoids)

4. Select a random point as the new medoid and swap it with the previous medoid. Repeat 2 and 3 steps.

5. If the total cost of the new medoid is less than that of the previous medoid, make the new medoid permanent and repeat step 4.

6. If the total cost of the new medoid is greater than the cost of the previous medoid, undo the swap and repeat step 4.

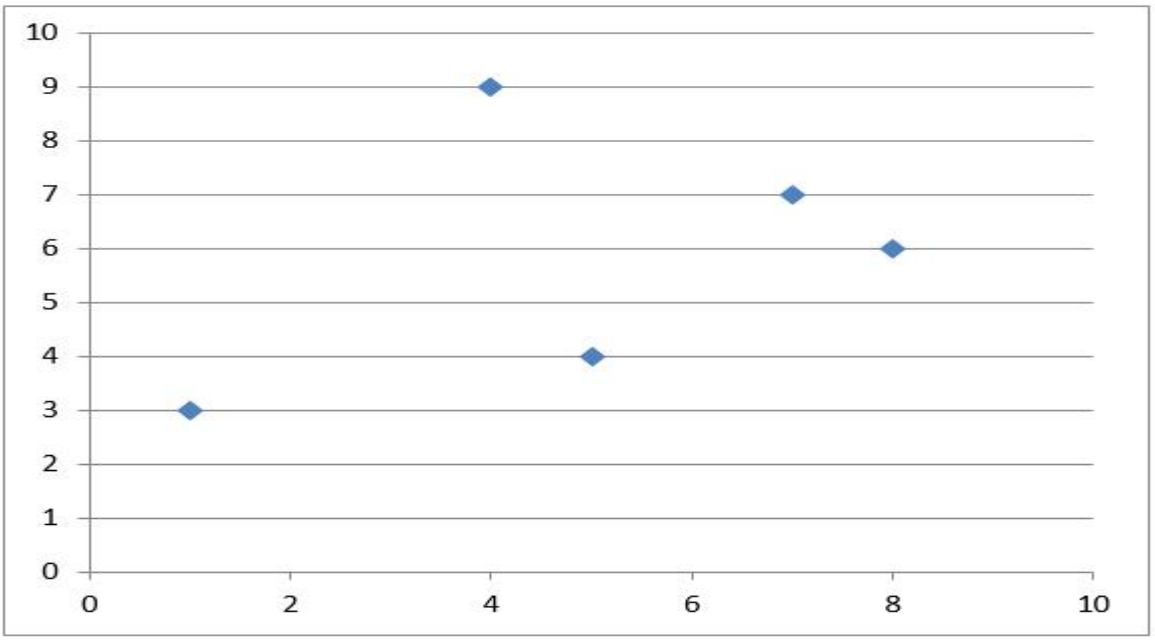
7. The Repetitions have to continue until no change is encountered with new medoids to classify data points.

Here is an example to make the theory clear:

Data set:

	x	y
0	5	4
1	7	7
2	1	3
3	8	6
4	4	9

Scatter plot:



If k is given as 2, we need to break down the data points into 2 clusters.

1. **Initial medoids: M1(1, 3) and M2(4, 9)**
2. Calculation of distances

Manhattan Distance: |x1 - x2| + |y1 - y2|

	x<	y	From M1(1, 3)	From M2(4, 9)
0	5	4	5	6
1	7	7	10	5
2	1	3	-	-
3	8	6	10	7
4	4	9	-	-

Cluster 1: 0

Cluster 2: 1, 3

1. Calculation of total cost:
   (5) + (5 + 7) = 17

2. Random medoid: (5, 4)

M1(5, 4) and M2(4, 9):

	x	y	From M1(5, 4)	From M2(4, 9)
0	5	4	-	-
1	7	7	5	5
2	1	3	5	9
3	8	6	5	7
4	4	9	-	-

Cluster 1: 2, 3

Cluster 2: 1

1. Calculation of total cost:
   (5 + 5) + 5 = 15
   Less than the previous cost
   New medoid: (5, 4).
2. Random medoid: (7, 7)

M1(5, 4) and M2(7, 7)

	x	y	From M1(5, 4)	From M2(7, 7)
0	5	4	-	-
1	7	7	-	-
2	1	3	5	10
3	8	6	5	2
4	4	9	6	5

Cluster 1: 2

Cluster 2: 3, 4

1. Calculation of total cost:
   (5) + (2 + 5) = 12
   Less than the previous cost
   New medoid: (7, 7).
2. Random medoid: (8, 6)

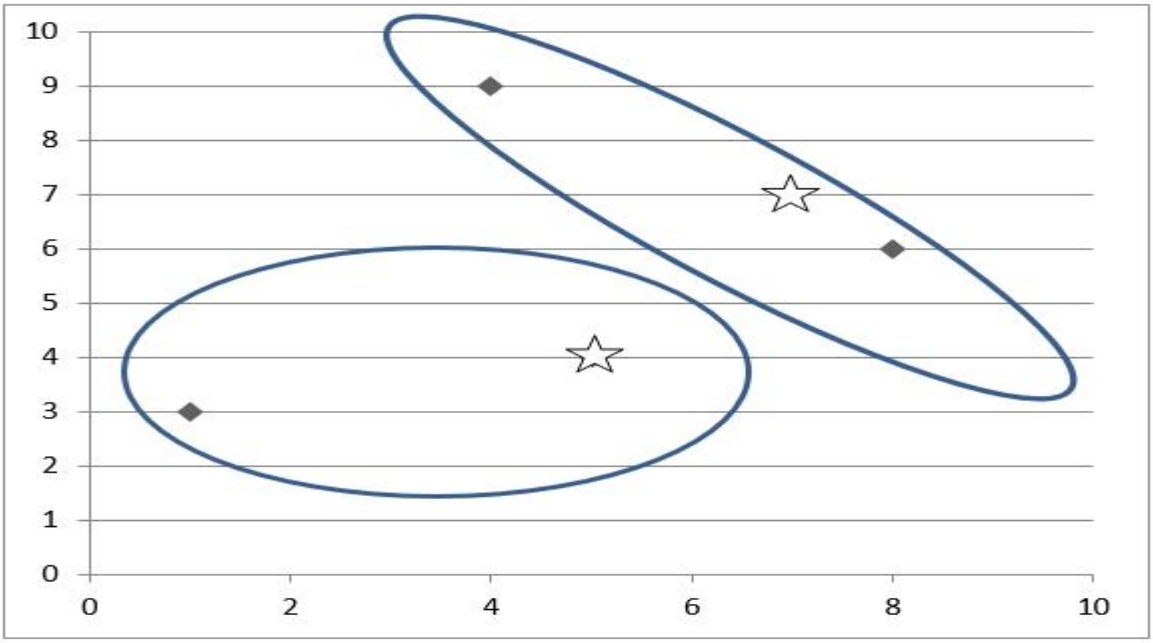
M1(7, 7) and M2(8, 6)

	x	y	From M1(7, 7)	From M2(8, 6)
0	5	4	5	5
1	7	7	-	-
2	1	3	10	10
3	8	6	-	-
4	4	9	5	7

Cluster 1: 4

Cluster 2: 0, 2

1. Calculation of total cost:
   (5) + (5 + 10) = 20
   Greater than the previous cost
   UNDO
   Hence, the final medoids: **M1(5, 4) and M2(7, 7)**
   Cluster 1: 2
   Cluster 2: 3, 4
   Total cost: 12
   Clusters:



Advantages of using K-Medoids:

1. Deals with noise and outlier data effectively
2. Easily implementable and simple to understand
3. Faster compared to other partitioning algorithms

Disadvantages:

1. Not suitable for Clustering arbitrarily shaped groups of data points.
2. As the initial medoids are chosen randomly, the results might vary based on the choice in different runs.

4.6 Basic Agglomerative Hierarchical Clustering Algorithm:

A **Hierarchical clustering** method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly executes the subsequent steps:

1. Identify the 2 clusters which can be closest together, and
2. Merge the 2 maximum comparable clusters. We need to continue these steps until all the clusters are merged together.

In Hierarchical Clustering, the aim is to produce a hierarchical series of nested clusters. A diagram called **Dendrogram** (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) graphically represents this hierarchy and is an inverted tree that describes the order in which factors are merged (bottom-up view) or clusters are broken up (top-down view).

Hierarchical clustering is a method of cluster analysis in data mining that creates a hierarchical representation of the clusters in a dataset. The method starts by treating each data point as a separate cluster and then iteratively combines the closest clusters until a stopping criterion is

reached. The result of hierarchical clustering is a tree-like structure, called a dendrogram, which illustrates the hierarchical relationships among the clusters.

Hierarchical clustering has a number of advantages over other clustering methods, including:

1. The ability to handle non-convex clusters and clusters of different sizes and densities.
2. The ability to handle missing data and noisy data.
3. The ability to reveal the hierarchical structure of the data, which can be useful for understanding the relationships among the clusters. However, it also has some drawbacks, such as:
4. The need for a criterion to stop the clustering process and determine the final number of clusters.
5. The computational cost and memory requirements of the method can be high, especially for large datasets.
6. The results can be sensitive to the initial conditions, linkage criterion, and distance metric used.
   In summary, Hierarchical clustering is a method of data mining that groups similar data points into clusters by creating a hierarchical structure of the clusters.
7. This method can handle different types of data and reveal the relationships among the clusters. However, it can have high computational cost and results can be sensitive to some conditions.

1. Agglomerative: Initially consider every data point as an **individual** Cluster and at every step, **merge** the nearest pairs of the cluster. (It is a bottom-up method). At first, every dataset is considered an individual entity or cluster. At every iteration, the clusters merge with different clusters until one cluster is formed.

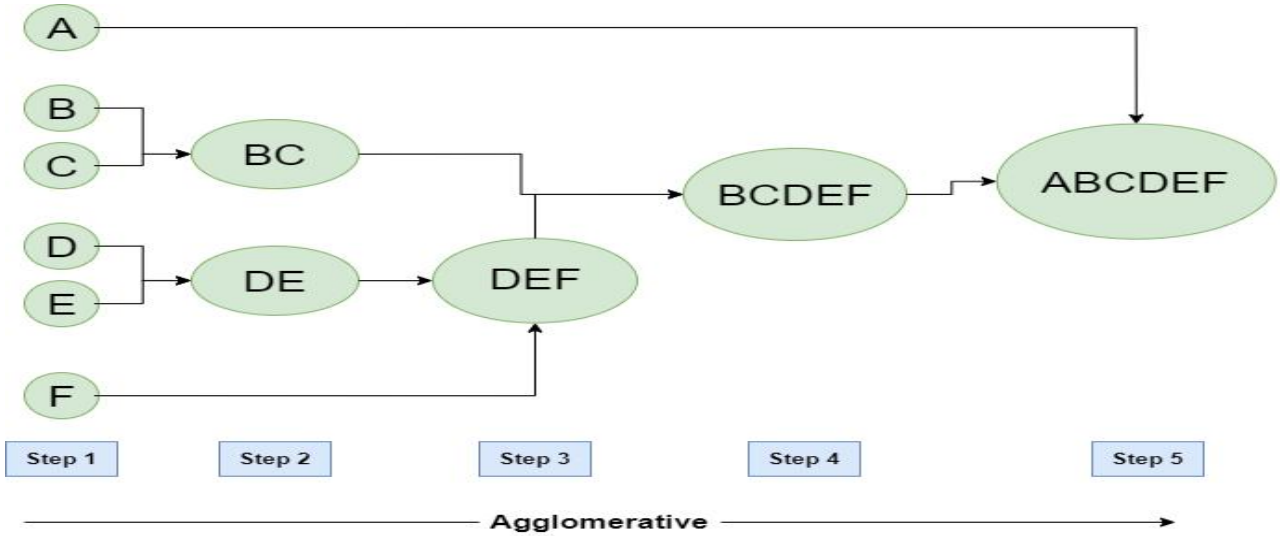
The algorithm for Agglomerative Hierarchical Clustering is:

- Calculate the similarity of one cluster with all the other clusters (calculate proximity matrix)
- Consider every data point as an individual cluster
- Merge the clusters which are highly similar or close to each other.
- Recalculate the proximity matrix for each cluster
- Repeat Steps 3 and 4 until only a single cluster remains.

Let's see the graphical representation of this algorithm using a dendrogram.

Note: This is just a demonstration of how the actual algorithm works no calculation has been performed below all the proximity among the clusters is assumed.

Let's say we have six data points **A, B, C, D, E, and F**.



Step-1: Consider each alphabet as a single cluster and calculate the distance of one cluster from all the other clusters.
Step-2: In the second step comparable clusters are merged together to form a single cluster. Let's say cluster (B) and cluster (C) are very similar to each other therefore we merge them in the second step similarly to cluster (D) and (E) and at last, we get the clusters [(A), (BC), (DE), (F)]
Step-3: We recalculate the proximity according to the algorithm and merge the two nearest clusters([(DE), (F)]) together to form new clusters as [(A), (BC), (DEF)]
Step-4: Repeating the same process; The clusters DEF and BC are comparable and merged together to form a new cluster. We're now left with clusters [(A), (BCDEF)].
Step-5: At last the two remaining clusters are merged together to form a single cluster [(ABCDEF)].

4.7 Key Issues in Hierarchical Clustering:

Lack of a Global Objective Function: agglomerative hierarchical clustering techniques perform clustering on a local level and as such there is no global objective function like in the K-Means algorithm. This is actually an advantage of this technique because the time and space complexity of global functions tends to be very expensive.

Ability to Handle Different cluster Sizes: we have to decide how to treat clusters of various sizes that are merged together.

Merging Decisions Are Final: one downside of this technique is that once two clusters have been merged they cannot be split up at a later time for a more favorable union..

4.8 Outlier Detection:

Outlier detection is the process of detecting outliers, or a data point that is far away from the average, and depending on what you are trying to accomplish, potentially removing or resolving them from the analysis to prevent any potential skewing. Outlier detection is one of the most important processes taken to create good, reliable data.

Outliers are extreme data points that are beyond the expected norms for their type. This can be a whole data set that is confounding, or extremities of a certain data set. Imagining a standard bell curve, the outliers are the data on the far right and left. These outliers can indicate fraud or some other anomaly you are trying to detect, but they can also be measurement errors, experimental problems, or a novel, one-off blip. Basically, it refers to a data point or set of data points that diverges dramatically from expected samples and patterns.

There are two types of outliers, multivariate and univariate. Univariate outliers are a data point that is extreme for one variable. A multivariate outlier is a combination of unusual data points, including at least two data points.

Point outliers: These are single data points that are far removed from the rest of the data points.

Contextual outliers: These are considered to be 'noise', such as punctuation symbols and commas in text, or background noise when performing speech recognition.

Collective outliers: These are subsets of unexpected data that show a deviation from conventional data, which may indicate a new phenomenon.

What Causes an Outlier?

There are eight main causes of outliers.

1. Incorrect data entry by humans
2. Codes used instead of values
3. Sampling errors, or data has been extracted from the wrong place or mixed with other data
4. Unexpected distribution of variables
5. Measurement errors caused by the application or system
6. Experimental errors in extracting the data or planning errors
7. Intentional dummy outliers inserted to test the detection methods
8. Natural deviations in data, not actually an error, that are indicate fraud or some other anomaly you are trying to detect

MODULE- V

Advanced Concepts: Basic concepts in Mining data streams–MiningTime–series data—Miningsequence patterns in Transactionaldatabases– Mining Object– Spatial– Multimedia–Text and Webdata –Spatial Datamining–Multimedia Datamining–TextMining–Mining theWorld Wide Web.

5.1 Basic concepts in Mining data streams:

Introduction to stream concepts :

A data stream is an existing, continuous, ordered (implicitly by entrance time or explicitly by timestamp) chain of items. It is unfeasible to control the order in which units arrive, nor it is feasible to locally capture stream in its entirety.

It is enormous volumes of data, items arrive at a high rate.

Types of Data Streams :

- **Data stream –**

A data stream is a(possibly unchained) sequence of tuples. Each tuple comprised of a set of attributes, similar to a row in a database table.

- **Transactional data stream –**

It is a log interconnection between entities

1. Credit card – purchases by consumers from producer
2. Telecommunications – phone calls by callers to the dialed parties
3. Web – accesses by clients of information at servers

- **Measurement data streams –**

1. Sensor Networks – a physical natural phenomenon, road traffic
2. IP Network – traffic at router interfaces
3. Earth climate – temperature, humidity level at weather stations

Examples of Stream Sources-

1. **Sensor Data –**

   In navigation systems, sensor data is used. Imagine a temperature sensor floating about in the ocean, sending back to the base station a reading of the surface temperature each hour. The data generated by this sensor is a stream of real numbers. We have 3.5 terabytes arriving every day and we for sure need to think about what we can be kept continuing and what can only be archived.

2. **Image Data –**

   Satellites frequently send down-to-earth streams containing many terabytes of images per day. Surveillance cameras generate images with lower resolution than satellites, but there can be numerous of them, each producing a stream of images at a break of 1 second each.

3. **Internet and Web Traffic –**

A bobbing node in the center of the internet receives streams of IP packets from many inputs and paths them to its outputs. Websites receive streams of heterogeneous types. For example, Google receives a hundred million search queries per day.

5.2 MiningTime–series:

A time series is a sequence of data points recorded at specific time points – most often in regular time intervals (seconds, hours, days, months etc.). Every organization generates a high volume of data every single day – be it sales figure, revenue, traffic, or operating cost. Time series data mining can generate valuable information for long-term business decisions, yet they are underutilized in most organizations. Below is a list of few possible ways to take advantage of time series datasets:

- **Trend analysis:** Just plotting data against time can generate very powerful insights. One very basic use of time-series data is just understanding temporal pattern/trend in what is being measured. In businesses it can even give an early indication on the overall direction of a typical business cycle.
- **Outlier/anomaly detection:** An outlier in a temporal dataset represents an anomaly. Whether desired (e.g. profit margin) or not (e.g. cost), outliers detected in a dataset can help prevent unintended consequences.
- **Examining shocks/unexpected variation:** Time-series data can identify variations (expected or unexpected) and abnormalities, detect signals in the noise.

- **Association analysis:** By plotting bivariate/multivariate temporal data it is easy (just visually) to identify associations between any two features (e.g. profit vs sales). This association may or may not imply causation, but this is a good starting point in selecting input features that impact output variables in more advanced statistical analysis.
- **Forecasting:** Forecasting future values using historical data is a common methodological approach – from simple extrapolation to sophisticated stochastic methods such as ARIMA.
- Predictive analytics: Advanced statistical analysis such as panel data models (fixed and random effects models) rely heavily on multi-variate longitudinal datasets. These types of analysis help in business forecasts, identify explanatory variables, or simply help understand associations between features in a dataset**.**

5.3 dataMiningsequence patterns in Transactionaldatabases:

GSP is a very important algorithm in data mining. It is used in sequence mining from large databases. Almost all sequence mining algorithms are basically based on a prior algorithm. GSP uses a level-wise paradigm for finding all the sequence patterns in the data. It starts with finding the frequent items of size one and then passes that as input to the next iteration of the GSP algorithm. The database is passed multiple times to this algorithm. In each iteration, GSP removes all the non-frequent itemsets. This is done based on a threshold frequency which is called support. Only those itemsets are kept whose frequency is greater than the support count. After the first pass, GSP finds all the frequent sequences of length-1 which are called 1-sequences. This makes the input to the next pass, it is the candidate for 2-sequences. At the end of this pass, GSP generates all frequent 2-sequences, which makes the input for candidate 3-sequences. The algorithm is recursively called until no more frequent itemsets are found.

Basic of Sequential Pattern (GSP) Mining:

- **Sequence:** A sequence is formally defined as the ordered set of items {s1, s2, s3, …, sn}. As the name suggests, it is the sequence of items occurring together. It can be considered as a transaction or purchased items together in a basket.
- **Subsequence:** The subset of the sequence is called a subsequence. Suppose {a, b, g, q, y, e, c} is a sequence. The subsequence of this can be {a, b, c} or {y, e}. Observe that the subsequence is not necessarily consecutive items of the sequence. From the sequences of databases, subsequences are found from which the generalized sequence patterns are found at the end.
- **Sequence pattern:** A sub-sequence is called a pattern when it is found in multiple sequences. The goal of the GSP algorithm is to mine the sequence patterns from the large database. The database consists of the sequences. When a subsequence has a frequency equal to more than the “support” value. For example: the pattern <a, b> is a sequence pattern mined from sequences {b, x, c, a}, {a, b, q}, and {a, u, b}.

Sequential Pattern (GSP) Mining uses:

Sequential pattern mining, also known as GSP (Generalized Sequential Pattern) mining, is a technique used to identify patterns in sequential data. The goal of GSP mining is to discover patterns in data that occur over time, such as customer buying habits, website navigation patterns, or sensor data.

Some of the main uses of GSP mining include:
Market basket analysis: GSP mining can be used to analyze customer buying habits and identify products that are frequently purchased together. This can help businesses to optimize their product placement and marketing strategies.

1. **Fraud detection:** GSP mining can be used to identify patterns of behavior that are indicative of fraud, such as unusual patterns of transactions or access to sensitive data.
2. **Website navigation:** GSP mining can be used to analyze website navigation patterns, such as the sequence of pages visited by users, and identify areas of the website that are frequently accessed or ignored.
3. **Sensor data analysis**: GSP mining can be used to analyze sensor data, such as data from IoT devices, and identify patterns in the data that are indicative of certain conditions or states.
4. **Social media analysis**: GSP mining can be used to analyze social media data, such as posts and comments, and identify patterns in the data that indicate trends, sentiment, or other insights.
5. **Medical data analysis**: GSP mining can be used to analyze medical data, such as patient records, and identify patterns in the data that are indicative of certain health conditions or trends.

Methods for Sequential Pattern Mining:

- Apriori-based Approaches
  - GSP
  - SPADE
- Pattern-Growth-based Approaches
  - FreeSpan
  - PrefixSpan

Sequence Database: A database that consists of ordered elements or events is called a sequence database. Example of a sequence database:

S.No.	SID	sequences
1.	100	<a(ab)(ac)d(cef)> or <a{ab}{ac}d{cef}>
2.	200	<(ad)c(bcd)(abe)>
3.	300	<(ef)(ab)(def)cb>
4.	400	<eg(adf)CBC>

Transaction: The sequence consists of many elements which are called transactions.
<a(ab)(ac)d(cef)> is a sequence whereas (a), (ab), (ac),

(d) and (cef) are the elements of the sequence.

These elements are sometimes referred as transactions.

An element may contain a set of items. Items within an element are unordered and we list them alphabetically.

For example, (cef) is the element and it consists of 3 items c, e and f.

Since, all three items belong to same element, their order does not matter. But we prefer to put them in alphabetical order for convenience.

The order of the elements of the sequence matters unlike order of items in same transaction.

k-length Sequence:
The number of items involved in the sequence is denoted by K. A sequence of 2 items is called a 2-len sequence. While finding the 2-length candidate sequence this term comes into use. Example of 2-length sequence is: {ab}, {(ab)}, {bc} and {(bc)}.

- {bc} denotes a 2-length sequence where b and c are two different transactions. This can also be written as {(b)(c)}
- {(bc)} denotes a 2-length sequence where b and c are the items belonging to the same transaction, therefore enclosed in the same parenthesis. This can also be written as {(cb)}, because the order of items in the same transaction does not matter.

Support in k-length Sequence:
Support means the frequency. The number of occurrences of a given k-length sequence in the sequence database is known as the support. While finding the support the order is taken care.

Illustration:
Suppose we have 2 sequences in the database.

s1: <a(bc)b(cd)>

s2: <b(ab)abc(de)>

We need to find the support of {ab} and {(bc)}

Finding support of {ab}:
This is present in first sequence.

s1: <a(bc)b(cd)>

Since, a and b belong to different elements, their order matters.

In second sequence {ab} is not found but {ba} is present.

s2: <b(ab)abc(de)> Thus we don't consider this.

Hence, support of {ab} is 1.

Finding support of {bc}:
Since, b and c are present in same element, their order does not matter.

s1: <a(bc)b(cd)>, first occurrence.

s2: <b(ab)abc(de)>, it seems correct, but is not. b and c are present in different elements here. So, we don't consider it.

Hence, support of {(bc)} is 1.

How to join L1 and L1 to give C2?
L1 is the final 1-length sequence after pruning. After pruning all the entries left in the set have supported greater than the threshold.

Case 1: Join {ab} and {ac}
s1: {ab}, s2: {ac}

After removing a from s1 and c from s2.

s1'={b}, s2'={a}

s1′ and s2′ are not same, so s1 and s2 can't be joined.

Case 2: Join {ab} and {be}
s1: {ab}, s2: {be}

After removing a from s1 and e from s2.

s1'={b}, s2'={b}

s1′ and s2′ are exactly same, so s1 and s2 be joined.

s1 + s2 = {abe}

Case 3: Join {(ab)} and {be}
s1: {(ab)}, s2: {be}

After removing a from s1 and e from s2.

s1'={(b)}, s2'={(b)}

s1′ and s2′ are exactly same, so s1 and s2 be joined.

s1 + s2 = {(ab)e}

s1 and s2 are joined in such a way that items belong to correct elements or transactions.

Pruning Phase: While building Ck (candidate set of k-length), we delete a candidate sequence that has a contiguous (k-1) subsequence whose support count is less than the minimum support (threshold). Also, delete a candidate sequence that has any subsequence without minimum support. {abg} is a candidate sequence of C3.

{abg} is a candidate sequence of C3.

To check if {abg} is proper candidate or not, without checking its support, we check the support of its subsets.

Because subsets of 3-length sequence will be 1 and 2 length sequences. We build the candidate sets increment like 1-length, 2-length and so on.

Subsets of {abg} are: {ab], {bg} and {ag}

Check support of all three subsets. If any of them have support less than minimum support then delete the sequence {abg} from the set C3 otherwise keep it.

Challenges in Generalized Sequential Pattern Data Mining

The database is passed many times to the algorithm recursively. The computational efforts are more to mine the frequent pattern. When the sequence database is very large and patterns to be mined are long then GSP encounters the problem in doing so effectively.

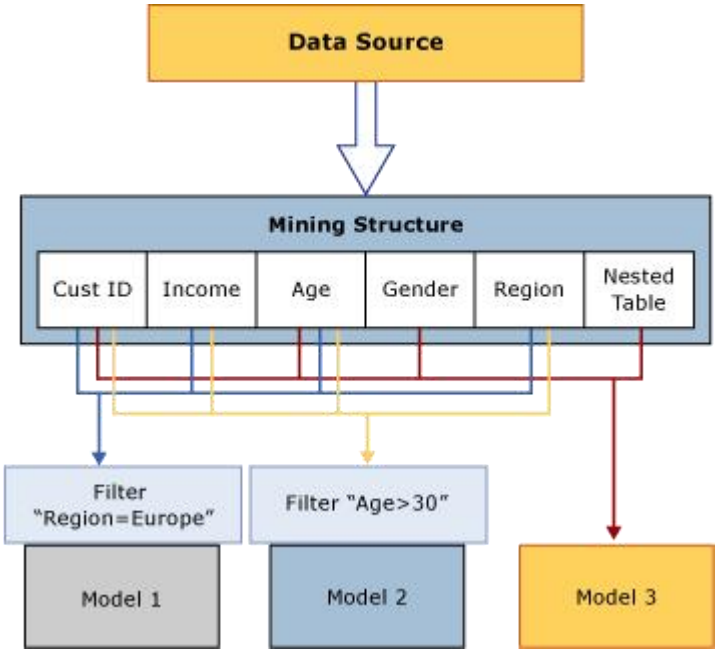
5.4 Mining Object– Spatial– Multimedia–Text and Webdata:

A data mining object is only an empty container until it has been processed. *Processing* a data mining model is also called *training*.

Processing mining structures: A mining structure gets data from an external data source, as defined by the column bindings and usage metadata, and reads the data. This data is read in full and then analyzed to extract various statistics. Analysis Services stores a compact representation of the data, which is suitable for analysis by data mining algorithms, in a local cache. You can either keep this cache or delete it after your models have been processed. By default, the cache is stored. For more information, see Process a Mining Structure.

Processing mining models: A mining model is empty, containing definitions only, until it is processed. To process a mining model, the mining structure that it is based on must have been processed. The mining model gets the data from the mining structure cache, applies any filters that may have been created on the model, and then passes the data set through the algorithm to detect patterns. After the model is processed, the model stores only the results of processing, not the data itself. For more information, see Process a Mining Model.

The following diagram illustrates the flow of data when a mining structure is processed, and when a mining model is processed.



Viewing the Results of Processing

After a mining structure has been processed, it contains a compact representation of the data for use in statistical analysis. If the cache has not been cleared, you can access the data in this cache in the following ways:

- Creating a Data Mining Extensions (DMX) query on the model and drilling through to the structure. For more information, see SELECT FROM <model>.CASES (DMX).

- Browsing a model based on the structure, and using one of the options in the user interface to drill through to structure cases. For more information, see Data Mining Model Viewers, or Drill Through to Case Data from a Mining Model.
- Creating a DMX query on the structure cases. For more information, see SELECT FROM <structure>.CASES.

After a mining model has been processed, it contains only the patterns that were derived from analysis, and mappings from the model results to the cached training data. You can browse or query the model results, called *model content*, or you can query the model and structure cases, if they have been cached.

The model content for each mining model depends on the algorithm that was used to create it. For example, if one model is a clustering model and another is a decision trees model, the model content is very different even though the models use exactly the same data. For more information, see Mining Model Content (Analysis Services - Data Mining).

Processing Requirements

Processing requirements may differ depending on whether your mining models are based solely on relational data, or on multidimensional data source.

For relational data source, processing requires only that you create training data and run mining algorithms on that data. However, mining models that are based on OLAP objects, such as dimensions and measures, require that the underlying data be in a processed state. This may requires that the multidimensional objects be processed to populate the mining model.

5.5 Spatial Datamining:

A spatial database saves a huge amount of space-related data, including maps, preprocessed remote sensing or medical imaging records, and VLSI chip design data. Spatial databases have several features that distinguish them from relational databases. They carry topological and/or distance information, usually organized by sophisticated, multidimensional spatial indexing structures that are accessed by spatial data access methods and often require spatial reasoning, geometric computation, and spatial knowledge representation techniques.

Spatial data mining refers to the extraction of knowledge, spatial relationships, or other interesting patterns not explicitly stored in spatial databases. Such mining demands the unification of data mining with spatial database technologies. It can be used for learning spatial records, discovering spatial relationships and relationships among spatial and nonspatial records, constructing spatial knowledge bases, reorganizing spatial databases, and optimizing spatial queries.

It is expected to have broad applications in geographic data systems, marketing, remote sensing, image database exploration, medical imaging, navigation, traffic control, environmental studies, and many other areas where spatial data are used.

A central challenge to spatial data mining is the exploration of efficient spatial data mining techniques because of the large amount of spatial data and the difficulty of spatial data types and spatial access methods. Statistical spatial data analysis has been a popular approach to analyzing spatial data and exploring geographic information.

The term geostatistics is often associated with continuous geographic space, whereas the term spatial statistics is often associated with discrete space. In a statistical model that manages non-spatial records, one generally considers statistical independence among different areas of data.

There is no such separation among spatially distributed records because, actually spatial objects are interrelated, or more exactly spatially co-located, in the sense that the closer the two objects are placed, the more likely they send the same properties. For example, natural resources, climate, temperature, and economic situations are likely to be similar in geographically closely located regions.

Such a property of close interdependency across nearby space leads to the notion of spatial autocorrelation. Based on this notion, spatial statistical modeling methods have been developed with success. Spatial data mining will create spatial statistical analysis methods and extend them for large amounts of spatial data, with more emphasis on effectiveness, scalability, cooperation with database and data warehouse systems, enhanced user interaction, and the discovery of new kinds of knowledge.

5.6 Introduction, webmining:

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:

Web mining is the process of discovering patterns, structures, and relationships in web data. It involves using data mining techniques to analyze web data and extract valuable insights. The applications of web mining are wide-ranging and include:

Personalized marketing:

Web mining can be used to analyze customer behavior on websites and social media platforms. This information can be used to create personalized marketing campaigns that target customers based on their interests and preferences.

E-commerce
Web mining can be used to analyze customer behavior on e-commerce websites. This information can be used to improve the user experience and increase sales by recommending products based on customer preferences.

Search engine optimization:
Web mining can be used to analyze search engine queries and search engine results pages (SERPs). This information can be used to improve the visibility of websites in search engine results and increase traffic to the website.

Fraud detection:
Web mining can be used to detect fraudulent activity on websites. This information can be used to prevent financial fraud, identity theft, and other types of online fraud.

Sentiment analysis:
Web mining can be used to analyze social media data and extract sentiment from posts, comments, and reviews. This information can be used to understand customer sentiment towards products and services and make informed business decisions.

Web content analysis:
Web mining can be used to analyze web content and extract valuable information such as keywords, topics, and themes. This information can be used to improve the relevance of web content and optimize search engine rankings.

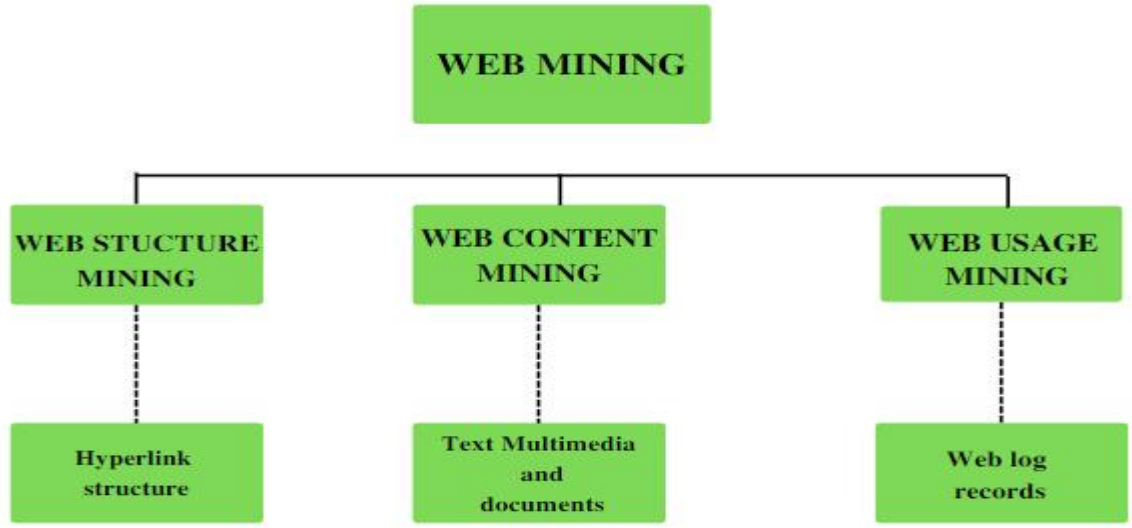
Customer service:
Web mining can be used to analyze customer service interactions on websites and social media platforms. This information can be used to improve the quality of customer service and identify areas for improvement.

Healthcare:
Web mining can be used to analyze health-related websites and extract valuable information about diseases, treatments, and medications. This information can be used to improve the quality of healthcare and inform medical research.

Process of Web Mining:

Web Mining Process

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.



Categories of Web 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.

5.2 web content mining:

Web Content Mining is one of the three different types of techniques in Web Mining. In this article, we will purely discuss Web Content Mining. Mining, extraction, and integration of useful data, information, and knowledge from Web page content are known as Web Mining.

It describes the discovery of useful information from web content. In simple words, it is the application of web mining that extracts relevant or useful information content from the Web. Web Content mining is somehow related but different from other mining techniques like data mining and text mining. Due to heterogeneity and the absence of web data, automated discovery of new knowledge patterns can be challenging to some extent.

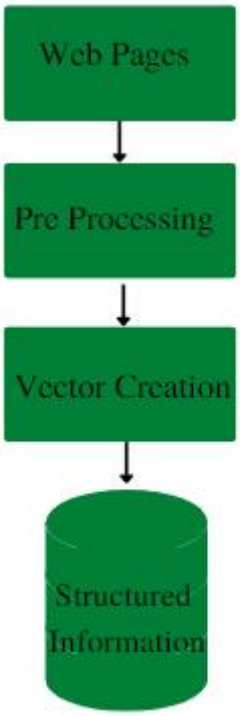
Web data are generally semi-structured and/or unstructured, while data mining is primarily concerned with structured data . It performs scanning and mining of text, image and images, and groups of web pages according to the content of input by displaying the list in search engines.

For Example: if the user is searching for a particular song then the search engine will display or provide suggestions relevant to it.

Web content mining deals with different kinds of data such as text, audio, video, image, etc.

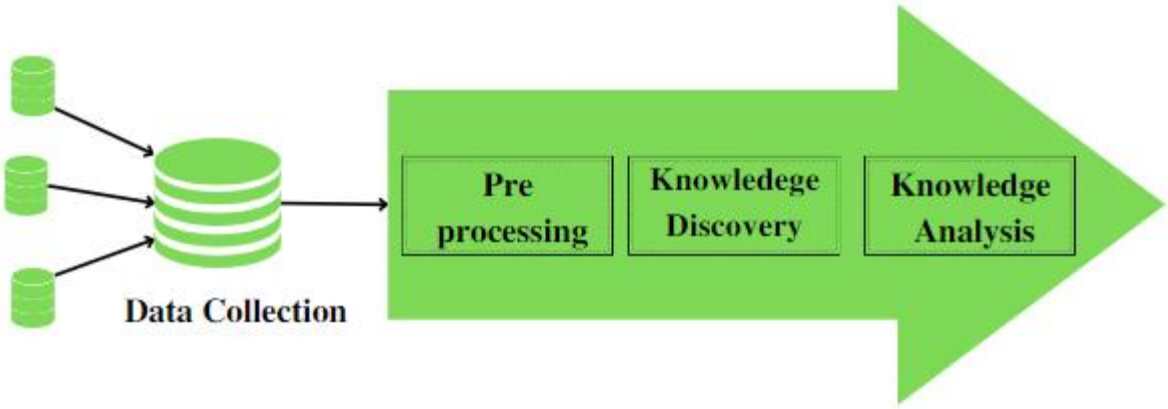
Unstructured Web Data Mining

Unstructured data includes data such as audio, video, etc, We convert these unstructured data into structured data,i.e., into useful information or structured information (which is known as Web Content Mining). the process of Conversion is mentioned as follows:



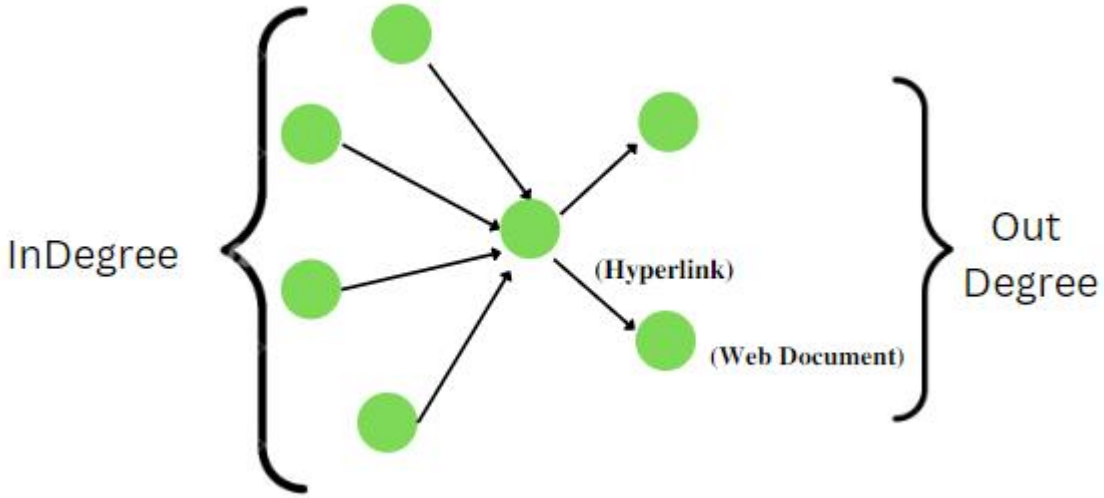
5.7 web structure mining:

Web Structure Mining is one of the three different types of techniques in Web Mining. In this article, we will purely discuss about the Web Structure Mining. Web Structure Mining is the technique of discovering structure information from the web. It uses graph theory to analyze the nodes and connections in the structure of a website.



Depending upon the type of Web Structural data, Web Structure Mining can be categorised into two types:

1.Extracting patterns from the hyperlink in the Web: The Web works through a system of hyperlinks using the hyper text transfer protocol (http). Hyperlink is a structural component that connects the web page according to different location. Any page can create a hyperlink of any other page and that page can also be linked to some other page. the intertwined or self-referral nature of web lends itself to some unique network analytical algorithms. The structure of Web pages could also be analyzed to examine the pattern of hyperlinks among pages.



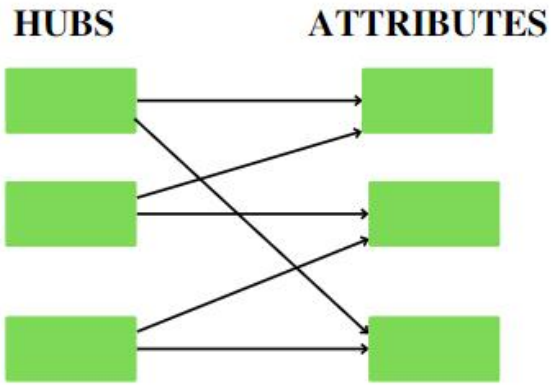
2. Mining the document structure. It is the analysis of tree like structure of web page to describe HTML or XML usage or the tags usage . There are different terms associated with Web Structure Mining :

- **Web Graph**: Web Graph is the directed graph representing Web.
- **Node**: Node represents the web page in the graph.
- **Edge(s)**: Edge represents the hyperlinks of the web page in the graph (Web graph)
- **In degree(s)**: It is the number of hyperlinks pointing to a particular node in the graph.
- **Degree(s)**: Degree is the number of links generated from a particular node. These are also called the Out Degrees.

All these terminologies will be more clear by looking at the following diagram of Web Graph:

Example of Web Structure Mining:
One of the techniques is the **Page rank Algorithm** that the **Google** uses to rank its web pages. The rank of a page is dependent on the number of pages and the quality of links pointing to the target node.
So, we can say that the Web Structure Mining is the type of Mining that can be performed either at the **document level** (intra-page) or at the **hyperlink level** (inter-page). The research done at the hyperlink level is called as Hyperlink Analysis. the Hyperlink Structure can be used to retrieve useful information on the Web.
Web structure Mining basically has two main approaches or there are two basic strategic models for successful websites:

- Page rank : refer Page Rank
- Hubs and Authorities



Hubs And Attributes

- **Hubs:** These are pages with large number of interesting links. They serve as a hub or a gathering point, where people visit to access a variety of information. More focused sites can aspire to become a hub for the new emerging areas. The pages on website themselves could be analyzed for quality of content that attracts most users.
- **Authorities:** People usually gravitate towards pages that provide the most complete and authentic information on a particular subject. This could be factual information, news, advice, etc. these websites would have the most number of inbound links from other websites.

5.8 WWW:

Web usage mining, a subset of Data Mining, is basically the extraction of various types of interesting data that is readily available and accessible in the ocean of huge web pages, Internet- or formally known as World Wide Web (WWW). Being one of the applications of data mining technique, it has helped to analyze user activities on different web pages and track them over a

period of time. Basically, Web Usage Mining can be divided into 2 major subcategories based on web usage data.

There are 3 main types of web data:

1. Web Content Data: The common forms of web content data are HTML, web pages, images audio-video, etc. The main being the HTML format. Though it may differ from browser to browser the common basic layout/structure would be the same everywhere. Since it's the most popular in web content data. XML and dynamic server pages like JSP, PHP, etc. are also various forms of web content data.
2. Web Structure Data: On a web page, there is content arranged according to HTML tags (which are known as intrapage structure information). The web pages usually have hyperlinks that connect the main webpage to the sub-web pages. This is called Inter-page structure information. So basically relationship/links describing the connection between webpages is web structure data.
3. Web Usage Data: The main source of data here is-Web Server and Application Server. It involves log data which is collected by the main above two mentioned sources. Log files are created when a user/customer interacts with a web page. The data in this type can be mainly categorized into three types based on the source it comes from:

- Server-side
- Client-side
- Proxy side.

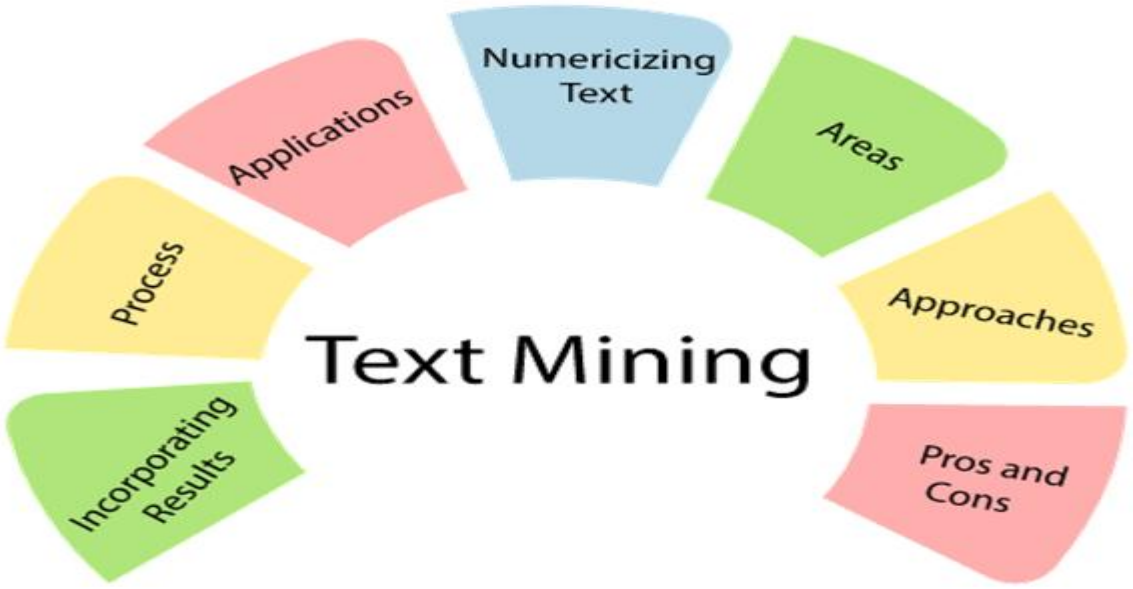
There are other additional data sources also which include cookies, demographics, etc.

Types of Web Usage Mining based upon the Usage Data:

1. Web Server Data: The web server data generally includes the IP address, browser logs, proxy server logs, user profiles, etc. The user logs are being collected by the web server data.
2. Application Server Data: An added feature on the commercial application servers is to build applications on it. Tracking various business events and logging them into application server logs is mainly what application server data consists of.
3. Application-level data: There are various new kinds of events that can be there in an application. The logging feature enabled in them helps us get the past record of the events.

5.9Text mining:

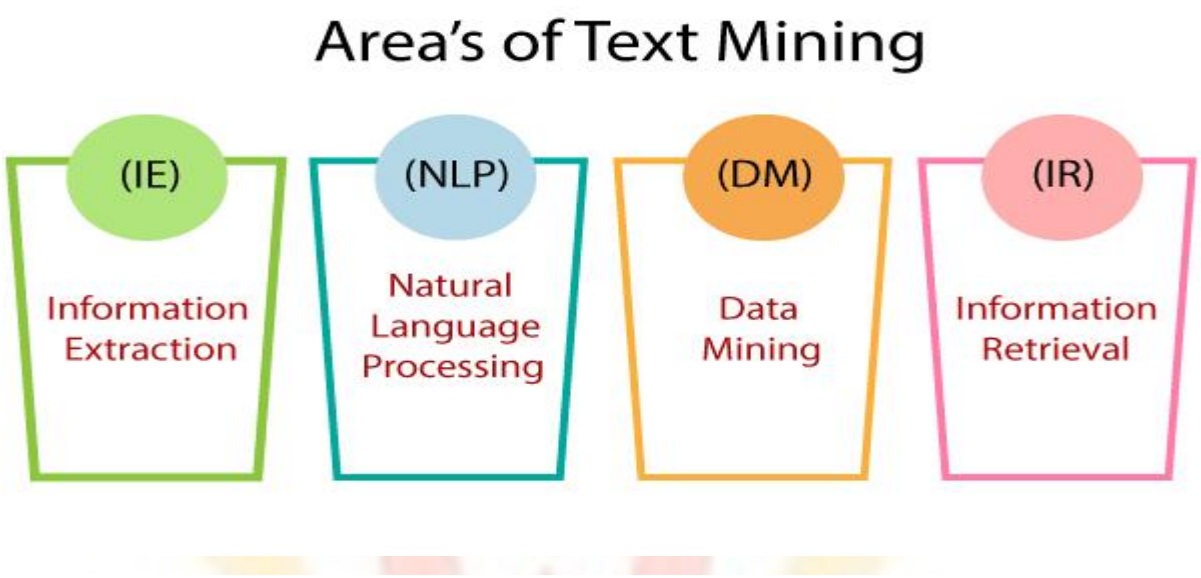
Text data mining can be described as the process of extracting essential data from standard language text. All the data that we generate via text messages, documents, emails, files are written in common language text. Text mining is primarily used to draw useful insights or patterns from such data.



The text mining market has experienced exponential growth and adoption over the last few years and also expected to gain significant growth and adoption in the coming future. One of the primary reasons behind the adoption of text mining is higher competition in the business market, many organizations seeking value-added solutions to compete with other organizations. With increasing completion in business and changing customer perspectives, organizations are making huge investments to find a solution that is capable of analyzing customer and competitor data to improve competitiveness. The primary source of data is e-commerce websites, social media platforms, published articles, survey, and many more. The larger part of the generated data is unstructured, which makes it challenging and expensive for the organizations to analyze with the help of the people. This challenge integrates with the exponential growth in data generation has led to the growth of analytical tools. It is not only able to handle large volumes of text data but also helps in decision-making purposes. Text mining software empowers a user to draw useful information from a huge set of data available sources.

Areas of text mining in data mining:

These are the following area of text mining :



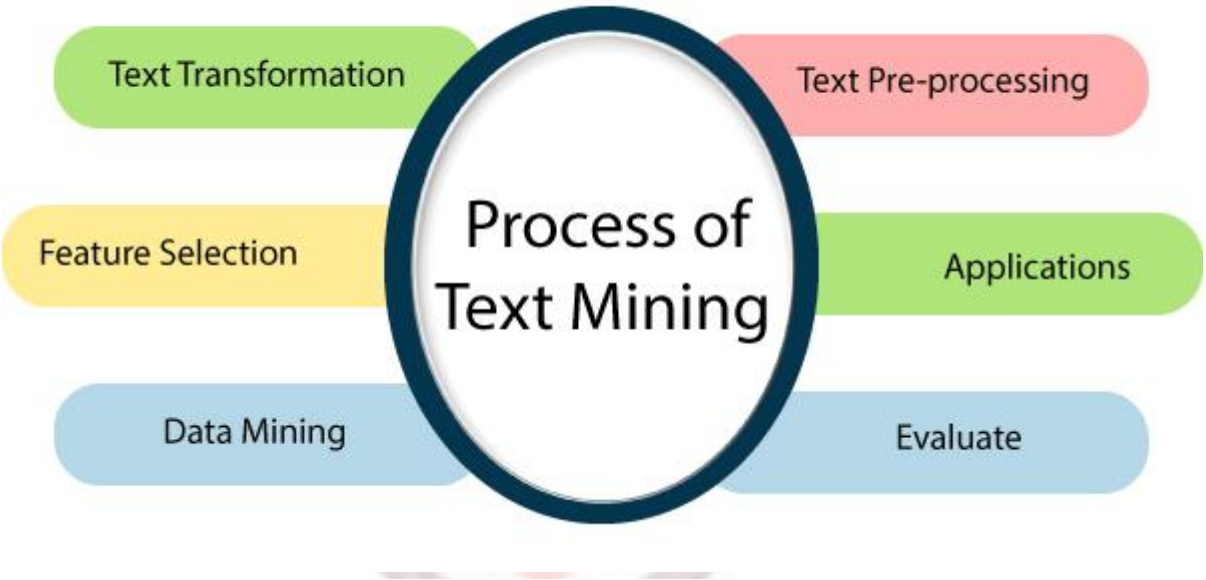
- **Information Extraction:**
  The automatic extraction of structured data such as entities, entities relationships, and attributes describing entities from an unstructured source is called information extraction.
- **Natural Language Processing:**
  NLP stands for Natural language processing. Computer software can understand human language as same as it is spoken. NLP is primarily a component of artificial intelligence(AI). The development of the NLP application is difficult because computers generally expect humans to "Speak" to them in a programming language that is accurate, clear, and exceptionally structured. Human speech is usually not authentic so that it can depend on many complex variables, including slang, social context, and regional dialects.
- **Data Mining:**
  Data mining refers to the extraction of useful data, hidden patterns from large data sets. Data mining tools can predict behaviors and future trends that allow businesses to make a better data-driven decision. Data mining tools can be used to resolve many business problems that have traditionally been too time-consuming.
- **Information Retrieval:**

Information retrieval deals with retrieving useful data from data that is stored in our systems. Alternately, as an analogy, we can view search engines that happen on websites such as e-commerce sites or any other sites as part of information retrieval.

Text Mining Process:

The text mining process incorporates the following steps to extract the data from the document.

Backward Skip 10sPlay VideoForward Skip 10s



- **Text transformation**
  A text transformation is a technique that is used to control the capitalization of the text. Here the two major way of document representation is given.
  a. Bag of words
  b. Vector Space
- **Text Pre-processing**
  Pre-processing is a significant task and a critical step in Text Mining, Natural Language Processing (NLP), and information retrieval(IR). In the field of text mining, data pre-processing is used for extracting useful information and knowledge from unstructured text data. Information Retrieval (IR) is a matter of choosing which documents in a collection should be retrieved to fulfill the user's need.
- **Feature selection:**
  Feature selection is a significant part of data mining. Feature selection can be defined as the process of reducing the input of processing or finding the essential information sources. The feature selection is also called variable selection.
- **Data Mining:**
  Now, in this step, the text mining procedure merges with the conventional process. Classic Data Mining procedures are used in the structural database.

- **Evaluate:**
  Afterward, it evaluates the results. Once the result is evaluated, the result abandon.
- **Applications:**
  These are the following text mining applications:
- **Risk Management:**
  Risk Management is a systematic and logical procedure of analyzing, identifying, treating, and monitoring the risks involved in any action or process in organizations. Insufficient risk analysis is usually a leading cause of disappointment. It is particularly true in the financial organizations where adoption of Risk Management Software based on text mining technology can effectively enhance the ability to diminish risk. It enables the administration of millions of sources and petabytes of text documents, and giving the ability to connect the data. It helps to access the appropriate data at the right time.
- **Customer Care Service:**
  Text mining methods, particularly NLP, are finding increasing significance in the field of customer care. Organizations are spending in text analytics programming to improve their overall experience by accessing the textual data from different sources such as customer feedback, surveys, customer calls, etc. The primary objective of text analysis is to reduce the response time of the organizations and help to address the complaints of the customer rapidly and productively.
- **Business Intelligence:**
  Companies and business firms have started to use text mining strategies as a major aspect of their business intelligence. Besides providing significant insights into customer behavior and trends, text mining strategies also support organizations to analyze the qualities and weaknesses of their opponent's so, giving them a competitive advantage in the market.
- **Social Media Analysis:**
  Social media analysis helps to track the online data, and there are numerous text mining tools designed particularly for performance analysis of social media sites. These tools help to monitor and interpret the text generated via the internet from the news, emails, blogs, etc. Text mining tools can precisely analyze the total no of posts, followers, and total no of likes of your brand on a social media platform that enables you to understand the response of the individuals who are interacting with your brand and content.

Text Mining Approaches in Data Mining:

These are the following text mining approaches that are used in data mining.

1. Keyword-based Association Analysis:

It collects sets of keywords or terms that often happen together and afterward discover the association relationship among them. First, it preprocesses the text data by parsing, stemming, removing stop words, etc. Once it pre-processed the data, then it induces association mining algorithms. Here, human effort is not required, so the number of unwanted results and the execution time is reduced.

2. Document Classification Analysis:

Automatic document classification:

This analysis is used for the automatic classification of the huge number of online text documents like web pages, emails, etc. Text document classification varies with the classification of relational data as document databases are not organized according to attribute values pairs.

Numericizing text:

- **Stemming algorithms**
  A significant pre-processing step before ordering of input documents starts with the stemming of words. The terms "stemming" can be defined as a reduction of words to their roots. For example, different grammatical forms of words and ordered are the same. The primary purpose of stemming is to ensure a similar word by text mining program.
- **Support for different languages:**
  There are some highly language-dependent operations such as stemming, synonyms, the letters that are allowed in words. Therefore, support for various languages is important.
- **Exclude certain character:**
  Excluding numbers, specific characters, or series of characters, or words that are shorter or longer than a specific number of letters can be done before the ordering of the input documents.
- **Include lists, exclude lists (stop-words):**
  A particular list of words to be listed can be characterized, and it is useful when we want to search for a specific word. It also classifies the input documents based on the frequencies with which those words occur. Additionally, "stop words," which means terms that are to be rejected from the ordering can be characterized. Normally, a default list of English stop words incorporates "the," "a," "since," etc. These words are used in the respective language very often but communicate very little data in the document.