##### DATA MINING CONCEPTS AND TECHNIQUES

##### M.GOPI SATYAVATHI

##### Asst professor

##### MCA DEPARTMENT

##### K.G.R.L P.G COLLEGE

# UNIT I

Data Warehouse and OLAP Technology: An overview Data Warehouse Basic Concepts, Data Warehouse Modelling: Data Cube and OLAP, Data Warehouse Implementation Data Pre-processing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization, From Data Warehousing to Data Mining

# UNIT II

Introduction to Data Mining: Motivation and importance, what is Data Mining, Data Mining on what kind of data, what kinds of patterns can be mined, which technologies are used, which kinds of applications are targeted, major issues in Data Mining. Getting to know your Data: Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring data Similarity and Dissimilarity

# UNIT III

Concept Description: Characterization and comparison What is Concept Description, Data Generalization by Attribute-Oriented Induction(AOI), AOI for Data Characterization, Efficient Implementation of AOI, AOI for Class comparisons. Mining Frequent Patterns, Associations and Correlations: Basic Concepts, Frequent Itemset Mining Methods: Apriori method, generating Association Rules, Improving the Efficiency of Apriori, Pattern-Growth Approach for mining Frequent Item sets, Mining Frequent Itemsets using vertical data format, Mining Closed and Max Patterns.

# UNIT IV

Classification Basic Concepts: Basic Concepts, Decision Tree Induction: Decision Tree Induction, Attribute Selection Measures, Tree Pruning, Bayes Classification Methods, Classification by Back Propagation, Support Vector Machines. Cluster Analysis: Cluster Analysis, Partitioning Methods, Hierarchal methods, Density based methods-DBSCAN and OPTICS.

# Text Book:

1. Data Mining Concepts and Techniques—JiaweiHan, MichelineKamber and Jian Pei, Morgan Kaufman Publications 3rd edition.

# Reference Books:

1. Introduction to Data Mining –Pang-Ning Tan, Michael Steinbach, Vipin Kumar
2. Introduction to Data Mining, Adriaan, Addison Wesley Publication

##### UNIT-I

## What Is a Data Warehouse?

A **data** **warehouse** is a database designed to enable business intelligence activities: it exists to help users understand and enhance their organization's performance. It is designed for query and analysis rather than for transaction processing, and usually contains historical data derived from transaction data, but can include data from other sources. Data warehouses separate analysis workload from transaction workload and enable an organization to consolidate data from several sources. This helps in:

* Maintaining historical records
* Analyzing the data to gain a better understanding of the business and to improve the business

In addition to a relational database, a data warehouse environment can include an extraction, transportation, transformation, and loading (ETL) solution, statistical analysis, reporting, data mining capabilities, client analysis tools, and other applications that manage the process of gathering data, transforming it into useful, actionable information, and delivering it to business users.

To achieve the goal of enhanced business intelligence, the data warehouse works with data collected from multiple sources. The source data may come from internally developed systems, purchased applications, third-party data syndicators and other sources. It may involve transactions, production, marketing, human resources and more. In today's world of big data, the data may be many billions of individual clicks on web sites or the massive data streams from sensors built into complex machinery.

Data warehouses are distinct from online transaction processing (OLTP) systems. With a data warehouse you separate analysis workload from transaction workload. Thus data warehouses are very much read-oriented systems. They have a far higher amount of data reading versus writing and updating. This enables far better analytical performance and avoids impacting your transaction systems. A data warehouse system can be optimized to consolidate data from many sources to achieve a key goal: it becomes your organization's "single source of truth". There is great value in having a consistent source of data that all users can look to; it prevents many disputes and enhances decision-making efficiency.

A data warehouse usually stores many months or years of data to support historical analysis. The data in a data warehouse is typically loaded through an extraction, transformation, and loading (ETL) process from multiple data sources. Modern data warehouses are moving toward an extract, load, transformation (ELT) architecture in which all or most data transformation is performed on the database that hosts the data warehouse. It is important to note that defining the ETL process is a very large part of the design effort of a data warehouse. Similarly, the speed and reliability of ETL operations are the foundation of the data warehouse once it is up and running.

Users of the data warehouse perform data analyses that are often time-related. Examples include consolidation of last year's sales figures, inventory analysis, and profit by product and by customer. But time-focused or not, users want to "slice and dice" their data however they see fit and a well-designed data warehouse will be flexible enough to meet those demands. Users will sometimes need highly aggregated data, and other times they will need to drill down to details. More sophisticated analyses include trend analyses and data mining, which use existing data to forecast trends or predict futures. The data warehouse acts as the underlying engine used by middleware business intelligence environments that serve reports, dashboards and other interfaces to end users.

Although the discussion above has focused on the term "data warehouse", there are two other important terms that need to be mentioned. These are the data mart and the operation data store (ODS).

A data mart serves the same role as a data warehouse, but it is intentionally limited in scope. It may serve one particular department or line of business. The advantage of a data mart versus a data warehouse is that it can be created much faster due to its limited coverage. However, data marts also create problems with inconsistency. It takes tight discipline to keep data and calculation definitions consistent across data marts. This problem has been widely recognized, so data marts exist in two styles. Independent data marts are those which are fed directly from source data. They can turn into islands of inconsistent information. Dependent data marts are fed from an existing data warehouse. Dependent data marts can avoid the problems of inconsistency, but they require that an enterprise-level data warehouse already exist.

Operational data stores exist to support daily operations. The ODS data is cleaned and validated, but it is not historically deep: it may be just the data for the current day. Rather than support the historically rich queries that a data warehouse can handle, the ODS gives data warehouses a place to get access to the most current data, which has not yet been loaded into the data warehouse. The ODS may also be used as a source to load the data warehouse. As data warehousing loading techniques have become more advanced, data warehouses may have less need for ODS as a source for loading data. Instead, constant trickle-feed systems can load the data warehouse in near real time.

A common way of introducing data warehousing is to refer to the characteristics of a data warehouse as set forth by William Inmon:

* [Subject Oriented](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-C3AD27A3-970A-442D-8B18-86B79D643F25__BABIBEAB)
* [Integrated](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-C3AD27A3-970A-442D-8B18-86B79D643F25__BABCEFAE)
* [Nonvolatile](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-C3AD27A3-970A-442D-8B18-86B79D643F25__BABFDJHA)
* [Time Varient](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-C3AD27A3-970A-442D-8B18-86B79D643F25__BABBGEDG)

**Subject Oriented**

Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a data warehouse that concentrates on sales. Using this data warehouse, you can answer questions such as "Who was our best customer for this item last year?" or "Who is likely to be our best customer next year?" This ability to define a data warehouse by subject matter, sales in this case, makes the data warehouse subject oriented.

**Integrated**

Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as naming conflicts and inconsistencies among units of measure. When they achieve this, they are said to be integrated.

**Nonvolatile**

Nonvolatile means that, once entered into the data warehouse, data should not change. This is logical because the purpose of a data warehouse is to enable you to analyze what has occurred.

**Time Varient**

A data warehouse's focus on change over time is what is meant by the term time variant. In order to discover trends and identify hidden patterns and relationships in business, analysts need large amounts of data. This is very much in contrast to [**online transaction processing (OLTP)**](https://docs.oracle.com/database/121/DWHSG/glossary.htm#GUID-356ED48C-6CBA-4F5A-B769-AE59798AEC86) systems, where performance requirements demand that historical data be moved to an archive.

### Key Characteristics of a Data Warehouse

The key characteristics of a data warehouse are as follows:

* Data is structured for simplicity of access and high-speed query performance.
* End users are time-sensitive and desire speed-of-thought response times.
* Large amounts of historical data are used.
* Queries often retrieve large amounts of data, perhaps many thousands of rows.
* Both predefined and ad hoc queries are common.
* The data load involves multiple sources and transformations.

In general, fast query performance with high data throughput is the key to a successful data warehouse.

## Contrasting OLTP and Data Warehousing Environments

There are important differences between an OLTP system and a data warehouse. One major difference between the types of system is that data warehouses are not exclusively in [**third normal form (3NF)**](https://docs.oracle.com/database/121/DWHSG/glossary.htm#GUID-6583088A-372F-44CE-A76B-6F9ADE28D3DC), a type of data normalization common in OLTP environments.

Data warehouses and OLTP systems have very different requirements. Here are some examples of differences between typical data warehouses and OLTP systems:

* Workload

Data warehouses are designed to accommodate ad hoc queries and data analysis. You might not know the workload of your data warehouse in advance, so a data warehouse should be optimized to perform well for a wide variety of possible query and analytical operations.

OLTP systems support only predefined operations. Your applications might be specifically tuned or designed to support only these operations.

* Data modifications

A data warehouse is updated on a regular basis by the ETL process (run nightly or weekly) using bulk data modification techniques. The end users of a data warehouse do not directly update the data warehouse except when using analytical tools, such as data mining, to make predictions with associated probabilities, assign customers to market segments, and develop customer profiles.

In OLTP systems, end users routinely issue individual data modification statements to the database. The OLTP database is always up to date, and reflects the current state of each business transaction.

* Schema design

Data warehouses often use partially denormalized schemas to optimize query and analytical performance.

OLTP systems often use fully normalized schemas to optimize update/insert/delete performance, and to guarantee data consistency.

* Typical operations

A typical data warehouse query scans thousands or millions of rows. For example, "Find the total sales for all customers last month."

A typical OLTP operation accesses only a handful of records. For example, "Retrieve the current order for this customer."

* Historical data

Data warehouses usually store many months or years of data. This is to support historical analysis and reporting.

OLTP systems usually store data from only a few weeks or months. The OLTP system stores only historical data as needed to successfully meet the requirements of the current transaction.

## Common Data Warehouse Tasks

As an Oracle data warehousing administrator or designer, you can expect to be involved in the following tasks:

* Configuring an Oracle database for use as a data warehouse
* Designing data warehouses
* Performing upgrades of the database and data warehousing software to new releases
* Managing schema objects, such as tables, indexes, and materialized views
* Managing users and security
* Developing routines used for the extraction, transformation, and loading (ETL) processes
* Creating reports based on the data in the data warehouse
* Backing up the data warehouse and performing recovery when necessary
* Monitoring the data warehouse's performance and taking preventive or corrective action as required

In a small-to-midsize data warehouse environment, you might be the sole person performing these tasks. In large, enterprise environments, the job is often divided among several DBAs and designers, each with their own specialty, such as database security or database tuning.

These tasks are illustrated in the following:

* For more information regarding partitioning, see [Oracle Database VLDB and Partitioning Guide](https://docs.oracle.com/database/121/VLDBG/toc.htm).
* For more information regarding database security, see [Oracle Database Security Guide](https://docs.oracle.com/database/121/DBSEG/toc.htm).
* For more information regarding database performance, see [Oracle Database Performance Tuning Guide](https://docs.oracle.com/database/121/TGDBA/toc.htm) and [Oracle Database SQL Tuning Guide](https://docs.oracle.com/database/121/TGSQL/toc.htm).
* For more information regarding backup and recovery, see Oracle Database Backup and Recovery User's Guide.
* For more information regarding ODI, see [Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator](http://www.oracle.com/pls/topic/lookup?ctx=E50529-01&id=ODIDG).

## Data Warehouse Architectures

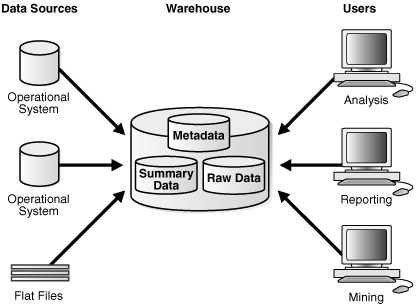
Data warehouses and their architectures vary depending upon the specifics of an organization's situation. Three common architectures are:

* [Data Warehouse Architecture: Basic](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-F90C7CB2-84A3-404C-B81C-1D71ADA7EA19)
* [Data Warehouse Architecture: with a Staging Area](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-34AD1205-06E3-46C2-9AAC-9E3FAF8D9622)
* [Data Warehouse Architecture: with a Staging Area and Data Marts](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-A53F7799-A01E-42F2-9B75-189D773479C5)

### Data Warehouse Architecture: Basic

[Figure 1-1](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-F90C7CB2-84A3-404C-B81C-1D71ADA7EA19__CHDGAFJD) shows a simple architecture for a data warehouse. End users directly access data derived from several source systems through the data warehouse.

**Figure 1-1 Architecture of a Data Warehouse**

  
[Description of "Figure 1-1 Architecture of a Data Warehouse"](https://docs.oracle.com/database/121/DWHSG/img_text/GUID-BFFE16AE-26ED-4D97-9A94-1193CC6B2D90-print.htm)

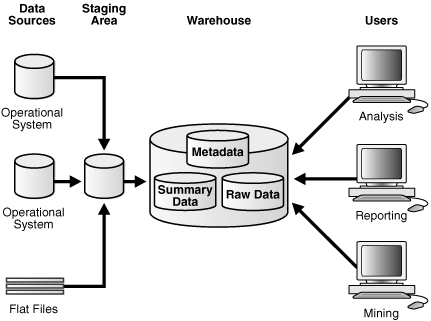
In [Figure 1-1](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-F90C7CB2-84A3-404C-B81C-1D71ADA7EA19__CHDGAFJD), the metadata and raw data of a traditional OLTP system is present, as is an additional type of data, summary data. Summaries are a mechanism to pre-compute common expensive, long-running operations for sub-second data retrieval. For example, a typical data warehouse query is to retrieve something such as August sales. A summary in an Oracle database is called a [**materialized view**](https://docs.oracle.com/database/121/DWHSG/glossary.htm#GUID-AC1E7D15-7178-4C7E-89CA-809D13AB2513).

The consolidated storage of the raw data as the center of your data warehousing architecture is often referred to as an Enterprise Data Warehouse (EDW). An EDW provides a 360-degree view into the business of an organization by holding all relevant business information in the most detailed format.

### Data Warehouse Architecture: with a Staging Area

You must clean and process your operational data before putting it into the warehouse, as shown in [Figure 1-2](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-34AD1205-06E3-46C2-9AAC-9E3FAF8D9622__i1006323). You can do this programmatically, although most data warehouses use a [**staging area**](https://docs.oracle.com/database/121/DWHSG/glossary.htm#GUID-A33DDFF2-93E1-4A99-A20B-7AB8E60B53A1) instead. A staging area simplifies data cleansing and consolidation for operational data coming from multiple source systems, especially for enterprise data warehouses where all relevant information of an enterprise is consolidated. [Figure 1-2](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-34AD1205-06E3-46C2-9AAC-9E3FAF8D9622__i1006323) illustrates this typical architecture.

**Figure 1-2 Architecture of a Data Warehouse with a Staging Area**

  
[Description of "Figure 1-2 Architecture of a Data Warehouse with a Staging Area"](https://docs.oracle.com/database/121/DWHSG/img_text/GUID-18665CE4-9FA1-4454-BD3D-024B3B288246-print.htm)

### Data Warehouse Architecture: with a Staging Area and Data Marts

Although the architecture in [Figure 1-2](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-34AD1205-06E3-46C2-9AAC-9E3FAF8D9622__i1006323) is quite common, you may want to customize your warehouse's architecture for different groups within your organization. You can do this by adding **data** **marts**, which are systems designed for a particular line of business. [Figure 1-3](https://docs.oracle.com/database/121/DWHSG/concept.htm#GUID-A53F7799-A01E-42F2-9B75-189D773479C5__i1006339) illustrates an example where purchasing, sales, and inventories are separated. In this example, a financial analyst might want to analyze historical data for purchases and sales or mine historical data to make predictions about customer behavior.

**DataWarehouseModeling:**

The interpretation and documentation of the current processes and transactions that exist during the software design and development is known as data modeling. The data modeling techniques and tools simplify the complicated system designs into easier data flows which can be used for re-engineering. It is used to create the logical and physical design of a data warehouse.

## What is the need for Data Modeling in a Data warehouse

### Collecting the Business Requirements

* Typically, a data warehouse is designed with the data architects and the business users determining the entities required in the data warehouse and the facts that need to be recorded. This initial design has much iteration before deciding the final model
* At this stage, we need to overcome the common drawbacks faced when designing. Since a data warehouse is implemented from an existing system the architects at times implement a larger part of the older system into the new design to save time or leave out details
* The logical model captures the business requirements efficiently and serves as a building block for the physical model

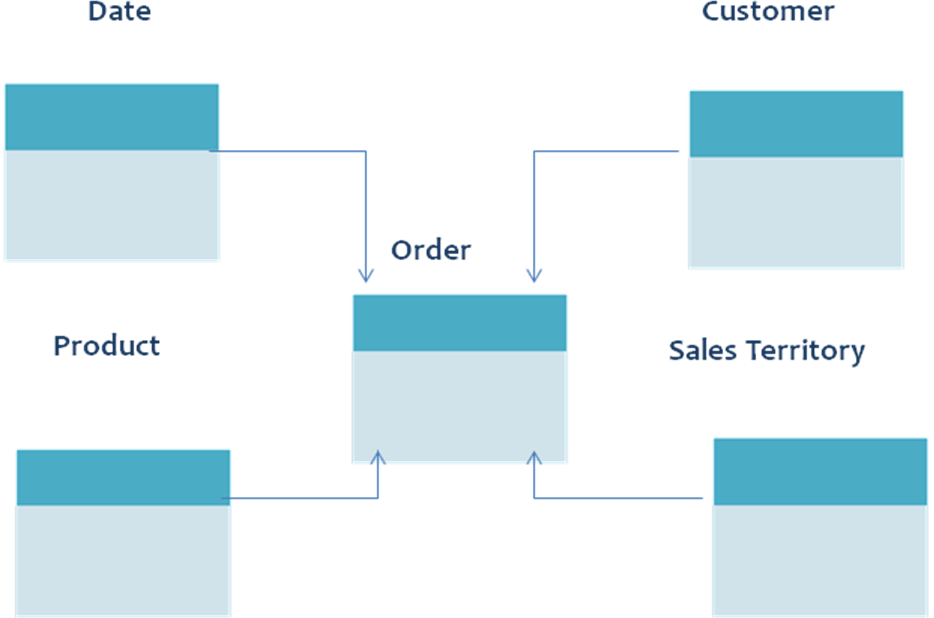
### Enhancing Database Performance

* Query performance is a vital feature of a data warehouse. Enormous data volumes are involved in a data warehouse, so using a data model product for management of the metadata and the data used by the BI users is very important
* The physical model adds indexing which optimize a database performance. At times the schemas too are changed. For example, a star schema can be changed to a snow flake schema if it promises a faster retrieval of data

### Offers Source and Target System Documentation

* In the process of designing an ETL system, it is very important to verify the physical and logical models of the source and target systems respectively
* The data modeling offers this documentation which acts as a reference for the future

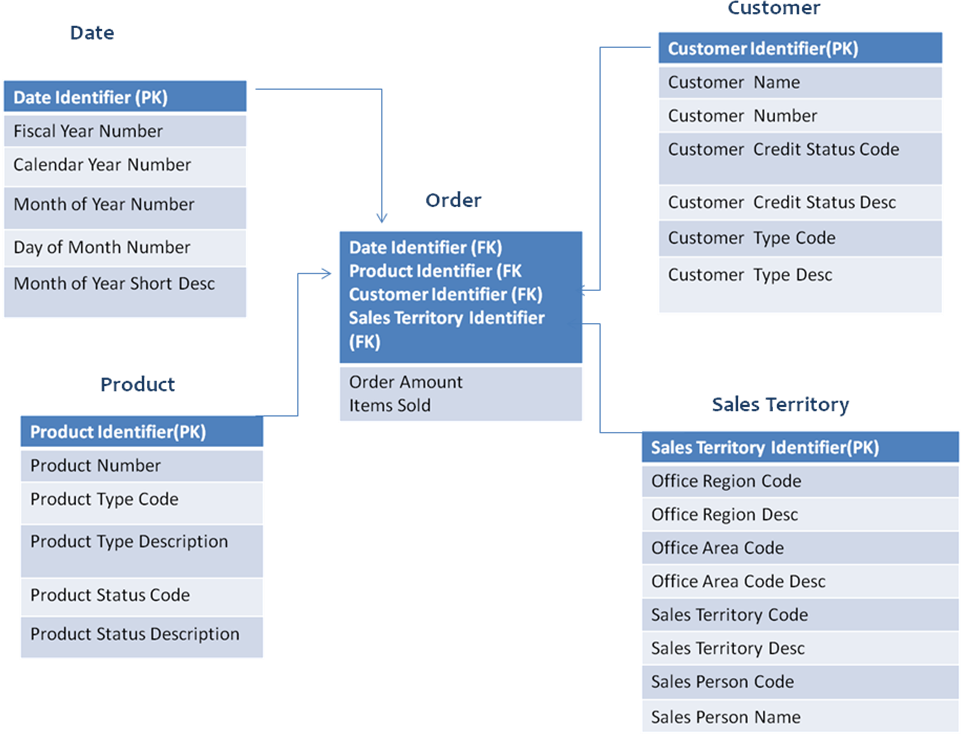
## Conceptual Data Model

A conceptual data model determines the highest-level relationships among the different entities.

* It is the primary step in creation of a data-model in top-down approach that is an exact representation of the business organization
* Conceives the overall structure of the database and gives information of the subject-areas
* Consists of entity types and relationships. The relationship between the subject areas are represented by symbolic notations (IDEF1X or IE). Cardinality in a data model exhibits the one to one relationship or many-to-many relationship
* No primary key is stated
* No attribute is specified

## Logical Data Model

Logical data model represents the specific particulars of the entities, attributes, and relationships involved in a business. It is the basis on which a physical model is designed.



* The development of a logical model begins after the sign-off of the conceptual data model by the functional team
* A logical model should systematize the physical design process by defining the data structures and the relationship between them
* The primary keys and foreign keys are established here
* Normalization occurs here
* Represents all the entities and the relationships between them

 Physical Data Model

Physical data model exhibits the model of the database that is to be built. It represents the table structures, column names, column data types, primary keys, and foreign keys.



* The physical data model is developed after receiving the acceptance of the logical data model by the functional team
* Physical data model might be different from the logical data model due to few physical constraints
* Physical data model differs for different databases. The data types change for different databases
* Denormalization takes place according to the user requirements
* The logical model is changed to physical data model by implementing the database rules, referential integrity , super types, and sub types

## Relational Data Model

Relational data modeling is used in OLTP systems which are transaction-oriented. The major characteristics of a relational data model are:

### Relationship among the tables

* All the data is stored in tables and each relation has rows and columns
* The table should have a header and a body. Header is the list of columns in the table and body consist of the values populated in the table. Tuple is the unique value generated from the junction of one column and one row

### Usage of keys

* Primary key is the most important key in a table. It is used as a unique identifier. The primary key is always a not null column
* Foreign key is used to relate to the primary key. They relate the data from one table to another table and establish a relationship

### Data Redundancy

* The relational data model applies rules to maintain data integrity
* It eliminates data redundancy. The data is not stored repetitively. This helps in maintaining data consistency and limited data storage

## Multi Dimensional Data Model

A multi dimensional data model is logical view of an enterprise that represents the important entities of a business and the relationship between them. It is not restricted to a physical database and tables. It’s not represented by E-R diagrams. The main components are:

### Attributes

* Attributes are the abstract terms devised for easier summarization of data on a report
* They can also be defined as the column headings that are not part of any calculations on a report

### Dimensions

* A dimension is a data set comprising individual, non overlapping data elements
* They enable end users to define, group and filter the data for display and browsing purposes

### Facts

* A fact is a table consisting of columns that are used for numeric purposes to answer the business questions
* They consist of additive, non-additive, and semi-additive measures

# https://www.wideskills.com/sites/default/files/subjects/Data%20Warehousing%20Tutorial/04/image1.png

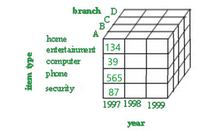
* It is the primary step in creation of a data-model in top-down approach that is an exact representation of the business organization
* Conceives the overall structure of the database and gives information of the subject-areas
* Consists of entity types and relationships. The relationship between the subject areas are represented by symbolic notations (IDEF1X or IE). Cardinality in a data model exhibits the one to one relationship or many-to-many relationship
* No primary key is stated
* No attribute is specified

DataCubeandOLAP:

# Data Cube or OLAP approach in Data Mining

* Last Updated : 01 Aug, 2021

Grouping of data in a multidimensional matrix is called data cubes. In Dataware housing, we generally deal with various multidimensional data models as the data will be represented by multiple dimensions and multiple attributes. This multidimensional data is represented in the data cube as the cube represents a high-dimensional space. The Data cube pictorially shows how different attributes of data are arranged in the data model. Below is the diagram of a general data cube.



Attention reader! Don’t stop learning now. Get hold of all the important CS Theory concepts for SDE interviews with the [**CS Theory Course**](https://practice.geeksforgeeks.org/courses/SDE-theory?vC=1) at a student-friendly price and become industry ready.

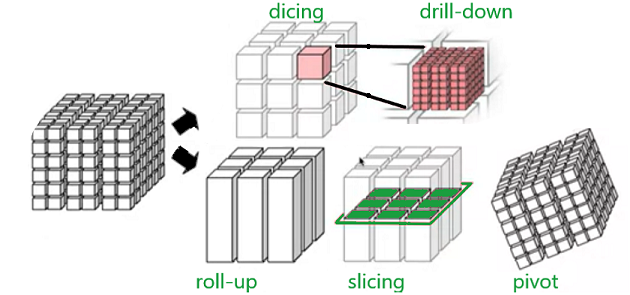
The example above is a 3D cube having attributes like branch(A,B,C,D),item type(home,entertainment,computer,phone,security), year(1997,1998,1999) .

### ****Data cube classification:****

The data cube can be classified into two categories:

* **Multidimensional data cube:**It basically helps in storing large amounts of data by making use of a multi-dimensional array. It increases its efficiency by keeping an index of each dimension. Thus, dimensional is able to retrieve data fast.
* **Relational data cube:**It basically helps in storing large amounts of data by making use of relational tables. Each relational table displays the dimensions of the data cube. It is slower compared to a Multidimensional Data Cube.

### Data cube operations:



Data cube operations are used to manipulate data to meet the needs of users. These operations help to select particular data for the analysis purpose. There are mainly 5 operations listed below-

* **Roll-up**: operation and aggregate certain similar data attributes having the same dimension together. For example, if the data cube displays the daily income of a customer, we can use a roll-up operation to find the monthly income of his salary.
* **Drill-down**: this operation is the reverse of the roll-up operation. It allows us to take particular information and then subdivide it further for coarser granularity analysis. It zooms into more detail. For example- if India is an attribute of a country column and we wish to see villages in India, then the drill-down operation splits India into states, districts, towns, cities, villages and then displays the required information.
* **Slicing**: this operation filters the unnecessary portions. Suppose in a particular dimension, the user doesn’t need everything for analysis, rather a particular attribute. For example, country=”jamaica”, this will display only about jamaica and only display other countries present on the countrylist.
* **Dicing**: this operation does a multidimensional cutting, that not only cuts only one dimension but also can go to another dimension and cut a certain range of it. As a result, it looks more like a subcube out of the whole cube(as depicted in the figure). For example- the user wants to see the annual salary of Jharkhand state employees.
* **Pivot**: this operation is very important from a viewing point of view. It basically transforms the data cube in terms of view. It doesn’t change the data present in the data cube. For example, if the user is comparing year versus branch, using the pivot operation, the user can change the viewpoint and now compare branch versus item type.

### Advantages of data cubes:

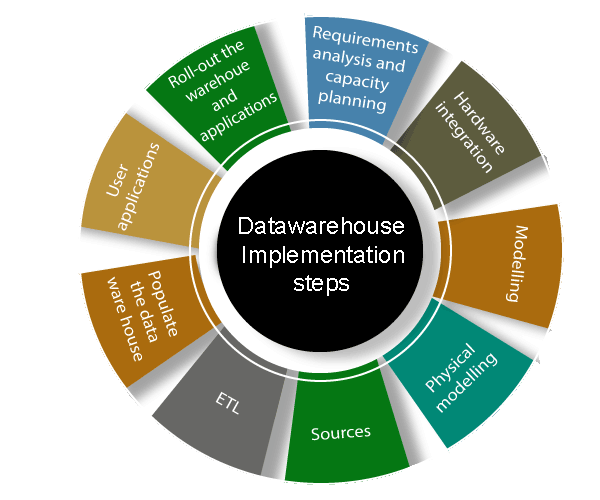
* Helps in giving a summarised view of data.
* Data cubes store large data in a simple way.
* Data cube operation provides quick and better analysis,
* Improve performance of data.
* OLAP (for online analytical processing) is software for performing multidimensional analysis at high speeds on large volumes of data from a [data warehouse](https://www.ibm.com/cloud/learn/data-warehouse), data mart, or some other unified, centralized data store.
* Most business data have multiple dimensions—multiple categories into which the data are broken down for presentation, tracking, or analysis. For example, sales figures might have several dimensions related to location (region, country, state/province, store), time (year, month, week, day), product (clothing, men/women/children, brand, type), and more.
* But in a data warehouse, data sets are stored in tables, each of which can organize data into just two of these dimensions at a time. OLAP extracts data from multiple relational data sets and reorganizes it into a multidimensional format that enables very fast processing and very insightful analysis.

OLAP approach in Data Mining:

* OLAP (for online analytical processing) is software for performing multidimensional analysis at high speeds on large volumes of data from a [data warehouse](https://www.ibm.com/cloud/learn/data-warehouse), data mart, or some other unified, centralized data store.
* Most business data have multiple dimensions—multiple categories into which the data are broken down for presentation, tracking, or analysis. For example, sales figures might have several dimensions related to location (region, country, state/province, store), time (year, month, week, day), product (clothing, men/women/children, brand, type), and more.
* But in a data warehouse, data sets are stored in tables, each of which can organize data into just two of these dimensions at a time. OLAP extracts data from multiple relational data sets and reorganizes it into a multidimensional format that enables very fast processing and very insightful analysis.

DataWarehouseImplementationDataPreprocessing:

here are various implementation in data warehouses which are as follows



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

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

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

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

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

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

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

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

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

**DataCleaning**:

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for your data cleaning process so you know you are doing it the right way every time.

**DataIntegration:**

Data integration is the practice of consolidating data from disparate sources into a single dataset with the ultimate goal of providing users with consistent access and delivery of data across the spectrum of subjects and structure types, and to meet the information needs of all applications and business processes. The data integration process is one of the main components in the overall data management process, employed with increasing frequency as big data integration and the need to share existing data continues to grow.

Data integration architects develop data integration software programs and data integration platforms that facilitate an automated data integration process for connecting and routing data from source systems to target systems. This can be achieved through a variety of data integration techniques, including:

* Extract, Transform and Load: copies of datasets from disparate sources are gathered together, harmonized, and loaded into a data warehouse or database
* Extract, Load and Transform: data is loaded as is into a big data system and transformed at a later time for particular analytics uses
* Change Data Capture: identifies data changes in databases in real-time and applies them to a data warehouse or other repositories
* Data Replication: data in one database is replicated to other databases to keep the information the information synchronized to operational uses and for backup
* Data Virtualization: data from different systems are virtually combined to create a unified view rather than loading data into a new repository
* Streaming Data Integration: a real time data integration method in which different streams of data are continuously integrated and fed into analytics systems and data stores

**DataReduction:**

Data Reduction In Data Mining

A database or date warehouse may store terabytes of data.So it may take very long to perform data analysis and mining on such huge amounts of data.

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume but still contain critical information.

Data Reduction Strategies:-

1 Data Cube Aggregation

Aggregation operations are applied to the data in the construction of a data cube.

2 Dimensionality Reduction

In dimensionality reduction redundant attributes are detected and removed which reduce the data set size.

3 Data Compression

Encoding mechanisms are used to reduce the data set size.

4 Numerosity Reduction

In numerosity reduction where the data are replaced or estimated by alternative.

5 Discretisation and concept hierarchy generation

**Data transformation and discretization**

As we know from the previous section, there are always some data formats that are best suited for specific data mining algorithms. Data transformation is an approach to transform the original data to preferable data format for the input of certain data mining algorithms before the processing.

**Data transformation**

Data transformation routines convert the data into appropriate forms for mining. They're shown as follows:

**Smoothing**: This uses binning, regression, and clustering to remove noise from the data

**Attribute construction**: In this routine, new attributes are constructed and added from the given set of attributes

**Aggregation**: In this summary or aggregation, operations are performed on the data

**Normalization**: Here, the attribute data is scaled so as to fall within a smaller range

**Discretization**: In this routine, the raw values of a numeric attribute are replaced by interval label or conceptual label

**Concept hierarchy generation for nominal data**: Here, attributes can be generalized to higher level concepts

**Normalization data transformation methods**

To avoid dependency on the choice of measurement units on data attributes, the data should be normalized. This means transforming or mapping the data to a smaller or common range. All attributes gain an equal weight after this process. There are many normalization methods. Let's have a look at some of them:

**Min-max normalization**: This preserves the relationships among the original data values and performs a linear transformation on the original data. The applicable ones of the actual maximum and minimum values of an attribute will be normalized.

**z-score normalization**: Here the values for an attribute are normalized based on the mean and standard deviation of that attribute. It is useful when the actual minimum and maximum of an attribute to be normalized are unknown.

**Normalization by decimal scaling**: This normalizes by moving the decimal point of values of attribute.

**Data discretization**

Data discretization transforms numeric data by mapping values to interval or concept labels. Discretization techniques include the following:

**Data discretization by binning**: This is a top-down unsupervised splitting technique based on a specified number of bins.

**Data discretization by histogram analysis**: In this technique, a histogram partitions the values of an attribute into disjoint ranges called buckets or bins. It is also an unsupervised method.

**Data discretization by cluster analysis**: In this technique, a clustering algorithm can be applied to discretize a numerical attribute by partitioning the values of that attribute into clusters or groups.

**Data discretization by decision tree analysis**: Here, a decision tree employs a top-down splitting approach; it is a supervised method. To discretize a numeric attribute, the method selects the value of the attribute that has minimum entropy as a split-point, and recursively partitions the resulting intervals to arrive at a hierarchical discretization.

**Data discretization by correlation analysis**: This employs a bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively. It is supervised method.

##### UNIT II

Introduction to Data Mining: Motivation and importance:

A host of technological advances have resulted in generating a huge amount of electronic data, and have enabled the data to be captured, processed, analyzed, and stored rather inexpensively. This capability has enabled industries and innovations such as • Banking, insurance, financial transactions - electronic banking, ATMs, credit cards, stock market data • Supermarket check-out scanner data, point-of-sale devices, barcode readers • Healthcare - pharmaceutical records • Communications - telephone-call detail records • Location data - GPS, cell phones • Internet and e-commerce - Web logs, click-streams that generate huge volumes of electronic data. For example, Walmart has 20 million transactions/day and a 10 terabyte database. Blockbuster has over 36 million household customers. The need to understand huge, complex, information-rich data sets is important to virtually all fields in business, science and engineering. The ability to extract useful knowledge hidden in these data and to act on that knowledge is becoming vital in today’s increasingly competitive world. Such data (typically terabytes in size) is often stored in data warehouses and data marts.

what is Data Mining:

Data mining is a process where hidden data are analyzed according to multiple perspectives and turning those data into useful information and can be put into action. Data Mining means collecting and assembling data from common areas, like data mining algorithms and data warehouses, look for patterns that can be used by businesses to improve customer service thereby increase their revenue. It is otherwise known as knowledge discovery or data discovery. It is very important in business intelligence to establish decisions that are driven by data.

Data Mining on what kind of data:

**Relational Databases: A database system**, also called a database management system(DBMS), consists of a collection of interrelated data, known as a database, and a set of software programs to manage and access the data.

**A relational database:**is a collection of tables, each of which is assigned a unique nameEach table consists of a set of attributes (columns or fields) and usually stores a large set of tuples (records or rows). Each tuple in a relational table represents an object identified by a unique key and described by a set of attribute values. A semantic data model, such as an entity-relationship (ER) data model, is often constructed for relational databases. An ER data model represents the database as a set of entities and their relationships.

**Data Warehouses:**A data warehouse is a repository of information collected from multiplesources, stored under a unified schema, and that usually resides at a single site. Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing.

* The data are stored to provide information from a historical perspective (such as from the past 5–10 years) and are typically summarized.
* A data warehouse is usually modeled by a multidimensional database structure, where each dimension corresponds to an attribute or a set of attributes in the schema, and each cell stores the value of some aggregate measure, such as count or sales amount
* The actual physical structure of a data warehouse may be a relational data store or a multidimensional data cube. A data cube provides a multidimensional view of data and allows the precomputation and fast accessing of summarized data

**What is the difference between a data warehouse and a data mart**?”you may ask.

* **A data warehouse**collects information about subjects that span anentire organization, andthus its scope isenterprise-wide.
* **A data mart**, on the other hand, is a department subset of a data warehouse. It focuses onselected subjects, and thus its scope is department-wide. Data warehouse systems are well suited for on-line analytical processing, or OLAP. OLAP operations use background knowledge regarding the domain of the data being studied in order to allow the presentation of data at different levels of abstraction. Such operations accommodate different user viewpoints.
* Examples of OLAP operations include drill-down and roll-up, which allow the user to view the data at differing degrees of summarization,

**Transactional Databases:**Transactional database consists of a file where each recordrepresents a transaction. A transaction typically includes a unique transaction identity number (trans ID) and a list of the items making up the transaction (such as items purchased in a store).

The transactional database may have additional tables associated with it, which contain other information regarding the sale, such as the date of the transaction, the customer ID number, the ID

number of the salesperson and of the branch at which the sale occurred, and so on.

**Advanced Data and Information Systems and Advanced Applications**

The new database applications include handling spatial data (such as maps), engineering design data (such as the design of buildings, system components, or integrated circuits), hypertext and multimedia data (including text, image, video, and audio data), time-related data (such as historical records or stock exchange data), stream data (such as video surveillance and sensor data, where data flow in and out like streams), and the WorldWideWeb (a huge, widely distributed information repository made available by the Internet).

These applications require efficient data structures and scalable methods for handling complex object structures; variable-length records; semi structured or unstructured data; text, spatiotemporal, and multimedia data; and database schemas with complex structures and dynamic changes.

**Object-Relational Databases:**Object-relational databases are constructed based on anobject-relational data model. This model extends the relational model by providing a rich data type for handling complex objects and object orientation object-relational databases are becoming increasingly popular in industry and applications.

The object-relational data model inherits the essential concepts of object-oriented databases Each object has associated with it the following:

**A set of variables**that describe the objects. These correspond to attributes in the entityrelationship and relational models.

**A set of messages**that the object can use to communicate with other objects, or with the restof the database system.

**A set of methods**, where each method holds the code to implement a message. Uponreceiving a message, the method returns a value in response. For instance, the method for the message get photo(employee) will retrieve and return a photo of the given employee object.

Objects that share a common set of properties can be grouped into an object class. Each object is an instance of its class. Object classes can be organized into class/subclass hierarchies so that each class represents properties that are common to objects in that class

**Temporal Databases, Sequence Databases, and Time-Series Databases**

**A temporal database**typically stores relational data that include time-related attributes.These attributes may involve several timestamps, each having different semantics.

**A sequence database**stores sequences of ordered events, with or without a concrete notionof time. Examples include customer shopping sequences, Web click streams, and biological sequences. A time series database stores sequences of values or events obtained over repeated measurements of time (e.g., hourly, daily, weekly). Examples include data collected from the stock exchange, inventory control, and the observation of natural phenomena (like temperature and wind).

**Spatial Databases and Spatiotemporal Databases**

**Spatial databases**contain spatial-related information. Examples include geographic (map)databases, very large-scale integration (VLSI) or computed-aided design databases, and medical and satellite image databases.

Spatial data may be represented in raster format, consisting of n-dimensional bit maps or pixel maps. For example, a 2-D satellite image may be represented as raster data, where each pixel registers the rainfall in a given area. Maps can be represented in vector format, where roads, bridges, buildings, and lakes are represented as unions or overlays of basic geometric constructs, such as points, lines, polygons, and the partitions and networks formed by these components.

“What kind of data mining can be performed on spatial databases?” you may ask. Data mining may uncover patterns describing the characteristics of houses located near a specified kind of location, such as a park, for instance. A spatial database that stores spatial objects that change with time is called a spatiotemporal database, from which interesting information can be mined

**Text Databases and Multimedia Databases**

**Text databases**are databases that contain word descriptions for objects. These worddescriptions are usually not simple keywords but rather long sentences or paragraphs, such as product specifications, error or bug reports, warning messages, summary reports, notes, or other documents.

Text databases may be highly unstructured (such as some Web pages on the WorldWideWeb). Some text databases may be somewhat structured, that is, semistructured (such as e-mail messages and many HTML/XML Web pages), whereas others are relatively well structured (such as library catalogue databases). Text databases with highly regular structures typically can be implemented using relational database systems.

“What can data mining on text databases uncover?” By mining text data, one may uncover general and concise descriptions of the text documents, keyword or content associations, as well as the clustering behavior of text objects.

**Multimedia databases**store image, audio, and video data. They are used in applications suchas picture content-based retrieval, voice-mail systems, video-on-demand systems, the World Wide Web, and speech-based user interfaces that recognize spoken commands. Multimedia databases must support large objects, because data objects such as video can require gigabytes of storage. Specialized storage and search techniques are also required. Because video and audio data require real-time retrieval at a steady and predetermined rate in order to avoid picture or sound gaps and system buffer overflows, such data are referred to as continuous-media data.

**Heterogeneous Databases and Legacy Databases**

**A heterogeneous database**consists of a set of interconnected, autonomous componentdatabases. The components communicate in order to exchange information and answer queries. Objects in one component database may differ greatly from objects in other component databases, making it difficult to assimilate their semantics into the overall heterogeneous database.

**A legacy database**is a group ofheterogeneous databasesthat combines different kinds of datasystems, such as relational or object-oriented databases, hierarchical databases, network databases, spreadsheets, multimedia databases, or file systems. The heterogeneous databases in a legacy database may be connected by intra or inter-computer networks.

**Data Streams**

Many applications involve the generation and analysis of a new kind of data, called stream data, where data flow in and out of an observation platform (or window) dynamically. Such data streams have the following unique features: huge or possibly infinite volume, dynamically changing, flowing in and out in a fixed order, allowing only one or a small number of scans, and demanding fast (often real-time) response time.

Typical examples of data streams include various kinds of scientific and engineering data, time-series data, and data produced in other dynamic environments, such as power supply, network traffic, stock exchange, telecommunications, Web click streams, video surveillance, and weather or environment monitoring.

Mining data streams involves the efficient discovery of general patterns and dynamic changes within stream data.

**The World Wide Web**

The World Wide Web and its associated distributed information services, such as Yahoo!, Google, America Online, and AltaVista, provide rich, worldwide, on-line information services, where data objects are linked together to facilitate interactive access. Users seeking information of interest traverse from one object via links to another. Such systems provide ample opportunities and challenges for data mining.

For example, understanding user access patterns will not only help improve system design (by providing efficient access between highly correlated objects), but also leads to better marketing decisions (e.g., by placing advertisements in frequently visited documents, or by providing better customer/user classification and behavior analysis). Capturing user access patterns in such distributed information environments is called Web usage mining (or Weblog mining).

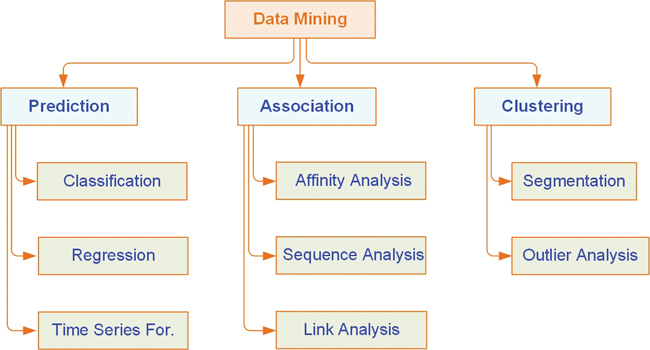
## What Kinds of Patterns Can Data Mining Discover?

Using the most relevant data (which may come from organizational databases or may be obtained from outside sources), data mining builds models to identify patterns among the attributes (i.e., variables or characteristics) that exist in a data set. Models are usually the mathematical representations (simple linear correlations and/or complex highly nonlinear relationships) that identify the relationships among the attributes of the objects (e.g., customers) described in the data set. Some of these patterns are explanatory (explaining the interrelationships and affinities among the attributes), whereas others are predictive (projecting future values of certain attributes). In general, data mining seeks to identify three major types of patterns:

* Associations find commonly co-occurring groupings of things, such as “beers and diapers” or “bread and butter” commonly purchased and observed together in a shopping cart (i.e., market-basket analysis). Another type of association pattern captures the sequences of things. These sequential relationships can discover time-ordered events, such as predicting that an existing banking customer who already has a checking account will open a savings account followed by an investment account within a year.
* Predictions tell the nature of future occurrences of certain events based on what has happened in the past, such as predicting the winner of the Super Bowl or forecasting the absolute temperature on a particular day.
* Clusters identify natural groupings of things based on their known characteristics, such as assigning customers in different segments based on their demographics and past purchase behaviors.

These types of patterns have been manually extracted from data by humans for centuries, but the increasing volume of data in modern times has created a need for more automatic approaches. As data sets have grown in size and complexity, direct manual data analysis has increasingly been augmented with indirect, automatic data processing tools that use sophisticated methodologies, methods, and algorithms. The manifestation of such evolution of automated and semi-automated means of processing large data sets is now commonly referred to as data mining.

As mentioned earlier, generally speaking, data mining tasks and patterns can be classified into three main categories: prediction, association, and clustering. Based on the way in which the patterns are extracted from the historical data, the learning algorithms of data mining methods can be classified as either supervised or unsupervised. With supervised learning algorithms, the training data includes both the descriptive attributes (i.e., independent variables or decision variables) and the class attribute (i.e., output variable or result variable). In contrast, with unsupervised learning, the training data includes only the descriptive attributes. [Figure 2.3](javascript:popUp('/content/images/chap2_9780136738510/elementLinks/02fig03_alt.jpg')) shows a simple taxonomy for data mining tasks, along with the learning methods and popular algorithms for each of the data mining tasks. Out of the three main categories of tasks, prediction patterns/models can be classified as the outcome of a supervised learning procedure, while association and clustering patterns/models can be classified as the outcome of unsupervised learning procedures.

[](javascript:popUp('/content/images/chap2_9780136738510/elementLinks/02fig03_alt.jpg'))

[**Figure 2.3**A Simple Taxonomy for Data Mining](javascript:popUp('/content/images/chap2_9780136738510/elementLinks/02fig03_alt.jpg'))

Prediction is commonly used to indicate telling about the future. It differs from simple guessing by taking into account the experiences, opinions, and other relevant information in conducting the task of foretelling. A term that is commonly associated with prediction is forecasting. Even though many people use these two terms synonymously, there is a subtle difference between them. Whereas prediction is largely experience and opinion based, forecasting is data and model based. That is, in the order of increasing reliability, one might list the relevant terms as guessing, predicting, and forecasting. In data mining terminology, prediction and forecasting are used synonymously, and the term prediction is used as the common representation of the act. Depending on the nature of what is being predicted, prediction can be named more specifically as classification (where the predicted thing, such as tomorrow’s forecast, is a class label such as “rainy” or “sunny”) or regression (where the predicted thing, such as tomorrow’s temperature, is a real number, such as “65 degrees”).

Classification, or supervised induction, is perhaps the most common of all data mining tasks. The objective of classification is to analyze the historical data stored in a database and automatically generate a model that can predict future behavior. This induced model consists of generalizations over the records of a training data set, which help distinguish predefined classes. The hope is that the model can then be used to predict the classes of other unclassified records and, more importantly, to accurately predict actual future events.

Common classification tools include neural networks and decision trees (from machine learning), logistic regression and discriminant analysis (from traditional statistics), and emerging tools such as rough sets, support vector machines, and genetic algorithms. Statistics-based classification techniques (e.g., logistic regression, discriminant analysis) have been criticized as making unrealistic assumptions about the data, such as independence and normality, which limit their use in classification-type data mining projects.

Neural networks involve the development of mathematical structures (somewhat resembling the biological neural networks in the human brain) that have the capability to learn from past experiences, presented in the form of well-structured data sets. They tend to be more effective when the number of variables involved is rather large and the relationships among them are complex and imprecise. Neural networks have disadvantages as well as advantages. For example, it is usually very difficult to provide a good rationale for the predictions made by a neural network. Also, neural networks tend to need considerable training. Unfortunately, the time needed for training tends to increase exponentially as the volume of data increases, and, in general, neural networks cannot be trained on very large databases. These and other factors have limited the applicability of neural networks in data-rich domains. (See Chapter 5, “Algorithms for Predictive Analytics,” for more detailed coverage of neural networks.)

Decision trees classify data into a finite number of classes, based on the values of the input variables. Decision trees are essentially a hierarchy of if–then statements and are thus significantly faster than neural networks. They are most appropriate for categorical and interval data. Therefore, incorporating continuous variables into a decision tree framework requires discretization—that is, the conversion of continuous valued numeric variables to ranges and categories.

A related category of classification tools is rule induction. Unlike with a decision tree, with rule induction, the if–then statements are induced from the training data directly, and they need not be hierarchical in nature. Other, more recent techniques such as SVM, rough sets, and genetic algorithms are gradually finding their way into the arsenal of classification algorithms and are covered in more detail in Chapter 5 as part of the discussion on data mining algorithms.

Using associations—which are commonly called association rules in data mining—is a popular and well-researched technique for discovering interesting relationships among variables in large databases. Thanks to automated data-gathering technologies such as use of bar code scanners, the use of association rules for discovering regularities among products in large-scale transactions recorded by point-of-sale systems in supermarkets has become a common knowledge-discovery task in the retail industry. In the context of the retail industry, association rule mining is often called market-basket analysis.

Two commonly used derivatives of association rule mining are link analysis and sequence mining. With link analysis, the links among many objects of interest are discovered automatically, such as the link between web pages and referential relationships among groups of academic publication authors. With sequence mining, relationships are examined in terms of their order of occurrence to identify associations over time. Algorithms used in association rule mining include the popular Apriori (where frequent item sets are identified), FP-Growth, OneR, ZeroR, and Eclat algorithms. Chapter 4, “Data and Methods for Predictive Analytics,” provides an explanation of Apriori.

Clustering involves partitioning a collection of things (e.g., objects, events, etc., presented in a structured data set) into segments (or natural groupings) whose members share similar characteristics. Unlike in classification, in clustering, the class labels are unknown. As the selected algorithm goes through the data set, identifying the commonalities of things based on their characteristics, the clusters are established. Because the clusters are determined using a heuristic-type algorithm, and because different algorithms may end up with different sets of clusters for the same data set, before the results of clustering techniques are put into use, it may be necessary for an expert to interpret and potentially modify the suggested clusters. After reasonable clusters have been identified, they can be used to classify and interpret new data.

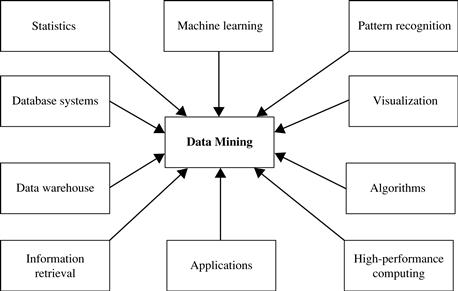
Not surprisingly, clustering techniques include optimization. The goal of clustering is to create groups so that the members within each group have maximum similarity and the members across groups have minimum similarity. The most commonly used clustering techniques include k-means (from statistics) and self-organizing maps (from machine learning), which is a unique neural network architecture developed by Kohonen (1982).

Firms often effectively use their data mining systems to perform market segmentation with cluster analysis. Cluster analysis is a means of identifying classes of items so that items in a cluster have more in common with each other than with items in other clusters. This type of analysis can be used in segmenting customers and directing appropriate marketing products to the segments at the right time in the right format at the right price. Cluster analysis is also used to identify natural groupings of events or objects so that a common set of characteristics of these groups can be identified to describe them.

Two techniques often associated with data mining are visualization and time-series forecasting. Visualization can be used in conjunction with other data mining techniques to gain a clearer understanding of underlying relationships. As the importance of visualization has increased in recent years, the term visual analytics has emerged. The idea is to combine analytics and visualization in a single environment for easier and faster knowledge creation. Visual analytics is covered in detail in Chapter 4. In time-series forecasting, the data consists of values of the same variable that is captured and stored over time, at regular intervals. This data is then used to develop forecasting models to extrapolate the future values of the same variable.

## Which Technologies Are Used?

As a highly application-driven domain, data mining has incorporated many techniques from other domains such as statistics, machine learning, pattern recognition, database and data warehouse systems, information retrieval, visualization, algorithms, high-performance computing, and many application domains The interdisciplinary nature of data mining research and development contributes significantly to the success of data mining and its extensive applications. In this section, we give examples of several disciplines that strongly influence the development of data mining methods.



## Which Kinds of Applications Are Targeted?

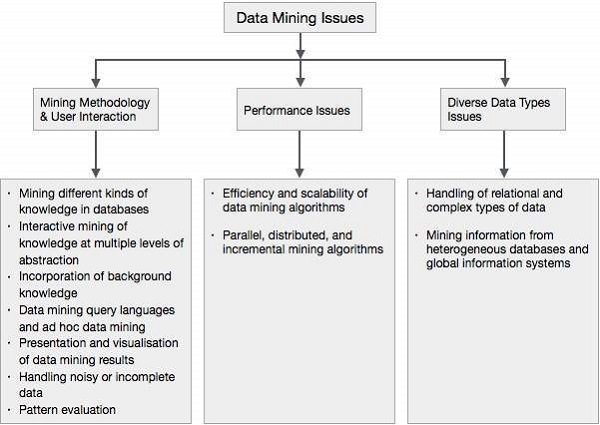
As a highly application-driven discipline, data mining has seen great successes in many applications. It is impossible to enumerate all applications where data mining plays a critical role. Presentations of data mining in knowledge-intensive application domains, such as bioinformatics and software engineering, require more in-depth treatment and are beyond the scope of this book. To demonstrate the importance of applications as a major dimension in data mining research and development, we briefly discuss two highly successful and popular application examples of data mining: business intelligence and search engines.

**Data Mining - Issues**

Data mining is not an easy task, as the algorithms used can get very complex and data is not always available at one place. It needs to be integrated from various heterogeneous data sources. These factors also create some issues. Here in this tutorial, we will discuss the major issues regarding −

* Mining Methodology and User Interaction
* Performance Issues
* Diverse Data Types Issues

The following diagram describes the major issues.



## Mining Methodology and User Interaction Issues

It refers to the following kinds of issues −

* **Mining different kinds of knowledge in databases** − Different users may be interested in different kinds of knowledge. Therefore it is necessary for data mining to cover a broad range of knowledge discovery task.
* **Interactive mining of knowledge at multiple levels of abstraction** − The data mining process needs to be interactive because it allows users to focus the search for patterns, providing and refining data mining requests based on the returned results.
* **Incorporation of background knowledge** − To guide discovery process and to express the discovered patterns, the background knowledge can be used. Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple levels of abstraction.
* **Data mining query languages and ad hoc data mining** − Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.
* **Presentation and visualization of data mining results** − Once the patterns are discovered it needs to be expressed in high level languages, and visual representations. These representations should be easily understandable.
* **Handling noisy or incomplete data** − The data cleaning methods are required to handle the noise and incomplete objects while mining the data regularities. If the data cleaning methods are not there then the accuracy of the discovered patterns will be poor.
* **Pattern evaluation** − The patterns discovered should be interesting because either they represent common knowledge or lack novelty.

## Performance Issues

There can be performance-related issues such as follows −

* **Efficiency and scalability of data mining algorithms** − In order to effectively extract the information from huge amount of data in databases, data mining algorithm must be efficient and scalable.
* **Parallel, distributed, and incremental mining algorithms** − The factors such as huge size of databases, wide distribution of data, and complexity of data mining methods motivate the development of parallel and distributed data mining algorithms. These algorithms divide the data into partitions which is further processed in a parallel fashion. Then the results from the partitions is merged. The incremental algorithms, update databases without mining the data again from scratch.

## Diverse Data Types Issues

* **Handling of relational and complex types of data** − The database may contain complex data objects, multimedia data objects, spatial data, temporal data etc. It is not possible for one system to mine all these kind of data.
* **Mining information from heterogeneous databases and global information systems** − The data is available at different data sources on LAN or WAN. These data source may be structured, semi structured or unstructured. Therefore mining the knowledge from them adds challenges to data mining.

## Data Objects and Attribute Types

Data sets are made up of data objects. A **data object** represents an entity—in a sales database, the objects may be customers, store items, and sales; in a medical database, the objects may be patients; in a university database, the objects may be students, professors, and courses. Data objects are typically described by attributes. Data objects can also be referred to as samples, examples, instances, data points, or objects. If the data objects are stored in a database, they are data tuples. That is, the rows of a database correspond to the data objects, and the columns correspond to the attributes. In this section, we define attributes and look at the various attribute types.

## What is an Attribute?

The attribute can be defined as a field for storing the data that represents the characteristics of a data object. The attribute is the property of the object. The attribute represents different features of the object.  For example, hair color is the attribute of a lady. Similarly, rollno, and marks are attributes of a student. An attribute vector is commonly known as a set of attributes that are used to describe a given object.  
Type of attributes  
We need to differentiate between different types of attributes during Data-preprocessing. So firstly, we need to differentiate between qualitative and quantitative attributes.  
1. **Qualitative Attributes** such as Nominal, Ordinal, and Binary Attributes.  
2. **Quantitative Attributes** such as Discrete and Continuous Attributes.  
There are different types of attributes. some of these attributes are mentioned below;

**Example of attribute**

In this example, RollNo, Name, and Result are attributes of the object named as a student.

|  |  |  |
| --- | --- | --- |
| **Rollo** | **Name** | **Result** |
| 1 | Ali | Pass |
| 2 | Akram | Fail |

## ****Types Of attributes****

* Binary
* Nominal

## Ordinal Attributes

* Numeric
  + Interval-scaled
  + Ratio-scaled

# Nominal Attributes

[Nominal data](https://t4tutorials.com/proximity-measure-for-nominal-attributes-in-data-mining/) is in alphabetical form and not in an integer. Nominal Attributes are Qualitative Attributes.

**Examples of Nominal attributes**

In this example, sates and colors are the attribute and New, Pending, Working, Complete, Finish and Black, Brown, White, and Red are the values.

|  |  |
| --- | --- |
| **Attribute** | **Value** |
| [Categorical data](https://t4tutorials.com/decision-tree-induction-calculation-on-categorical-attributes-in-data-mining/) | Lecturer, [Assistant Professor](https://t4tutorials.com/mcqs-for-subject-test-chemistry-assistant-professor/), Professor |
| States | New, Pending, Working, Complete, Finish |
| Colors | Black, Brown, White, Red |

# Binary Attributes

Binary data have only two values/states. For example, here HIV detected can be only Yes or No.  
Binary Attributes are Qualitative Attributes.

|  |  |
| --- | --- |
| **Attribute** | **Value** |
| HIV detected | Yes, No |
| Result | Pass, Fail |

**Examples of Binary Attributes**

The [binary attribute](https://t4tutorials.com/proximity-measure-for-binary-attributes-in-data-mining/) is of two types;

1. [Symmetric binary](https://t4tutorials.com/distance-measure-for-symmetric-binary-variables/)
2. [Asymmetric binary](https://t4tutorials.com/jaccard-coefficient-similarity-measure-for-asymmetric-binary-variables/)

# Examples of ****Symmetric data****

Both values are equally important. For example, if we have open admission to our university, then it does not matter, whether you are a male or a female.

**Example:**

|  |  |
| --- | --- |
| **Attribute** | **Value** |
| Gender | Male, Female |

# Examples of ****Asymmetric data****

Both values are not equally important. For example, HIV detected is more important than HIV not detected. If a patient is with HIV and we ignore him, then it can lead to death but if a person is not HIV detected and we ignore it, then there is no special issue or risk.

**Example**

|  |  |
| --- | --- |
| **Attribute** | **Value** |
| HIV detected | Yes, No |
| Result | Pass, Fail |

## Ordinal Attributes

All Values have a meaningful order.  For example, Grade-A means highest marks, B means marks are less than A, C means marks are less than grades A and B, and so on. Ordinal Attributes are Quantitative Attributes.

**Examples of Ordinal Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Value** |
| Grade | A, B, C, D, F |
| BPS- Basic pay scale | 16, 17, 18 |

**Discrete Attributes**

Discrete data have a finite value. It can be in numerical form and can also be in a categorical form. Discrete Attributes are Quantitative Attributes.

**Examples of Discrete Data**

|  |  |
| --- | --- |
| **Attribute** | **Value** |
| Profession | Teacher, Bussiness Man, Peon etc |
| Postal Code | 42200, 42300 etc |

**Example of Continuous Attribute**

Continuous data technically have an infinite number of steps.

Continuous data is in float type. There can be many numbers in between 1 and 2. These attributes are Quantitative Attributes.

**Example of Continuous Attribute**

|  |  |
| --- | --- |
| **Attribute** | **Value** |
| Height | 5.4…, 6.5….. etc |
| Weight | 50.09….  etc |

# Basic Statistical Descriptions of Data - Mean, Median, Mode & Midrange:

The basic statistical descriptions of data help us measure some very special properties of the data. One of these properties is the central tendency. Measuring the central tendency helps us know, where does most of the data lie taking into account the whole set of data.

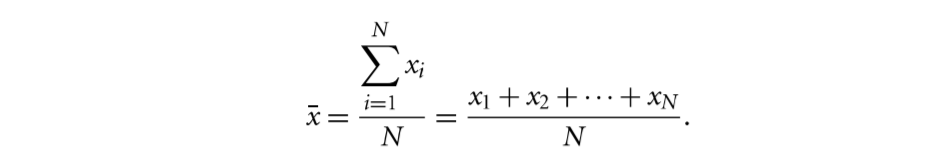
Let us take a use case. Suppose that we have a set of values and we want to find a value that has the capability of replacing the whole dataset and still achieve a relevant result. Finding the central tendency helps us achieve this use case.

Let us discuss some of the central tendencies we can use -

**Mean:**Suppose that we have a dataset, in which, we have an attribute “age” of supposing 100 people. Let the corresponding ages be a1, a2, a3…..an.

The mean of the ages of these 100 people means the mean-age of the people, which is equivalent to answering, “what age do most of the people belong to?”

Mathematically, the mean of n values can be defined as:



Along with the benefits of finding the mean of the data, there are some drawbacks. One of them is when there are some extreme values in the data.

Let us take a case. Suppose that out of 100 people in a company. 95 have a salary in the range 2 Lakhs to 5 Lakhs, but 5 people have age above 100 Lakhs. In this case, the mean salary will be around 8 Lakhs. But as we can see that most of the people have salaries between 2 Lakhs and 5 Lakhs, so this result did have much significance and was not at all useful. We cannot replace the whole dataset with the mean in this scenario. In such cases, we have another measure of central tendency which is the Median of the data.

**Median:**When our dataset has skewness (data is asymmetric), calculating the Median could prove to be more beneficial than Mean.

Median is defined as the centermost value of an ordered numerical dataset. For calculating the Median, it is important for the dataset to be in some order, i.e. it should be sorted.

Let us take the same above case again. Out of 100 people in a company, 95 have a salary in the range 2 Lakhs to 5 Lakhs, but 5 people have age above 100 Lakhs. The Median of this dataset will still lie between 2 Lakhs and 5 Lakhs. So we can see that the Median of the dataset is not affected by extreme values in the dataset. Therefore in such scenarios, the Median of the dataset has more significance.

**Mode:**This is another measure of central tendency. The mode for a set of data is the value that occurs most frequently in the set. Hence, it can be calculated for both qualitative and quantitative attributes.

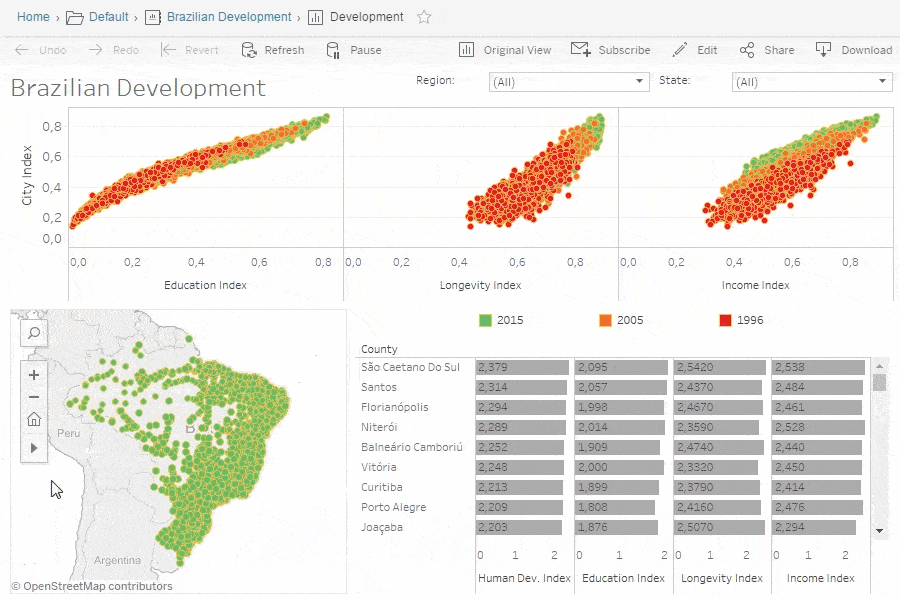
There is an equal possibility that a dataset might have two modes. Such datasets are known as bimodal. In general, a dataset with two or more modes is known as multimodal.

**Midrange:** This is defined as the average of the largest and smallest values in the set of values.

Data visualization is the graphical representation of information and data. By using [visual elements like charts, graphs, and maps](https://www.tableau.com/data-insights/reference-library/visual-analytics), data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

## The advantages and benefits of good data visualization

Our eyes are [drawn to colors and patterns](https://www.tableau.com/learn/whitepapers/tableau-visual-guidebook). We can quickly identify red from blue, square from circle. Our culture is visual, including everything from art and advertisements to TV and movies. Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message. When we see a chart, we [quickly see trends and outliers](https://www.tableau.com/reports/business-intelligence-trends). If we can see something, we internalize it quickly. It’s storytelling with a purpose. If you’ve ever stared at a massive spreadsheet of data and couldn’t see a trend, you know how much more effective a visualization can be.

### Big Data is here and we need to know what it says

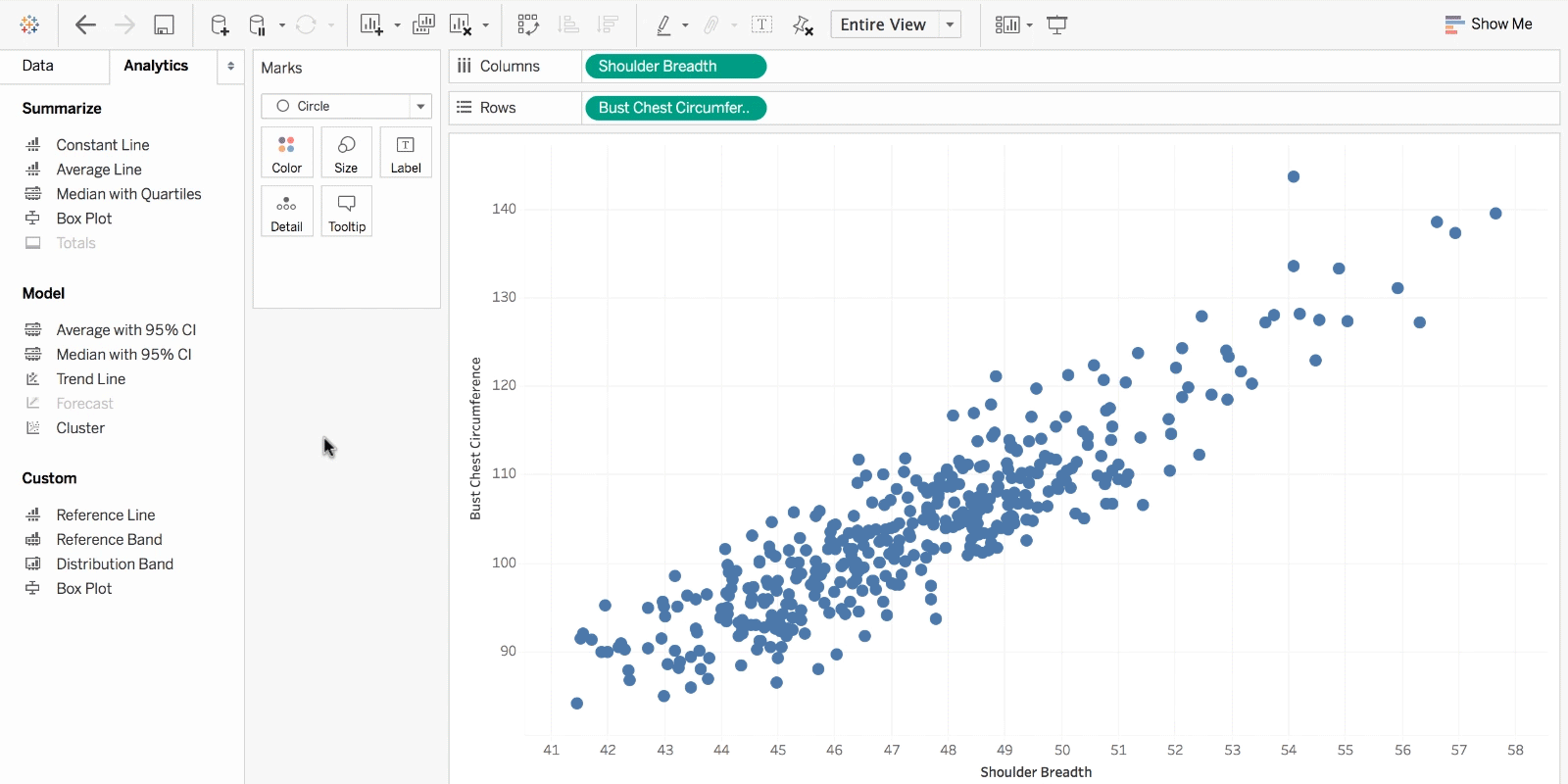
As the [“age of Big Data” kicks into high-gear](https://www.economist.com/news/leaders/21721656-data-economy-demands-new-approach-antitrust-rules-worlds-most-valuable-resource), visualization is an increasingly key tool to make sense of the trillions of rows of data generated every day. Data visualization helps to tell stories by curating data into a form easier to understand, highlighting the trends and outliers. A good visualization tells a story, removing the noise from data and highlighting the useful information. However, it’s not simply as easy as just dressing up a graph to make it look better or slapping on the “info” part of an infographic. Effective data visualization is a delicate balancing act between form and function. The plainest graph could be too boring to catch any notice or it make tell a powerful point; the most stunning visualization could utterly fail at conveying the right message or it could speak volumes. The data and the visuals need to work together, and there’s an art to combining great analysis with great storytelling.

### Why data visualization is important for any career

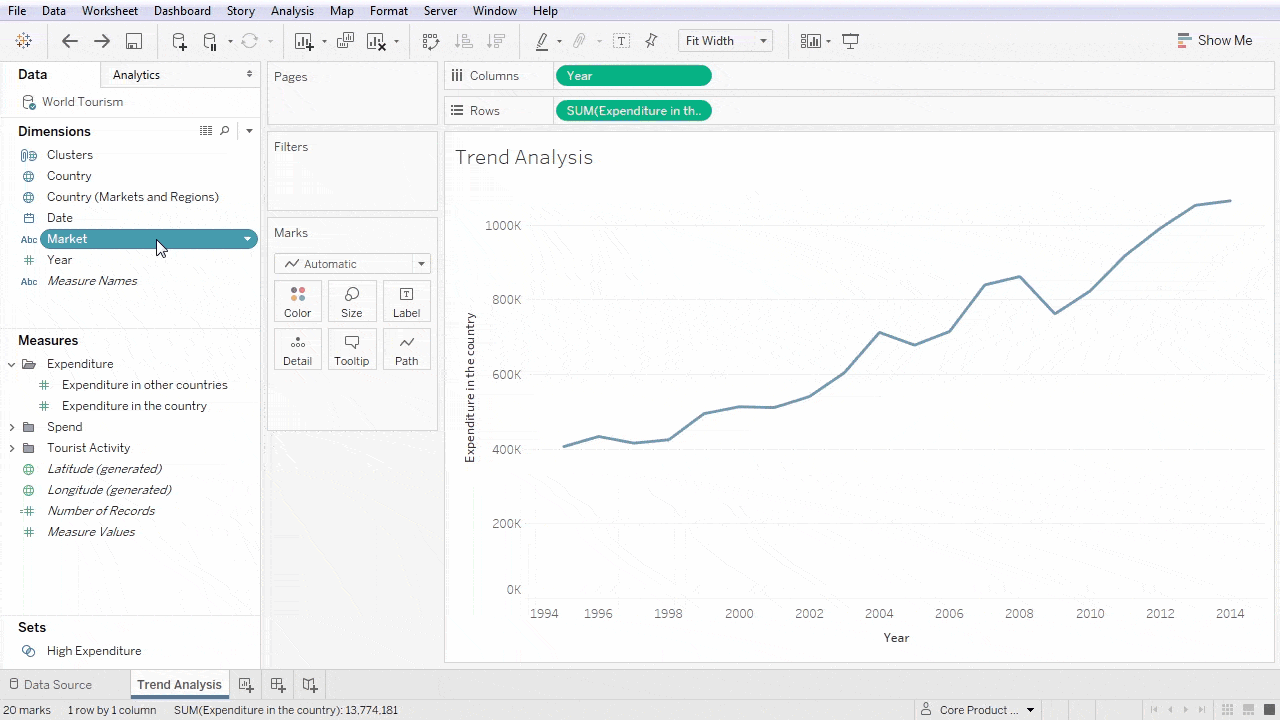
It’s hard to think of a professional industry that doesn’t benefit from [making data more understandable](https://www.forbes.com/sites/jeffkauflin/2017/07/20/the-five-most-in-demand-skills-for-data-analysis-jobs/#3e300312c7ce). Every STEM field benefits from understanding data—and so do fields in government, finance, marketing, history, consumer goods, service industries, education, sports, and so on. While we’ll always wax poetically about data visualization (you’re on the Tableau website, after all) there are practical, real-life applications that are undeniable. And, since visualization is so prolific, it’s also one of the most useful professional skills to develop. The better you can convey your points visually, whether in a dashboard or a slide deck, the better you can leverage that information. The concept of [the citizen data scientist is on the rise](https://www.gartner.com/newsroom/id/3570917). Skill sets are changing to accommodate a data-driven world. It is increasingly valuable for professionals to be able to use data to make decisions and use visuals to tell stories of when data informs the who, what, when, where, and how. While traditional education typically draws a distinct line between creative storytelling and technical analysis, the modern professional world also values those who can cross between the two: data visualization sits right in the middle of analysis and visual storytelling.

## Examples of data visualization in action

##### visualizations

Of course, one of the best ways to understand data visualization is to see it. What a crazy concept! With public data visualization galleries and data everywhere online, it can be overwhelming to know where to start. We’ve collected [10 of the best examples of data visualization of all time](https://www.tableau.com/learn/articles/best-beautiful-data-visualization-examples), with examples that map historical conquests, analyze film scripts, reveal hidden causes of mortality, and more. Tableau’s own [public gallery](https://public.tableau.com/s/) shows off loads of visualizations made with the free Tableau Public tool, we feature some common starter business dashboards as usable templates, and [Viz of the Day collects some of the best community creations](https://public.tableau.com/en-us/s/gallery). Plus, there are [tons of great blogs](https://www.tableau.com/learn/articles/best-data-visualization-blogs) and [books about data visualization](https://www.tableau.com/learn/articles/books-about-data-visualization)containing excellent examples, explanations, and information about best practices.

### The different types of visualizations

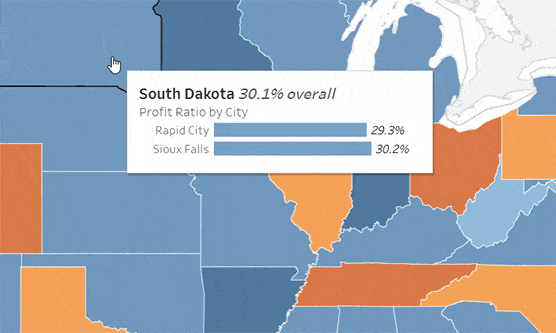
When you think of data visualization, your first thought probably immediately goes to simple bar graphs or [pie charts](https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/pie-charts). While these may be an integral part of visualizing data and a common baseline for many data graphics, the right visualization must be paired with the right set of information. [Simple graphs are only the tip of the iceberg](https://www.tableau.com/learn/whitepapers/which-chart-or-graph-is-right-for-you). There’s a whole selection of visualization methods to present data in effective and interesting ways. **Common general types of data visualization:**

* Charts
* Tables
* Graphs
* Maps
* Infographics
* Dashboards

**More specific examples of methods to visualize data:**

* Area Chart
* Bar Chart
* Box-and-whisker Plots
* Bubble Cloud
* [Bullet Graph](https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/bullet-graphs)
* Cartogram
* Circle View
* Dot Distribution Map
* Gantt Chart
* Heat Map
* Highlight Table
* Histogram
* Matrix
* Network
* Polar Area
* Radial Tree
* Scatter Plot (2D or 3D)
* Streamgraph
* Text Tables
* Timeline
* Treemap
* Wedge Stack Graph
* Word Cloud
* And any mix-and-match combination in a dashboard!

## Learn more about data visualizations (and how to create your own)

 If you’re feeling inspired or want to learn more, there are tons of resources to tap into. Data visualization and data journalism are full of enthusiastic practitioners eager to share their tips, tricks, theory, and more.

### Blogs about data visualization are a perfect place to start

See our list of great [data visualization blogs full of examples](https://www.tableau.com/learn/articles/best-data-visualization-blogs), inspiration, and educational resources. The experts who write books and teach classes about the theory behind data visualization also tend to keep blogs where they analyze the latest trends in the field and discuss new vizzes. Many will offer critique on modern graphics or write tutorials to create effective visualizations. Others will collect many different data visualizations from around the web in order to highlight the most intriguing ones. Blogs are a great way to learn more about specific subsets of data visualization or to look for relatable inspiration from well-done projects.

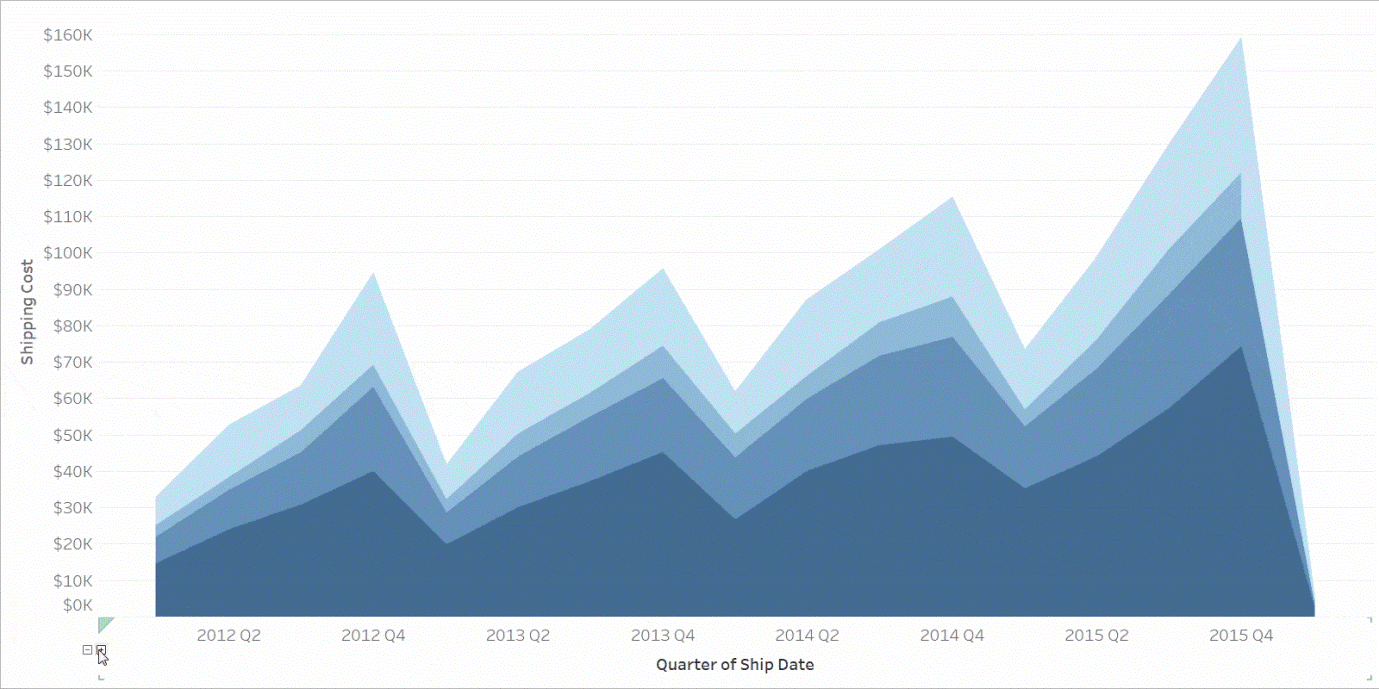
### Learn about historical examples and theory from books

Read our list of great [books about data visualization theory and practice](https://www.tableau.com/learn/articles/books-about-data-visualization). While blogs can keep up with the changing field of data visualization, books focus on where the theory stays constant. Humans have been trying to present data in a visual form throughout our entire existence. One of the earlier books about data visualization, originally published in 1983, set the stage for data visualization to come and still remains relevant to this day. More current books still deal with theory and techniques, offering up timeless examples and practical tips. Some even take completed projects and present the visual graphics in book-form as an archival display.

### There are loads of free courses and paid training programs

There are plenty of great paid and free courses and resources on data visualization out there, including [right here on the Tableau website](https://www.tableau.com/learn). There are videos, articles, and whitepapers for everyone from beginner to data rockstar. When it comes to third-party courses, however, we won’t provide specific suggestions in this article at this time.

**A note on data visualization tools and software**

There are [dozens of tools for data visualization and data analysis](http://bigdata-madesimple.com/review-of-20-best-big-data-visualization-tools/). These range from simple to complex, from intuitive to obtuse. Not every tool is right for every person looking to learn visualization techniques, and not every tool can scale to industry or enterprise purposes. If you’d like to learn more about the options, feel free to [read up here](https://www.tableau.com/compare) or dive into [detailed third-party analysis like the Gartner Magic Quadrant](https://www.tableau.com/reports/gartner). Also, remember that good data visualization theory and skills will transcend specific tools and products. When you’re learning this skill, focus on best practices and explore your own personal style when it comes to visualizations and dashboards. Data visualization isn’t going away any time soon, so it’s important to build a foundation of analysis and storytelling and exploration that you can carry with you regardless of the tools or software you end up using.

## Introduction

We consider similarity and dissimilarity in many places in data science.

**Similarity** measure

* is a numerical measure of how alike two data objects are.
* higher when objects are more alike.
* often falls in the range [0,1]

Similarity might be used to identify

* duplicate data that may have differences due to typos.
* equivalent instances from different data sets. E.g. names and/or addresses that are the same but have misspellings.
* groups of data that are very close (clusters)

**Dissimilarity** measure

* is a numerical measure of how different two data objects are
* lower when objects are more alike
* minimum dissimilarity is often 0 while the upper limit varies depending on how much variation can be

Dissimilarity might be used to identify

* outliers
* interesting exceptions, e.g. credit card fraud
* boundaries to clusters

**Proximity** refers to either a similarity or dissimilarity

## Single attribute sim/dissim measures



**Nominal** is binary if two values are equal or not

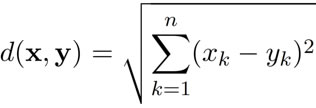
**Ordinal** is the difference between two values, normalized by the maximum distance

**Quantitative** dissimilarity is just a distance between, similarity attempts to scale that distance to [0,1]

## Distance between instances with multiple attributes.

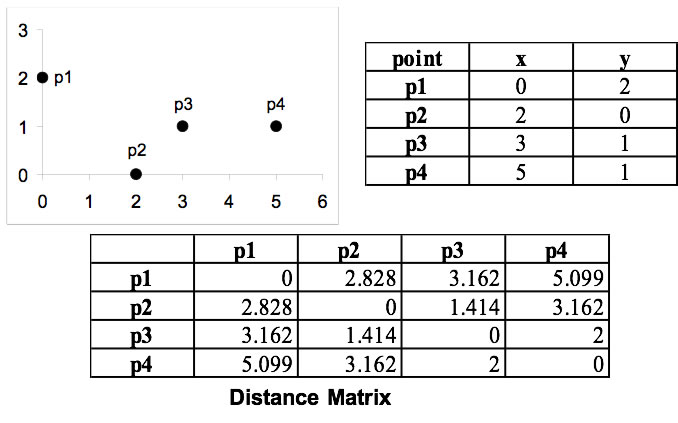
Attributes are naturally numbers or ordinal, but nominal must resort to the binary 0 or 1 if match or not

Euclidean distance



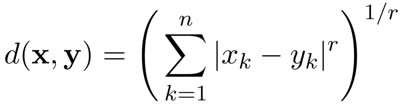
where n is the number of dimensions (attributes) and xk and yk  are, respectively, the k-th attributes (components) or data objects **x** and **y**

Standardization/normalization may be necessary to ensure an attribute does not skew the distances due to different scales.



## Minkowski Distance

is a generalization of Euclidean Distance



where r is a parameter, n is the number of dimensions (attributes) and xk and yk  are, respectively, the k-th attributes (components) or data objects **x** and **y.**

### Examples

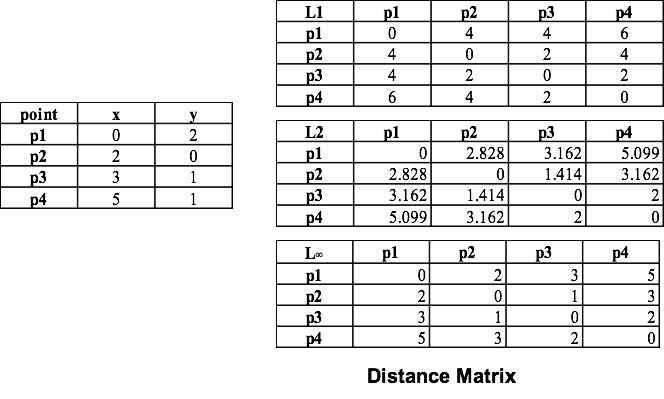
**r = 1.** "City block",  "Manhattan", "taxicab", L1 norm distance.

* Another example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors

**r = 2.**Euclidean distance (L2 norm)

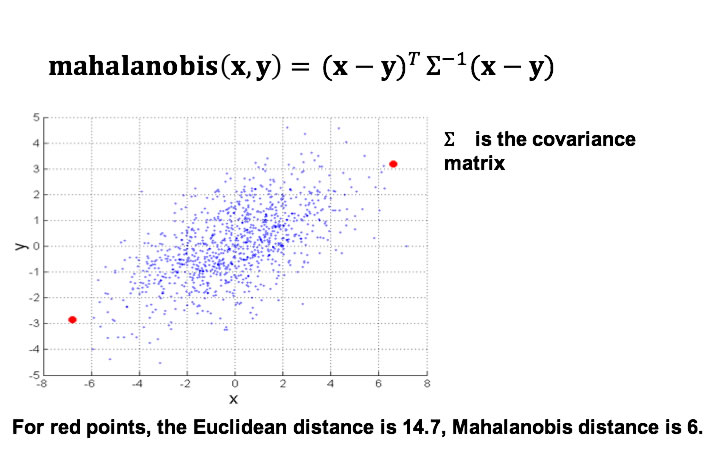
**r = ∞.** “supremum” (Lmax norm, L∞ norm) distance. This is the maximum difference between any component of the vectors

Do not confuse r with n, i.e., all these distance measures are defined for all numbers of dimensions.



## Mahalanobis Distance

Essentially this measures distance from the centroid of the cluster and there's significant correlation among the attributes.



## Common Properties of Distance

Distances, such as the Euclidean distance, have some well known properties.

1. **Positivity:** d(x, y) ≥ 0 for all x and **y**, and d(x, y) = 0 only if **x** = **y**.
2. **Symmetry**: d(x, y) = d(y, x) for all x and **y**.
3. **Triangle Inequality:**d(x, z) ≤ d(x, y) + d(y, z) for all points x, y, and z.

where d(x, y) is the distance (dissimilarity) between points (data objects), x and y.

A distance that satisfies these properties is a **metric**.

## Similarity Properties

Similarities, also have some well known properties.

1. s(x, y) = 1 (or maximum similarity) only if x = y (0 ≤ s ≤ 1)
2. s(x, y) = s(y, x) for all x and y. (Symmetry)

where s(x, y) is the similarity between points (data objects), x and y.

## Similarity Between Binary Vectors

A common situation is that two objects, p and q, have only binary attributes.  Example: A set of products bought or not.

Compute similarities using the following quantities (counts)

f01 = the number of attributes where p was 0 and q was 1  
f10 = the number of attributes where p was 1 and q was 0  
f00 = the number of attributes where p was 0 and q was 0  
f11 = the number of attributes where p was 1 and q was 1

### Simple Matching and Jaccard Coefficients

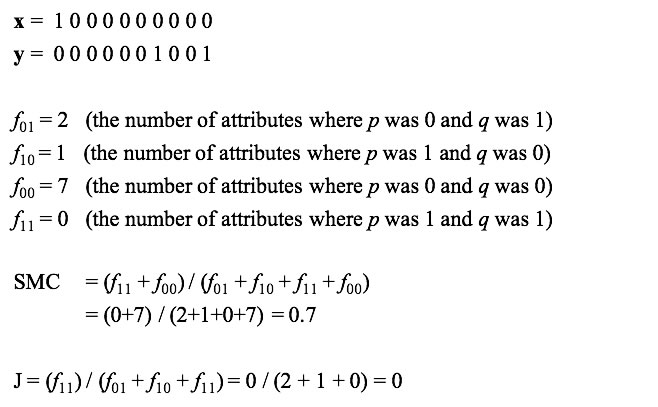
Simple Matching Coef. SMC is measuring similarity in a symmetric setting, i.e., both 0 and 1 matches

* SMC = number of matches / number of attributes = (f11 + f00) / (f01 + f10 + f11 + f00)

Jaccard measures an asymmetric setting, i.e., only consider matching of 1's, ignoring the 0's also in the denominator

J = number of 11 matches / number of non-zero attributes = (f11) / (f01 + f10 + f11)

### Example SMC v Jaccard



### Example 2

x = 1 0 1 1 0 1 0 0 0 1  
y = 1 1 0 1 0 0 0 0 1 1

f01= 2,  f10= 2, f00=3, f11= 3

SMC = 7/10 and J = 3/7

## Cosine Similarity

When you have a set of quantified attributes for each instance-- an alternative to Minkowski distances.

If **d1** and **d2** are two document vectors, then **cos**( **d1**, **d2** ) = <**d1**,**d2**> / ||**d1**|| ||**d2**|| ,   
where <**d1**,**d2**> indicates inner product or vector dot product  **d1**'**d2** of vectors  **d1** and **d2**, and || d || is the length of vector d.

Example:

**d1** = 3 2 0 5 0 0 0 2 0 0   
**d2** = 1 0 0 0 0 0 0 1 0 2

<**d1**, **d2**> = 3\*1 + 2\*0 + 0\*0 + 5\*0 + 0\*0 + 0\*0 + 0\*0 + 2\*1 + 0\*0 + 0\*2 = 5  
|| **d1** || = (3\*3+2\*2+0\*0+5\*5+0\*0+0\*0+0\*0+2\*2+0\*0+0\*0)1/2 = (42) 1/2 = 6.481  
|| **d2** || = (1\*1+0\*0+0\*0+0\*0+0\*0+0\*0+0\*0+1\*1+0\*0+2\*2) 1/2 = (6)1/2 = 2.449

**cos**(**d1**, **d2** ) = 0.3150

### 

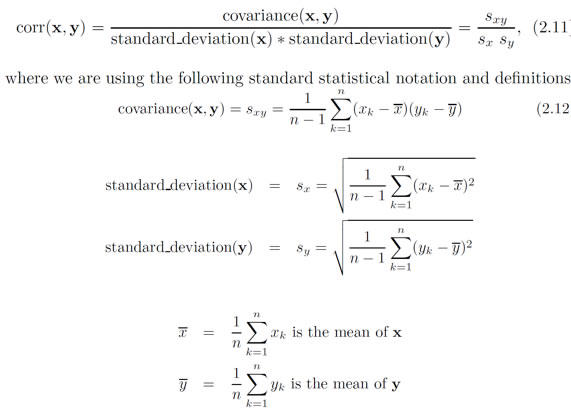
### Extended Jaccard Coefficient (Tanimoto)

Variation of Jaccard for continuous or count attributes

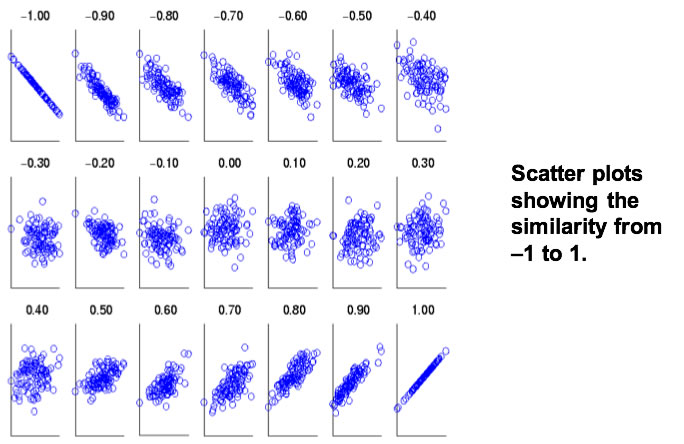
* Reduces to Jaccard for binary attributes



## Correlation



### Visually evaluating correlation



### Counterexample of correlation computation

x = (-3, -2, -1, 0, 1, 2, 3)  
y = (9, 4, 1, 0, 1, 4, 9)

In this case, notice that **yi = xi2** so there's clearly a functional relationship between x and y

μ(x) = 0, μ(y) = 4  
σ(x) = 2.16, σ(y) = 3.74

But the correlation = (-3)(5)+(-2)(0)+(-1)(-3)+(0)(-4)+(1)(-3)+(2)(0)+3(5) / ( 6 \* 2.16 \* 3.74 ) = 0

## Comparison of Proximity Measures

How to choose among the proximity measures?

Domain of application often drives choice

* Similarity measures tend to depend on the type of attribute and data
* Record data, images, graphs, sequences, 3D-protein structure, etc. will use different measures

However, one can talk about various properties that you would like a proximity measure to have

* Symmetry is a common one
* Tolerance to noise and outliers is another
* Ability to find more types of patterns?
* Many others possible

The measure must be applicable to the data and produce results that agree with domain knowledge.

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Cosine** | **Correlation** | **Minkowski** |
| Invariant to scaling (multiply) | Yes | Yes | No |
| Invariant to translations (add) | Yes | Yes | No |
| x=(1,2,4,3,0,0) and y=(1,2,3,4,0,0,0) | 0.9667 | 0.9429 | 1.412 (r=2) |
| x same and s = (2,4,6,8,0,0,0) scaled x | 0.9667 | 0.9429 | 5.831 (r=2) |
| x samd and t = (6,7,8,9,5,5,5) translated x | 0.794 | 0.9429 | 14.2127 (r=2) |

Information Based Measures

**Information theory** is a well-developed and fundamental disciple with broad applications

Information relates to possible outcomes of an event

E,g, transmission of a message, flip of a coin, or measurement of a piece of data

The more certain an outcome, the less information that it contains and vice-versa

* For example, if a coin has two heads, then an outcome of heads provides no information

More quantitatively, the information is [inversely] related the probability of an outcome

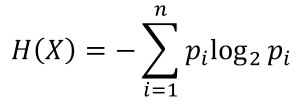
* The smaller the probability of an outcome, the more information it provides and vice-versa

## Entropy

is the commonly used measure for information

For a variable (event), X, with n possible values (outcomes), **x1, x2 …, xn**   
and each outcome having probability, **p1, p2 …, pn**

the entropy of X , **H(X)**, is given by



**0 ≤ H(X) ≤ log2n**and is measured in **bits**.

Thus, entropy is a measure of how many bits it takes to represent an observation of X on average. Not likely an integer!

### Examples

For a coin with probability p for heads and probability q = 1 – p for tails

**H = - p log2 p - q log2 q**

For a fair coin, p= 0.5, q = 0.5**H = 1**  
For a weighted coin, p = 1 or q = 1,**H = 0**

A more realistic example

|  |  |  |  |
| --- | --- | --- | --- |
| **Hair Color** | **Count** | **p** | **-p log2p** |
| **Black** | 75 | 0.75 | 0.3113 |
| **Brown** | 15 | 0.15 | 0.4105 |
| **Blond** | 5 | 0.05 | 0.2161 |
| **Red** | 0 | 0.00 | 0 |
| **Other** | 5 | 0.05 | 0.2161 |
| **Total** | **100** | **1.0** | **1.1540** |

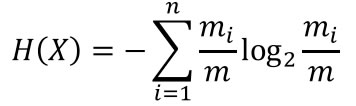
Maximum entropy is log25 = 2.3219, that is, we need more than 2 bits for 5 unique values, but not more than 3

In general, suppose we have a number of observations (**m**) of some attribute, X, e.g., the hair color of students in the class,

where there are **n** different possible values.

And the number of observations in the **ith** category is **mi** , thus the probability is **mi / m**

Then, for this sample, the entropy is



For continuous data, the calculation is harder.

## Density

Measures the degree to which data objects are close to each other in a specified area

The notion of density is closely related to that of proximity

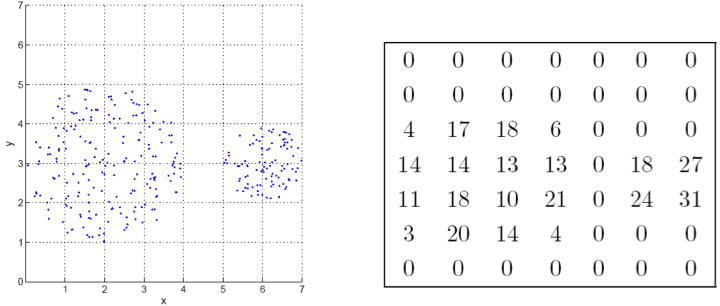
Concept of density is typically used for clustering and anomaly detection

Examples:

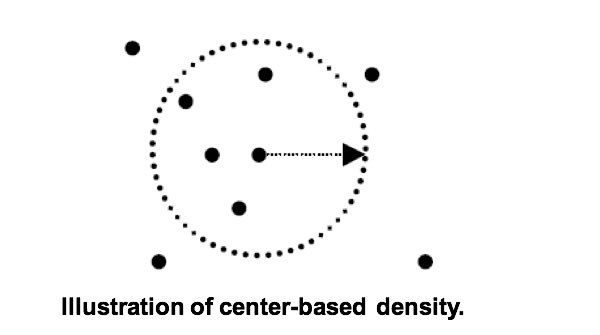
* **Euclidean density** = number of points per unit volume
* **Probability density**: Estimate what the distribution of the data looks like
* **Graph-based density**: Connectivity

### Euclidean Density: Grid-based Approach

Simplest approach is to divide region into a number of rectangular cells of equal volume,   
then define density as # of points the cell contains.



Euclidean density is the number of points within a specified radius of the point





**UNIT III**

Concept Description:

Descriptive vs. predictive data mining – Descriptive mining: describes concepts or task‐relevant data sets in concise, summarative, informative, discriminative forms – Predictive Predictive mining: mining: Based on data and analysis analysis, constructs constructs models for the database, and predicts the trend and properties of unknown data • Concept description: – Characterization: provides a concise and succinct s mmari ation summarization of the gi en v collection collection of data – Comparison: provides descriptions comparing two or more collections of data What is Concept Concept Description? Description? 3 Concept Description vs. OLAP • Concept description: – can h dl an e compl d f h ex data types of the attributes and their aggregations – a more aut td oma e process • OLAP: – restricted to a small number of dimension and measure types – user‐controlled process

## Data Generalization by Attribute-Oriented Induction

Conceptually, the data cube can be viewed as a kind of multidimensional data generalization. In general, data generalizationsummarizes data by replacing relatively low-level values (e.g., numeric values for an attribute age) with higher-level concepts (e.g., young, middle-aged, and senior), or by reducing the number of dimensions to summarize data in concept space involving fewer dimensions (e.g., removingbirth\_dateand telephone number when summarizing the behavior of a group of students). Given the large amount of data stored in databases, it is useful to be able to describe concepts in concise and succinct terms at generalized (rather than low) levels of abstraction. Allowing data sets

## Attribute-Oriented Induction

The Attribute-Oriented Induction (AOI) approach to data generalization and summarization – based characterization was first proposed in 1989 (KDD ‘89 workshop) a few years before the introduction of the data cube approach.   
  
The data cube approach can be considered as a data warehouse – based, pre computational – oriented, materialized approach.  
  
It performs off-line aggregation before an OLAP or data mining query is submitted for processing.    
  
On the other hand, the attribute oriented induction approach, at least in its initial proposal, a relational database query – oriented, generalized – based, on-line data analysis technique.  
  
However, there is no inherent barrier distinguishing the two approaches based on online aggregation versus offline precomputation.  
  
Some aggregations in the data cube can be computed on-line, while off-line precomputation of multidimensional space can speed up attribute-oriented induction as well.  
  
It was proposed in 1989 (KDD ‘89 workshop).  
  
It is not confined to categorical data nor particular measures.  
  
Collect the task-relevant data( initial relation) using a relational database query

* Perform generalization by attribute removal or attribute generalization.
* Apply aggregation by merging identical, generalized tuples and accumulating their respective counts.
* Reduces the size of the generalized data set.
* Interactive presentation with users.

## Basic Principles Of Attribute Oriented Induction

### Data focusing:

* Analyzing task-relevant data, including dimensions, and the result is the initial relation.

### Attribute-removal:

* To remove attribute A if there is a large set of distinct values for A but (1) there is no generalization operator on A, or (2) A’s higher-level concepts are expressed in terms of other attributes.

Attribute-generalization:

* If there is a large set of distinct values for A, and there exists a set of generalization operators on A, then select an operator and generalize A.

Attribute-threshold control:

* Typical 2-8, specified/default.

Generalized relation threshold control (10-30):

* To control the final relation/rule size.

## Algorithm for Attribute Oriented Induction

### InitialRel:

* It is nothing but query processing of task-relevant data and deriving the initial relation.

PreGen:

* It is based on the analysis of the number of distinct values in each attribute and to determine the generalization plan for each attribute: removal? or how high to generalize?

PrimeGen:

* It is based on the PreGen plan and performing the generalization to the right level to derive a “prime generalized relation” and also accumulating the counts.

### Presentation:

* User interaction: (1) adjust levels by drilling, (2) pivoting, (3) mapping into rules, cross tabs, visualization presentations.

Example

Let's say there is a University database that is to be characterized, for that its corresponding DMQL will be

**use**University\_DB  
**mine characteristics as** “Science\_Students”  
**in relevance to** name, gender, major, birth\_place, birth\_date, residence, phone\_no, GPA  
**from** student

Its corresponding SQL statement can be:

**Select**name, gender, major, birth\_place, birth\_date, residence, phone\_no, GPA  
**from**student  
**where**status in {“Msc”, “MBA”, “Ph.D.” }  
 Now for this database let's create a characterized view:   
 InitialRel:

* From this table, we are querying task-relevant data.
* From this table, we also removed a few attributes like name and phoneno, because they make no sense in concluding insights.

PreGen

* Now, we have generalized these results by removing a few attributes and retaining important attributes.

# Efficient Algorithms for Attribute-Oriented Induction:

Data mining or knowledge discovery in databases is the search for relationships and global patterns that exist but are hidden in large databases. Many different methods have been proposed and one of them is the attribute-oriented induction method. In this method, domain knowledge in the form of concept hierarchies helps to generalize the concepts of the attributes in the database relations. This approach has been generalized to the rule-based attribute-oriented induction. The time complexity of the original algorithms is given by O(N log N), where N is the number of relevant tuples in the database. In this paper, we make use of the static property of the database schema and the concept hierarchies to derive more efficient algorithms. Given that the concept hierarchies and the resulting knowledge are small in size compared to the database, the complexity of our algorithm is O(N). The amount of disk I/O is decreased by O(log N) times compared to the previous methods. We believe that this improvement in performance will give extra power to the attribute-oriented method.

Mining Frequent Patterns, Associations and Correlations: Basic Concepts, Frequent Itemset Mining Methods: Apriori method, generating Association Rules

• Item set:- set of items. Example- {computer, printer, MS office software} is 3- item set. { milk, bread} is 2-item set. similarly set of K items is called k-item set.

• Frequent patterns are patterns that appear frequently in a data set. Patterns may be itemsets, subsequences or substructures. Example: A set of items, such as Milk & Butter that appear together in a transaction data set. ( Also called Frequent Item set).

• Frequent item set mining leads to the discovery of associations and correlations among items in large transactional (or) relational data sets.

• This helps in many business decision- making processes like Catalog design, and customer shopping behavior analysis, etc.

• Market Basket Analysis: This is the example of frequent item set mining. This process analyzes customer buying habits by finding associations between different items that customer places in their shopping baskets.

• Retailers can use the result by placing the items that are frequently purchased together in proximity to further encourage the combined sale of such items. In our example(in the figure), Milk and bread is frequent, so it can be kept in proximity. Other example is, if customers who purchase computers also tend to buy printer at the same time, then placing the hardware display close to the printer may increase the sale of both the items.

Association rules: Let I= { I1 , I2 , I3,……….,Im } be an item set. D= { T1 , T2 , T3,……….,Tn } be a set of n transactions where each transaction Ti is non- empty item set such that T ⊑ I. [or] for each i, Ti ≠ Φ and Ti ⊑ I Let A and B are set of items. [ ex- A= { I1 , I3 , I7,I8 } and B= { I4 , I5 , I6 } ] An Association rule is an implication of the form A ⟹ B where A ⊂ I, B ⊂ I, A ≠ Φ and B ≠ Φ & A ⋂ B ≠ Φ The rule A ⟹ B holds in the transaction set D with Support s and Confidence c. Support: This is the percentage of transaction in D that contain A⋃B. Here A⋃B means every item in A and every item in B. Support is also written as P(A⋃B). It is also called Relative support. [ Note: (A⋃B) ≠ A or B]

• Therefore, Support (A ⟹ B) = P(A⋃B). Confidence: This is the percentage of transactions in D containing A that also contain B. It is also written as P(B/A). Confidence(A ⟹ B) = P(B/A). = support(A⋃B) support(A) = support count (A⋃B) support count (A) Support count or Frequency: Number of transactions that contain the item set. It is also called Absolute support.

• Any association rules that satisfy both a minimum support threshold(min\_sup) and minimum confidence threshold (min\_conf) are called strong association.

• We have seen in the previous slide that the confidence can easily be derived from the support counts. i.e. If support counts of A, B and A⋃B are found, then we can derive corresponding association rules A ⟹ B and B ⟹ A and check whether they are strong or not.

• Hence mining association rules can be viewed as a two step process: 1. Finding all frequent item sets and 2. Generate strong association rules from the frequent item sets. [Note: frequent item set are those item sets that satisfies the min\_sup]

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

* Maximal Frequent itemset: An itemset X is a maximal frequent itemset in a data set D if X is frequent and there exist no super-itemset Y such that X⊂Y and Y is frequent in D. Example: Let T1 = (a1 ,a2 ,a3 ,a4 ,a5 ) and T2= (a1 ,a2 ,a3 ) and minimum support count threshold min\_sup=1 Therefore, Set of closed frequent itemset C= { {a1 ,a2 ,a3 }= 2; {a1 ,a2 ,a3 ,a4 ,a5 }=1}. and Set of maximal frequent itemset M= {{a1 ,a2 ,a3 ,a4 ,a5 }=1}. Apriori algorithm: (For finding frequent itemsets) It is an iterative approach where k-itemsets are used to explore (k+1) itemsets.
* Steps: [1] The set of frequent 1-itemset is found by scanning the data base and selecting those whose support count satisfy the minimum support. And denote this set as L1.
* [2] L1 is used to find set of frequent 2-itemset say L2 .
* [3] Further L2 is used to find L3 and so on until no more frequent k-itemset can be found. [Note: the finding of each Lk requires one full scan of the database.] Finding Lk (k>=2) : [1] Join step: { Assumption: 1. itemsets are sorted in lexicographic order. 2. l i [j] means j th item in l i . } The join (Lk-1 ⋈ Lk-1 ) (say it Ck ) is performed where members of Lk-1 are joinable if their first (k-2) items are in common. i.e. Members l1 and l2 of Lk-1 are joined if (l1 [1] = l2 [1] ^ l1 [2]= l2 [2] ^ l1 [3]= l2 [3] ^ ……………. ^ l1 [k-2]= l2 [k-2] ^ l1 [k-1] < l2 [k-1] ) [condition l1 [k-1] < l2 [k-1] ensures no duplicity] Therefore, resulting itemset formed by joining l1 and l2 is {l1 [1], l1 [2], l1 [3], l1 [k-2], l2 [k-1]} Example: Let L2 = [{I1 , I2 }, {I1 , I3 }, {I1 , I5 }] then, L2 ⋈ L2 (i.e. C3 ) = [{I1 , I2 , I3 }, {I1 , I2 , I5 },{I1 , I3 , I5 }] [2] prune step: The support count of each itemset in Ck is calculated and determine Lk by putting all those itemsets which satisfy the min\_sup in Ck . [Note: To determine the support count of each candidate in Ck a complete database scan is needed. Therefore to reduce the size of Ck the Apriori property is used. Apriori property: if an itemset I does not satisfy the minimum support threshold then (I ⋃ A) also will not satisfy the min\_sup.]
* Therefore if any (k-1) subset of a candidate k-itemset is not in Lk-1 , then the candidate can’t be frequent (i.e. does not satisfy min\_sup) hence can be removed from Ck . Example: Consider the following dataset and for this we have to find frequent itemsets and also have to generate association rules for them Let min\_sup = 2 Transactional Dataset D Step 1: create a table C1 that contain support count of each item present in the dataset D.
* TID List of items\_IDs T1 I1,I2,I5 T2 I2,I4 T3 I2,I3 T4 I1,I2,I4 T5 I1,I3 T6 I2,I3 T7 I1,I3 T8 I1,I2,I3,I5 T9 I1,I2,I3 C1 L1 Step 2: Generate C2 candidates from L1 (join step), and scan D for count of each candidate. C2 L2 Itemset Support count {I1} 6 {I2} 7 {I3} 6 {I4} 2 {I5} 2 Now, Compare candidate support count with minimum support count. This gives itemset L1. Itemset Support count {I1,I2} 4 {I1,I3} 4 {I1,I4} 1 {I1,I5} 2 {I2,I3} 4 {I2,I4} 2 {I2,I5} 2 {I3,I4} 0 {I3,I5} 1 {I4,I5} 0 Itemset Support count {I1,I2} 4 {I1,I3} 4 {I1,I5} 2 {I2,I3} 4 {I2,I4} 2 {I2,I5} 2 Compare candidate support count with minimum support count. This gives itemset L2.
* Step 3: Generate candidate set C3 using L2 (join step). And scan D for count of each candidate. L2 ⋈ L2 Therefore, C3 L3 {I1,I2,I3} {I1,I2,I5} {I1,I3,I5} {I2,I3,I4} {I2,I4,I5} {I2,I3,I5} But, using Apriori property we can remove {I1, I3, I5},{I2, I3, I4},{I2, I4, I5} and {I2, I3, I5} because every subsets of these sets are not frequent. Example- for itemset {I1,I3,I5} subset {I3,I5} is not frequent. And for {I2, I3, I4} subset {I3, I4} is not frequent. Itemset Support count {I1,I2,I3} 2 {I1,I2,I5} 2 Compare candidate support count with minimum support count. This gives itemset L3.

• Step 4: Generate candidate set C4 using L3 (join step). And scan D for count each candidate. L3 ⋈ L3 Therefore C4 = Φ Because the subset {I1, I3, I5} of itemset {I1, I2, I3, I5} is not frequent so there is no itemset in C4. Hence algorithm terminated . we have discovered all the frequent item-sets. In next lecture we will see the generation of strong association rules and pseudocode for Apriori algorithm.

Improving the Efficiency of Apriori:

**Techniques to improve the efficiency of Apriori algorithm**

* Hash based technique
* Transaction Reduction
* Portioning
* Sampling
* Dynamic item counting

### Apriori Algorithm – ****Frequent Pattern Algorithms****

Apriori algorithm was the first algorithm that was proposed for frequent itemset mining. It was later improved by R Agarwal and R Srikant and came to be known as Apriori. This algorithm uses two steps “join” and “prune” to reduce the search space. It is an iterative approach to discover the most frequent item sets.

**Apriori says:**

The probability that item I is not frequent is if:

* P(I) < minimum support threshold, then I is not frequent.
* P (I+A) < minimum support threshold, then I+A is not frequent, where A also belongs to itemset.
* If an itemset set has value less than minimum support then all of its supersets will also fall below min support, and thus can be ignored. This property is called the Antimonotone property.

**The steps followed in the Apriori Algorithm of data mining are:**

1. **Join Step**: This step generates (K+1) itemset from K-itemsets by joining each item with itself.
2. **Prune Step**: This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.

# Frequent Pattern (FP) Growth Algorithm In Data Mining:

this [**Data Mining Tutorial Series**](https://www.softwaretestinghelp.com/data-mining/), we had a look at the [**Decision Tree Algorithm**](https://www.softwaretestinghelp.com/decision-tree-algorithm-examples-data-mining/) in our previous tutorial.

There are several methods for Data Mining such as association, correlation, classification & clustering.

This tutorial primarily focuses on mining using association rules. By association rules, we identify the set of items or attributes that occur together in a table.

## What Is An Itemset?

A set of items together is called an itemset. If any itemset has k-items it is called a k-itemset. An itemset consists of two or more items. An itemset that occurs frequently is called a frequent itemset. **Thus frequent itemset mining is a data mining technique to identify the items that often occur together.**

**For Example**, Bread and butter, Laptop and Antivirus software, etc.

### What Is A Frequent Itemset?

A set of items is called frequent if it satisfies a minimum threshold value for support and confidence. Support shows transactions with items purchased together in a single transaction. Confidence shows transactions where the items are purchased one after the other.

For frequent itemset mining method, we consider only those transactions which meet minimum threshold support and confidence requirements. Insights from these mining algorithms offer a lot of benefits, cost-cutting and improved competitive advantage.

There is a tradeoff time taken to mine data and the volume of data for frequent mining. The frequent mining algorithm is an efficient algorithm to mine the hidden patterns of itemsets within a short time and less memory consumption.

### Frequent Pattern Mining (FPM)

The frequent pattern mining algorithm is one of the most important techniques of data mining to discover relationships between different items in a dataset. These relationships are represented in the form of association rules. It helps to find the irregularities in data.

FPM has many applications in the field of data analysis, software bugs, cross-marketing, sale campaign analysis, market basket analysis, etc.

Frequent itemsets discovered through Apriori have many applications in data mining tasks. Tasks such as finding interesting patterns in the database, finding out sequence and Mining of association rules is the most important of them.

Association rules apply to supermarket transaction data, that is, to examine the customer behavior in terms of the purchased products. Association rules describe how often the items are purchased together.

### Association Rules

**Association Rule Mining is defined as:**

**“Let I= { …} be a set of ‘n’ binary attributes called items. Let D= { ….} be set of transaction called database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of form X->Y where X, Y? I and X?Y=?. The set of items X and Y are called antecedent and consequent of the rule respectively.”**

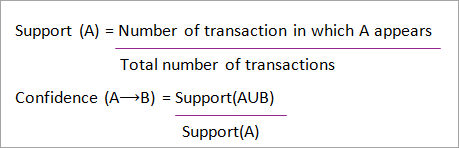
Learning of Association rules is used to find relationships between attributes in large databases. An association rule, A=> B, will be of the form” for a set of transactions, some value of itemset A determines the values of itemset B under the condition in which minimum support and confidence are met”.

**Support and Confidence can be represented by the following example:**

Bread=> butter [support=2%, confidence-60%]

The above statement is an example of an association rule. This means that there is a 2% transaction that bought bread and butter together and there are 60% of customers who bought bread as well as butter.

**Support and Confidence for Itemset A and B are represented by formulas:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/09/Support-and-Confidence-for-Itemset-A-and-B.png)

**Association rule mining consists of 2 steps:**

1. Find all the frequent itemsets.
2. Generate association rules from the above frequent itemsets.

## Why Frequent Itemset Mining?

Frequent itemset or pattern mining is broadly used because of its wide applications in mining association rules, correlations and graph patterns constraint that is based on frequent patterns, sequential patterns, and many other data mining tasks.

### Apriori Algorithm – ****Frequent Pattern Algorithms****

Apriori algorithm was the first algorithm that was proposed for frequent itemset mining. It was later improved by R Agarwal and R Srikant and came to be known as Apriori. This algorithm uses two steps “join” and “prune” to reduce the search space. It is an iterative approach to discover the most frequent itemsets.

**Apriori says:**

The probability that item I is not frequent is if:

* P(I) < minimum support threshold, then I is not frequent.
* P (I+A) < minimum support threshold, then I+A is not frequent, where A also belongs to itemset.
* If an itemset set has value less than minimum support then all of its supersets will also fall below min support, and thus can be ignored. This property is called the Antimonotone property.

**The steps followed in the Apriori Algorithm of data mining are:**

1. **Join Step**: This step generates (K+1) itemset from K-itemsets by joining each item with itself.
2. **Prune Step**: This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.

#### Steps In Apriori

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database. This data mining technique follows the join and the prune steps iteratively until the most frequent itemset is achieved. A minimum support threshold is given in the problem or it is assumed by the user.

**#1)** In the first iteration of the algorithm, each item is taken as a 1-itemsets candidate. The algorithm will count the occurrences of each item.

**#2)** Let there be some minimum support, min\_sup( eg 2). The set of 1 – itemsets whose occurrence is satisfying the min sup are determined. Only those candidates which count more than or equal to min\_sup, are taken ahead for the next iteration and the others are pruned.

**#3)** Next, 2-itemset frequent items with min\_sup are discovered. For this in the join step, the 2-itemset is generated by forming a group of 2 by combining items with itself.

**#4)** The 2-itemset candidates are pruned using min-sup threshold value. Now the table will have 2 –itemsets with min-sup only.

**#5)** The next iteration will form 3 –itemsets using join and prune step. This iteration will follow antimonotone property where the subsets of 3-itemsets, that is the 2 –itemset subsets of each group fall in min\_sup. If all 2-itemset subsets are frequent then the superset will be frequent otherwise it is pruned.

**#6)** Next step will follow making 4-itemset by joining 3-itemset with itself and pruning if its subset does not meet the min\_sup criteria. The algorithm is stopped when the most frequent itemset is achieved.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/09/AprioriSteps.png)

[image [source](https://www.hackerearth.com/)]

**Example of Apriori: Support threshold=50%, Confidence= 60%**

**TABLE-1**

| **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 |

**Solution:**

Support threshold=50% => 0.5\*6= 3 => min\_sup=3

**1. Count Of Each Item**

**TABLE-2**

| **Item** | **Count** |
| --- | --- |
| I1 | 4 |
| I2 | 5 |
| I3 | 4 |
| I4 | 4 |
| I5 | 2 |

**2.** **Prune Step:** **TABLE -2** shows that I5 item does not meet min\_sup=3, thus it is deleted, only I1, I2, I3, I4 meet min\_sup count.

**TABLE-3**

| **Item** | **Count** |
| --- | --- |
| I1 | 4 |
| I2 | 5 |
| I3 | 4 |
| I4 | 4 |

**3.** **Join Step:** Form 2-itemset. From **TABLE-1**find out the occurrences of 2-itemset.

**TABLE-4**

| **Item** | **Count** |
| --- | --- |
| I1,I2 | 4 |
| I1,I3 | 3 |
| I1,I4 | 2 |
| I2,I3 | 4 |
| I2,I4 | 3 |
| I3,I4 | 2 |

**4.** **Prune Step:** **TABLE -4**shows that item set {I1, I4} and {I3, I4} does not meet min\_sup, thus it is deleted.

**TABLE-5**

| **Item** | **Count** |
| --- | --- |
| I1,I2 | 4 |
| I1,I3 | 3 |
| I2,I3 | 4 |
| I2,I4 | 3 |

**5.** **Join and Prune Step:** Form 3-itemset. From the **TABLE- 1** find out occurrences of 3-itemset. From **TABLE-5**, find out the 2-itemset subsets which support min\_sup.

We can see for itemset {I1, I2, I3} subsets, {I1, I2}, {I1, I3}, {I2, I3} are occurring in **TABLE-5** thus {I1, I2, I3} is frequent.

We can see for itemset {I1, I2, I4} subsets, {I1, I2}, {I1, I4}, {I2, I4}, {I1, I4} is not frequent, as it is not occurring in **TABLE-5** thus {I1, I2, I4} is not frequent, hence it is deleted.

**TABLE-6**

| **Item** |
| --- |
| I1,I2,I3 |
| I1,I2,I4 |
| I1,I3,I4 |
| I2,I3,I4 |

**Only {I1, I2, I3} is frequent**.

**6. Generate Association Rules:** From the frequent itemset discovered above the association could be:

{I1, I2} => {I3}

Confidence = support {I1, I2, I3} / support {I1, I2} = (3/ 4)\* 100 = 75%

{I1, I3} => {I2}

Confidence = support {I1, I2, I3} / support {I1, I3} = (3/ 3)\* 100 = 100%

{I2, I3} => {I1}

Confidence = support {I1, I2, I3} / support {I2, I3} = (3/ 4)\* 100 = 75%

{I1} => {I2, I3}

Confidence = support {I1, I2, I3} / support {I1} = (3/ 4)\* 100 = 75%

{I2} => {I1, I3}

Confidence = support {I1, I2, I3} / support {I2 = (3/ 5)\* 100 = 60%

{I3} => {I1, I2}

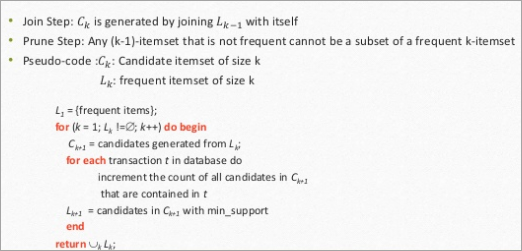
Confidence = support {I1, I2, I3} / support {I3} = (3/ 4)\* 100 = 75%

This shows that all the above association rules are strong if minimum confidence threshold is 60%.

**The Apriori Algorithm: Pseudo Code**

C: Candidate item set of size k

L: Frequent itemset of size k

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/09/Psudocode.png)

#### Advantages

1. Easy to understand algorithm
2. Join and Prune steps are easy to implement on large itemsets in large databases

#### Disadvantages

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

## Methods To Improve Apriori Efficiency

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

1. **Hash-Based Technique:** This method uses a hash-based structure called a hash table for generating the k-itemsets and its corresponding count. It uses a hash function for generating the table.
2. **Transaction Reduction:** This method reduces the number of transactions scanning in iterations. The transactions which do not contain frequent items are marked or removed.
3. **Partitioning:** This method requires only two database scans to mine the frequent itemsets. It says that for any itemset to be potentially frequent in the database, it should be frequent in at least one of the partitions of the database.
4. **Sampling:** This method picks a random sample S from Database D and then searches for frequent itemset in S. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min\_sup.
5. **Dynamic Itemset Counting:** This technique can add new candidate itemsets at any marked start point of the database during the scanning of the database.

## Applications Of Apriori Algorithm

**Some fields where Apriori is used:**

1. **In Education Field:** Extracting association rules in data mining of admitted students through characteristics and specialties.
2. **In the Medical field:** For example Analysis of the patient’s database.
3. **In Forestry:** Analysis of probability and intensity of forest fire with the forest fire data.
4. Apriori is used by many companies like Amazon in the **Recommender System** and by Google for the auto-complete feature.

## Conclusion

Apriori algorithm is an efficient algorithm that scans the database only once.

It reduces the size of the itemsets in the database considerably providing a good performance. Thus, data mining helps consumers and industries better in the decision-making process.

Mining Frequent Itemsets using vertical data format:

**Introduction:** Both the Apriori and FP-growth methods mine frequent patterns from a set of transactions in TID-itemset format (that is, {TID :itemsetg), where TID is a transaction-id and itemset is the set of items bought in transaction TID. This data format is known as horizontal data format. Alternatively, data can also be presented in item-TID set format (that is, fitem : TID setg), where item is an item name, and TID set is the set of transaction identifiers containing the item.

This format is known as vertical data format. In this section, we look at how frequent itemsets can also be mined efficiently using vertical data format, which is the essence of the ECLAT (Equivalence CLASS Transformation) algorithm developed by Zaki [Zak00].

**Algorithm**: FP growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

**Input:**

* D, a transaction database;
* min sup, the minimum support count threshold.

**Output:** The complete set of frequent patterns.

**Method:**

1. The FP-tree is constructed in the following steps:

(a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the list of frequent items.

(b) Create the root of an FP-tree, and label it as “null.” For each transaction Trans in D do the following. Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [pjP], where p is the first element and P is the remaining list. Call insert tree([pjP], T), which is performed as follows. If T has a child N such that N.item-name=p.item-name, then increment N’s count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree(P, N) recursively.

2. The FP-tree is mined by calling FP growth(FP tree, null), which is implemented as follows.

procedure FP growth(Tree,**α**)

(1) if Tree contains a single path P then

(2) for each combination (denoted as β) of the nodes in the path P

(3) generate pattern β∪**α** with support count = minimum support count o f nodes in β;

(4) else for each ai in the header of Tree {

(5) generate pattern β = ai∪ a with support count = ai:support\_count;

(6) construct β’s conditional pattern base and then β’s conditional FP tree Treeb;

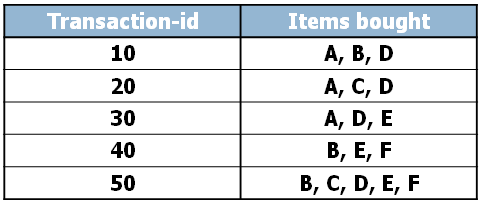
(7) if Treeβ ≠0 then

(8) call FP growth(Treeβ, β); g

Mining Closed and Max Patterns:

# lose and Max Patterns

Compress frequent patterns using closed patterns and max patterns



When we have many items, the total number of itemsets grow exponentially to the number of items in the database: |itemsets| = 2n. This immediately causes both space and time complexity problems in finding frequent patterns.

For example, the table above contains five transaction records and six items. This small database can have 26 = 64 possible itemsets. To deal with this problem, we now define a series of sets.

## Sub-pattern (proper subset)

To find efficient algorithms for mining frequent patterns, we will start with two basic concepts of patterns. A pattern is an itemset, therefore the order of the items are not important. What will be useful is the properties of subsets of supersets of frequent patterns.

Let us start with subsets. We call any proper subset of a pattern, a sub-pattern. Here is an example,

* If a pattern X contains, 3 items: X={a, b, c}, any proper subsets, such as sets containing a, or b, or c, or ab, or ac, or bc, are all sub patterns.

Now, suppose we have a pattern with 100 items. How many sub-patterns can we find for this pattern?

* A lot, it is more than 10 to the power of 30.

That means if this pattern is a frequent pattern, there are 10 to the power of 30 frequent patterns.

## Super-pattern (proper superset)

Another concept is of course super-pattern. A super pattern of a pattern is a proper super set of the pattern.

Here are some examples of the pattern containing the 3 items: a, b, c

* abcd, abce, abcde, …

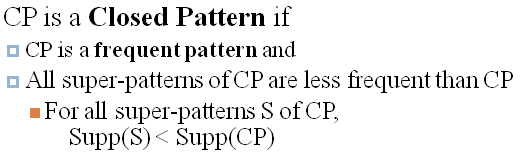
## Close Pattern

Now, we are ready for two important concepts of frequent patterns. Let’s start with closed patterns.

A **closed pattern** is

1. a frequent pattern. So it meets the minimum support criteria.
2. In addition to that, all super-patterns of a closed pattern are less frequent than the closed pattern.

Formally, this is defined as:



Let’s see some examples.

1. Suppose, the minimum support count is 2.
2. For the first example, suppose there are a total of 3 items: a, b, c.
3. Suppose a pattern ab has support count of 2 and a pattern abc has support count of 2.
4. Is the pattern ab is a closed pattern?

Pattern ab is a frequent pattern, but it has a super-pattern that is NOT less frequent than ab.

For the second example,

1. suppose there are a total of 3 items: x, y, z.
2. suppose a pattern xy has support count of 3 and a pattern xyz has support count of 2.
3. Is the pattern xy is a closed pattern?

Pattern xy is a frequent pattern and also the only super-pattern xyz is less frequent than xy.

Therefore, xy is a closed pattern.

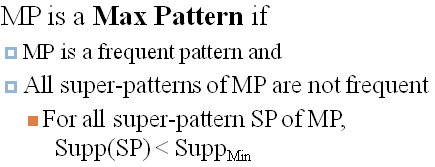
## Max Pattern

Next, let’s define max pattern.

A **max pattern** is

1. a frequent pattern. So it also meets the minimum support criteria like closed pattern
2. In addition, but unlike closed pattern, all super-patterns of a max pattern are NOT frequent patterns.

This is formally written as



Let’s see some examples as well.

1. Suppose, the minimum support count is 2.
2. Like before, for the first example, suppose there are a total of 3 items: a, b, c.
3. Suppose a pattern ab has support count of 3 and a pattern abc has support count of 2.
4. Is the pattern ab is a max pattern?

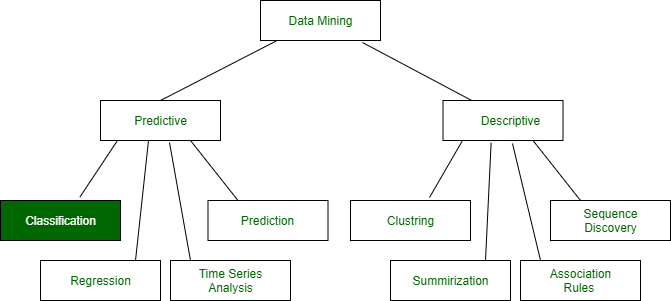
Pattern ab is a frequent pattern, but it has a super-pattern that is a frequent pattern as well. So, pattern ab is NOT a max pattern.

For the second example,

1. suppose there are a total of 3 items: x, y, z.
2. Suppose a pattern xy has support count of 3 and a pattern xyz has support count of 1.
3. Is the pattern xy is a max pattern?Pattern xy is a frequent pattern and also the only super-pattern xyz is NOT a frequent pattern. Therefore, xy is a max pattern.

**UNIT IV**

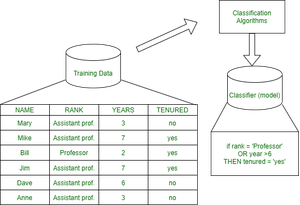
**Data Mining**: Data mining in general terms means mining or digging deep into data that is in different forms to gain patterns, and to gain knowledge on that pattern. In the process of data mining, large data sets are first sorted, then patterns are identified and relationships are established to perform data analysis and solve problems.



Attention reader! Don’t stop learning now. Get hold of all the important Machine Learning Concepts with the [**Machine Learning Foundation Course**](https://practice.geeksforgeeks.org/courses/machine-learning?utm_source=geeksforgeeks&utm_medium=article&utm_campaign=GFG_Article_Bottom_Python_ML) at a student-friendly price and become industry ready.

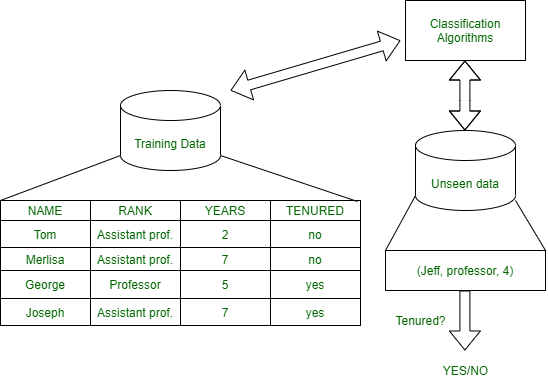
**Classification**: It is a data analysis task, i.e. the process of finding a model that describes and distinguishes data classes and concepts. Classification is the problem of identifying to which of a set of categories (subpopulations), a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.   
**Example**: Before starting any project, we need to check its feasibility. In this case, a classifier is required to predict class labels such as ‘Safe’ and ‘Risky’ for adopting the Project and to further approve it. It is a two-step process such as : 

1. **Learning Step (Training Phase)**: Construction of Classification Model   
   Different Algorithms are used to build a classifier by making the model learn using the training set available. The model has to be trained for the prediction of accurate results.



Test data are used to estimate the accuracy of the classification rule

1. **Classification Step**: Model used to predict class labels and testing the constructed model on test data and hence estimate the accuracy of the classification rules.



Test data are used to estimate the accuracy of the classification rule

**Training and Testing:**   
Suppose there is a person who is sitting under a fan and the fan starts falling on him, he should get aside in order not to get hurt. So, this is his training part to move away. While Testing if the person sees any heavy object coming towards him or falling on him and moves aside then the system is tested positively and if the person does not move aside then the system is negatively tested.   
Same is the case with the data, it should be trained in order to get the accurate and best results.

There are certain data types associated with data mining that actually tells us the format of the file (whether it is in text format or in numerical format).   
Attributes – Represents different features of an object. Different types of attributes are:

1. **Binary**: Possesses only two values i.e. True or False   
   Example: Suppose there is a survey evaluating some products. We need to check whether it’s useful or not. So, the Customer has to answer it in Yes or No.   
   Product usefulness: Yes / No
   * **Symmetric**: Both values are equally important in all aspects
   * **Asymmetric**: When both the values may not be important.
2. **Nominal**: When more than two outcomes are possible. It is in Alphabet form rather than being in Integer form.   
   **Example**: One needs to choose some material but of different colors. So, the color might be Yellow, Green, Black, Red.   
   Different Colors: Red, Green, Black, Yellow
   * **Ordinal**: Values that must have some meaningful order.   
     Example: Suppose there are grade sheets of few students which might contain different grades as per their performance such as A, B, C, D   
     Grades: A, B, C, D
   * **Continuous**: May have an infinite number of values, it is in float type   
     Example: Measuring the weight of few Students in a sequence or orderly manner i.e. 50, 51, 52, 53   
     Weight: 50, 51, 52, 53
   * **Discrete**: Finite number of values.   
     Example: Marks of a Student in a few subjects: 65, 70, 75, 80, 90   
     Marks: 65, 70, 75, 80, 90

**Syntax:**

* Mathematical Notation: Classification is based on building a function taking input feature vector “X” and predicting its outcome “Y” (Qualitative response taking values in set C)
* Here Classifier (or model) is used which is a Supervised function, can be designed manually based on expert’s knowledge. It has been constructed to predict class labels (Example: Label – “Yes” or “No” for the approval of some event).

Classifiers can be categorized into two major types: 

1. **Discriminative**: It is a very basic classifier and determines just one class for each row of data. It tries to model just by depending on the observed data, depends heavily on the quality of data rather than on distributions.   
   **Example**: Logistic Regression   
   Acceptance of a student at a University (Test and Grades need to be considered)   
   Suppose there are few students and the Result of them are as follows :
2. **Generative**: It models the distribution of individual classes and tries to learn the model that generates the data behind the scenes by estimating assumptions and distributions of the model. Used to predict the unseen data.   
   **Example**: Naive Bayes Classifier   
   Detecting Spam emails by looking at the previous data. Suppose 100 emails and that too divided in 1:4 i.e. Class A: 25%(Spam emails) and Class B: 75%(Non-Spam emails). Now if a user wants to check that if an email contains the word cheap, then that may be termed as Spam.   
   It seems to be that in Class A(i.e. in 25% of data), 20 out of 25 emails are spam and rest not.   
   And in Class B(i.e. in 75% of data), 70 out of 75 emails are not spam and rest are spam.   
   So, if the email contains the word cheap, what is the probability of it being spam ?? (= 80%)

**Classifiers Of Machine Learning:**

1. Decision Trees
2. Bayesian Classifiers
3. Neural Networks
4. K-Nearest Neighbour
5. Support Vector Machines
6. Linear Regression
7. Logistic Regression

**Associated Tools and Languages:** Used to mine/ extract useful information from raw data.

* **Main Languages used**: R, SAS, Python, SQL
* **Major Tools used**: RapidMiner, Orange, KNIME, Spark, Weka
* **Libraries used**: Jupyter, NumPy, Matplotlib, Pandas, ScikitLearn, NLTK, TensorFlow, Seaborn, Basemap, etc.

**Real**–**Life Examples :**

* **Market Basket Analysis:**   
  It is a modeling technique that has been associated with frequent transactions of buying some combination of items.   
  **Example**: Amazon and many other Retailers use this technique. While viewing some products, certain suggestions for the commodities are shown that some people have bought in the past.
* **Weather Forecasting:**   
  Changing Patterns in weather conditions needs to be observed based on parameters such as temperature, humidity, wind direction. This keen observation also requires the use of previous records in order to predict it accurately.

**Advantages:**

* Mining Based Methods are cost-effective and efficient
* Helps in identifying criminal suspects
* Helps in predicting the risk of diseases
* Helps Banks and Financial Institutions to identify defaulters so that they may approve Cards, Loan, etc.

**Disadvantages:**  
Privacy: When the data is either are chances that a company may give some information about their customers to other vendors or use this information for their profit.   
Accuracy Problem: Selection of Accurate model must be there in order to get the best accuracy and result.

**APPLICATIONS:** 

* Marketing and Retailing
* Manufacturing
* Telecommunication Industry
* Intrusion Detection
* Education System
* Fraud Detection

**GIST OF DATA MINING :**

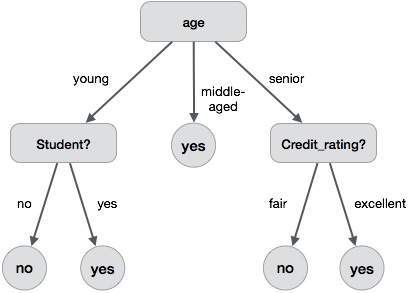
1. Choosing the correct classification method, like decision trees, Bayesian networks, or neural networks.
2. Need a sample of data, where all class values are known. Then the data will be divided into two parts, a training set, and a test set.

Now, the training set is given to a learning algorithm, which derives a classifier. Then the classifier is tested with the test set, where all class values are hidden.   
If the classifier classifies most cases in the test set correctly, it can be assumed that it works accurately also on the future data else it may be a wrong model chosen.

# Decision Tree Induction:

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy\_computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



The benefits of having a decision tree are as follows −

* It does not require any domain knowledge.
* It is easy to comprehend.
* The learning and classification steps of a decision tree are simple and fast.

## Decision Tree Induction Algorithm

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner.

Generating a decision tree form training tuples of data partition D

**Algorithm : Generate\_decision\_tree**

**Input:**

Data partition, D, which is a set of training tuples

and their associated class labels.

attribute\_list, the set of candidate attributes.

Attribute selection method, a procedure to determine the

splitting criterion that best partitions that the data

tuples into individual classes. This criterion includes a

splitting\_attribute and either a splitting point or splitting subset.

**Output:**

A Decision Tree

**Method**

create a node N;

if tuples in D are all of the same class, C then

return N as leaf node labeled with class C;

if attribute\_list is empty then

return N as leaf node with labeled

with majority class in D;|| majority voting

apply attribute\_selection\_method(D, attribute\_list)

to find the best splitting\_criterion;

label node N with splitting\_criterion;

if splitting\_attribute is discrete-valued and

multiway splits allowed then // no restricted to binary trees

attribute\_list = splitting attribute; // remove splitting attribute

for each outcome j of splitting criterion

// partition the tuples and grow subtrees for each partition

letDj be the set of data tuples in D satisfying outcome j; // a partition

ifDj is empty then

attach a leaf labeled with the majority

class in D to node N;

else

attach the node returned by Generate

decision tree(Dj, attribute list) to node N;

end for

return N;

## Tree Pruning

Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.

### Tree Pruning Approaches

There are two approaches to prune a tree −

* **Pre-pruning** − The tree is pruned by halting its construction early.
* **Post-pruning** - This approach removes a sub-tree from a fully grown tree.

## Cost Complexity

The cost complexity is measured by the following two parameters −

* Number of leaves in the tree, and
* Error rate of the tree.

Attribute Selection Measures:

• It is a heuristic approach to select the best splitting criterion that separates a given data partition, D, of class-labeled training tuples into individual classes.

• Splitting criterion is called the best when after splitting, each partition will be pure.

• A partition is called pure when all the tuples that fall into the partition belongs to the same class.

• Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split.

• First, a rank is provided for each attribute that describes the training tuples. And the attribute having the best score for the measure is chosen as the splitting attribute for the given tuples.

• If the splitting attribute is continuous-valued or if we are restricted to binary trees, then respectively either a split point or a splitting subset must also be determined as part of the splitting criterion. partition scenarios Examples 1. 2. 3. 1. A is discrete-valued: In this case, the outcomes of the test at node N correspond directly to the known values of A. A branch is created for each known value, aj , of A and labeled with that value (as in the figure). Partition Dj is the subset of class-labeled tuples in D having value aj of A. 2. A is continuous-valued: In this case, the test at node N has two possible outcomes, corresponding to the conditions A ≤ split point and A > split point, respectively, where split point is the split-point returned by Attribute selection method as part of the splitting criterion. 3. If A is discrete-valued and a binary tree must be produced, then the test is of the form A ∈SA , where SA is the splitting subset for A.

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 Classification Methods,

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 −

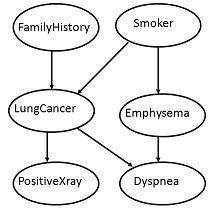
* Directed acyclic graph
* A set of conditional probability tables

## 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

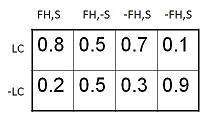
The following diagram shows a directed acyclic graph for six Boolean variables.



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 −



**Classification by Backpropagation**

Backpropagation: A **neural network** learning algorithm

Started by psychologists and neurobiologists to develop and test computational analogues of neurons

A neural network: A set of connected input/output units where each connection has a **weight**associated with it

During the learning phase, the **network learns by adjusting the weights** so as to be able to predict the correct class label of the input tuples

Also referred to as **connectionist learning** due to the connections between units

**Neural Network as a Classifier**

       Weakness

o   Long training time

 o   Require a number of parameters typically best determined empirically, e.g., the network topology or ``structure."

 o   Poor interpretability: Difficult to interpret the symbolic meaning behind the learnedweights and of ``hidden units" in the networkStrength

o   High tolerance to noisy data

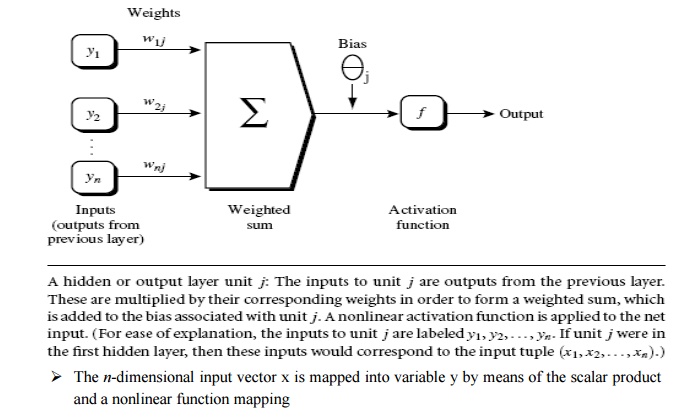
o   Ability to classify untrained patterns

o   Well-suited for continuous-valued inputs and outputs o Successful on a wide array of real-world data

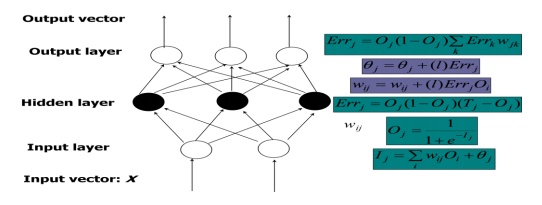
o   Algorithms are inherently parallel

o   Techniques have recently been developed for the extraction of rules from trained neural networks

**ANeuron(=aperceptron)**



**A Multi-Layer Feed-Forward Neural Network**



o   The inputs to the network correspond to the attributes measured for each training tuple

oInputs are fed simultaneously into the units making up the input layer

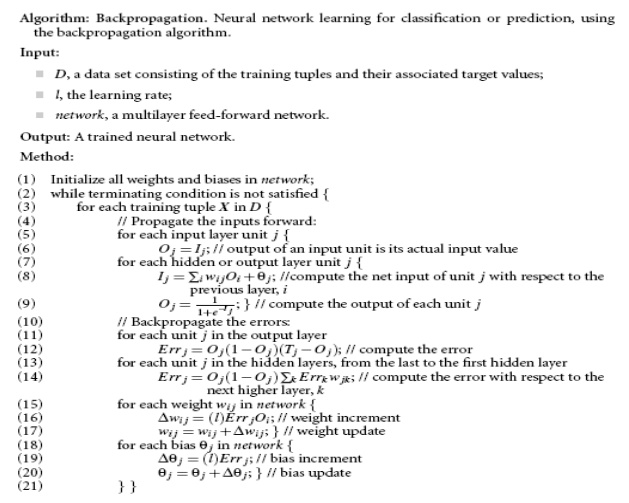
oThey are then weighted and fed simultaneously to a hidden layer

oThe number of hidden layers is arbitrary, although usually only one

oThe weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction

oThe network is feed-forward in that none of the weights cycles back to an input unit or to an output unit of a previous layer

o   From a statistical point of view, networks perform nonlinear regression: Given enough hidden units and enough training samples, they can closely approximate any function

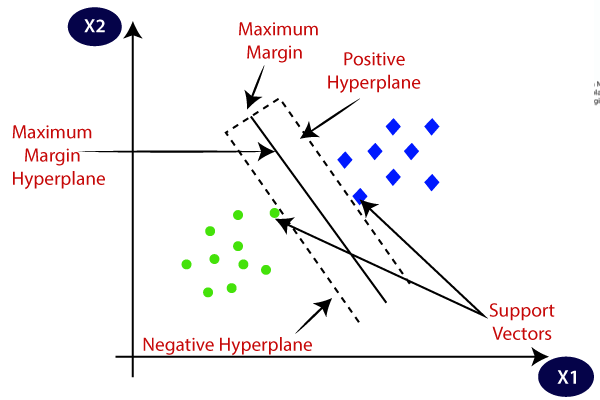


# Support Vector Machine Algorithm

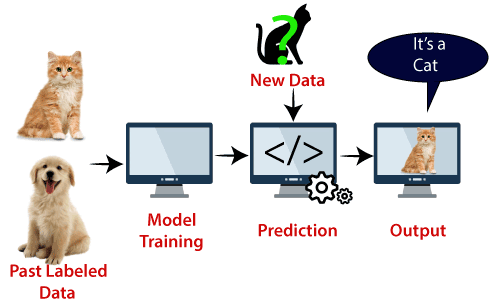
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



**Example:** SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

## Types of SVM

**SVM can be of two types:**

* **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
* **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

## Hyperplane and Support Vectors in the SVM algorithm:

**Hyperplane:** There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

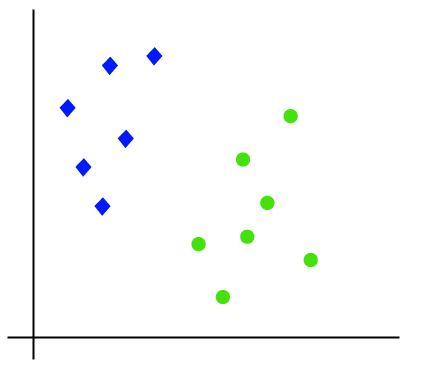
**Support Vectors:**

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

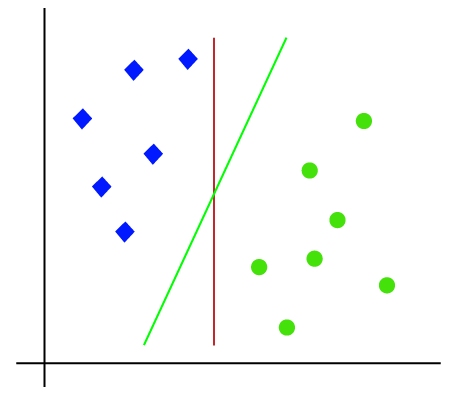
## How does SVM works?

**Linear SVM:**

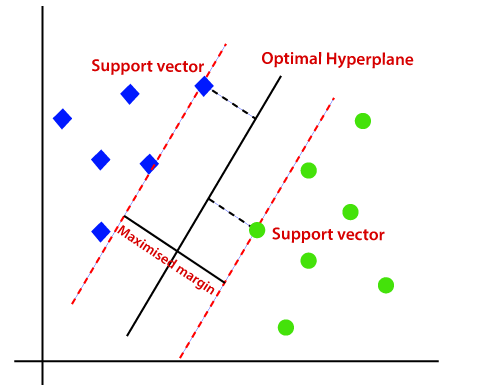
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

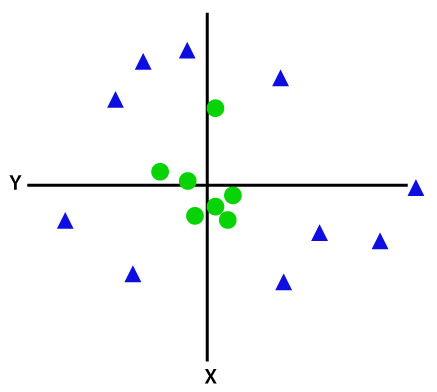


Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.



**Non-Linear SVM:**

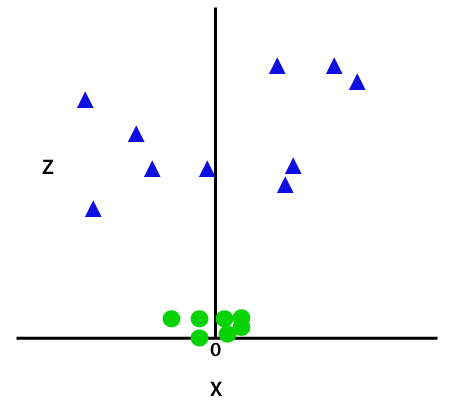
If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



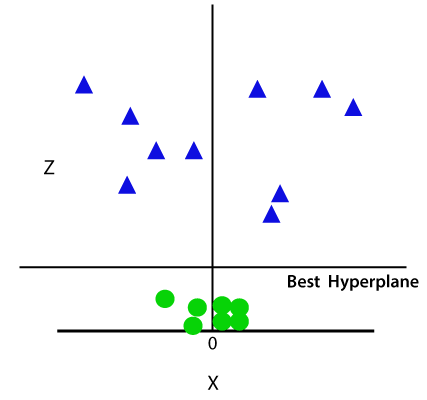
So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

z=x2 +y2

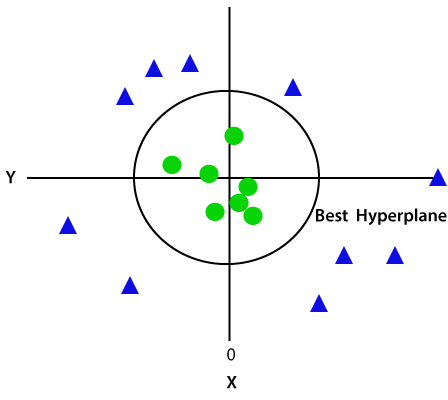
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Hence we get a circumference of radius 1 in case of non-linear data.

**Python Implementation of Support Vector Machine**

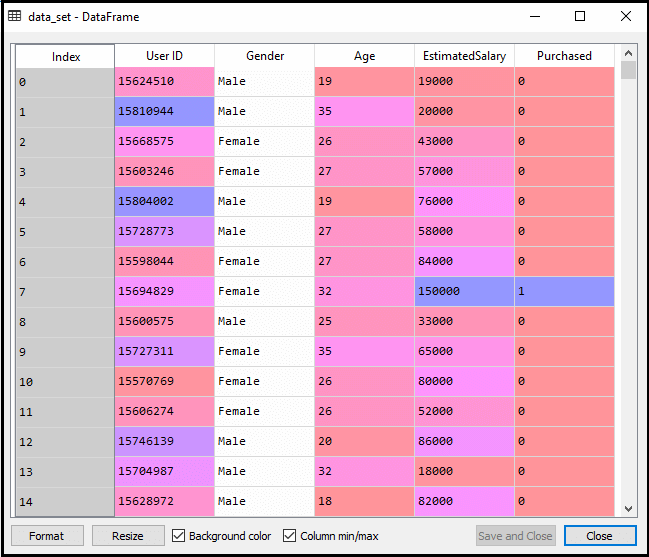
Now we will implement the SVM algorithm using Python. Here we will use the same dataset **user\_data**, which we have used in Logistic regression and KNN classification.

* **Data Pre-processing step**

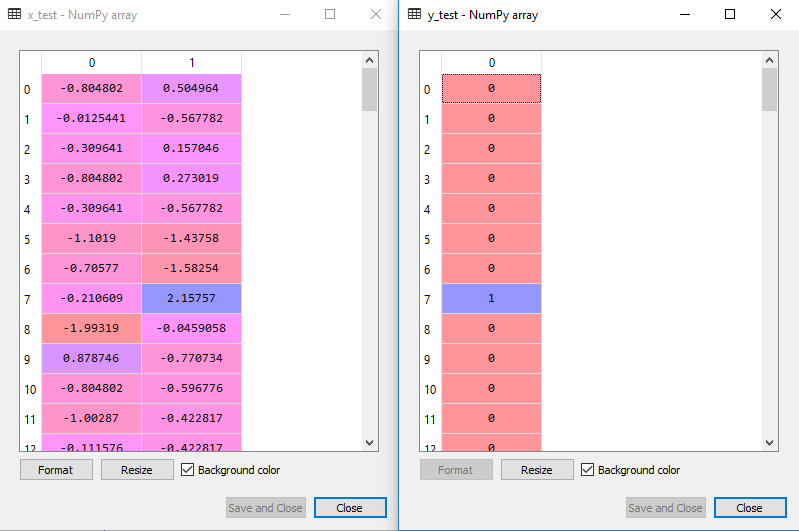
Till the Data pre-processing step, the code will remain the same. Below is the code:

1. #Data Pre-processing Step
2. # importing libraries
3. **import** numpy as nm
4. **import** matplotlib.pyplot as mtp
5. **import** pandas as pd
7. #importing datasets
8. data\_set= pd.read\_csv('user\_data.csv')
10. #Extracting Independent and dependent Variable
11. x= data\_set.iloc[:, [2,3]].values
12. y= data\_set.iloc[:, 4].values
14. # Splitting the dataset into training and test set.
15. from sklearn.model\_selection **import** train\_test\_split
16. x\_train, x\_test, y\_train, y\_test= train\_test\_split(x, y, test\_size= 0.25, random\_state=0)
17. #feature Scaling
18. from sklearn.preprocessing **import** StandardScaler
19. st\_x= StandardScaler()
20. x\_train= st\_x.fit\_transform(x\_train)
21. x\_test= st\_x.transform(x\_test)

After executing the above code, we will pre-process the data. The code will give the dataset as:



The scaled output for the test set will be:



**Fitting the SVM classifier to the training set:**

Now the training set will be fitted to the SVM classifier. To create the SVM classifier, we will import **SVC**class from **Sklearn.svm** library. Below is the code for it:

1. from sklearn.svm **import** SVC # "Support vector classifier"
2. classifier = SVC(kernel='linear', random\_state=0)
3. classifier.fit(x\_train, y\_train)

In the above code, we have used **kernel='linear'**, as here we are creating SVM for linearly separable data. However, we can change it for non-linear data. And then we fitted the classifier to the training dataset(x\_train, y\_train)

**Output:**

Out[8]:

SVC(C=1.0, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma='auto\_deprecated',

kernel='linear', max\_iter=-1, probability=False, random\_state=0,

shrinking=True, tol=0.001, verbose=False)

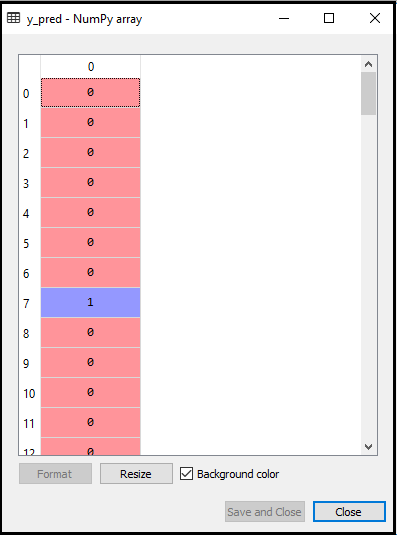
The model performance can be altered by changing the value of **C(Regularization factor), gamma, and kernel**.

* **Predicting the test set result:**  
  Now, we will predict the output for test set. For this, we will create a new vector y\_pred. Below is the code for it:

1. #Predicting the test set result
2. y\_pred= classifier.predict(x\_test)

After getting the y\_pred vector, we can compare the result of **y\_pred** and **y\_test** to check the difference between the actual value and predicted value.

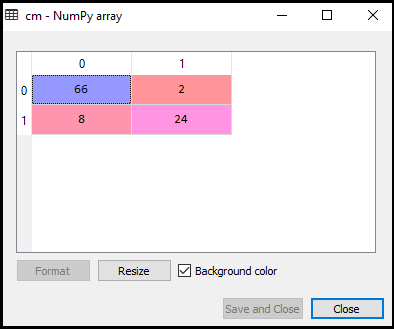
**Output:** Below is the output for the prediction of the test set:



* **Creating the confusion matrix:**  
  Now we will see the performance of the SVM classifier that how many incorrect predictions are there as compared to the Logistic regression classifier. To create the confusion matrix, we need to import the **confusion\_matrix** function of the sklearn library. After importing the function, we will call it using a new variable **cm**. The function takes two parameters, mainly **y\_true**( the actual values) and **y\_pred** (the targeted value return by the classifier). Below is the code for it:

1. #Creating the Confusion matrix
2. from sklearn.metrics **import** confusion\_matrix
3. cm= confusion\_matrix(y\_test, y\_pred)

**Output:**



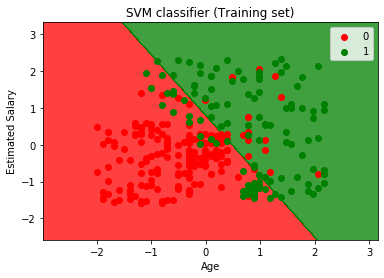
As we can see in the above output image, there are 66+24= 90 correct predictions and 8+2= 10 correct predictions. Therefore we can say that our SVM model improved as compared to the Logistic regression model.

* **Visualizing the training set result:**  
  Now we will visualize the training set result, below is the code for it:

1. from matplotlib.colors **import** ListedColormap
2. x\_set, y\_set = x\_train, y\_train
3. x1, x2 = nm.meshgrid(nm.arange(start = x\_set[:, 0].min() - 1, stop = x\_set[:, 0].max() + 1, step  =0.01),
4. nm.arange(start = x\_set[:, 1].min() - 1, stop = x\_set[:, 1].max() + 1, step = 0.01))
5. mtp.contourf(x1, x2, classifier.predict(nm.array([x1.ravel(), x2.ravel()]).T).reshape(x1.shape),
6. alpha = 0.75, cmap = ListedColormap(('red', 'green')))
7. mtp.xlim(x1.min(), x1.max())
8. mtp.ylim(x2.min(), x2.max())
9. **for** i, j in enumerate(nm.unique(y\_set)):
10. mtp.scatter(x\_set[y\_set == j, 0], x\_set[y\_set == j, 1],
11. c = ListedColormap(('red', 'green'))(i), label = j)
12. mtp.title('SVM classifier (Training set)')
13. mtp.xlabel('Age')
14. mtp.ylabel('Estimated Salary')
15. mtp.legend()
16. mtp.show()

**Output:**

By executing the above code, we will get the output as:



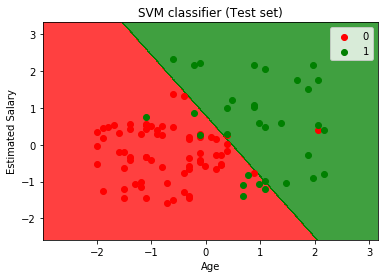
As we can see, the above output is appearing similar to the Logistic regression output. In the output, we got the straight line as hyperplane because we have **used a linear kernel in the classifier**. And we have also discussed above that for the 2d space, the hyperplane in SVM is a straight line.

* **Visualizing the test set result:**

1. #Visulaizing the test set result
2. from matplotlib.colors **import** ListedColormap
3. x\_set, y\_set = x\_test, y\_test
4. x1, x2 = nm.meshgrid(nm.arange(start = x\_set[:, 0].min() - 1, stop = x\_set[:, 0].max() + 1, step  =0.01),
5. nm.arange(start = x\_set[:, 1].min() - 1, stop = x\_set[:, 1].max() + 1, step = 0.01))
6. mtp.contourf(x1, x2, classifier.predict(nm.array([x1.ravel(), x2.ravel()]).T).reshape(x1.shape),
7. alpha = 0.75, cmap = ListedColormap(('red','green' )))
8. mtp.xlim(x1.min(), x1.max())
9. mtp.ylim(x2.min(), x2.max())
10. **for** i, j in enumerate(nm.unique(y\_set)):
11. mtp.scatter(x\_set[y\_set == j, 0], x\_set[y\_set == j, 1],
12. c = ListedColormap(('red', 'green'))(i), label = j)
13. mtp.title('SVM classifier (Test set)')
14. mtp.xlabel('Age')
15. mtp.ylabel('Estimated Salary')
16. mtp.legend()
17. mtp.show()

**Output:**

By executing the above code, we will get the output as:



As we can see in the above output image, the SVM classifier has divided the users into two regions (Purchased or Not purchased). Users who purchased the SUV are in the red region with the red scatter points. And users who did not purchase the SUV are in the green region with green scatter points. The hyperplane has divided the two classes into Purchased and not purchased variable.

## What is Cluster Analysis?

Cluster analysis is a statistical method used to group similar objects into respective categories. It can also be referred to as segmentation analysis, taxonomy analysis, or clustering.

The goal of performing a cluster analysis is to sort different objects or data points into groups in a manner that the degree of association between two objects is high if they belong to the same group, and low if they belong to different groups.

Cluster analysis differs from many other statistical methods due to the fact that it’s mostly used when researchers do not have an assumed principle or fact that they are using as the foundation of their research.

This analysis technique is typically performed during the exploratory phase of research, since unlike techniques such as [factor analysis](https://www.alchemer.com/analyzing-data/factor-analysis/), it doesn’t make any distinction between dependent and independent variables. Instead, cluster analysis is leveraged mostly to discover structures in data without providing an explanation or interpretation.

Put simply, cluster analysis discovers structures in data without explaining why those structures exist.

For example, when cluster analysis is performed as part of [market research](https://www.alchemer.com/market-research-survey-solutions/), specific groups can be identified within a population. The analysis of these groups can then determine how likely a population cluster is to purchase products or services. If these groups are defined clearly, a marketing team can then target varying cluster with tailored, targeted communication.

## Common Applications of Cluster Analysis

### Marketing

Marketers commonly use cluster analysis to develop market segments, which allow for better positioning of products and messaging.  company to better position itself, explore new markets, and development products that specific clusters find relevant and valuable.

### Insurance

Insurance companies often leverage cluster analysis if there are a high number of claims in a given region. This enables them to learn exactly what is driving this increase in claims.

### Geology

For cities on fault lines, geologists use cluster analysis to evaluate seismic risk and the potential weaknesses of earthquake-prone regions. By considering the results of this research, residents can do their best to prepare mitigate potential damage.

## Putting Clustering into Context

It’s easy to overthink cluster analysis, but our brains naturally cluster data on a regular basis in order to simplify the world around us. Whether we realize it or not, we deal with clustering in practically every aspect of our day-to-day lives.

For example, a group of friends sitting at the same table in a restaurant can be considered a cluster.

In grocery stores, goods of a similar nature are grouped together in order to make shopping more convenient and efficient.

This list of events during which we use clustering in our everyday lives could go on forever, but perhaps it makes more sense to consider a more classic, archetypal example.

In biology, humans belong to the following clusters: primates, mammals, amniotes, vertebrates, and animals. In this example, note that as we move down the chain of clusters, humans show less and less similarities to the other members of the group. Humans have more in common with primates than they do with other mammals, and more in common with mammals than they do with all animals in general.

## The Benefits of Cluster Analysis

Clustering allows researchers to identify and define patterns between data elements.

Revealing these patterns between data points helps to distinguish and outline structures which might not have been apparent before, but which give significant meaning to the data once they are discovered.

Once a clearly defined structure emerges from the dataset at hand, informed decision-making becomes much easier.

## The Different Types of Cluster Analysis

There are three primary methods used to perform cluster analysis:

### Hierarchical Cluster

This is the most common method of clustering. It creates a series of models with cluster solutions from 1 (all cases in one cluster) to n (each case is an individual cluster). This approach also works with variables instead of cases. Hierarchical clustering can group variables together in a manner similar to [factor analysis](https://www.alchemer.com/analyzing-data/factor-analysis/).

Finally, hierarchical cluster analysis can handle nominal, ordinal, and scale data. But, remember not to mix different levels of measurement into your study.

### K-Means Cluster

This method is used to quickly cluster large datasets. Here, researchers define the number of clusters prior to performing the actual study. This approach is useful when testing different models with a different assumed number of clusters.

### Two-Step Cluster

This method uses a cluster algorithm to identify groupings by performing pre-clustering first, and then performing hierarchical methods. Two-step clustering is best for handling larger datasets that would otherwise take too long a time to calculate with strictly hierarchical methods.

Essentially, two-step cluster analysis is a combination of hierarchical and k-means cluster analysis. It can handle both scale and ordinal data, and it automatically selects the number of clusters.

## What Does The Clustering Process Look Like?

### Step #1: Build and Distribute a Survey

Your survey should be designed to include multiple measures of propensity to purchase and the preferences for the product at hand. It should be distributed to your population of interest, and your sample size should be large enough to inform statistically-based decisions.

### Step #2: Analyze Response Data

It’s considered best practice to perform a factor analysis on your survey to minimize the factors being clustered. If after your factor analysis it’s concluded that a handful of questions are measuring the same thing, you should combine these questions prior to performing your cluster analysis.

After reducing your data by factoring, perform the cluster analysis and decide how many clusters seem appropriate, and record those cluster assignments. You’ll now be able to view the means of all of your factors across clusters.

### Step #3: Take Informed Action!

Comb through your data to identify differences in the means of factors, and name your clusters based on these differences. These differences between clusters are then able to inform your marketing, allowing you to target precise groups of customers with the right message, at the right time, in the right manner.

# Partitioning Method (K-Mean) in Data Mining

* **Last Updated :** 05 Feb, 2020

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

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

Attention reader! Don’t stop learning now. Get hold of all the important CS Theory concepts for SDE interviews with the [**CS Theory Course**](https://practice.geeksforgeeks.org/courses/SDE-theory?vC=1) at a student-friendly price and become industry ready.

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

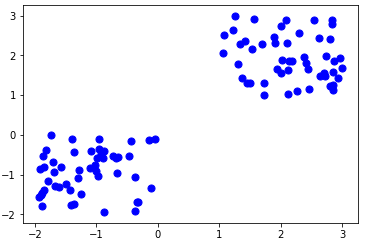
D: A dataset containing N number of objects

**Output:**

A dataset of K clusters

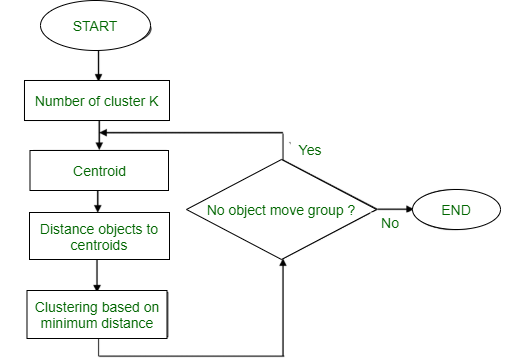
**Method:**

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 4 until no change occurs.



**Figure –** K-mean Clustering

**Flowchart:**



**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.

**Like**0

# Hierarchical Clustering in Data Mining

A **Hierarchical clustering** method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data points 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 cluster are break up (top-down view).

Attention reader! Don’t stop learning now. Get hold of all the important CS Theory concepts for SDE interviews with the [**CS Theory Course**](https://practice.geeksforgeeks.org/courses/SDE-theory?vC=1) at a student-friendly price and become industry ready.

The basic method to generate hierarchical clustering are:

**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 everydata set set is considered as individual entity or cluster. At every iteration, the clusters merge with different clusters until one cluster is formed.

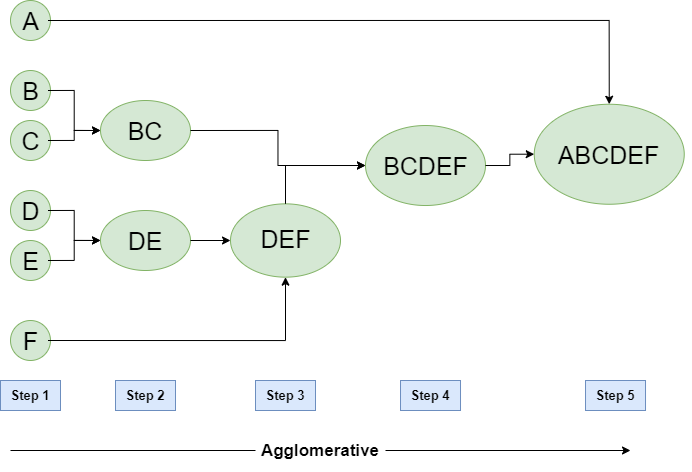
Algorithm for Agglomerative Hierarchical Clustering is:

* Calculate the similarity of one cluster with all the other clusters (calculate proximity matrix)
* Consider every data point as a individual cluster
* Merge the clusters which are highly similar or close to each other.
* Recalculate the proximity matrix for each cluster
* Repeat Step 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 are assumed.

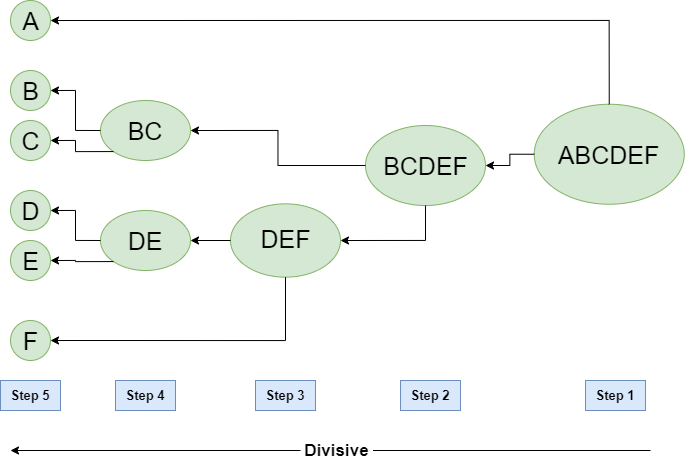
Let’s say we have six data points **A, B, C, D, E, F**.



**Figure –** Agglomerative Hierarchical clustering

* **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 with 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)].

**2. Divisive:**  
We can say that the Divisive Hierarchical clustering is precisely the **opposite** of the Agglomerative Hierarchical clustering. In Divisive Hierarchical clustering, we take into account all of the data points as a single cluster and in every iteration, we separate the data points from the clusters which aren’t comparable. In the end, we are left with N clusters.



**Figure –** Divisive Hierarchical clustering

Density based methods-DBSCAN andOPTICS:

## Density-Based Clustering

Density-Based Clustering method is one of the clustering methods based on density (local cluster criterion), such as density-connected points.

The basic ideas of density-based clustering involve a number of new definitions. We intuitively present these definitions and then follow up with an example.  
  
The neighborhood within a radius ε of a given object is called the ε-neighborhood of the object.  
  
If the ε-neighborhood of an object contains at least a minimum number, MinPts, of objects, then the object is called a core object.  
  
(Partitional Clustering - [read here](https://www.datamining365.com/2020/03/partitional-clustering-k-means.html))   
    
(Hierarchical Clustering - [read here](https://www.datamining365.com/2020/03/hierarchical-clustering.html))  
    
(Grid-based Clustering - [read here](https://www.datamining365.com/2020/04/grid-based-clustering.html))    
Density-reachable:

* A point p is density-reachable from a point q wrt. Eps, MinPts if there is a chain of points p1, …, pn, p1 = q, pn = p such that pi+1 is directly density-reachable from pi

### Density-connected

* A point p is density-connected to a point q wrt. Eps, MinPts if there is a point o such that both, p and q are density-reachable from o wrt. Eps and MinPts.

## Working Of Density-Based Clustering

Given a set of objects, D' we say that an object p is directly density-reachable from object q if p is within the ε-neighborhood of q, and q is a core object.  
An object p is density-reachable from object q with respect to ε and MinPts in a set of objects, D' if there is a chain of objects p1,.,.,.pn, where p1 = q and pn = p such that pi+1 is directly density-reachable from pi with respect to e and MinPts, for 1/n, pi € D.  
An object p is density-connected to object q with respect to ε and MinPts in a set of objects, D', if there is an object o, belongs D such that both p and q are density-reachable from o with respect to ε and MinPts.

## Density-Based Clustering - Background

Two parameters:

* Eps: Maximum radius of the neighborhood.
* MinPts: Minimum number of points in an Eps-neighbourhood of that point.

NEps(p): {q belongs to D | dist(p,q) <= Eps}  
Directly density-reachable: A point p is directly density-reachable from a point q wrt. Eps, MinPts if

* p belongs to NEps(q)
* core point condition:|NEps (q)| >= MinPts

## Major features:

It is used to discover clusters of arbitrary shape.

It is also used to handle noise in the data clusters.

It is a one scan method.

It needs density parameters as a termination condition.

## Density-Based Methods

**DBSCAN**: Ester, et al. (KDD’96)

**OPTICS**: Ankerst, et al (SIGMOD’99).

**DENCLUE**: Hinneburg& D. Keim  (KDD’98)

**CLIQUE**: Agrawal, et al. (SIGMOD’98)

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

It relies on a density-based notion of cluster:  A cluster is defined as a maximal set of density-connected points.

It discovers clusters of arbitrary shape in spatial databases with noise.

## DBSCAN Algorithm

Arbitrary select a point p.

Retrieve all points density-reachable from p wrtEps and MinPts.

If p is a core point, a cluster is formed.

If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.

Continue the process until all of the points have been processed.

say, let MinPts = 3.   
  
Of the labeled points, m, p, o, and r are core objects because each is in an ε-neighborhood containing at least three points.

q is directly density-reachable from m. m is directly density-reachable from p and vice versa.

q is (indirectly) density-reachable from p  because q is directly density-reachable from m and m is directly density-reachable from p.

However, p is not density-reachable from q  because q is not a core object.

Similarly, r and s are density-reachable from o, and o is density-reachable from o, and o is density-reachable from R.   
OTICS **–**

**A Cluster-Ordering Method**

OPTICS: Ordering Points To Identify the Clustering Structure.

It produces a special order of the database with respect to its density-based clustering structure.

This cluster-ordering contains info equivalent to the density-based clusterings corresponding to a broad range of parameter settings.

It is good for both automatic and interactive cluster analysis, including finding an intrinsic clustering structure.

It can be represented graphically or using visualization techniques.

Core-distance and reachability-distance: The figure illustrates the concepts of core-distance and reachability-distance.

Suppose that e=6 mm and MinPts=5.   
The core distance of p is the distance, e0, between p and the fourth closest data object.

The reachability-distance of q1 with respect to p is the core-distance of p (i.e., e0 =3 mm) because this is greater than the Euclidean distance from p to q1.

The reachability distance of q2 with respect to p is the Euclidean distance from p to q2 because this is greater than the core-distance of p.  
  
DENCLUE - Using Density Functions

**DEN**sity-based **CLU**st**E**ring by Hinneburg&Keim  (KDD’98)

**Major Features**

It ha got a solid mathematical foundation.

It is definitely good for data sets with large amounts of noise.

It allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets.

It is significantly faster than the existing algorithm (faster than DBSCAN by a factor of up to 45).

But it needs a large number of parameters.

DENCLUE - Technical Essence

It uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure.

**Influence function:**This describes the impact of a data point within its neighborhood.

The Overall density of the data space can be calculated as the sum of the influence function of all data points.

The Clusters can be determined mathematically by identifying density attractors.

The Density attractors are local maxima of the overall density function.

THE END