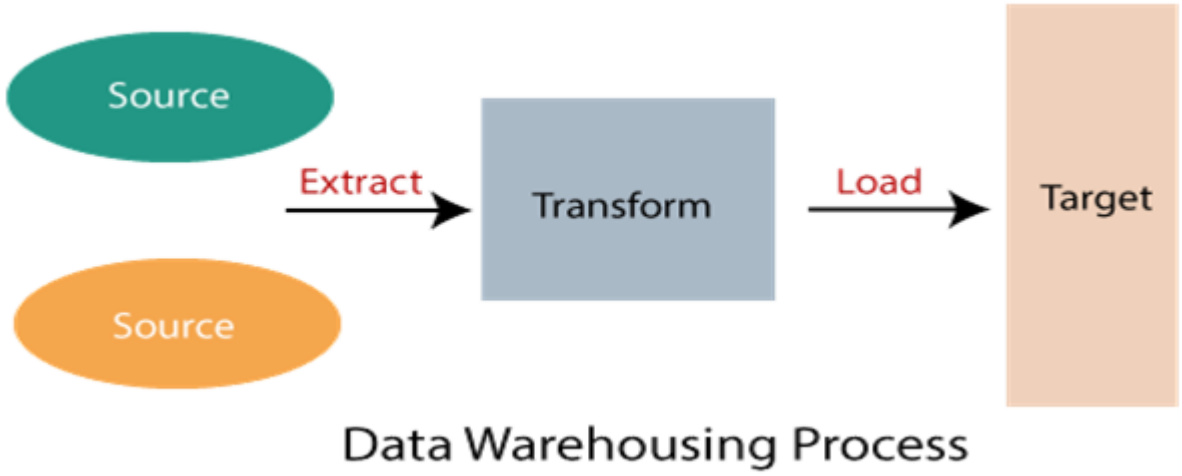
Data Warehousing and Data Mining

Data warehouse refers to the process of compiling and organizing data into one common database, whereas **data mining** refers to the process of extracting useful data from the databases. The data mining process depends on the data compiled in the data warehousing phase to recognize meaningful patterns. A data warehousing is created to support management systems.

Data Warehouse:

A **Data Warehouse** refers to a place where data can be stored for useful mining. It is like a quick computer system with exceptionally huge data storage capacity. Data from the various organization's systems are copied to the Warehouse, where it can be fetched and conformed to delete errors. Here, advanced requests can be made against the warehouse storage of data.



Data Warehousing Process

Data warehouse combines data from numerous sources which ensure the data quality, accuracy, and consistency. Data warehouse boosts system execution by separating analytics processing from transnational databases. Data flows into a data warehouse from different databases. A data warehouse works by sorting out data into a pattern that depicts the format and types of data. Query tools examine the data tables using patterns.

Data warehouses and **databases** both are relative data systems, but both are made to serve different purposes. A data warehouse is built to store a huge amount of historical data and empowers fast requests over all the data, typically using **Online Analytical Processing** (OLAP). A database is made to store current transactions and allow quick access to specific transactions for ongoing business processes, commonly known as **Online Transaction Processing** (OLTP).

Data modelling for Data Warehouses

A data model for a data warehouse (DW) is a conceptual representation of the structure and relationships between the data elements that make up the DW. The model is used as a starting point for the creation of a data repository for business facts; it's also a way to inform stakeholders how data will be organized, stored, and accessed.

The notation and elements of the entity-relationship diagram (ER diagram or ERD) are commonly used to model data warehouses, although ERDs are mostly used to design transactional databases.

Creating a Data Model for a Data Warehouse

When faced with the task of creating a data model for a warehouse, you may be tempted to use your favorite SQL client and start creating tables right away. This may work for a very small data warehouse. But data warehouses are not exactly known for being small.

For this reason, using the "quick & dirty" methodology to create a data warehouse is decidedly a bad idea. This is mainly because, once the data warehouse is full of data, modifying its structure is quite a difficult task.

We must keep in mind that a data warehouse – even more than a database – is a strategic and critical tool for business. For this reason, we must minimize all possible risks during its construction. To do so, I recommend you follow the steps below. As a result you will have a data warehouse built to last.

Step 1: Understand Business Objectives and Processes

The first phase of creating a data model for a data warehouse involves requirements engineering work, in which you gain an overall understanding of the information and results you expect from using the data warehouse. As a result of this first phase, you should get a detailed description of data warehouse requirements; this serves as input for the next phase.

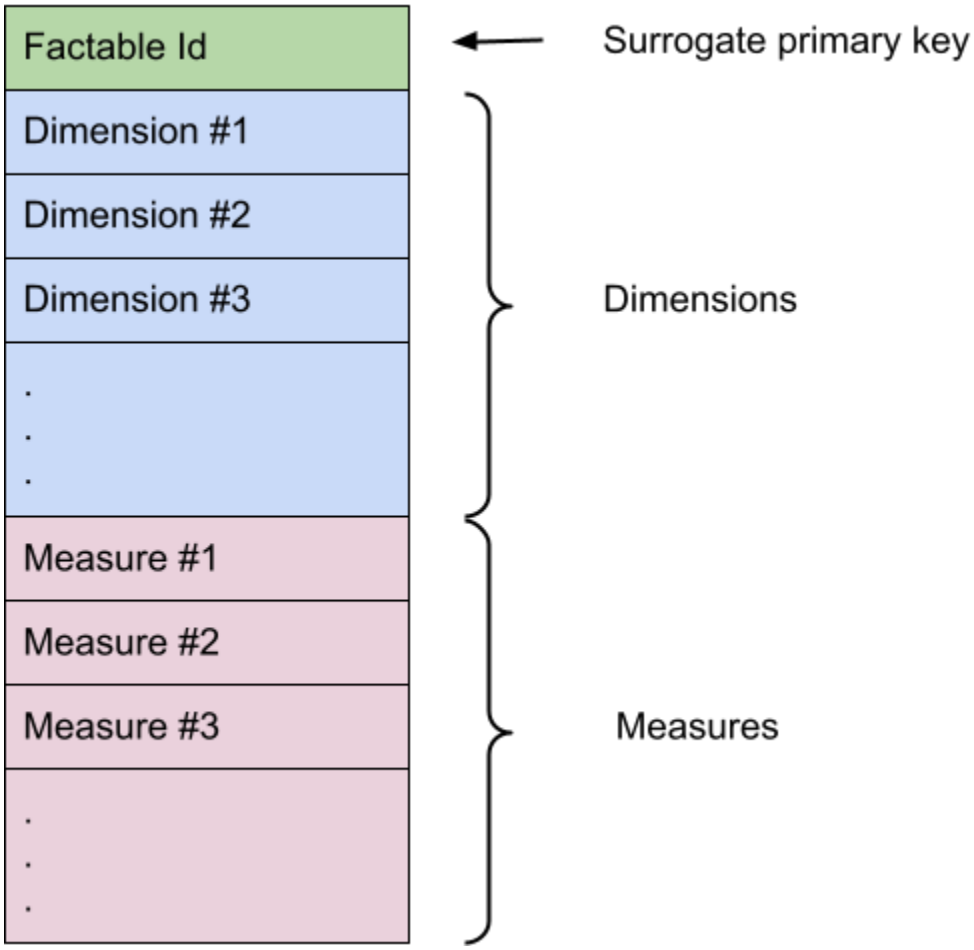
Step 2: Create a Conceptual Model

Using the detailed requirements obtained in the initial phase, start building a conceptual model that provides an overview of the two main types of tables in any data warehouse: fact tables and dimension tables. You can learn a lot about them by reading our article on FACTS AND DIMENSIONS IN A DATA WAREHOUSE.

The fact tables are central to the diagram. They contain two fundamental types of attributes: numerical measures and dimension identifiers. Numerical measures are aggregate values (totals, averages, etc.) for each combination of dimension identifiers.

Dimension identifiers are usually foreign keys to the dimension tables surrounding the fact tables. There are many different types of dimension tables, which types you use depends on how the tables are maintained and the kind of information they store. You can learn more about this by reading our article on THE MOST COMMON TYPES OF DIMENSION TABLES.

For simplicity, you can think of dimension tables as lookup tables for identifiers that appear in fact tables, such as product SKUs, customer codes, vendor codes, etc.



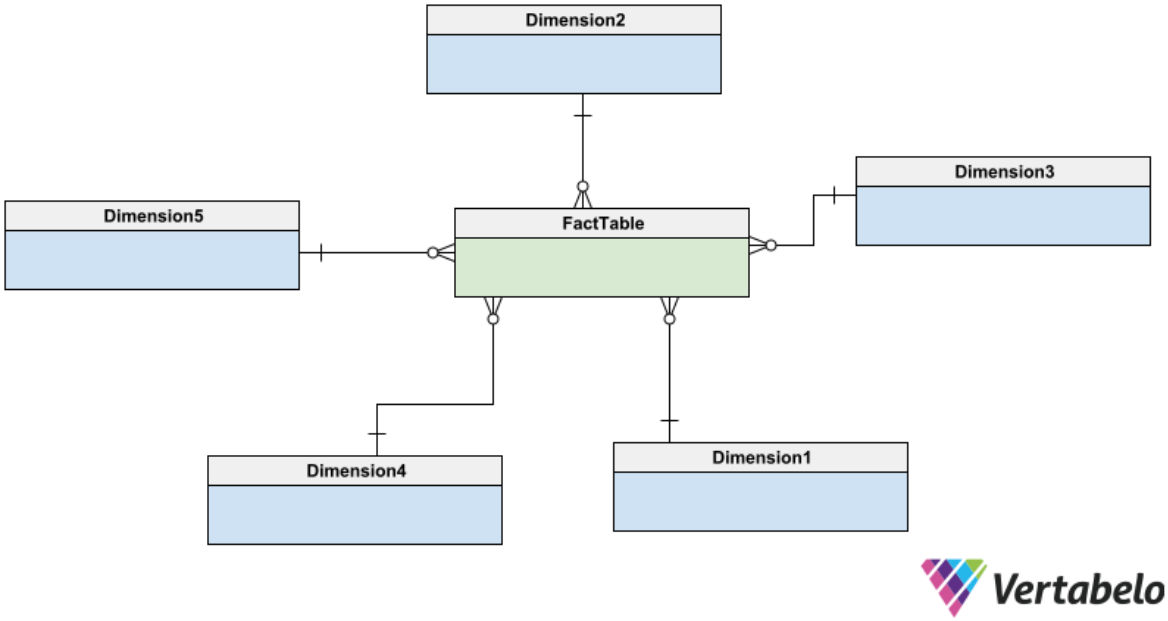
The basic structure of a fact table consists of a set of dimensions and a set of measures containing aggregate values.

Step 3: Define the Shape of the Data Model

The conceptual model should show the shape that the data model will have. This shape will be determined by the distribution of the fact and dimension tables.

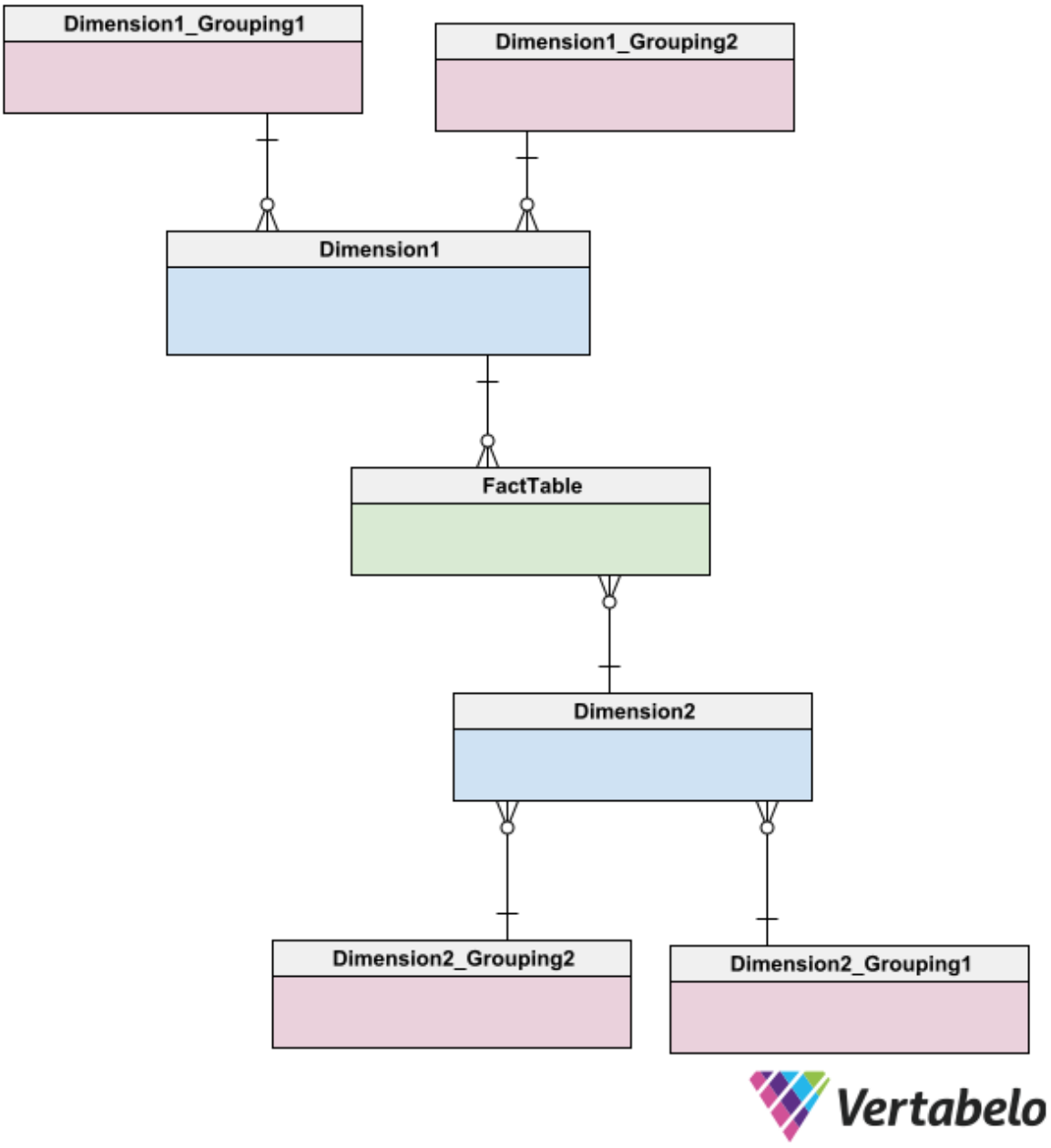
Three fundamental types of structures are recognized, named for the similarity of their shapes: star, snowflake, and constellation.

In a star data warehouse schema, a single fact table is placed in the center of the diagram. All dimension tables are related to the fact table by a foreign key in the fact table. In the diagram, the dimension tables surround the fact table, giving it a star-like shape. Find more information in THIS ARTICLE ON THE STAR SCHEMA.



Basic form of a star schema with one fact table (green) and five dimension tables (blue).

In a snowflake schema, the fact table is surrounded by small clusters formed by hierarchies of tables. Each of these hierarchies is a normalized sub-schema that is associated with a dimension of the fact table. For more information you can read THIS ARTICLE ON SNOWFLAKE SCHEMAS or this one on the DIFFERENCES BETWEEN STAR AND SNOWFLAKE SCHEMAS.



Basic form of a snowflake schema with one fact table (green) and two dimension tables (blue), each one related to two grouping tables (red).

In constellation schemas – also called galaxy schemas – several fact tables appear. Each of them responds to a different business information need and has a set of dimensions surrounding it. Some dimension tables may be shared between the different fact tables.

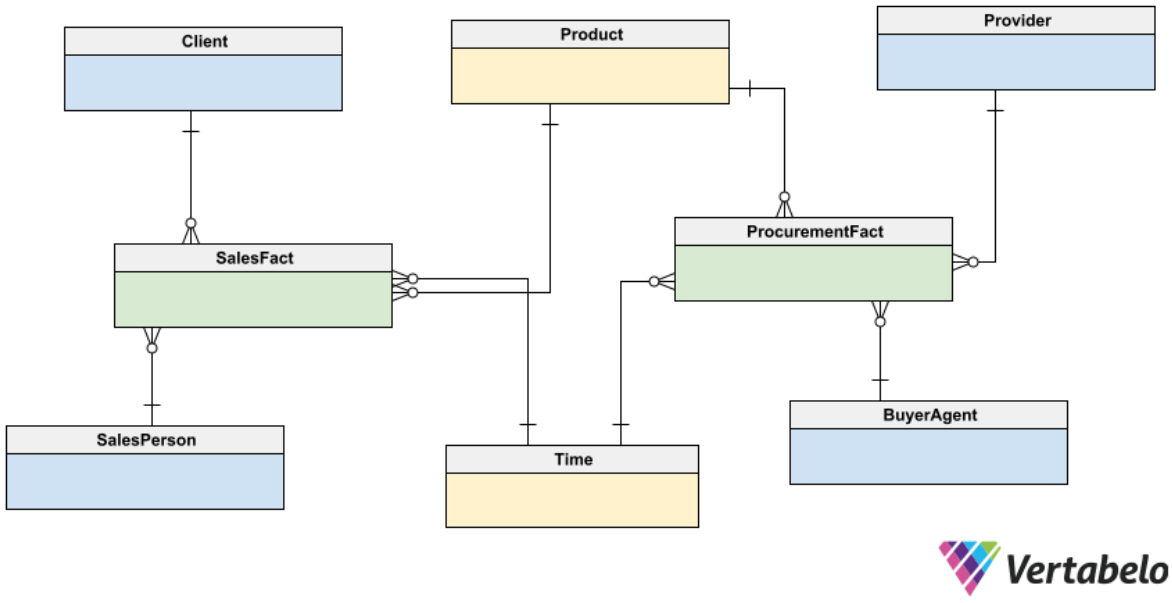
In the model we will build below, you will see an example of a constellation schema. It will have two fact tables: one for sales and one for procurement (i.e. purchases made by the company).

Step 4: Design the Conceptual Data Model

For the construction of the conceptual model, we do not need to define the diagram down to the smallest detail. It is enough to show the fact tables, the dimension tables, and the relationships between them. This diagram will help us explain the model to users and stakeholders. The goal is to obtain feedback and approval so that when the database is running there is no possibility of misunderstandings and complaints.

To illustrate the process, we will use the VERTABELO platform to create a conceptual model. The advantage of using Vertabelo is that it allows us to work in a top-down manner – i.e. starting with a conceptual model, then making a copy of it to create a logical model, and using the subsequent logical model to derive the physical model. Finally, Vertabelo allows us to generate the DDL scripts that will enable the creation of the data warehouse in any DBMS (Database Management System) or warehouse tool.

As mentioned above, our data model will have a galaxy or constellation shape:



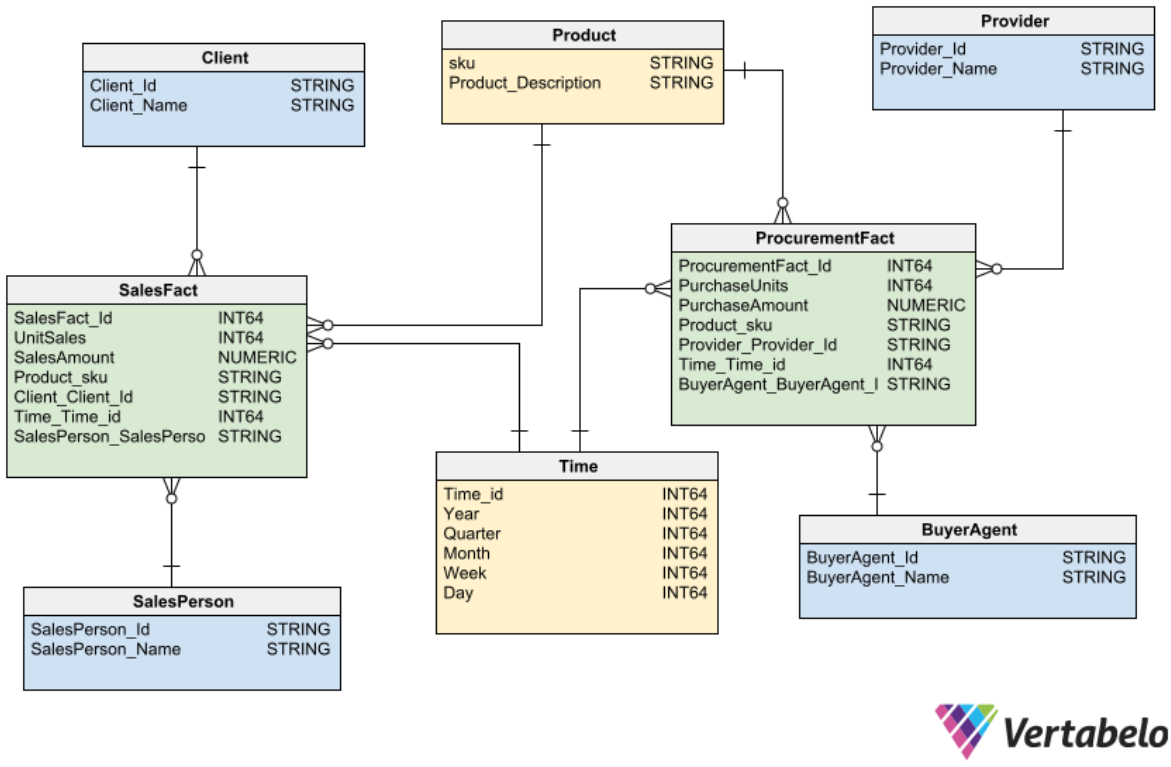
Vertabelo allows us to use different colors in our model to identify different types of tables. In this constellation schema, fact tables are green, shared dimension tables are yellow, and the rest of the dimension tables are blue.

Step 5: Create the Logical Data Model

The conceptual model allows everyone involved in the process to verify that the model meets the requirements and to give their approval for further development. Vertabelo allows you to share your model with different users and give each one the privileges they need (i.e. either to view the design or to make modifications to it). You can see the options for SHARING YOUR MODELS HERE.

Using Vertabelo, we can create the logical model from the conceptual one. There are two ways to do this. We can create a replica of the conceptual model and extend that replica to build the logical model, or we can tag the version of the conceptual model and then continue working on it. By creating a tag, we can revert to the tagged version at any moment (e.g. if we need to revise the conceptual model). You will find more information about VERSION CONTROL IN VERTABELO HERE.

To complete the logical model for our data warehouse, we need an entity for each fact table and for each dimension table. Then we need to establish the corresponding relationships between them. In the logical model, we must include all the entities and all the attributes. Don't overlook any of them!



The logical model includes all the entities and all the attributes that make up our data warehouse.

Our data model for a warehouse includes two fact tables: one for `Sales` and one for `Procurement`. Both share the dimensions `Time` and `Product`, since the same data is used to characterize both sales and procurement facts.

Then, both fact tables are related to dimensions that are specific to each business process:

- The `Sales` fact table relates to the `SalesPerson` and `Client` dimension tables.
- The `Procurement` fact table relates to the `BuyerAgent` and `Provider` dimension tables.

At this point, it is important to submit the data model to a validation process that will give us the greatest possible assurance that the database can be implemented without errors. This validation process will also minimize risks to information integrity. For this purpose, we can use the LIVE MODEL VALIDATION FEATURE OFFERED BY VERTABELO. This functionality will allow us to automatically detect problems, such as:

- Entity name repetition.
- Attribute name repetition within the same entity.
- Entities without attributes.
- Entities without primary identifiers.
- Attributes with different data types involved in a relation.
- And many other possible errors that could cause future problems.

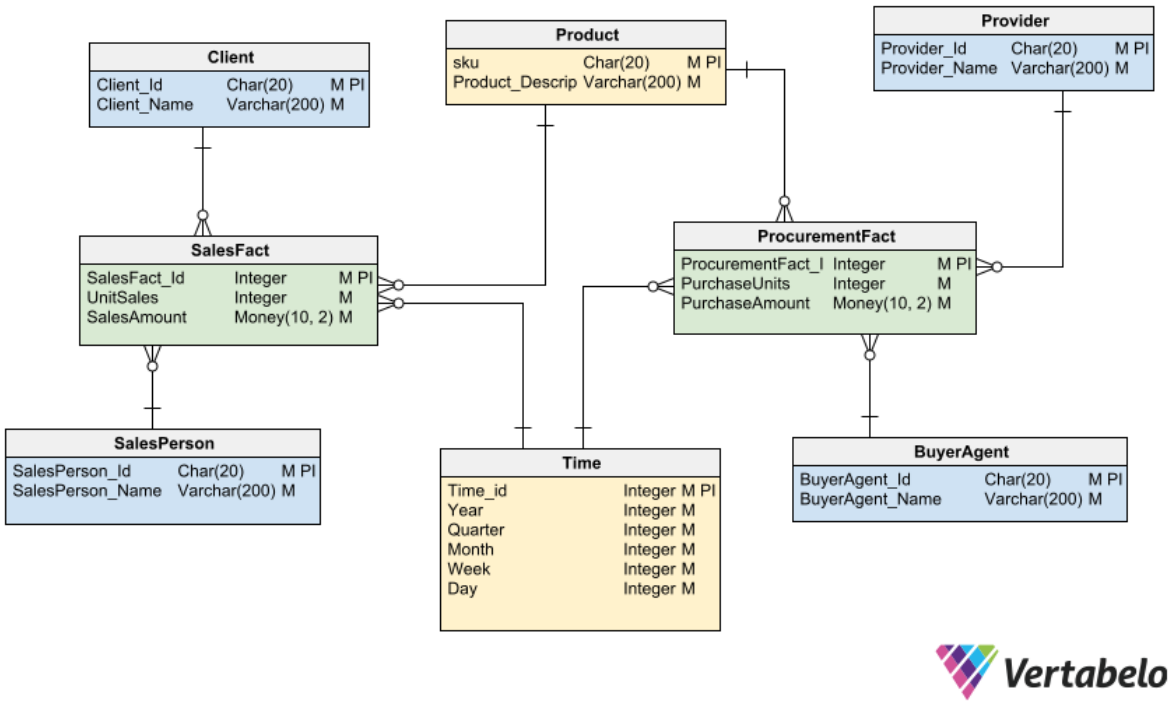
Live validation should also be applied to the physical model, since this model could be affected by errors of its own.

Step 6: Create the Physical Data Model

Having the logical model verified by Vertabelo's live validation feature, the only missing step before implementing the model on a database engine is to generate a physical model from the logical one.

This task is performed automatically by Vertabelo. The only human intervention needed is to tell Vertabelo which database engine will be used to implement the database. Read this article about the MAIN DATA

WAREHOUSE TOOLS to find out which option to choose when creating your physical data model.



The automatically generated physical model is almost ready to be implemented. In this case, we picked Google BigQuery as the target database engine.

The live validation feature can also be used on the physical model to detect (before it is too late) problems that may affect the integrity of the data or the database itself. An example is columns with data types that are incompatible with the warehouse tool on which the model will be implemented.

Step 7: Implement the Model

The last step required to build our data warehouse is to implement the physical model on the target DBMS. This step consists of generating the scripts that must be executed on the DBMS to create the database.

Concept Hierarchy

In a data warehouse, a **concept hierarchy** is used to organize data from multiple sources into a single, consistent, and meaningful structure. It is a tree-like structure that represents the organization of data, where each level of the hierarchy represents a concept that is more general than the level below it . This hierarchical organization of data allows for more efficient and effective data analysis, as well as the ability to drill down to more specific levels of detail when needed.

There are several types of concept hierarchies, including:

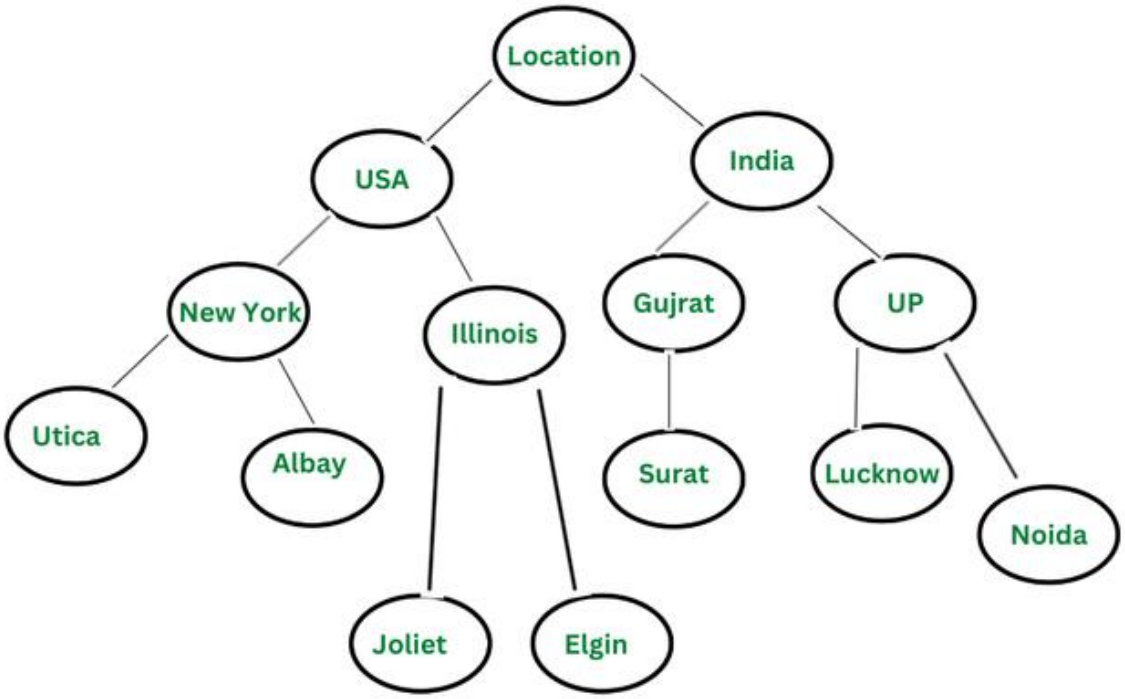
- **Schema Hierarchy**: Used to organize the schema of a database in a logical and meaningful way, grouping similar objects together .
- **Set-Grouping Hierarchy**: Based on set theory, where each set in the hierarchy is defined in terms of its membership in other sets. Set-grouping hierarchy can be used for data cleaning, data pre-processing, and data integration .
- **Operation-Derived Hierarchy**: Represented by a set of operations on the data. These operations are defined by users, professionals, or the data mining system. These hierarchies are usually represented for mathematical attributes.
- **Rule-based Hierarchy**: Either a whole concept hierarchy or an allocation of it is represented by a set of rules and is computed dynamically based on the current information and rule definition. A lattice-like architecture is used for graphically defining this type of hierarchy .

Concept hierarchy enables raw information to be managed at a higher and more generalized level of abstraction [2]. It can be used in business intelligence to organize and classify data in a way that makes it more understandable and easier to analyze [1].

Concept Hierarchy in Data Mining

In data mining, the concept of a concept hierarchy refers to the organization of data into a tree-like structure, where each level of the hierarchy represents a concept that is more general than the level below it. This hierarchical organization of data allows for more efficient and effective data analysis, as well as the ability to drill down to more specific levels of detail when needed. The concept of hierarchy is used to organize and classify data in a way that makes it more understandable and easier to analyze. The main idea behind the concept of hierarchy is that the same data can have different levels of granularity or levels of detail and that by organizing the data in a hierarchical fashion, it is easier to understand and perform analysis.

Example:



Concept Hierarchy for Dimension Location

Hierarchical Structure for Dimension Location

OLAP and OLTP

OLAP stands for Online Analytical Processing. OLAP systems have the capability to analyze database information of multiple systems at the current time. The primary goal of OLAP Service is data analysis and not data processing.

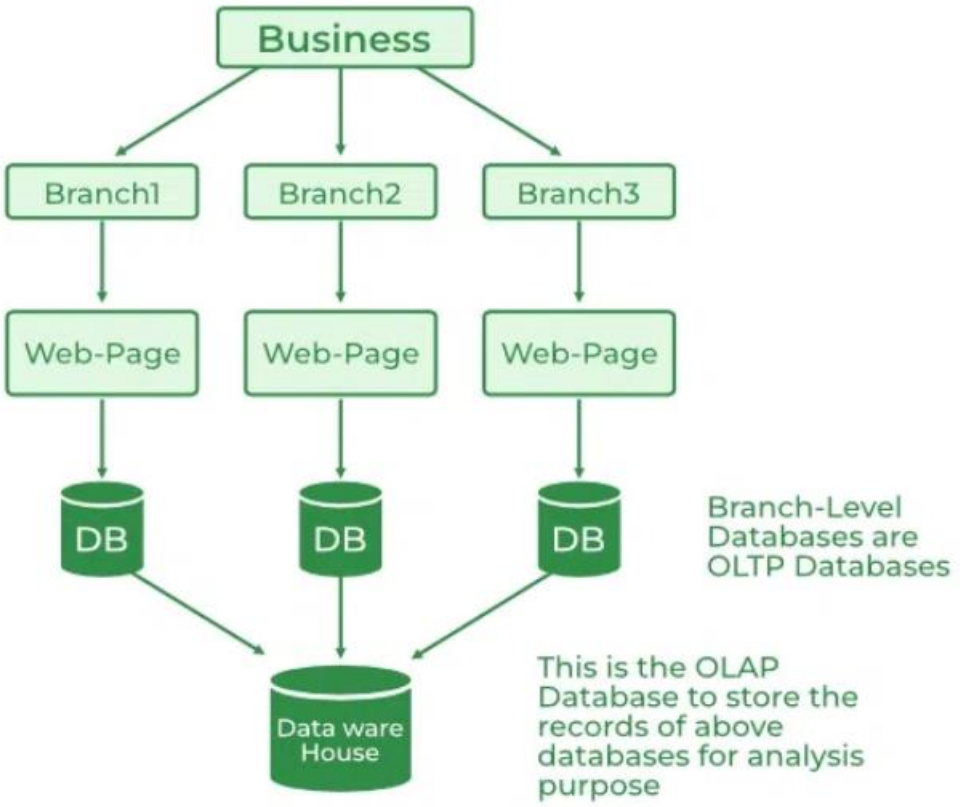
OLTP stands for Online Transaction Processing. OLTP has the work to administer day-to-day transactions in any organization. The main goal of OLTP is data processing not data analysis.

Online Analytical Processing (OLAP)
Online Analytical Processing (OLAP) consists of a type of software tool that is used for data analysis for business decisions. OLAP provides an environment to get insights from the database retrieved from multiple database systems at one time.

OLAP Examples

Any type of Data Warehouse System is an OLAP system. The uses of the OLAP System are described below.

- Spotify analyzed songs by users to come up with a personalized homepage of their songs and playlist.
- Netflix movie recommendation system.



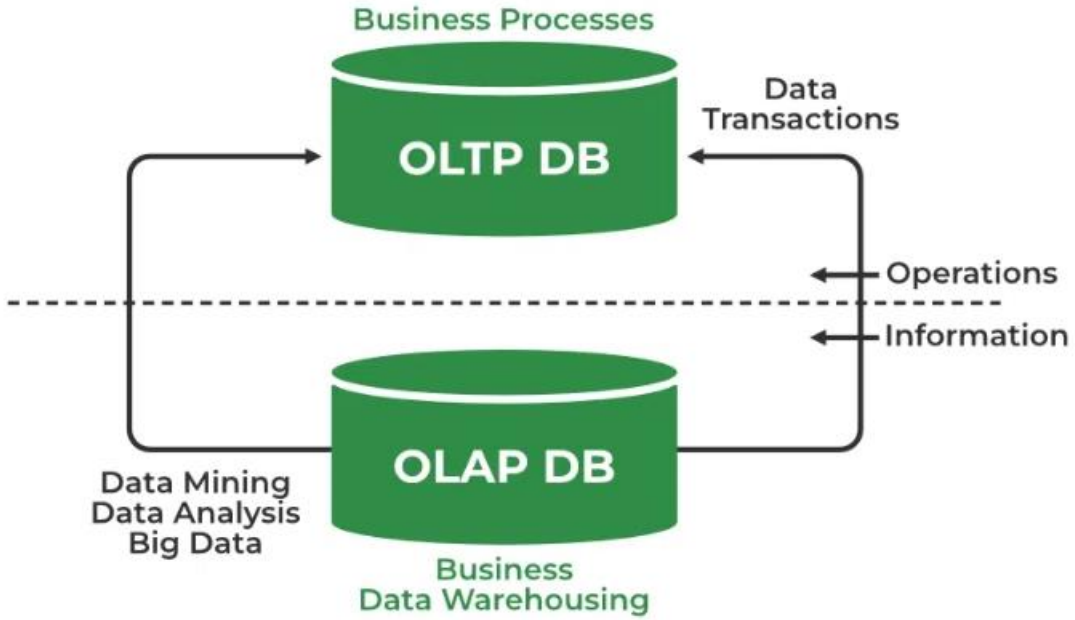
OLAP

Online Transaction Processing (OLTP)
Online transaction processing provides transaction-oriented applications in a 3-tier architecture. OLTP administers the day-to-day transactions of an organization.

OLTP Examples

An example considered for OLTP System is ATM Center a person who authenticates first will receive the amount first and the condition is that the amount to be withdrawn must be present in the ATM. The uses of the OLTP System are described below.

- ATM center is an OLTP application.
- OLTP handles the ACID properties during data transactions via the application.
- It's also used for Online banking, Online airline ticket booking, sending a text message, add a book to the shopping cart.



OLTP vs OLAP

Association Rules

Association rule learning is a machine learning method for discovering interesting relationships between variables in large databases. It is designed to detect strong rules in the database based on some interesting metrics. For any given multi-item transaction, association rules aim to obtain rules that determine how or why certain items are linked.

Association rules are created by searching for information on common if-then patterns and using specific criteria with support and trust to define what the key relationships are. They help to show the frequency of an item in a given data since confidence is defined by the number of times an if-then statement is found to be true. However, a third criterion called lift is often used to compare expected and actual confidence. Lift shows how many times the if-then statement was predicted to be true. Create association rules to compute itemsets based on data created by two or more items. Association rules usually consist of rules that are well represented by the data.

There are different types of data mining techniques that can be used to find out the specific analysis and result like Classification analysis, Clustering analysis, and multivariate analysis. Association rules are mainly used to analyze and predict customer behavior.

- In Classification analysis, it is mostly used to question, make decisions, and predict behavior.
- In Clustering analysis, it is mainly used when no assumptions are made about possible relationships in the data.
- In Regression analysis, it is used when we want to predict an infinitely dependent value of a set of independent variables.

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.[1] In any given transaction with a variety of items, association rules are meant to discover the rules that determine how or why certain items are connected.

To illustrate the concepts, we use a small example from the supermarket domain. Table 2 shows a small database containing the items where, in each entry, the value 1 means the presence of the item in the corresponding transaction, and the value 0 represents the absence of an item in that transaction. The set of items is

I={milk, bread, butter, beer, diapers, eggs, fruit}.

An example rule for the supermarket could be {butter, bread}⟹ {milk} meaning that if butter and bread are bought, customers also buy milk.

In order to select interesting rules from the set of all possible rules, constraints on various measures of significance and interest are used. The best-known constraints are minimum thresholds on support and confidence.

Let X, Y be itemsets, $X \Rightarrow Y$ an association rule and T a set of transactions of a given database.

Note: this example is extremely small. In practical applications, a rule needs a support of several hundred transactions before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

Support

Support is an indication of how frequently the itemset appears in the dataset.

In our example, it can be easier to explain support by writing

Support=P(A U B)= (number of transactions containing A and B) / (total number of transactions)

where A and B are separate item sets that occur in at the same time in a transaction.

Using Table 2 as an example, the itemset X= {beer, diapers} has a support of $1/5=0.2$ since it occurs in 20% of all transactions (1 out of 5 transactions). The argument of *support of X* is a set of preconditions, and thus becomes more restrictive as it grows (instead of more inclusive).[13]

Furthermore, the itemset $Y=\{milk, bread, butter\}$ has a support of $1/5=0.2$ as it appears in 20% of all transactions as well.

When using antecedents and consequents, it allows a data miner to determine the support of multiple items being bought together in comparison to the whole data set. For example, Table 2 shows that if milk is bought, then bread is bought has a support of 0.4 or 40%. This because in 2 out 5 of the transactions, milk as well as bread are bought. In smaller data sets like this example, it is harder to see a strong correlation when there are few samples, but when the data set grows larger, support can be used to find correlation between two or more products in the supermarket example.

Minimum support thresholds are useful for determining which itemsets are preferred or interesting.

If we set the support threshold to ≥0.4 in Table 3, then the $\{milk\} \Rightarrow$ {eggs} would be removed since it did not meet the minimum threshold of 0.4. Minimum threshold is used to remove samples where there is not a strong enough support or confidence to deem the sample as important or interesting in the dataset.

Another way of finding interesting samples is to find the value of (support)X(confidence); this allows a data miner to see the samples where support and confidence are high enough to be highlighted in the dataset and prompt a closer look at the sample to find more information on the connection between the items.

Support can be beneficial for finding the connection between products in comparison to the whole dataset, whereas confidence looks at the connection between one or more items and another item. Below is a table that shows the comparison and contrast between support and support x confidence, using the information from Table 4 to derive the confidence values.

Table 3. Example of Support, and support X confidence

if Antecedent then Consequent	support	support X confidence
if buy milk, then buy bread	2/5= 0.4	0.4X1.0= 0.4
if buy milk, then buy eggs	1/5= 0.2	0.2X0.5= 0.1
if buy bread, then buy fruit	2/5= 0.4	0.4X0.66= 0.264

Table 3. Example of Support, and support X confidence

if Antecedent then Consequent	support	support X confidence
if buy fruit, then buy eggs	2/5= 0.4	0.4X0.66= 0.264
if buy milk and bread, then buy fruit	2/5= 0.4	0.4X1.0= 0.4

The support of X with respect to T is defined as the proportion of transactions in the dataset which contains the itemset X. Denoting a transaction by (i,t) where i is the unique identifier of the transaction and t is its itemset, the support may be written as:

$$\textbf{support of X} = |\{(i,t) \in T : X \subseteq t\}| \; / \; |T|$$

This notation can be used when defining more complicated datasets where the items and itemsets may not be as easy as our supermarket example above. Other examples of where support can be used is in finding groups of genetic mutations that work collectively to cause a disease, investigating the number of subscribers that respond to upgrade offers, and discovering which products in a drug store are never bought together.[12]

Confidence

Confidence is the percentage of all transactions satisfying X that also satisfy Y.

With respect to T, the confidence value of an association rule, often denoted as $X \Longrightarrow Y$, is the ratio of transactions containing both X and Y to the total amount of X values present, where X is the antecedent and Y is the consequent.

Confidence can also be interpreted as an estimate of the conditional probability $P(E_Y \mid E_X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.[13][15]

It is commonly depicted as:

$$\text{conf}(X \Longrightarrow Y) = P(Y|X) = \text{supp}(X \cup Y) \; / \; \text{supp}(X) = \text{number of transactions containing X and Y} \; / \; \text{number of transactions containing X}$$

The equation illustrates that confidence can be computed by calculating the co-occurrence of transactions X and Y within the dataset in ratio to transactions containing only X. This means that the number of transactions in both X and Y is divided by those just in X .

For example, Table 2 shows the rule $\{\text{butter, bread}\} \Longrightarrow \{\text{milk}\}$ which has a confidence of

$1/5 \; / \; 1/5 = 0.2/0.2 = 1.0$ in the dataset, which denotes that every time a customer buys butter and bread, they also buy milk. This particular example demonstrates the rule being correct 100% of the time for transactions containing both butter and bread. The rule $\{\text{fruit}\} \Longrightarrow \{\text{eggs}\}$, however, has a confidence of

$2/5 \; / \; 3/5 = 0.4 \; / \; 0.6 = 0.67$. This suggests that eggs are bought 67% of the times that fruit is brought. Within this particular dataset, fruit is purchased a total of 3 times, with two of those times consisting of egg purchases.

For larger datasets, a minimum threshold, or a percentage cutoff, for the confidence can be useful for determining item relationships. When applying this method to some of the data in Table 2, information that does not meet the requirements are removed. Table 4 shows association rule examples where the minimum threshold for confidence is 0.5 (50%). Any data that does not have a confidence of at least 0.5 is omitted. Generating thresholds allow for the association between items to become stronger as the data is further researched by emphasizing those that co-occur the most. The table uses the confidence information from Table 3 to implement the Support x Confidence column, where the relationship between items via their both confidence and support, instead of just one concept, is highlighted. Ranking the rules by Support x Confidence multiples the confidence of a particular rule to its support and is often implemented for a more in-depth understanding of the relationship between the items.

Table 4. Example of Confidence and Support x Confidence

if Antecedent then Consequent	Confidence	Support x Confidence
if buy milk, then buy bread	2/2= 1.0	0.4X1.0= 0.4
if buy milk, then buy eggs	1/2= 0.5	0.2X0.5= 0.1
if buy bread, then buy fruit	2/3= 0.66	0.4X0.66= 0.264
if buy fruit, then buy eggs	2/3= 0.66	0.4X0.66= 0.264
if buy milk and bread, then buy fruit	2/2= 1.0	0.4X1.0= 0.4

Overall, using confidence in association rule mining is great way to bring awareness to data relations. Its greatest benefit is highlighting the relationship between particular items to one another within the set, as it compares co-occurrences of items to the total occurrence of the antecedent in the specific rule. However, confidence is not the optimal method for every concept in association rule mining. The disadvantage of using it is that it does not offer multiple difference outlooks on the associations. Unlike support, for instance, confidence does not provide the perspective of relationships between certain items in comparison to the entire dataset, so while milk and bread, for example, may occur 100% of the time for confidence, it only has a support of 0.4 (40%). This is why it is important to look at other viewpoints, such as Support x Confidence, instead of solely relying on one concept incessantly to define the relationships.

Lift

The *lift* of a rule is defined as:

$$\text{lift}\ (X \Longrightarrow Y) = \text{supp}\ (X \cup Y) \ /\ \text{supp}(X) \times \text{supp}(Y)$$

or the ratio of the observed support to that expected if X and Y were independent.

For example, the rule {milk, bread} $\Longrightarrow$ {butter} has a lift of 0.2 / 0.4x0.4=1.25.

If the rule had a lift of 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events.

If the lift is > 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.

If the lift is < 1, that lets us know the items are substitute to each other. This means that presence of one item has negative effect on presence of other item and vice versa.

The value of lift is that it considers both the support of the rule and the overall data set.[13]

Conviction

The *conviction* of a rule is defined as $\mathrm{conv}(X \Rightarrow Y) = 1-\mathrm{supp}(Y) / 1- \mathrm{conf}(X \Rightarrow Y)$

For example, the rule $\{\text{milk, bread}\} \Rightarrow \{\text{butter}\}$ has a conviction of 1-0.4 /1-0.5=1.2, and can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions. In this example, the conviction value of 1.2 shows that the rule $\{\text{milk, bread}\} \Rightarrow \{\text{bread}\}$ would be incorrect 20% more often (1.2 times as often) if the association between X and Y was purely random chance.

Classification, Clustering, and Regression

Machine learning is a field of study that involves training machines to learn from data and make predictions. There are three main types of machine learning: **classification**, **clustering**, and **regression** .

Classification is a type of supervised learning that involves predicting a categorical label for a given input. For example, given an image of a cat or dog, the algorithm would predict whether the image contains a cat or a dog.

Clustering is an unsupervised learning technique that involves grouping similar data points together. For example, clustering can be used to group customers based on their purchasing behavior.

Regression is another type of supervised learning that involves predicting a continuous value for a given input. For example, regression can be used to predict the price of a house based on its features such as location, number of bedrooms, etc.

Classification Algorithm in Machine Learning

As we know, the Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms. In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.

What is the Classification Algorithm?

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, **Yes or No, 0 or 1, Spam or Not Spam, cat or dog,** etc. Classes can be called as targets/labels or categories.

Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.

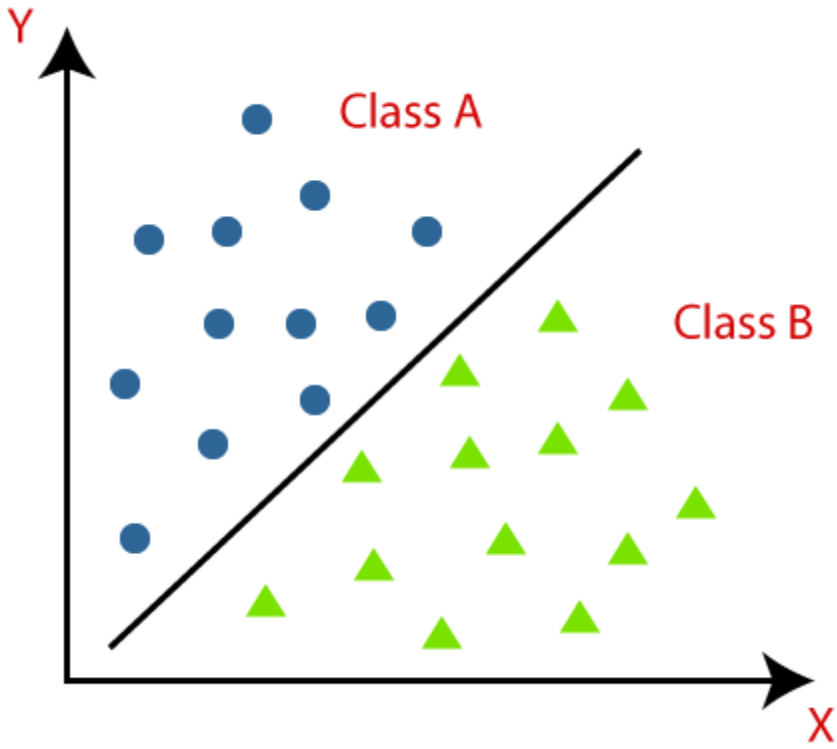
In classification algorithm, a discrete output function(y) is mapped to input variable(x).

1. y=f(x), where y = categorical output

The best example of an ML classification algorithm is **Email Spam Detector**.

The main goal of the Classification algorithm is to identify the category of a given dataset, and these algorithms are mainly used to predict the output for the categorical data.

Classification algorithms can be better understood using the below diagram. In the below diagram, there are two classes, class A and Class B. These classes have features that are similar to each other and dissimilar to other classes.



The algorithm which implements the classification on a dataset is known as a classifier.
There are two types of Classifications:

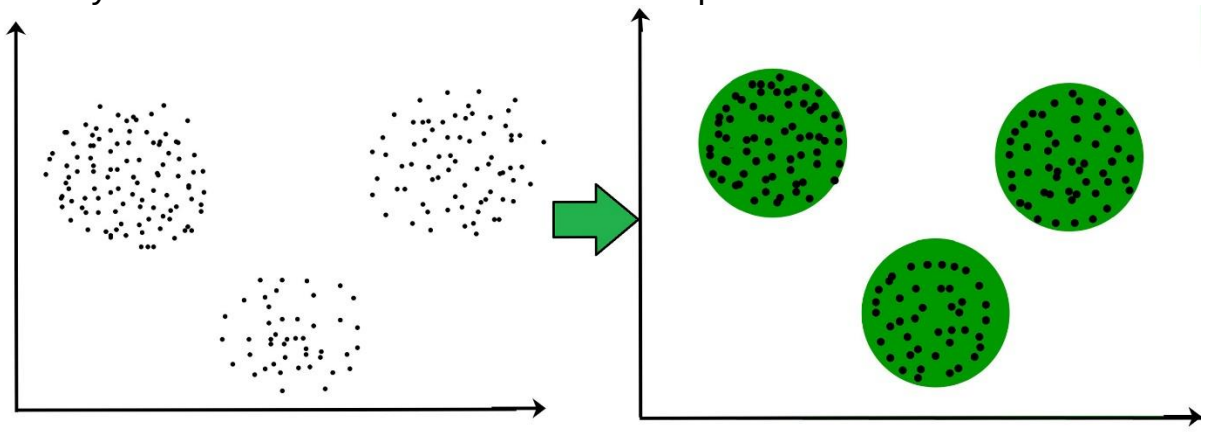
Binary Classifier: If the classification problem has only two possible outcomes, then it is called as Binary Classifier.
Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.
Multi-class Classifier: If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.
Example: Classifications of types of crops, Classification of types of music.

Clustering:

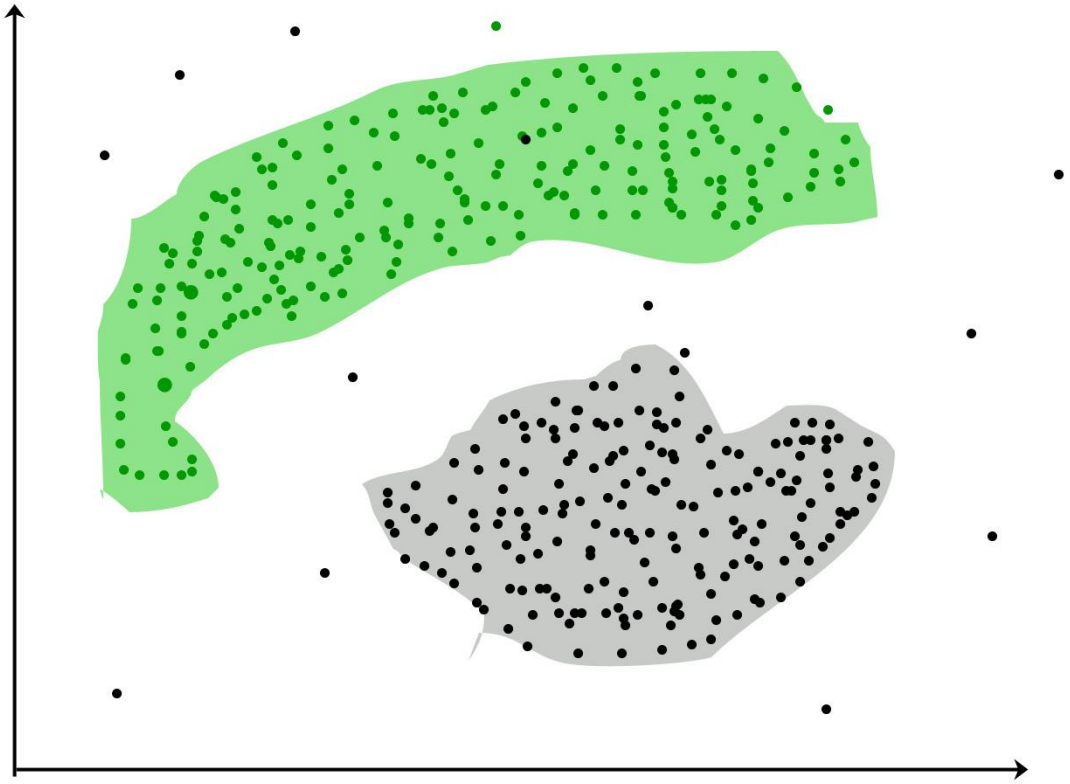
It is basically a type of *unsupervised learning method*. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.
Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.

It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For example The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.



It is not necessary for clusters to be spherical as depicted below:



DBSCAN: Density-based Spatial Clustering of Applications with Noise

These data points are clustered by using the basic concept that the data point lies within the given constraint from the cluster center. Various distance methods and techniques are used for the calculation of the outliers.

Regression:

Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables).

Also called simple regression or ordinary least squares (OLS), linear regression is the most common form of this technique. Linear regression establishes the linear relationship between two variables based on a line of best fit. Linear regression is thus graphically depicted using a straight line with the slope defining how the change in one variable impacts a change in the other. The y-intercept of a linear regression relationship represents the value of one variable when the value of the other is zero. Non-linear regression models also exist, but are far more complex.

Regression analysis is a powerful tool for uncovering the associations between variables observed in data, but cannot easily indicate causation. It is used in several contexts in business, finance, and economics. For instance, it is used to help investment managers value assets and understand the relationships between factors such as commodity prices and the stocks of businesses dealing in those commodities.

Regression as a statistical technique should not be confused with the concept of regression to the mean (mean reversion).

Linear Regression

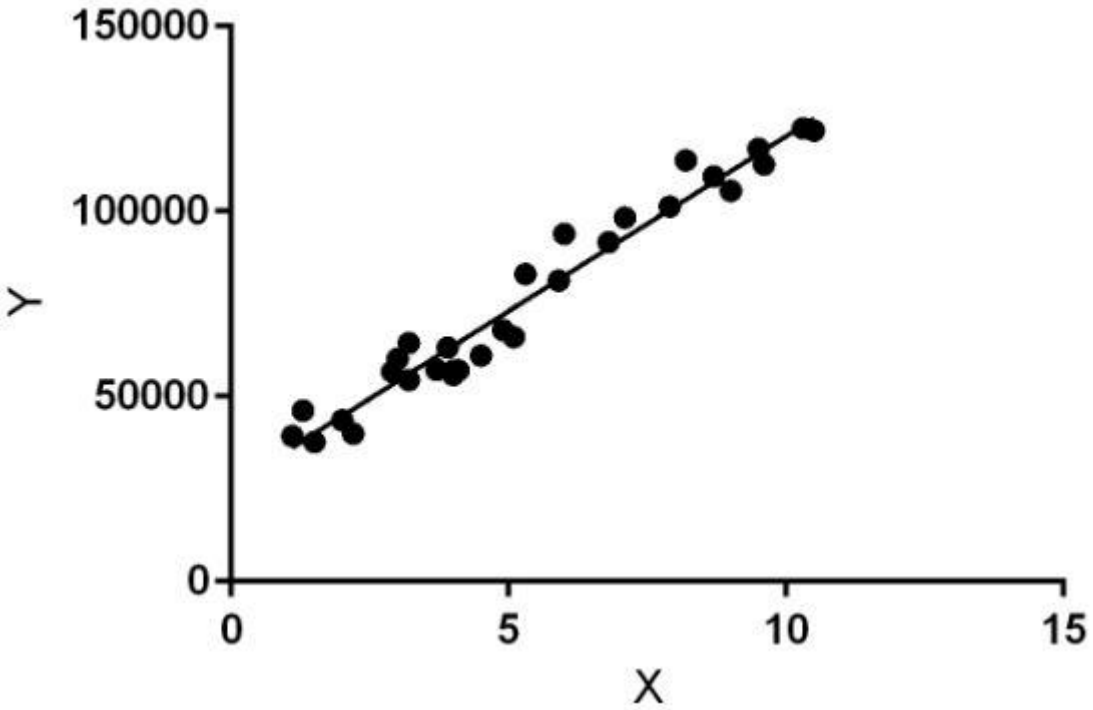
Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features. When the number of the independent feature, is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression. The goal of the algorithm is to find the best linear equation that can predict the value of the dependent variable based on the independent variables. The equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s).

Linear regression is used in many different fields, including finance, economics, and psychology, to understand and predict the behavior of a particular variable. For example, in finance, linear regression might be used to understand the relationship between a company's stock price and its

earnings or to predict the future value of a currency based on its past performance.

One of the most important supervised learning tasks is regression. In regression set of records are present with X and Y values and these values are used to learn a function so if you want to predict Y from an unknown X this learned function can be used. In regression we have to find the value of Y, So, a function is required that predicts continuous Y in the case of regression given X as independent features.

Here Y is called a dependent or target variable and X is called an independent variable also known as the predictor of Y. There are many types of functions or modules that can be used for regression. A linear function is the simplest type of function. Here, X may be a single feature or multiple features representing the problem.



Linear Regression

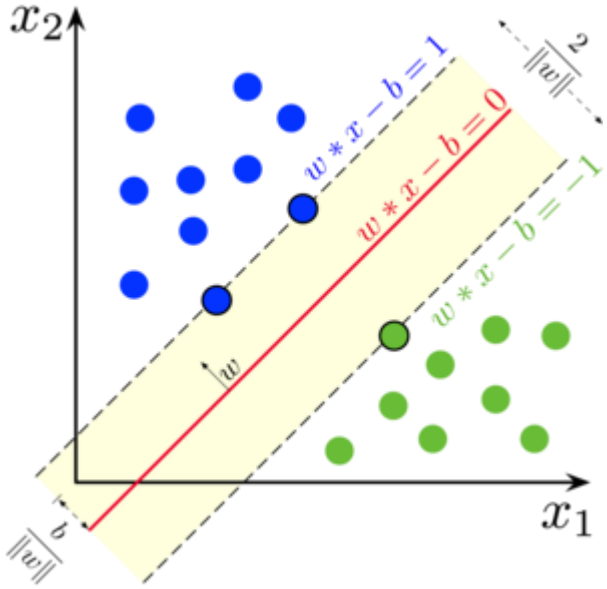
Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best-fit line for our model.

4 Support Vector Machine

In machine learning, **support vector machines** (**SVMs**, also **support vector networks**[1]) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik (1982, 1995) and Chervonenkis (1974). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Linear SVM



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

We are given a training dataset of n points of the form

$(x_1, y_1), \ldots\ldots\ldots, (x_n, y_n),$

where the y_i are either 1 or −1, each indicating the class to which the point x_i belongs. Each x_i is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points x_i for which $y_i=1$ from the group of points for which $y_i=-1$, which is defined so that the distance between the hyperplane and the nearest point x_i from either group is maximized.

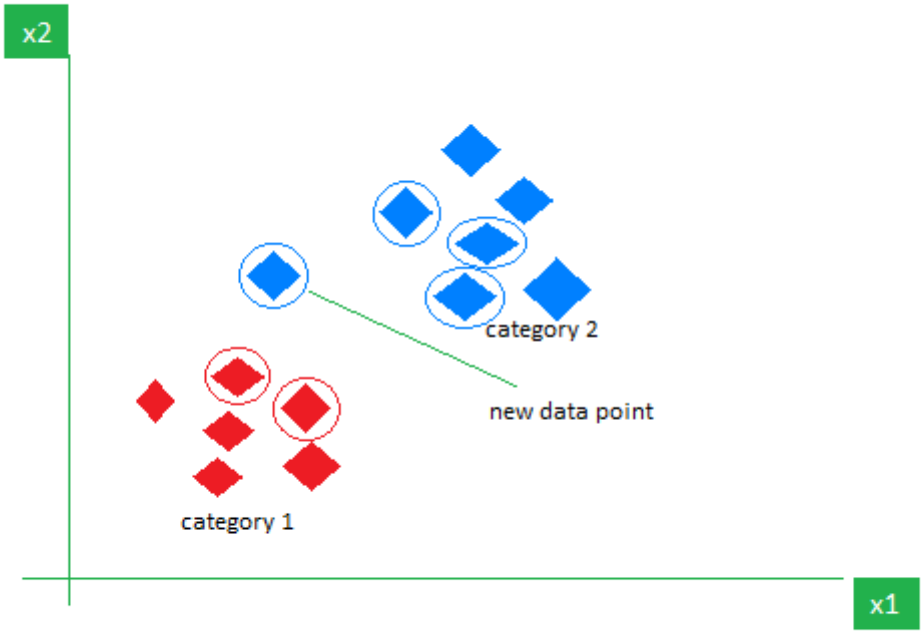
Any hyperplane can be written as the set of points x satisfying

$W^T X - b = 0$,

where w is the (not necessarily normalized) normal vector to the hyperplane. This is much like Hesse normal form, except that w is not necessarily a unit vector. The parameter $b/||w||$ determines the offset of the hyperplane from the origin along the normal vector w.

K-Nearest Neighbour

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.
It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.
As an example, consider the following table of data points containing two features:



KNN Algorithm working visualization

Now, given another set of data points (also called testing data), allocate these points to a group by analyzing the training set. Note that the unclassified points are marked as ‘White’.

Intuition Behind KNN Algorithm
If we plot these points on a graph, we may be able to locate some clusters or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbors belong to. This means a point close to a cluster of points classified as ‘Red’ has a higher probability of getting classified as ‘Red’.

Intuitively, we can see that the first point (2.5, 7) should be classified as ‘Green’ and the second point (5.5, 4.5) should be classified as ‘Red’.

Distance Metrics Used in KNN Algorithm

As we know that the KNN algorithm helps us identify the nearest points or the groups for a query point. But to determine the closest groups or the nearest points for a query point we need some metric. For this purpose, we use below distance metrics:

- Euclidean Distance
- Manhattan Distance
- Minkowski Distance

Euclidean Distance

This is nothing but the cartesian distance between the two points which are in the plane/hyperplane. Euclidean distance can also be visualized as the length of the straight line that joins the two points which are into consideration. This metric helps us calculate the net displacement done between the two states of an object.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattan Distance

This distance metric is generally used when we are interested in the total distance traveled by the object instead of the displacement. This metric is calculated by summing the absolute difference between the coordinates of the points in n-dimensions.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Minkowski Distance

We can say that the Euclidean, as well as the Manhattan distance, are special cases of the Minkowski distance.

$$d(x, y) = \left(\sum_{i=1}^{n} (x_i - y_i)^p\right)^{\frac{1}{p}}$$

From the formula above we can say that when p = 2 then it is the same as the formula for the Euclidean distance and when p = 1 then we obtain the formula for the Manhattan distance.

The above-discussed metrics are most common while dealing with a Machine Learning problem but there are other distance metrics as well like Hamming Distance which come in handy while dealing with problems that require overlapping

comparisons between two vectors whose contents can be boolean as well as string values.

Hidden Markov Model in Machine Learning

Hidden Markov Models (HMMs) are a type of probabilistic model that are commonly used in machine learning for tasks such as speech recognition, natural language processing, and bioinformatics. They are a popular choice for modelling sequences of data because they can effectively capture the underlying structure of the data, even when the data is noisy or incomplete. In this article, we will give a comprehensive overview of Hidden Markov Models, including their mathematical foundations, applications, and limitations.

What are Hidden Markov Models?

A Hidden Markov Model (HMM) is a probabilistic model that consists of a sequence of hidden states, each of which generates an observation. The hidden states are usually not directly observable, and the goal of HMM is to estimate the sequence of hidden states based on a sequence of observations. An HMM is defined by the following components:

- A set of N hidden states, S = {s1, s2, ..., sN}.
- A set of M observations, O = {o1, o2, ..., oM}.
- An initial state probability distribution, ? = {?1, ?2, ..., ?N}, which specifies the probability of starting in each hidden state.
- A transition probability matrix, A = [aij], defines the probability of moving from one hidden state to another.
- An emission probability matrix, B = [bjk], defines the probability of emitting an observation from a given hidden state.

The basic idea behind an HMM is that the hidden states generate the observations, and the observed data is used to estimate the hidden state sequence. This is often referred to as the **forward-backwards algorithm**. ey applications of HMMs, including speech recognition, natural language processing, bioinformatics, and finance.

Example

Drawing balls from hidden urns**[edit]**

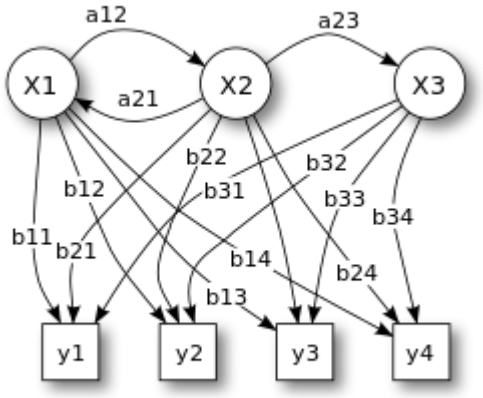


Figure 1. Probabilistic parameters of a hidden Markov model (example)
X — states
y — possible observations
a — state transition probabilities
b — output probabilities

In its discrete form, a hidden Markov process can be visualized as a generalization of the urn problem with replacement (where each item from the urn is returned to the original urn before the next step).[7] Consider this example: in a room that is not visible to an observer there is a genie. The room contains urns X1, X2, X3, ... each of which contains a known mix of balls, each ball labeled y1, y2, y3, ... The genie chooses an urn in that room and randomly draws a ball from that urn. It then puts the ball onto a conveyor belt, where the observer can observe the sequence of the balls but not the sequence of urns from which they were drawn. The genie has some procedure to choose urns; the choice of the urn for the *n*-th ball depends only upon a random number and the choice of the urn for the (*n* − 1)-th ball. The choice of urn does not directly depend on the urns chosen before this single previous urn; therefore, this is called a Markov process. It can be described by the upper part of Figure 1.

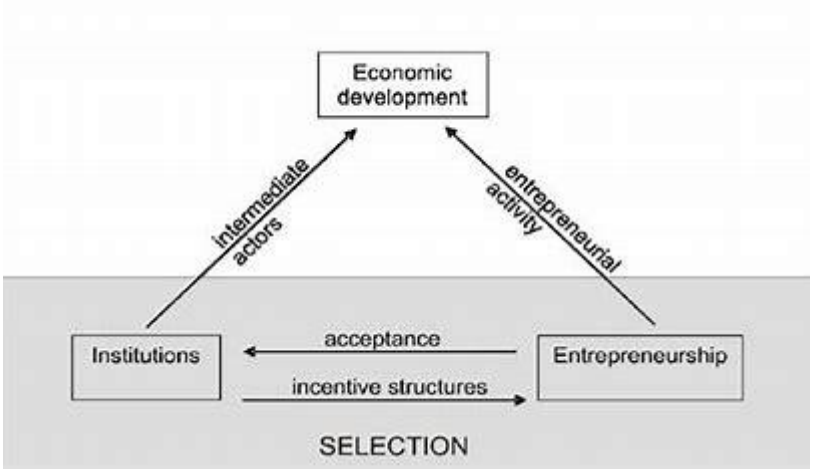
The Markov process itself cannot be observed, only the sequence of labeled balls, thus this arrangement is called a "hidden Markov process". This is illustrated by the lower part of the diagram shown in Figure 1, where one can see that balls y1, y2, y3, y4 can be drawn at each state. Even if the observer knows the composition of the urns and has just observed a sequence of three balls, *e.g.* y1, y2 and y3 on the conveyor belt, the observer still cannot be *sure* which urn (*i.e.*, at which state) the genie has drawn the third ball from. However, the observer can work out other information, such as the likelihood that the third ball came from each of the urns.

Summarization:

Summarization AI is a technology that uses artificial intelligence to create short summaries of large texts, audio or video files. There are different types of summarization AI, such as abstractive, extractive, or hybrid. Some summarization AI tools can also generate summaries in different styles, such as paragraphs, bullet points, or one-liners. Summarization AI can help you save time, improve comprehension, and reduce reading time.

Dependency modelling :

The term "dependency model" can refer to different concepts depending on the context. In the context of relationship breakup decisions, the dependence model **proposes that the primary issue in understanding breakup decisions is the degree of dependence on a relationship**[1]. In the context of economic underdevelopment, dependency theory is an approach that emphasizes the constraints imposed by the global political and economic order. It holds that underdevelopment is mainly caused by the peripheral position of affected countries in the world economy.



Link Analysis

Link analysis is a data mining technique that reveals the structure and content of a body of information by representing it as a set of interconnected, linked objects or entities. Often link analysis allows an investigator to identify association patterns, new emerging groups, and connections between suspects. Through the visualization of these entities and links, an investigator can gain an understanding of the strength of relationships and the frequency of contacts and discover new hidden associations. For this reason, link analysis is typically used by criminal investigators in such fields as fraud detection and money laundering, as well as by intelligence analysts in the study of terrorist networks. Link analysis is the first level of data mining. It is a manual interactive technique for forming and examining a visual network of relationships (see Figure 3.1).

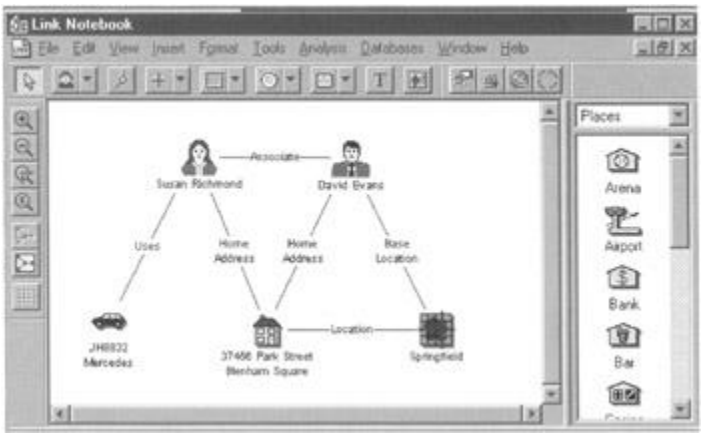


Figure 3.1: A financial link analysis network.

Link analysis begins with data that can be represented as a network and attempts to infer useful knowledge from the nodes and links of that network from which an investigator or analyst can discover associations. Many of the current link analysis tools are highly specialized, interactive graphical software—with some having the capability of incorporating multimedia and some interactive *what-if* scenarios. While these visual-link networks have proven useful to investigators, their manual construction has proven difficult when it involves hundred of thousands of transactions.

Linkage data is typically modeled as a graph with nodes representing suspects of interest to the analyst and the links representing relationships or transactions. Examples might be a collection of telephone toll data with phone numbers, times of calls, and durations of calls subpoenaed for a criminal investigation; a collection of cash transactions to and from certain domestic and foreign bank accounts; a collection of sightings of individuals' meetings and their addresses, trips to foreign countries, points of entry, wire transfers, schools or churches attended, Web sites visited, and other related commercial or social interactions. The events can be a few meetings or conversations or a large number of toll calls or bank deposits or withdrawals. However, if the observations are very voluminous, the value of link analysis will begin to deteriorate.

Sequential pattern mining:

Sequential pattern mining is a topic of data mining concerned with finding statistically relevant patterns between data examples where the values are delivered in a sequence. It is usually presumed that the values are discrete, and thus time series mining is closely related, but usually considered a different activity. Sequential pattern mining is a special case of structured data mining.

There are several key traditional computational problems addressed within this field. These include building efficient databases and indexes for sequence information, extracting the frequently occurring patterns, comparing sequences for similarity, and recovering missing sequence members. In general, sequence mining problems can be classified as *string mining* which is typically based on string processing algorithms and *itemset mining* which is typically based on association rule learning. *Local process models* [3] extend sequential pattern mining to more complex patterns that can include (exclusive) choices, loops, and concurrency constructs in addition to the sequential ordering construct.

social network analysis data mining:

Social network analysis (SNA) is a process of investigating social structures through the use of networks and graph theory [1]. It characterizes networked structures in terms of nodes (individual actors, people, or things within the network) and the ties, edges, or links (relationships or interactions) that connect them [1]. Social media networks, meme spread, information circulation, friendship and acquaintance networks, peer learner networks, business networks, knowledge networks, difficult working relationships, collaboration graphs, kinship, disease transmission, and sexual relationships are some examples of social structures commonly visualized through social network analysis .

Data mining is the process of collecting and processing data from a heap of unprocessed data. When the patterns are established, various relationships between the datasets can be identified and they can be presented in a summarized format which helps in statistical analysis in various industries . In data mining, graphs are used to find subgraph patterns for discrimination, classification, clustering of data, etc. The graph is used in network analysis. By linking the various nodes, graphs form network-like communications, web and computer networks, social networks, etc. In multi-relational data mining, graphs or networks are used because of the varied interconnected relationship between the datasets in a relational database .

The combination of SNA and data mining is often referred to as social network analysis and mining (SNAM). SNAM is a powerful tool to disclose relevant information hidden in large volumes of raw data . It has been applied to several research fields such as anthropology, biology, demography, communication studies, economics, geography, history, information science, organizational studies, political science, public health, social psychology, development studies, sociolinguistics and computer science [13]. Some commonly used social media data mining techniques include classification, association tracking patterns predictive analytics keyword extraction sentiment analysis and market/trend analysis .