UNIVERSITY OF GONDAR

COLLEGE OF INFORMATICS

DEPARTMENT OF INFORMATION SYSTEMS

INTRODUCTION TO MACHINE LEARNING MODULE

Prepared By:

Dr.Ibraih Gashew and Mr.Setegn Asnakew

Table of Contents

INTRODUCTION TO MACHINE LEARNING MODULE

CHAPTER ONE

1. Introduction

We are living in the 'age of data' that is enriched with better computational power and more storage resources. This data or information is increasing day by day, but the real challenge is to make sense of all the data. Businesses & organizations are trying to deal with it by building intelligent systems using the concepts and methodologies from Data science, Data Mining and Machine learning. Among them, machine learning is the most exciting field of computer science [1]. It would not be wrong if we call machine learning the application and science of algorithms that provides sense to the data.

An "intelligent" computer uses AI to think like a human and perform tasks on its own. Machine learning is how a computer system develops its intelligence. One way to train a computer to mimic human reasoning is to use a neural network, which is a series of algorithms that are modeled after the human brain. Artificial intelligence is a technology using which we can create intelligent systems that can simulate human intelligence but Machine learning is a subfield of artificial intelligence, which enables machines to learn from past data or experiences without being explicitly programmed. Machine learning enables a computer system to make predictions or take some decisions using historical data without being explicitly programmed. Machine learning uses a massive amount of structured and semi-structured data so that a machine learning model can generate accurate result or give predictions based on that data.

1.1 What is machine learning?

Machine Learning (ML) is that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do [1]. In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method. The main focus of ML is to allow computer systems learn from experience without being explicitly programmed or human intervention. Furthermore, Machine learning (ML) is the scientific study

of algorithms and statistical models that computer systems use to perform a specific task without being explicitly programmed.

In addition to the above definition of machine learning, machine learning is defined as a technology that is used to train machines to perform various actions such as predictions, recommendations, estimations, etc., based on historical data or past experience.

Machine Learning enables computers to behave like human beings by training them with the help of past experience and predicted data. There are three key aspects of Machine Learning, which are as follows:

- **Task:** A task is defined as the main problem in which we are interested. This task/problem can be related to the predictions and recommendations and estimations, etc.
- **Experience:** It is defined as learning from historical or past data and used to estimate and resolve future tasks.
- **Performance:** It is defined as the capacity of any machine to resolve any machine learning task or problem and provide the best outcome for the same. However, performance is dependent on the type of machine learning problems.

1.2 Foundation of machine learning?

The foundations of machine learning is a research area within Georgia Tech's School of Computer Science (SCS) that focuses on the development of algorithms that leverage data and statistical tools to solve complex human tasks, to explore novel applications of such tools, and to better understand the apparent success of a machine [2].

1.3 History and relationships to other fields

Machine learning is an area related to both cybernetics and computer science (or Control Science and Computer Science) attracting recently an overwhelming interest both of professionals and of the general public [3]. In the last few years thanks to successes of computer science (the emergence of GPUs, leading to significant improvements in the performance of computers and development of special software, allowing to work with big data) machine learning is often attributed to computer science. However, historically, learning algorithms that provide convergence and sufficient convergence rate of the learning process arose within cybernetics/control.

As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. Already in the early days of AI as an academic discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what were then termed "neural networks"; these were mostly perceptrons and other models that were later found to be reinventions of the generalized linear models of statistics. Probabilistic reasoning was also employed, especially in automated medical diagnosis.

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning [4]. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation. By 1980, expert systems had come to dominate AI, and statistics was out of favor. Work on symbolic/knowledge-based learning did continue within AI, leading to inductive logic programming, but the more statistical line of research was now outside the field of AI proper, in pattern recognition and information retrieval. Neural networks research had been abandoned by AI and computer science around the same time. This line, too, was continued outside the AI/CS field, as "connectionism", by researchers from other disciplines including Hopfield, Rumelhart and Hinton. Their main success came in the mid-1980s with the reinvention of back propagation.

Machine learning and statistics are closely related fields [4]. According to Michael I. Jordan, the ideas of machine learning, from methodological principles to theoretical tools, have had a long pre-history in statistics. He also suggested the term data science as a placeholder to call the overall field. Leo Breiman distinguished two statistical modelling paradigms: data model and algorithmic model, wherein 'algorithmic model' means more or less the machine learning algorithms like Random forest. Some statisticians have adopted methods from machine learning, leading to a combined field that they call statistical learning

1.4 Applications of machine learning

Machine Learning is widely being used in approximately every sector, including healthcare, marketing, finance, infrastructure, automation, etc. There are some important real-world examples of machine learning, which are as follows:

- **Healthcare and Medical Diagnosis:** Machine Learning is used in healthcare industries that help in generating neural networks. These self-learning neural networks help

specialists for providing quality treatment by analyzing external data on a patient's condition, X-rays, CT scans, various tests, and screenings. Other than treatment, machine learning is also helpful for cases like automatic billing, clinical decision supports, and development of clinical care guidelines, etc.

- **Marketing:** Machine learning helps marketers to create various hypotheses, testing, evaluation, and analyze datasets. It helps us to quickly make predictions based on the concept of big data. It is also helpful for stock marketing as most of the trading is done through bots and based on calculations from machine learning algorithms. Various Deep Learning Neural network helps to build trading models such as Convolutional Neural Network, Recurrent Neural Network, Long-short term memory, etc.
- **Self-driving cars:** This is one of the most exciting applications of machine learning in today's world. It plays a vital role in developing self-driving cars. Various automobile companies like Tesla, Tata, etc., are continuously working for the development of self-driving cars. It also becomes possible by the machine learning method (supervised learning), in which a machine is trained to detect people and objects while driving.
- **Speech Recognition:** Speech Recognition is one of the most popular applications of machine learning. Nowadays, almost every mobile application comes with a voice search facility. This "Search by Voice" facility is also a part of speech recognition. In this method, voice instructions are converted into text, which is known as Speech to text" or "Computer speech recognition.
- **Traffic Prediction:** Machine Learning also helps us to find the shortest route to reach our destination by using Google Maps. It also helps us in predicting traffic conditions, whether it is cleared or congested, through the real-time location of the Google Maps app and sensor.
- **Image Recognition:** Image recognition is also an important application of machine learning for identifying objects, persons, places, etc. Face detection and auto friend tagging suggestion is the most famous application of image recognition used by Facebook, Instagram, etc. Whenever we upload photos with our Facebook friends, it automatically suggests their names through image recognition technology.
- **Product Recommendations:** Machine Learning is widely used in business industries for the marketing of various products. Almost all big and small companies like Amazon,

Alibaba, Walmart, Netflix, etc., are using machine learning techniques for products recommendation to their users. Whenever we search for any products on their websites, we automatically get started with lots of advertisements for similar products. This is also possible by Machine Learning algorithms that learn users' interests and, based on past data, suggest products to the user.

- **Automatic Translation:** Automatic language translation is also one of the most significant applications of machine learning that is based on sequence algorithms by translating text of one language into other desirable languages. Google GNMT (Google Neural Machine Translation) provides this feature, which is Neural Machine Learning. Further, you can also translate the selected text on images as well as complete documents through Google Lens.
- **Virtual Assistant:** A virtual personal assistant is also one of the most popular applications of machine learning. First, it records out voice and sends to cloud-based server then decode it with the help of machine learning algorithms. All big companies like Amazon, Google, etc., are using these features for playing music, calling someone, opening an app and searching data on the internet, etc.
- **Email Spam and Malware Filtering:** Machine Learning also helps us to filter various Emails received on our mailbox according to their category, such as important, normal, and spam. It is possible by ML algorithms such as Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier.

1.5 Types of machine learning techniques

Machine learning tasks are typically classified into three broad categories, depending on the nature of the learning "signal" or "feedback" available to a learning system [4]. These are:

- **Supervised learning:** The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. Supervised learning is applicable when a machine has sample data, i.e., input as well as output data with correct labels. Correct labels are used to check the correctness of the model using some labels and tags. Supervised learning technique helps us to predict future events with the help of past experience and labeled examples. Initially, it analyses the known training dataset, and later it introduces an inferred function that makes

predictions about output values. Further, it also predicts errors during this entire learning process and also corrects those errors through algorithms. Example: Let's assume we have a set of images tagged as "dog". A machine learning algorithm is trained with these dog images so it can easily distinguish whether an image is a dog or not.

- **Unsupervised learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end. Unsupervised learning, a machine is trained with some input samples or labels only, while output is not known. The training information is neither classified nor labeled; hence, a machine may not always provide correct output compared to supervise learning. Although unsupervised learning is less common in practical business settings, it helps in exploring the data and can draw inferences from datasets to describe hidden structures from unlabeled data. Example: Let's assume a machine is trained with some set of documents having different categories (Type A, B, and C), and we have to organize them into appropriate groups. Because the machine is provided only with input samples or without output, so, it can organize these datasets into type A, type B, and type C categories, but it is not necessary whether it is organized correctly or not.
- **Reinforcement learning:** A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle), without a teacher explicitly telling it whether it has come close to its goal or not. Another example is learning to play a game by playing against an opponent. Reinforcement Learning is a feedback-based machine learning technique. In such type of learning, agents (computer programs) need to explore the environment, perform actions, and on the basis of their actions, they get rewards as feedback. For each good action, they get a positive reward, and for each bad action, they get a negative reward. The goal of a Reinforcement learning agent is to maximize the positive rewards. Since there is no labeled data, the agent is bound to learn by its experience only.
- **Semi-supervised Learning:** Semi-supervised Learning is an intermediate technique of both supervised and unsupervised learning. It performs actions on datasets having few labels as well as unlabeled data. However, it generally contains unlabeled data. Hence, it also reduces the cost of the machine learning model as labels are costly, but for corporate

purposes, it may have few labels. Further, it also increases the accuracy and performance of the machine learning model. Sem-supervised learning helps data scientists to overcome the drawback of supervised and unsupervised learning. Speech analysis, web content classification, protein sequence classification, text documents classifiers etc., are some important applications of Semi-supervised learning.

1.6. Overview of Data mining and KDD process:

In an increasingly data-driven world, there would seem to never be such a thing as too much data. However, data is only valuable when you can parse, sort, and sift through it in order to extrapolate the actual value. Most industries collect massive volumes of data, but without a filtering mechanism that graphs, charts, and trends data models, pure data itself has little use. However, the sheer volume of data and the speed with which it is collected makes sifting through it challenging. Thus, it has become economically and scientifically necessary to scale up our analysis capability to handle the vast amount of data that we now obtain. Since computers have allowed humans to collect more data than we can process, we naturally turn to computational techniques to help us extract meaningful patterns and structures from vast amounts of data.

Machine learning and data mining often employ the same methods and overlap significantly. They can be roughly distinguished as follows:

- Machine learning focuses on prediction, based on known properties learned from the training data.
- Data mining focuses on the discovery of (previously) unknown properties in the data. This is the analysis step of Knowledge Discovery in Databases.

Knowledge Discovery in Databases (KDD) is the non-trivial extraction of implicit, previously unknown and potentially useful knowledge from data. Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data.

1.7. Prediction vs. Description modeling

A descriptive model will exploit the past data that are stored in databases and provide you with the accurate report. In a Predictive model, it identifies patterns found in past and transactional data to find risks and future outcomes.

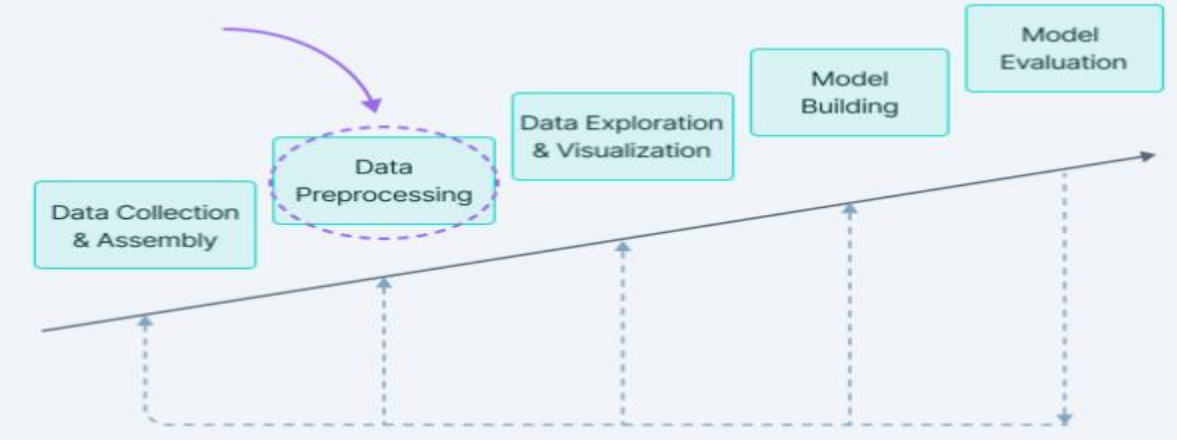
The descriptive and predictive data mining techniques have huge applications in data mining; they are used to mine the types of patterns. The descriptive analysis is used to mine data and specify the current data on past events. In contrast, the predictive analysis gives the answers to all queries related to recent or previous data that move across using historical data as the main principle for decision. The task of data mining can be predictive, descriptive and prescriptive. In this article, we will discuss the two terms, predictive data mining and descriptive data mining, separately. In laymen language, you can say that descriptive mining involves finding interesting patterns or associations relating to data. In contrast, predictive mining involves the prediction and classification of the data gathered in past or current. Read the article to learn the difference between descriptive and predictive data mining.

- **Descriptive Data Mining**: Descriptive mining is generally used to provide correlation, cross-tabulation, frequency, etc. These methods are used to decide the data's regularities and to reveal patterns. It focuses on the summarization and conversion of records into significant data for reporting and monitoring. Descriptive mining "describes" the data. Once the data is captured, it can modify it into human interpretable form. In descriptive data mining, an association technique that uses Apriori algorithms to characterize student performance to find co-relations between a set of items. The Apriori algorithm is used in the database including academic records of several students and tries to extract association rules to profile students based on several parameters such as exam scores, term work grades, attendance, and practical.
- **Predictive Data Mining:** The term 'Predictive' defines to predict something, so predictive data mining is the analysis done to predict the future event or multiple data or trends. Predictive data mining can allow business analysts to create decisions and insert a value into the analytics team efforts. Predictive data mining provides predictive analytics. In predictive analytics, it is the use of data to predict outcomes. The main goal of predictive mining is to predict future results rather than current behavior. It includes the supervised learning services used for the prediction of the focus value. The approaches that fall under this mining element are classification, time-sequence analysis, and regression. Data modeling is the fundamental of predictive analysis, which works by using some variables to anticipate the unknown future data values for other variables.

CHAPTER TWO

2. Data preprocessing

Data Preprocessing includes the steps we need to follow to transform or encode data so that it may be easily parsed by the machine [5]. The main agenda for a model to be accurate and precise in predictions is that the algorithm should be able to easily interpret the data's features.



2.1 Why preprocess the data?

The majority of the real-world datasets for machine learning are highly susceptible to be missing, inconsistent, and noisy due to their heterogeneous origin. Applying data mining algorithms on this noisy data would not give quality results as they would fail to identify patterns effectively. Data Processing is, therefore, important to improve the overall data quality. Duplicate or missing values may give an incorrect view of the overall statistics of data. Outliers and inconsistent data points often tend to disturb the model's overall learning, leading to false predictions. Quality decisions must be based on quality data. Data Preprocessing is important to get this quality data, without which it would just be a Garbage In, Garbage Out scenario. In general, real world data are generally Incomplete (lacking attribute values, lacking certain attributes of interest, or containing only aggregate data), Noisy (containing errors or outliers), and Inconsistent: (containing discrepancies in codes or names)

- **Features in machine learning**: Individual independent variables that operate as an input in our machine learning model are referred to as features. They can be thought of as representations or attributes that describe the data and help the models to predict the classes/labels.

2.2 Major Tasks in Data Preprocessing

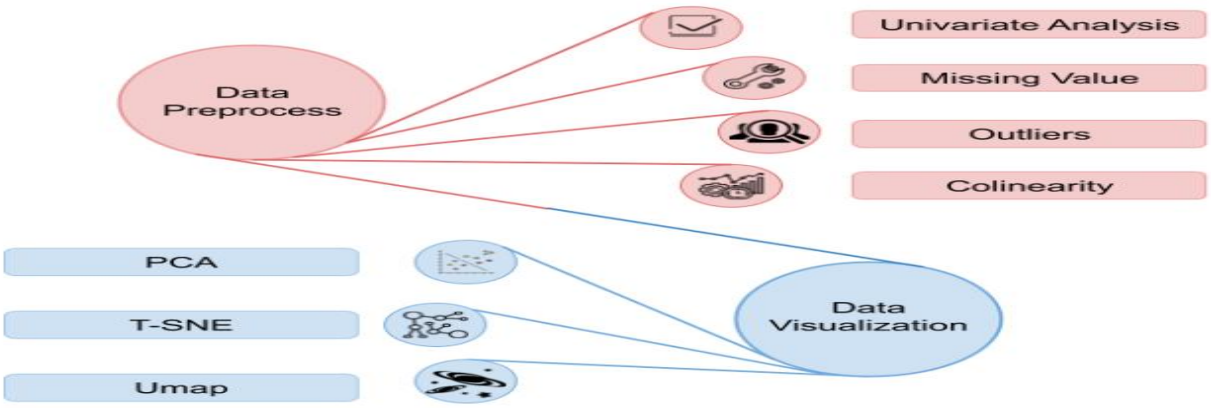
Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms. The major tasks in data preprocessing are data cleaning, data integration, data understanding, data exploration, data reduction and data transformation.

- **Data cleaning** is a process to clean the data in such a way that data can be easily integrated.
- **Data integration** is a process to integrate/combine all the data.
- **Data reduction** is a process to reduce the large data into smaller once in such a way that data can be easily transformed further.
- **Data transformation** is a process to transform the data into a reliable shape.
- **Data discretization** converts a large number of data values into smaller once, so that data evaluation and data management becomes very easy.

2.2.1 Data Exploration

Data exploration, also known as exploratory data analysis (EDA), is a process where users look at and understand their data with statistical and visualization methods. This step helps identifying patterns and problems in the dataset, as well as deciding which model or algorithm to use in subsequent steps.

Data exploration can be divided into data preprocessing and data visualization. For data preprocessing, we focus on four methods: univariate analysis, missing value treatment, outlier treatment, and collinearity treatment. For data visualization, we discuss dimensionality reduction methods including PCA, T-SNE, and UMAP as shown belowfigure.



2.2.2 Data understanding

Quality data is fundamental to any data science engagement. To gain actionable insights, the appropriate data must be sourced and cleansed. There are two key stages of Data Understanding: a Data Assessment which is evaluates what data is available and how it aligns to the business problem and Data Exploration which is data preprocessing and data visualization.

2.2.3 Data cleaning and reduction

Data Cleaning is particularly done as part of data preprocessing to clean the data by filling missing values, smoothing the noisy data, resolving the inconsistency, and removing outliers.

- **Missing values:** Here are a few ways to solve this issue:

  Ignore those tuples: This method should be considered when the dataset is huge and numerous missing values are present within a tuple.

  Fill in the missing values: There are many methods to achieve this, such as filling in the values manually, predicting the missing values using regression method, or numerical methods like attribute mean.

- **Noisy Data:** It involves removing a random error or variance in a measured variable. It can be done with the help of the following techniques:
  - ✓ **Binning**: It is the technique that works on sorted data values to smoothen any noise present in it. The data is divided into equal-sized bins, and each bin/bucket is dealt with independently. All data in a segment can be replaced by its mean, median or boundary values.
  - ✓ **Regression**: This data mining technique is generally used for prediction. It helps to smoothen noise by fitting all the data points in a regression function. The linear regression equation is used if there is only one independent attribute; else Polynomial equations are used.

  Clustering: Creation of groups/clusters from data having similar values. The values that don't lie in the cluster can be treated as noisy data and can be removed.

- **Removing outliers:** Clustering techniques group together similar data points. The tuples that lie outside the cluster are outliers/inconsistent data.

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data. That is, mining on

the reduced data set should be more efficient yet produce the same (or almost the same) analytical results. In this section, we first present an overview of data reduction strategies, followed by a closer look at individual techniques. Data reduction strategies include dimensionality reduction, numerosity reduction, and data compression.

- **Dimensionality reduction:** This process is necessary for real-world applications as the data size is big. In this process, the reduction of random variables or attributes is done so that the dimensionality of the data set can be reduced. Combining and merging the attributes of the data without losing its original characteristics. This also helps in the reduction of storage space and computation time is reduced. When the data is highly dimensional the problem called "Curse of Dimensionality" occurs.
- **Numerosity Reduction:** In this method, the representation of the data is made smaller by reducing the volume. There will not be any loss of data in this reduction.
- **Data compression:** The compressed form of data is called data compression. This compression can be lossless or lossy. When there is no loss of information during compression it is called lossless compression. Whereas lossy compression reduces information but it removes only the unnecessary information.

2.2.4 Data Integration and Transformation

Data Integration is one of the data preprocessing steps that are used to merge the data present in multiple sources into a single larger data store like a data warehouse.

Data Integration: it is needed especially when we are aiming to solve a real-world scenario like detecting the presence of nodules from CT scan images. The only option is to integrate the images from multiple medical nodes to form a larger database. We might run into some issues while adopting Data Integration as one of the Data Preprocessing steps:

- Schema integration and object matching: The data can be present in different formats, and attributes that might cause difficulty in data integration.
- Removing redundant attributes from all data sources.
- Detection and resolution of data value conflicts.

Data Transformation: Once data clearing has been done, we need to consolidate the quality data into alternate forms by changing the value, structure, or format of data using the below-mentioned Data Transformation strategies.

- **Generalization:** The low-level or granular data that we have converted to high-level information by using concept hierarchies. We can transform the primitive data in the address like the city to higher-level information like the country.
- **Normalization:** It is the most important Data Transformation technique widely used. The numerical attributes are scaled up or down to fit within a specified range. In this approach, we are constraining our data attribute to a particular container to develop a correlation among different data points. Normalization can be done in multiple ways, which are highlighted here:
    - ✓ Min-max normalization
    - ✓ Z-Score normalization
    - ✓ Decimal scaling normalization

Attribute Selection: New properties of data are created from existing attributes to help in the data mining process. For example, date of birth, data attribute can be transformed to another property like is_senior_citizen for each tuple, which will directly influence predicting diseases or chances of survival, etc.

Aggregation: It is a method of storing and presenting data in a summary format. For example sales, data can be aggregated and transformed to show as per month and year format.

2.2.5 Discretization and concept hierarchy generation

Data discretization transforms numeric data by mapping values to interval or concept labels. Such methods can be used to automatically generate concept hierarchies for the data, which allows for mining at multiple levels of granularity. Discretization techniques include binning, histogram analysis, cluster analysis, decision-tree analysis, and correlation analysis. For nominal data, concept hierarchies may be generated based on schema definitions as well as the number of distinct values per attribute.

Data discretization is used to divide the attributes of the continuous nature into data with intervals. This is done because continuous features tend to have a smaller chance of correlation with the target variable. Thus, it may be harder to interpret the results. After discretizing a variable, groups corresponding to the target can be interpreted. For example, attribute age can be discretized into bins like below 18, 18-44, 44-60, above 60.

CHAPTER THREE

3. Classification and prediction

Classification is the process of identifying which category a new observation belongs to base on a training data set containing observations whose category membership is known. Predication is the process of identifying the missing or unavailable numerical data for a new observation [6]. There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends [6]. These two forms are as follows:

- **Classification**
- **Prediction**

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

3.1 What is classification? What is prediction?

Classification:

Classification is to identify the category or the class label of a new observation. First, a set of data is used as training data. The set of input data and the corresponding outputs are given to the algorithm. So, the training data set includes the input data and their associated class labels. Using the training dataset, the algorithm derives a model or the classifier. The derived model can be a decision tree, mathematical formula, or a neural network. In classification, when unlabeled data is given to the model, it should find the class to which it belongs. The new data provided to the model is the test data set.

- ✓ predicts categorical class labels (discrete or nominal)
- ✓ Classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data

There are two stapes in classification which are Model construction (describing a set of predetermined classes) and Model usage (for classifying future or unknown objects) [6].

Prediction:

Another process of data analysis is prediction. It is used to find a numerical output. Same as in classification, the training dataset contains the inputs and corresponding numerical output values. The algorithm derives the model or a predictor according to the training dataset. The model should find a numerical output when the new data is given. Unlike in classification, this method does not have a class label. The model predicts a continuous-valued function or ordered value. Regression is generally used for prediction. Predicting the value of a house depending on the facts such as the number of rooms, the total area, etc., is an example for prediction.

For example, suppose the marketing manager needs to predict how much a particular customer will spend at his company during a sale. We are bothered to forecast a numerical value in this case. Therefore, an example of numeric prediction is the data processing activity. In this case, a model or a predictor will be developed that forecasts a continuous or ordered value function.

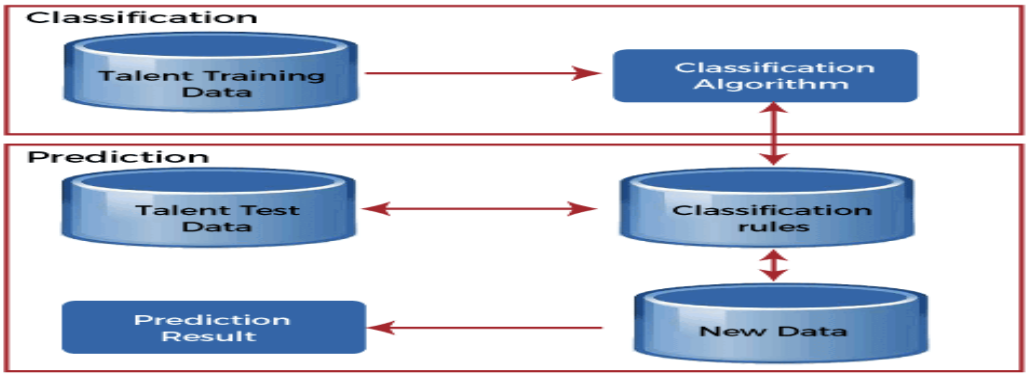
- ✓ Models continuous-valued functions, i.e., predicts unknown or missing values

Typical applications of classification and prediction

- ✓ Credit/loan approval, Medical diagnosis: if a tumor is cancerous or being, Fraud detection: if a transaction is fraudulent and Web page categorization: which category it is

Difference between Classification and Prediction

The decision tree, applied to existing data, is a classification model. We can get a class prediction by applying it to new data for which the class is unknown. The assumption is that the new data comes from a distribution similar to the data we used to construct our decision tree. In many instances, this is a correct assumption, so we can use the decision tree to build a predictive model. Classification of prediction is the process of finding a model that describes the classes or concepts of information. The purpose is to predict the class of objects whose class label is unknown using this model. Below are some major differences between classification and prediction.



Classification	Prediction
Classification is the process of identifying which category a new observation belongs to based on a training data set containing observations whose category membership is known.	Predication is the process of identifying the missing or unavailable numerical data for a new observation.
In classification, the accuracy depends on finding the class label correctly.	In prediction, the accuracy depends on how well a given predictor can guess the value of a predicated attribute for new data.
In classification, the model can be known as the classifier.	In prediction, the model can be known as the predictor.
A model or the classifier is constructed to find the categorical labels.	A model or a predictor will be constructed that predicts a continuous-valued function or ordered value.
For example, the grouping of patients based on their medical records can be considered a classification.	**For example**, We can think of prediction as predicting the correct treatment for a particular disease for a person.

3.2 Issues regarding classification and prediction

The major issue is preparing the data for Classification and Prediction. Preparing the data involves the following activities:

- **Data Cleaning:** Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.
- **Relevance Analysis:** Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- **Data Transformation and reduction:** The data can be transformed by any of the following methods.

- ✓ **Normalization:** The data is transformed using normalization. Normalization involves scaling all values for given attribute in order to make them fall within a small specified range. Normalization is used when in the learning step, the neural networks or the methods involving measurements are used.
- ✓ **Generalization:** The data can also be transformed by generalizing it to the higher concept. For this purpose we can use the concept hierarchies.

3.3 Classification by decision tree induction

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

3.4 Bayesian classification

Thomas Bayes, who proposed the Bayes Theorem so, it named Bayesian theorem.

- It is statistical method & supervised learning method for classification.
- It can solve problems involving both categorical and continuous valued attributes.
- Bayesian classification is used to find conditional probabilities.

In numerous applications, the connection between the attribute set and the class variable is non-deterministic. In other words, we can say the class label of a test record cant be assumed with certainty even though its attribute set is the same as some of the training examples. These

circumstances may emerge due to the noisy data or the presence of certain confusing factors that influence classification, but it is not included in the analysis. For example, consider the task of predicting the occurrence of whether an individual is at risk for liver illness based on individuals eating habits and working efficiency. Although most people who eat healthly and exercise consistently having less probability of occurrence of liver disease, they may still do so due to other factors. For example, due to consumption of the high-calorie street foods and alcohol abuse. Determining whether an individual's eating routine is healthy or the workout efficiency is sufficient is also subject to analysis, which in turn may introduce vulnerabilities into the leaning issue.

Bayesian classification uses Bayes theorem to predict the occurrence of any event. Bayesian classifiers are the statistical classifiers with the Bayesian probability understandings. The theory expresses how a level of belief, expressed as a probability. Bayes theorem came into existence after Thomas Bayes, who first utilized conditional probability to provide an algorithm that uses evidence to calculate limits on an unknown parameter. Bayes's theorem is expressed mathematically by the following equation that is given below.

Here X and Y are the events and P (Y) $\neq$ 0

P(X/Y) is a **conditional probability** that describes the occurrence of event **X** is given that **Y** is true.

P(Y/X) is a **conditional probability** that describes the occurrence of event **Y** is given that **X** is true.

P(X) and P(Y) are the probabilities of observing X and Y independently of each other. This is known as the **marginal probability**.

Bayesian interpretation:

In the Bayesian interpretation, probability determines a "**degree of belief**." Bayes theorem connects the degree of belief in a hypothesis before and after accounting for evidence. For example, lets us consider an example of the coin. If we toss a coin, then we get either heads or tails, and the percent of occurrence of either heads or tails is 50%. If the coin is flipped numbers of times, and the outcomes are observed, the degree of belief may rise, fall, or remain the same depending on the outcomes.

For proposition X and evidence Y,

- P(X), the prior, is the primary degree of belief in X

- P(X/Y), the posterior is the degree of belief having accounted for Y.
- The quotient $\frac{P(Y/X)}{P(Y)}$ represents the supports Y provides for X.

Bayes theorem can be derived from the conditional probability:

$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

Where P (X∩Y) is the **joint probability** of both X and Y being true, because

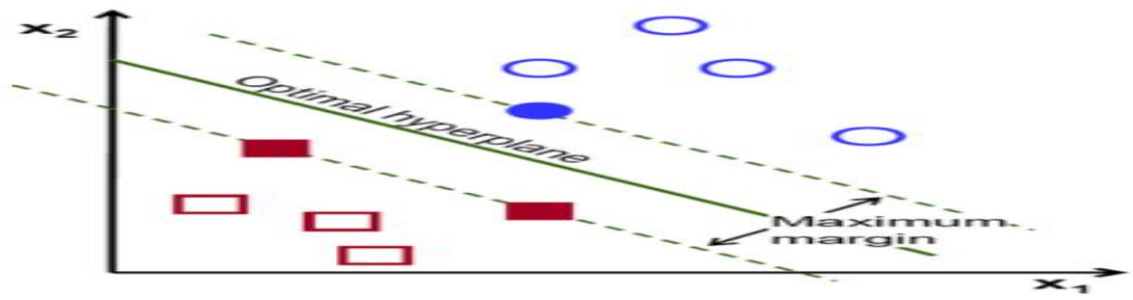
$$P(Y \cap X) = P(X \cap Y)$$

$$\text{or, } P(X \cap Y) = P(X/Y)P(Y) = P(Y/X)P(X)$$

$$\text{or, } P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}, \text{ if } P(Y) \neq 0$$

3.5 Support vector machines

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text. Compared to newer algorithms like neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands). This makes the algorithm very suitable for text classification problems, where it's common to have access to a dataset of at most a couple of thousands of tagged samples. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N- the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.



3.6 Classification by back propagation

Back propagation is a widely used algorithm for training feed forward neural networks. It computes the gradient of the loss function with respect to the network weights. It is very efficient, rather than naively directly computing the gradient concerning each weight. This efficiency makes it possible to use gradient methods to train multi-layer networks and update weights to minimize loss; variants such as gradient descent or stochastic gradient descent are often used.

Furthermore, back propagation is an algorithm that back propagates the errors from the output nodes to the input nodes. Therefore, it is simply referred to as the backward propagation of errors. It uses in the vast applications of neural networks in data mining like Character recognition, Signature verification, etc. The back propagation algorithm works by computing the gradient of the loss function with respect to each weight via the chain rule, computing the gradient layer by layer, and iterating backward from the last layer to avoid redundant computation of intermediate terms in the chain rule.

Features of Back propagation: it is the gradient descent method as used in the case of simple perceptron network with the differentiable unit. It is different from other networks in respect to the process by which the weights are calculated during the learning period of the network.

- training is done in the three stages :
- the feed-forward of input training pattern
- the calculation and back propagation of the error
- updating of the weight

Working of Back propagation: Neural networks use supervised learning to generate output vectors from input vectors that the network operates on. It Compares generated output to the desired output and generates an error report if the result does not match the generated output vector. Then it adjusts the weights according to the bug report to get your desired output.

Back propagation Algorithm:

Step 1: Inputs X, arrive through the preconnected path.

Step 2: The input is modeled using true weights W. Weights are usually chosen randomly.

Step 3: Calculate the output of each neuron from the input layer to the hidden layer to the output layer.

Step 4: Calculate the error in the outputs

Back propagation Error= Actual Output – Desired Output

Step 5: From the output layer, go back to the hidden layer to adjust the weights to reduce the error.

Step 6: Repeat the process until the desired output is achieved.

3.7 Other classification methods

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.

3.7.1 K-nearest neighbor classifier

The k-nearest neighbor's classifier (kNN) is a non-parametric supervised machine learning algorithm. It's distance-based: it classifies objects based on their proximate neighbors' classes. kNN is most often used for classification, but can be applied to regression problems as well.

Non-parametric means that there is no fine-tuning of parameters in the training step of the model. Although k can be considered an algorithm parameter in some sense, it's actually a hyper parameter. It's selected manually and remains fixed at both training and inference time.

The k-nearest neighbor's algorithm is also non-linear. In contrast to simpler models like linear regression, it will work well with data in which the relationship between the independent variable (x) and the dependent variable (y) is not a straight line.

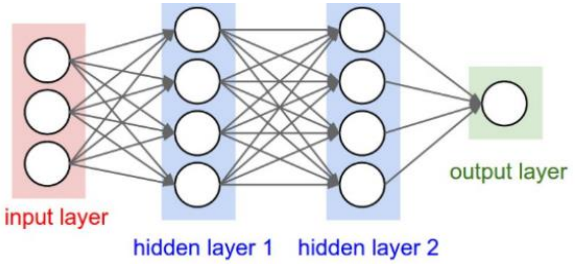
The parameter k in kNN refers to the number of labeled points (neighbors) considered for classification. The value of k indicates the number of these points used to determine the result. Our task is to calculate the distance and identify which categories are closest to our unknown entity

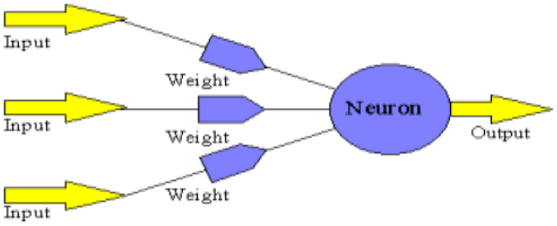
3.7.2 Neural Network

The Artificial Neural Network (ANN) bases its assimilation of data on the way that the human brain processes information [7] [8]. The brain has billions of cells called neurons that process

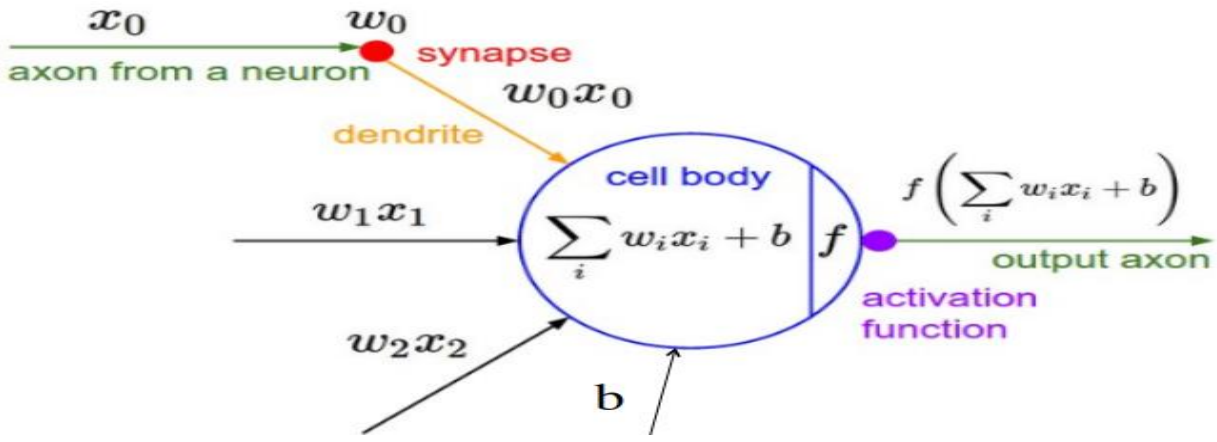
information in the form of electric signals. External information, or stimuli, is received, after which the brain processes it, and then produces an output. Additionally, a neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning that uses interconnected nodes or neurons in a layered structure that resembles the human brain. Neural computing requires a number of neurons, to be connected together into a "neural network". Neurons are arranged in layers [9].

A Neural Network is a function. It (generally) comprised of: Neurons which pass input values through functions and output the result and Weights which carry values between neurons. We group neurons into layers. There are 3 main types of layers: Input Layer, Hidden Layer(s) and Output Layer





Each neuron within the network is usually a simple processing unit which takes one or more inputs and produces an output. At each neuron, every input has an associated "weight" which modifies the strength of each input. The neuron simply adds together all the inputs and calculates an output to be passed on. Bellow diagram shows that the mathematical model of the neural in neural network.



3.7.3 Genetic algorithm

Genetic algorithm in machine learning is mainly adaptive heuristic or search engine algorithms that provide solutions for search and optimization problems in machine learning. It is a methodology that solves unconstrained and constrained optimization problems based on natural selection. Besides, a genetic algorithm is a search-based algorithm used for solving optimization problems in machine learning. This algorithm is important because it solves difficult problems that would take a long time to solve. It has been used in various real-life applications such as data centers, electronic circuit design, code-breaking, image processing, and artificial creativity.

Genetic algorithms are applied in the following fields:

- **Transport:** Genetic algorithms are used in the traveling salesman problem to develop transport plans that reduce the cost of travel and the time taken. They are also used to develop an efficient way of delivering products.
- **DNA Analysis:** They are used in DNA analysis to establish the DNA structure using spectrometric information.
- **Multimodal Optimization:** They are used to provide multiple optimum solutions in multimodal optimization problems.
- **Aircraft Design:** They are used to develop parametric aircraft designs. The parameters of the aircraft are modified and upgraded to provide better designs.
- **Economics:** They are used in economics to describe various models such as the game theory, cobweb model, asset pricing, and schedule optimization.

Limitations of genetic algorithms

- They are not effective in solving simple problems.
- Lack of proper implementation may make the algorithm converge to a solution that is not optimal.
- The quality of the final solution is not guaranteed.
- Repetitive calculation of fitness values may make some problems to experience computational challenges.

3.8. Classifier accuracy

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

- It works well only if there are equal number of samples belonging to each class.
- For example, consider that there are 98% samples of class A and 2% samples of class B in our training set. Then our model can easily get **98% training accuracy** by simply predicting every training sample belonging to class A.
- When the same model is tested on a test set with 60% samples of class A and 40% samples of class B, then the **test accuracy would drop down to 60%.** Classification Accuracy is great, but gives us the false sense of achieving high accuracy.
- The real problem arises, when the cost of misclassification of the minor class samples are very high. If we deal with a rare but fatal disease, the cost of failing to diagnose the disease of a sick person is much higher than the cost of sending a healthy person to more tests.

Evaluating your machine learning algorithm is an essential part of any project. Your model may give you satisfying results when evaluated using a metric say accuracy_score but may give poor results when evaluated against other metrics such as logarithmic_loss or any other such metric. Most of the times we use classification accuracy to measure the performance of our model, however it is not enough to truly judge our model. In this post, we will cover different types of evaluation metrics available such as Classification Accuracy, Logarithmic Loss, Confusion Matrix, Area under Curve, F1 Score, Mean Absolute Error and Mean Squared Error.

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Chapter 4: Cluster analysis

What is cluster analysis?

Cluster analysis: is the grouping of objects such that objects in the same cluster are more similar to each other than they are to objects in another cluster. The classification into clusters is done using criteria such as smallest distances, density of data points, graphs, or various statistical distributions. Cluster analysis has wide applicability, including in unsupervised machine learning, data mining, statistics, Graph Analytics, image processing, and numerous physical and social science applications.

Data scientists and others use clustering to gain important insights from data by observing what groups (or clusters) the data points fall into when they apply a clustering algorithm to the data. By definition, unsupervised learning is a type of machine learning that searches for patterns in a data set with no pre-existing labels and a minimum of human intervention. Clustering can also be used for anomaly detection to find data points that are not part of any cluster, or outliers.

Clustering is used to identify groups of similar objects in datasets with two or more variable quantities. In practice, this data may be collected from marketing, biomedical, or geospatial databases, among many other places.

Types of data in cluster analysis

The data used in cluster analysis can be interval, ordinal or categorical. However, having a mixture of different types of variable will make the analysis more complicated. This is because in cluster analysis you need to have some way of measuring the distance between observations and the type of measure used will depend on what type of data you have.

Categorization of major clustering methods

Broadly, there are 2 types of cluster analysis methods. On the basis of the categorization of data sets into a particular cluster, cluster analysis can be divided into 2 types - hard and soft clustering. They are as follows;

1. Hard Clustering

In a given dataset, it is possible for a data researcher to organize clusters in a manner that a single dataset is placed in only one of the total number of given clusters. This implies that a hard-core classification of datasets is required in order to organize and classify data accordingly. For instance, a clustering algorithm classifies data points in one cluster such that they have the maximum similarity. However, there are no other grounds of similarity with data sets belonging to other clusters.

2. Soft Clustering

The second class of cluster analysis is Soft Clustering. Unlike hard clustering that requires a given data point to belong to only a cluster at a time, soft clustering follows a different rule. In the case of soft clustering, a given data point can belong to more than one cluster at a time. This means that a fuzzy classification of datasets characterizes soft clustering. Fuzzy Clustering Algorithm in Machine learning is a renowned unsupervised algorithm for processing data into soft clusters.

There are a number of different methods that can be used to carry out a cluster analysis; these methods can be classified as Hierarchical and Non-Hierarchical methods:

- **Hierarchical methods** – Agglomerative methods, in which subjects start in their own separate cluster. The two 'closest' (most similar) clusters are then combined and this is done repeatedly until all subjects are in one cluster. At the end, the optimum number of clusters is then chosen out of all cluster solutions. – Divisive methods, in which all subjects start in the same cluster and the above strategy is applied in reverse until every subject is in a separate cluster. Agglomerative methods are used more often than divisive methods, so this handout will concentrate on the former rather than the latter.

Within this approach to cluster analysis there are a number of different methods used to determine which clusters should be joined at each stage. The main methods are summarised below.

- **Nearest neighbor method** (single linkage method): In this method the distance between two clusters is defined to be the distance between the two closest members, or neighbors. This method is relatively simple but is often criticised

because it doesn't take account of cluster structure and can result in a problem called chaining whereby clusters end up being long and straggly. However, it is better than the other methods when the natural clusters are not spherical or elliptical in shape.

- **Furthest neighbor method** (complete linkage method) In this case the distance between two clusters is defined to be the maximum distance between members — i.e. the distance between the two subjects that are furthest apart. This method tends to produce compact clusters of similar size but, as for the nearest neighbour method, does not take account of cluster structure. It is also quite sensitive to outliers.

- **Average (between groups) linkage method** (sometimes referred to as UPGMA) The distance between two clusters is calculated as the average distance between all pairs of subjects in the two clusters. This is considered to be a fairly robust method.

- **Centroid method:** Here the centroid (mean value for each variable) of each cluster is calculated and the distance between centroids is used. Clusters whose centroids are closest together are merged. This method is also fairly robust.

- **Ward's method:** In this method all possible pairs of clusters are combined and the sum of the squared distances within each cluster is calculated. This is then summed over all clusters. The combination that gives the lowest sum of squares is chosen. This method tends to produce clusters of approximately equal size, which is not always desirable. It is also quite sensitive to outliers. Despite this, it is one of the most popular methods, along with the average linkage method

- **Non-hierarchical methods** (often known as k-means clustering methods)

In these methods the desired number of clusters is specified in advance and the 'best' solution is chosen. The steps in such a method are as follows:

1. Choose initial cluster centres (essentially this is a set of observations that are far apart each subject forms a cluster of one and its centre is the value of the variables for that subject).

2. Assign each subject to its ’nearest’ cluster, defined in terms of the distance to the centroid.
3. Find the centroids of the clusters that have been formed
4. Re-calculate the distance from each subject to each centroid and move observations that are not in the cluster that they are closest to.
5. Continue until the centroids remain relatively stable.

Non-hierarchical cluster analysis tends to be used when large data sets are involved. It is sometimes preferred because it allows subjects to move from one cluster to another (this is not possible in hierarchical cluster analysis where a subject, once assigned, cannot move to a different cluster). Two disadvantages of non-hierarchical cluster analysis are: (1) it is often difficult to know how many clusters you are likely to have and therefore the analysis may have to be repeated several times and (2) it can be very sensitive to the choice of initial cluster centres. Again, it may be worth trying different ones to see what impact this has. One possible strategy to adopt is to use a hierarchical approach initially to determine how many clusters there are in the data and then to use the cluster centres obtained from this as initial cluster centres in the non-hierarchical method.

Partitioning methods

This clustering method classifies the information into multiple groups based on the characteristics and similarity of the data. Its the data analysts to specify the number of clusters that has to be generated for the clustering methods. In the partitioning method when database(D) that contains multiple(N) objects then the partitioning method constructs user-specified(K) partitions of the data in which each partition represents a cluster and a particular region. There are many algorithms that come under partitioning method some of the popular ones are K-Mean, PAM(K-Medoids), CLARA algorithm (Clustering Large Applications) etc.

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

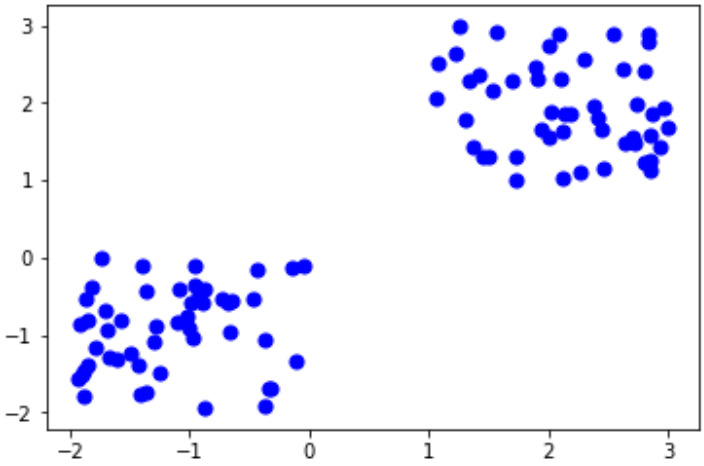
Method:

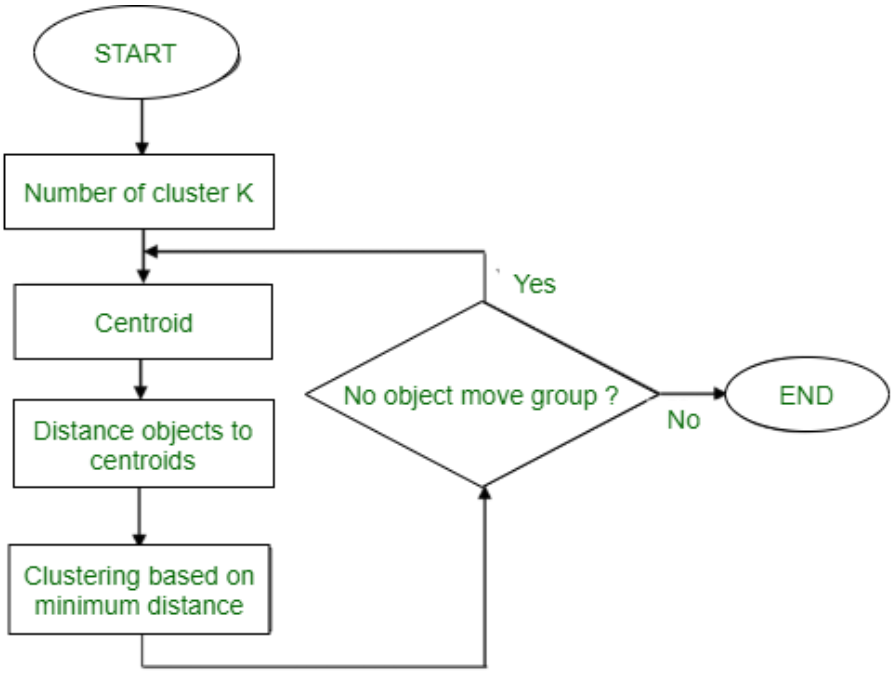
1.Randomly assign K objects from the dataset (D) as cluster centers (C)

2.(Re) Assign each object to which object is most similar based upon mean values.

3.Update Cluster means, i.e., Recalculate the mean of each cluster with the updated values.

4.Repeat Step 2 until no change occurs.





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.

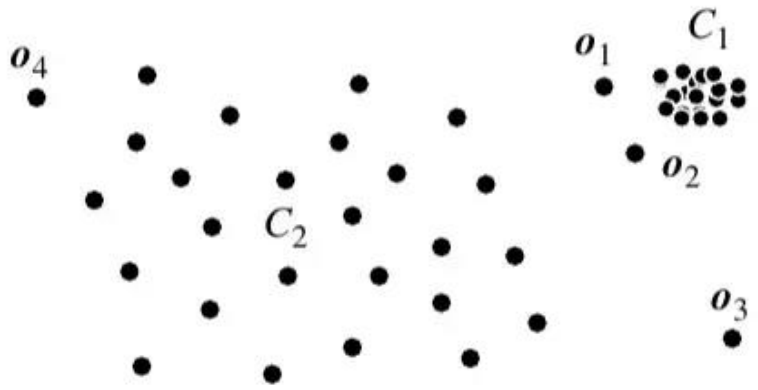
Density based methods & Outlier analysis

Density-based outlier detection method investigates the density of an object and that of its neighbors. Here, an object is identified as an outlier if its density is relatively much lower than that of its neighbors.

Many real-world data sets demonstrate a more complex structure, where objects may be considered outliers with respect to their local neighborhoods, rather than with respect to the global data distribution.

Some partitioning methods cluster objects based on the distance among objects. Such methods can discover only spherical-shaped clusters and encounter difficulty in finding clusters of arbitrary shapes. Other clustering methods have been created based on the concept of density.

DBSCAN is a typical density-based method that increases clusters according to a density threshold. OPTICS is a density-based method that evaluates an augmented clustering ordering for automatic and interactive cluster analysis.



Consider the example above, distance-based methods are able to detect o3, but as for o1 and o2 it is not as evident.

The idea of density-based is that we need to compare the density around an object with the density around its local neighbors. The basic assumption of density-based outlier detection methods is that the density around a nonoutlier object is similar to the density around its neighbors, while the density around an outlier object is significantly different from the density around its neighbors.

dist_k(o)-is a distance between object o and k-nearest neighbors. The k-distance neighborhood of o contains all objects of the distance to o is not greater than dist_k(o) the k-th distance of o

$$N_k(o) = [o' | o' \in D, dist(o, o') \leq dist_k(o)]$$

We can use the average distance from the objects in Nk(o) to **o** as the measure of the local density of **o**. If **o** has very close neighbors **o'** such that dist(o,o') is very small, the statistical fluctuations of the distance measure can be undesirably high. To overcome this problem, we can switch to the following reachability distance measure by adding a smoothing effect.

$$reachdist_k(o, o') = max[dist_k(o), dist(o, o')]$$

K is a user-specified parameter that controls the smoothing effect. Essentially, k specifies the minimum neighborhood to be examined to determine the local density of an object. Reachability distance is not symmetric.

Local reachability density of an object o is $$lrd_k(o) = \frac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} reachdist_k(o, o')}$$

We calculate the local reachability density for an object and compare it with that of its neighbors to quantify the degree to which the object is considered an outlier.

$$LOF_k(\boldsymbol{o}) = \frac{\sum_{\boldsymbol{o}' \in N_k(\boldsymbol{o})} \frac{lrd_k(\boldsymbol{o}')}{lrd_k(\boldsymbol{o})}}{\|N_k(\boldsymbol{o})\|} = \sum_{\boldsymbol{o}' \in N_k(\boldsymbol{o})} lrd_k(\boldsymbol{o}') \cdot \sum_{\boldsymbol{o}' \in N_k(\boldsymbol{o})} reachdist_k(\boldsymbol{o}' \leftarrow \boldsymbol{o}).$$

The local outlier factor is the average of the ratio of the local reachability density of o and those of o's k-nearest neighbors. The lower local reachability density of o and the higher the local reachability densities of the k-nearest neighbors of o, the higher the LOF value is. This exactly captures a local outlier of which the local density is relatively low compared to the local densities of its k-nearest neighbors.

Chapter 5: Mining association rules in large databases

Data mining uses a technique of Association Rule Mining to generate rules in an efficient way. Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. The main problem that occurs is, handling the large

databases.

5.1 Overview of Pattern Discovery

Patern discovery is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.

- **valid**: to a certain degree the discovered patterns should also hold for new, previously unseen problem instances.
- **novel**: at least to the system and preferable to the use
- **Potentially useful:** they should lead to some benefit to the user or task
- **Ultimately understandable**: the end user should be able to interpret the patterns either immediately or after some postprocessing

5.2 Pattern finding and association rules discovery techniques

Frequent pattern-mining can be classified in various ways, based on the following criteria:

1. Based on the completeness of patterns to be mined:
We can mine the complete set of frequent itemsets, the closed frequent itemsets, and the maximal frequent itemsets, given a minimum support threshold.
We can also mine constrained frequent itemsets, approximate frequent itemsets, near-match frequent itemsets, top-k frequent itemsets and so on.
2. Based on the levels of abstraction involved in the rule set:
Some methods for association-rule mining can find rules at differing levels of abstraction.
For example, suppose-that a set of association rules mined includes the following rules
where X is a variable representing a customer:

buys(X, —”computer”))=>buys(X, —”HP printer”) (1)

buys(X, —”laptop computer”)) =>buys(X, —”HP printer”) (2)

In rule (1) and (2), the items bought are referenced at different levels of abstraction (e.g., “computer” is a higher-level abstraction of —“laptop computer”).

3. Based on the number of data dimensions involved in the rule:

If the items or attributes in an association rule reference only one dimension, then it is a single-dimensional association rule.

buys(X, — “computer”))=>buys(X, — “antivirus software”)

If a rule references two or more dimensions, such as the dimensions age, income, and buys, then it is amultidimensional association rule. The following rule is an exampleof a multidimensional rule:

age(X, — “30,31…39”) ^ income(X, — “42K,…48K”))=>buys(X, — “high resolution TV”)

4. Based on the types of values handled in the rule:

If a rule involves associations between the presence or absence of items, it is a Boolean association rule. If a rule describes associations between quantitative items or attributes, then it is a quantitative association rule.

5. Based on the kinds of rules to be mined:

Frequent pattern analysis can generate various kinds of rules and other interesting relationships.

Association rule mining cangenerate a large number of rules, many of which are redundant or do not indicatea correlation relationship among itemsets. The discovered associations can be further analyzed to uncover statistical correlations, leading to correlation rules.

6. Based on the kinds of patterns to be mined:

Many kinds of frequent patterns can be mined from different kinds of data sets. Sequential pattern mining searches for frequent subsequences in a sequence data set, where a sequence records an ordering of events. For example, with sequential pattern mining, we can study the order in which items are frequently purchased. For instance, customers may tend to first buy a PC, followed by a

digitalcamera,and then a memory card. Structured pattern mining searches for frequent substructuresin a structured data set. Single items are the simplest form of structure. Each element of an itemsetmay contain a subsequence, a subtree, and so on. Therefore, structuredpattern mining can be considered as the most general formof frequent pattern mining.

Efficient Frequent Itemset Mining Methods:

Finding Frequent Itemsets Using Candidate Generation:**The Apriori Algorithm**

- Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.
- The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties.
- Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets.
- First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L1.Next, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found.
- The finding of each Lkrequires one full scan of the database.
- A two-step process is followed in Aprioriconsisting of joinand prune action.

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

- D, a database of transactions;
- *min_sup*, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

```
(1)   L1 = find_frequent_1-itemsets(D);
(2)   for (k = 2; Lk−1 ≠ ϕ; k++) {
(3)       Ck = apriori_gen(Lk−1);
(4)       for each transaction t ∈ D { // scan D for counts
(5)           Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)           for each candidate c ∈ Ct
(7)               c.count++;
(8)       }
(9)       Lk = {c ∈ Ck | c.count ≥ min_sup}
(10)  }
(11)  return L = ∪k Lk;

procedure apriori_gen(Lk−1:frequent (k − 1)-itemsets)
(1)   for each itemset l1 ∈ Lk−1
(2)       for each itemset l2 ∈ Lk−1
(3)           if (l1[1] = l2[1]) ∧ (l1[2] = l2[2]) ∧ ... ∧ (l1[k − 2] = l2[k − 2]) ∧ (l1[k − 1] < l2[k − 1]) then {
(4)               c = l1 ⋈ l2; // join step: generate candidates
(5)               if has_infrequent_subset(c, Lk−1) then
(6)                   delete c; // prune step: remove unfruitful candidate
(7)               else add c to Ck;
(8)           }
(9)       return Ck;

procedure has_infrequent_subset(c: candidate k-itemset;
          Lk−1: frequent (k − 1)-itemsets); // use prior knowledge
(1)   for each (k − 1)-subset s of c
(2)       if s ∉ Lk−1 then
(3)           return TRUE;
(4)   return FALSE;
```

Example:

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

There are nine transactions in this database, that is, |D| = 9

Steps:

1. In the first iteration of the algorithm, each item is a member of the set of candidate1-itemsets, C1. The algorithm simply scans all of the transactions in order to countthe number of occurrences of each item.
2. Suppose that the minimum support count required is 2, that is, min sup = 2. The set of frequent 1-itemsets, L1, can thenbe determined. It consists of the candidate 1-itemsets satisfying minimum support.In our example, all of the candidates in C1 satisfy minimum= support.
3. To discover the set of frequent 2-itemsets, L2, the algorithm uses the join L1 on L1 togenerate a candidate set of 2-itemsets, C2.No candidates are removed fromC2 during the prune step because each subset of thecandidates is also frequent.
4. Next, the transactions inDare scanned and the support count of each candidate itemsetInC2 is accumulated.
5. The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate2-itemsets in C2 having minimum support.
6. The generation of the set of candidate 3-itemsets,C3, Fromthejoin step, we first getC3 =L2x L2 = ({I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4},{I2, I3, I5}, {I2, I4, I5}.

Based on the Apriori property that all subsets of a frequentitemsetmust also be frequent, we can determine that the four latter candidates cannotpossibly be frequent.

7. The transactions in D are scanned in order to determine L3, consisting of those candidate 3-itemsets in C3 having minimum support.
8. The algorithm uses L3x L3 to generate a candidate set of 4-itemsets, C4.

Pattern-Growth Approach

The two primary drawbacks of the Apriori Algorithm are:

1. At each step, candidate sets have to be built.
2. To build the candidate sets, the algorithm has to repeatedly scan the database.

These two properties inevitably make the algorithm slower. To overcome these redundant steps, a new association-rule mining algorithm was developed named Frequent Pattern Growth Algorithm. It overcomes the disadvantages of the Apriori algorithm by storing all the transactions in a Trie Data Structure.

Pattern-growth is one of several influential frequent pattern mining methodologies, where a pattern (e.g., an itemset, a subsequence, a subtree, or a substructure) is *frequent* if its occurrence frequency in a database is no less than a specified *minimum_support* threshold. The (frequent) pattern-growth method mines the data set in a divide-and-conquer way: It first derives the set of size-1 frequent patterns, and for each pattern *p*, it derives *p*'s projected (or conditional) database by data set partitioning and mines the projected database recursively. Since the data set is decomposed progressively into a set of much smaller, pattern-related projected data sets, the pattern-growth method effectively reduces the search space and leads to high efficiency and scalability.

Consider the following data:-

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

The above-given data is a hypothetical dataset of transactions with each letter representing an item. The frequency of each individual item is computed:-

Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
0	3
U	1
Y	3

Let the minimum support be 3. A **Frequent Pattern set** is built which will contain all the elements whose frequency is greater than or equal to the minimum support. These elements are stored in descending order of their respective frequencies. After insertion of the relevant items, the set L looks like this:-

L = {K : 5, E : 4, M : 3, O : 3, Y : 3}

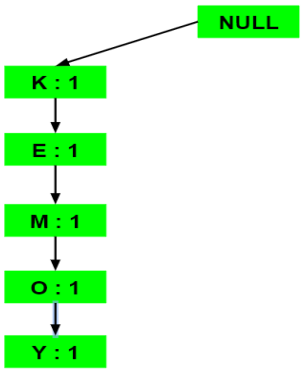
Now, for each transaction, the respective **Ordered-Item set** is built. It is done by iterating the Frequent Pattern set and checking if the current item is contained in the transaction in question. If the current item is contained, the item is inserted in the Ordered-Item set for the current transaction. The following table is built for all the transactions:

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

Now, all the Ordered-Item sets are inserted into a Trie Data Structure.

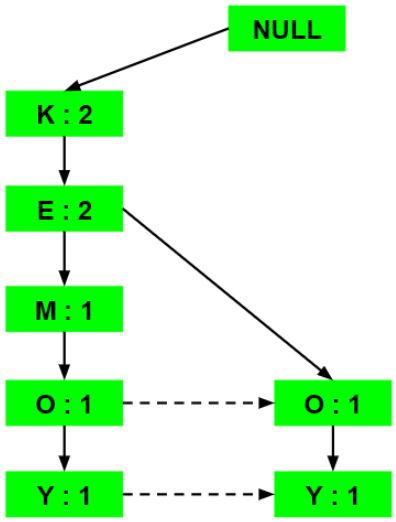
a) **Inserting the set {K, E, M, O, Y}:**

Here, all the items are simply linked one after the other in the order of occurrence in the set and initialize the support count for each item as 1.



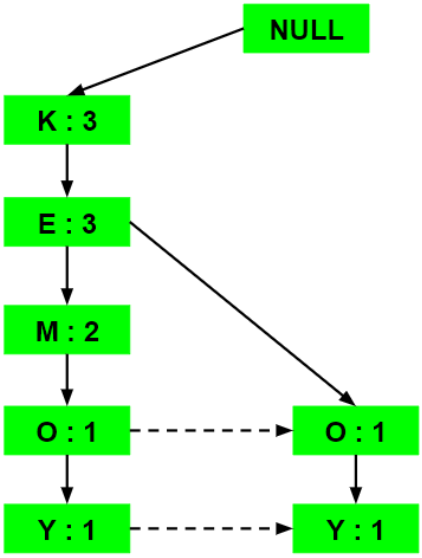
b) **Inserting the set {K, E, O, Y}:**

Till the insertion of the elements K and E, simply the support count is increased by 1. On inserting O we can see that there is no direct link between E and O, therefore a new node for the item O is initialized with the support count as 1 and item E is linked to this new node. On inserting Y, we first initialize a new node for the item Y with support count as 1 and link the new node of O with the new node of Y.



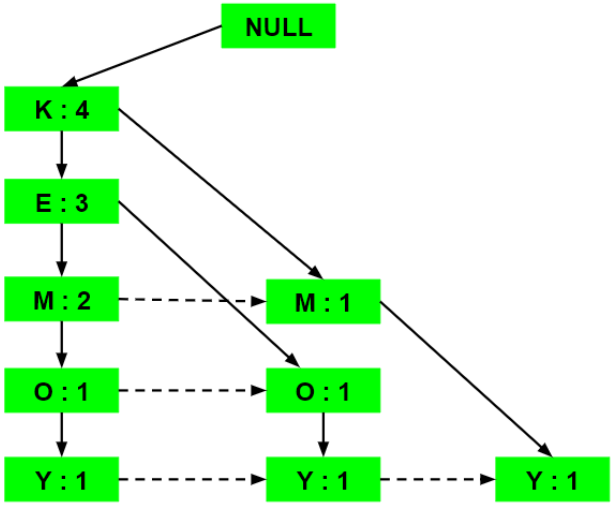
c) **Inserting the set {K, E, M}:**

Here simply the support count of each element is increased by 1.



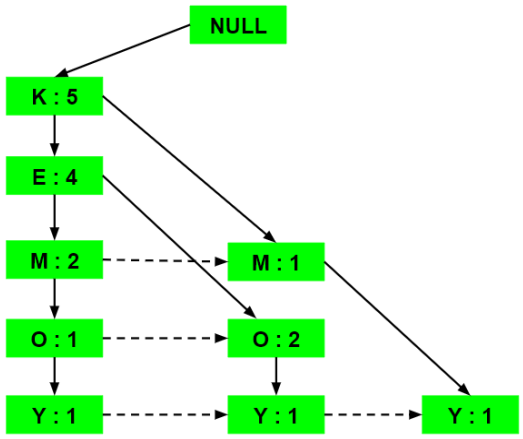
d) Inserting the set {K, M, Y}:

Similar to step b), first the support count of K is increased, then new nodes for M and Y are initialized and linked accordingly.



e) **Inserting the set {K, E, O}:**

Here simply the support counts of the respective elements are increased. Note that the support count of the new node of item O is increased.



Now, for each item, the **Conditional Pattern Base** is computed which is path labels of all the paths which lead to any node of the given item in the frequent-pattern tree. Note that the items in the below table are arranged in the ascending order of their frequencies.

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

Now for each item, the **Conditional Frequent Pattern Tree is built.** It is done by taking the set of elements that is common in all the paths in the Conditional Pattern Base of that item and calculating its support count by summing the support counts of all the paths in the Conditional Pattern Base.

Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

From the Conditional Frequent Pattern tree, the **Frequent Pattern rules** are generated by pairing the items of the Conditional Frequent Pattern Tree set to the corresponding to the item as given in the below table.

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

For each row, two types of association rules can be inferred for example for the first row which contains the element, the rules K -> Y and Y -> K can be inferred. To determine the valid rule, the confidence of both the rules is calculated and the one with confidence greater than or equal to the minimum confidence value is retained.

5.3. Mining single-dimensional Boolean association rules from transactional databases

If the items or attributes in an association rule reference only one dimension, then it is a **single-dimensional association rule**.

For example, the rule

computer => antivirus_software [support = 2%, confidence = 60% could be written as

buys(X, "computer”) = buys(X, “antivirus software")

If a rule references two or more dimensions, such as the dimensions age, income and buys, then it is a multidimensional association rule. The following rule is an example of a **multidimensional rule –**

age(X, “30...39") ^ income(X, “42K...48K”) = buys(X, “high resolution TV”)

If a rule involves associations between the presence or absence of items, it is a **Boolean association rule**. For example,

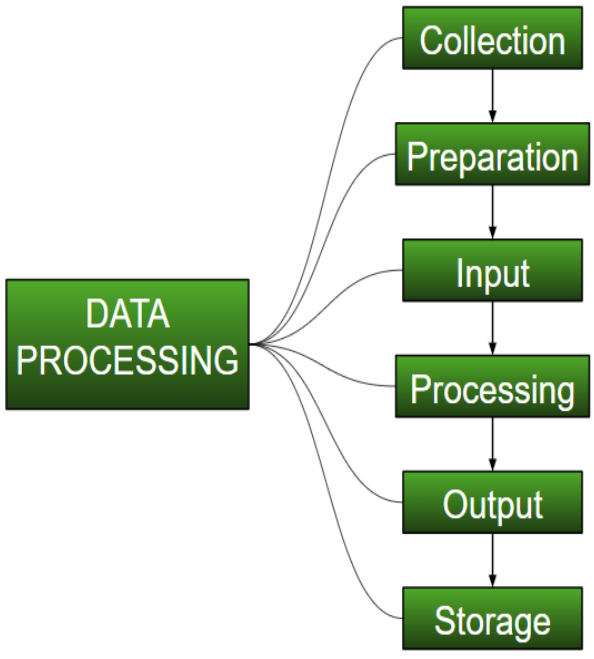
buys(X, “laptop computer”) = buys(X, “HP_printer”)

Apriori is one of single-dimensional boolean association rule algorithms.

Chapter Six: Model Evaluation

Data Processing

Data Processing is the task of converting data from a given form to a much more usable and desired form i.e. making it more meaningful and informative. Using Machine Learning algorithms, mathematical modeling, and statistical knowledge, this entire process can be automated. The output of this complete process can be in any desired form like graphs, videos, charts, tables, images, and many more, depending on the task we are performing and the requirements of the machine. This might seem to be simple but when it comes to massive organizations like Twitter, Facebook, Administrative bodies like Parliament, UNESCO, and health sector organizations, this entire process needs to be performed in a very structured manner. So, the steps to perform are as follows:



Collection : The most crucial step when starting with ML is to have data of good quality

and accuracy. Data can be collected from any authenticated source like data.gov.in, Kaggle or UCI dataset repository. For example, while preparing for a competitive exam, students study from the best study material that they can access so that they learn the best to obtain the best results. In the same way, high-quality and accurate data will make the learning process of the model easier and better and at the time of testing, the model would yield state-of-the-art results.

A huge amount of capital, time and resources are consumed in collecting data. Organizations or researchers have to decide what kind of data they need to execute their tasks or research.

Example: Working on the Facial Expression Recognizer, needs numerous images having a variety of human expressions. Good data ensures that the results of the model are valid and can be trusted upon.

Preparation : The collected data can be in a raw form which can't be directly fed to the machine. So, this is a process of collecting datasets from different sources, analyzing these datasets and then constructing a new dataset for further processing and exploration. This preparation can be performed either manually or from the automatic approach. Data can also be prepared in numeric forms also which would fasten the model's learning.

Example: An image can be converted to a matrix of N X N dimensions, the value of each cell will indicate the image pixel.

Input : Now the prepared data can be in the form that may not be machine-readable, so to convert this data to the readable form, some conversion algorithms are needed. For this task to be executed, high computation and accuracy is needed. Example: Data can be collected through the sources like MNIST Digit data(images), Twitter comments, audio files, video clips.

Processing : This is the stage where algorithms and ML techniques are required to perform the instructions provided over a large volume of data with accuracy and optimal computation.

Output : In this stage, results are procured by the machine in a meaningful manner which can be inferred easily by the user. Output can be in the form of reports, graphs, videos, etc

Storage : This is the final step in which the obtained output and the data model data and all the useful information are saved for future use.

Data cleaning and transforming

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.

What is the difference between data cleaning and data transformation?

Data cleaning is the process that removes data that does not belong in your dataset. Data transformation is the process of converting data from one format or structure into another. Transformation processes can also be referred to as data wrangling, or data munging, transforming and mapping data from one "raw" data form into another format for warehousing and analyzing. This article focuses on the processes of cleaning that data.

While the techniques used for data cleaning may vary according to the types of data your company stores, you can follow these basic steps to map out a framework for your organization.

Step 1: Remove duplicate or irrelevant observations

Remove unwanted observations from your dataset, including duplicate observations or irrelevant observations. Duplicate observations will happen most often during data collection. When you combine data sets from multiple places, scrape data, or receive data from clients or multiple departments, there are opportunities to create duplicate data. De-duplication is one of the largest areas to be considered in this process. Irrelevant observations are when you notice observations that do not fit into the specific problem you are trying to analyze. For example, if you want to analyze data regarding millennial customers, but your dataset includes older generations, you might remove those irrelevant observations. This can make analysis more efficient and minimize distraction from your primary target—as well as creating a more manageable and more performant dataset.

Step 2: Fix structural errors

Structural errors are when you measure or transfer data and notice strange naming conventions, typos, or incorrect capitalization. These inconsistencies can cause mislabeled categories or

classes. For example, you may find “N/A” and “Not Applicable” both appear, but they should be analyzed as the same category.

Step 3: Filter unwanted outliers

Often, there will be one-off observations where, at a glance, they do not appear to fit within the data you are analyzing. If you have a legitimate reason to remove an outlier, like improper data-entry, doing so will help the performance of the data you are working with. However, sometimes it is the appearance of an outlier that will prove a theory you are working on. Remember: just because an outlier exists, doesn’t mean it is incorrect. This step is needed to determine the validity of that number. If an outlier proves to be irrelevant for analysis or is a mistake, consider removing it.

Step 4: Handle missing data

You can’t ignore missing data because many algorithms will not accept missing values. There are a couple of ways to deal with missing data. Neither is optimal, but both can be considered.

1. As a first option, you can drop observations that have missing values, but doing this will drop or lose information, so be mindful of this before you remove it.
2. As a second option, you can input missing values based on other observations; again, there is an opportunity to lose integrity of the data because you may be operating from assumptions and not actual observations.
3. As a third option, you might alter the way the data is used to effectively navigate null values.

Step 5: Validate and QA

At the end of the data cleaning process, you should be able to answer these questions as a part of basic validation:

- Does the data make sense?
- Does the data follow the appropriate rules for its field?
- Does it prove or disprove your working theory, or bring any insight to light?
- Can you find trends in the data to help you form your next theory?
- If not, is that because of a data quality issue?

False conclusions because of incorrect or “dirty” data can inform poor business strategy and decision-making. False conclusions can lead to an embarrassing moment in a reporting meeting when you realize your data doesn’t stand up to scrutiny. Before you get there, it is important to create a culture of quality data in your organization. To do this, you should document the tools you might use to create this culture and what data quality means to you.

Obviously Machine Learning requires a structured dataset to get meaningful prediction outcomes. The following are some the checklist to ensure your data is well structured and machine learning ready.

- Dataset must have at least 1,000 rows
- Dataset must have at least 5 columns
- The first column must be an identifier column, such as a name, customer_id, etc.
- The first row should be column names
- The data should be aggregated in a single file or table
- The data must have as less missing values as possible
- No personally identifiable information is required, such as phone numbers, addresses, etc.
- No long text phrases—only use discrete values for text columns

Feature selection and visualization

Machine learning models follow a simple rule: whatever goes in, comes out. If we put garbage into our model, we can expect the output to be garbage too. In this case, garbage refers to noise in our data.

To train a model, we collect enormous quantities of data to help the machine learn better. Usually, a good portion of the data collected is noise, while some of the columns of our dataset might not contribute significantly to the performance of our model. Further, having a lot of data can slow down the training process and cause the model to be slower. The model may also learn from this irrelevant data and be inaccurate.

Feature selection is what separates good data scientists from the rest. Given the same model and computational facilities, why do some people win in competitions with faster and more accurate models? The answer is Feature Selection. Apart from choosing the right model for our data, we need to choose the right data to put in our model.

Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data.

It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve. We do this by including or excluding important features without changing them. It helps in cutting down the noise in our data and reducing the size of our input data.

Feature Selection Models

Feature selection models are of two types:

1. **Supervised Models:** Supervised feature selection refers to the method which uses the output label class for feature selection. They use the target variables to identify the variables which can increase the efficiency of the model
2. **Unsupervised Models:** Unsupervised feature selection refers to the method which does not need the output label class for feature selection. We use them for unlabelled data.

We can further divide the supervised models into three :

1. Filter Method: In this method, features are dropped based on their relation to the output, or how they are **correlating** to the output. We use correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly. Eg: Information Gain, Chi-Square Test, Fisher's Score, etc.

2. Wrapper Method: We split our data into subsets and train a model using this. Based on the output of the model, we add and subtract features and train the model again. It forms the subsets using a greedy approach and evaluates the accuracy of all the possible combinations of features. Eg: Forward Selection, Backwards Elimination, etc.

3. Intrinsic Method: This method combines the qualities of both the Filter and Wrapper method to create the best subset. This method takes care of the machine training iterative process while maintaining the computation cost to be minimum. Eg: Lasso and Ridge Regression.

How to Choose a Feature Selection Model?

How do we know which feature selection model will work out for our model? The process is relatively simple, with the model depending on the types of input and output variables.

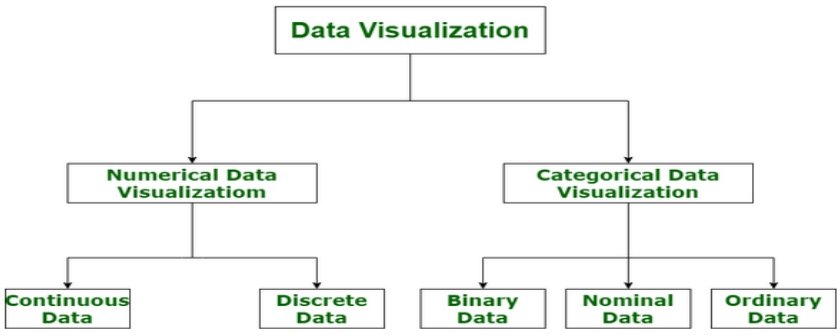
Variables are of two main types:

- Numerical Variables: Which include integers, float, and numbers.
- Categorical Variables: Which include labels, strings, boolean variables, etc.

Data visualization: is the graphical representation of information and data in a pictorial or graphical format(Example: charts, graphs, and maps). Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers. Data visualization tools and technologies are essential to analyzing massive amounts of information and making data-driven decisions. The concept of using pictures is to understand data that has been used for centuries. General types of data visualization are Charts, Tables, Graphs, Maps, Dashboards.

Categories of Data Visualization

Data visualization is very critical to market research where both numerical and categorical data can be visualized, which helps in an increase in the impact of insights and also helps in reducing the risk of analysis paralysis. So, data visualization is categorized into the following categories:



Advantages of Data Visualization

1. Better Agreement: In business, for numerous periods, it happens that we need to look at the exhibitions of two components or two situations. A conventional methodology is to experience the massive information of both the circumstances and afterward examine it. This will clearly take a great deal of time.

2. A Superior Method: It can tackle the difficulty of placing the information of both perspectives into the pictorial structure. This will unquestionably give a superior comprehension of the circumstances. For instance, Google patterns assist us with understanding information identified with top ventures or inquiries in pictorial or graphical structures.

3. Simple Sharing of Data: With the representation of the information, organizations present another arrangement of correspondence. Rather than sharing the cumbersome information, sharing the visual data will draw in and pass on across the data which is more absorbable.

4. Deals Investigation: With the assistance of information representation, a salesman can, without much of a stretch, comprehend the business chart of items. With information perception instruments like warmth maps, he will have the option to comprehend the causes that are pushing the business numbers up just as the reasons that are debasing the business numbers. Information representation helps in understanding the patterns and furthermore, different variables like sorts of clients keen on purchasing, rehash clients, the impact of topography, and so forth.

5. Discovering Relations Between Occasions: A business is influenced by a lot of elements. Finding a relationship between these elements or occasions encourages chiefs to comprehend the issues identified with their business. For instance, the online business market is anything but another thing today. Each time during certain happy seasons, like Christmas or Thanksgiving, the diagrams of online organizations go up. Along these lines, state if an online organization is doing a normal $1 million business in a specific quarter and the business ascends straightaway, at that point they can rapidly discover the occasions compared to it.

6. Investigating Openings and Patterns: With the huge loads of information present, business chiefs can discover the profundity of information in regard to the patterns and openings around them. Utilizing information representation, the specialists can discover examples of the conduct of their clients, subsequently preparing for them to investigate patterns and open doors for business.

Model selection and tuning

A machine learning model usually contains parameters. Some deep learning models even have over hundred parameters. Model selection is to select the best model with the best hyperparameters. This can be done manually or automatically.

What is Model Selection?

- Model selection: how to select the final model from potential candidate models based on their performance on the training dataset
    - select a model from different types of models, e.g. from three classifiers: logistic

regression, decision tree and KNN

    - select a model from the same model with different model hyperparameters, e.g., from the decision tree model with different tree depths

- Model evaluation: how to evaluate the performance of a candidate models using certain metrics, e.g., accuracy

Considerations in Model Selection

- In a business project, model selection is not only a technical problem
- There are many considerations such as
    - business requirements
    - cost
    - performance
    - state-of-the-art
    - maintenance
- It is difficult or impossible to find the best model meeting all criteria. Usually, turn to find a satisfactory model
- The best model in terms of technical performance may not be the best in business

Parameter and Hyperparameter

machine learning model always has a few parameters. Some deep learning models even have over hundred parameters

Parameters of a model can be classified into two types:

1. **parameter**: internal coefficients or weights for a model

- e.g., Logistic regression: coefficients ($\beta_0, \cdots, \beta_n$) $$ \pi(x) = \frac{e^{\beta_0 +\beta_1 x_1+ \cdots + \beta_n x_n}}{1+e^{\beta_0 +\beta_1 x+\cdots+\beta_n x_n}}$$
- learned in model fitting
- fitting is to find optimal internal model parameters

2. **hyperparameter**: specified by practitioner when configuring the model
    - e.g. decision tree: depth of a tree
    - tuned by practitioner

- Tuning hyper-parameters may significantly improve the performance of a machine learning model
- Manual tuning: use the accuracy curve or other curves to select a best hyper-parameter
- Automatic tuning: a more powerful hyper-parameter tuning technique through search

Dataset for Tuning Hyperparameter

- On the test dataset
    - Test dataset is regarded as new data to a model
    - A model cannot see class labels of test data. In Kaggle competitions, class labels of test data are set unseen to participants
    - Thus, it is impossible to tune parameters on the test dataset
    - In other words, hyperparameter tuning in our previous labs is wrong
- On the training dataset
    - A hyperparameter must be tuned on training data set
    - However, this is bad practice because it leads to overfitting

Methods of dimensional reduction

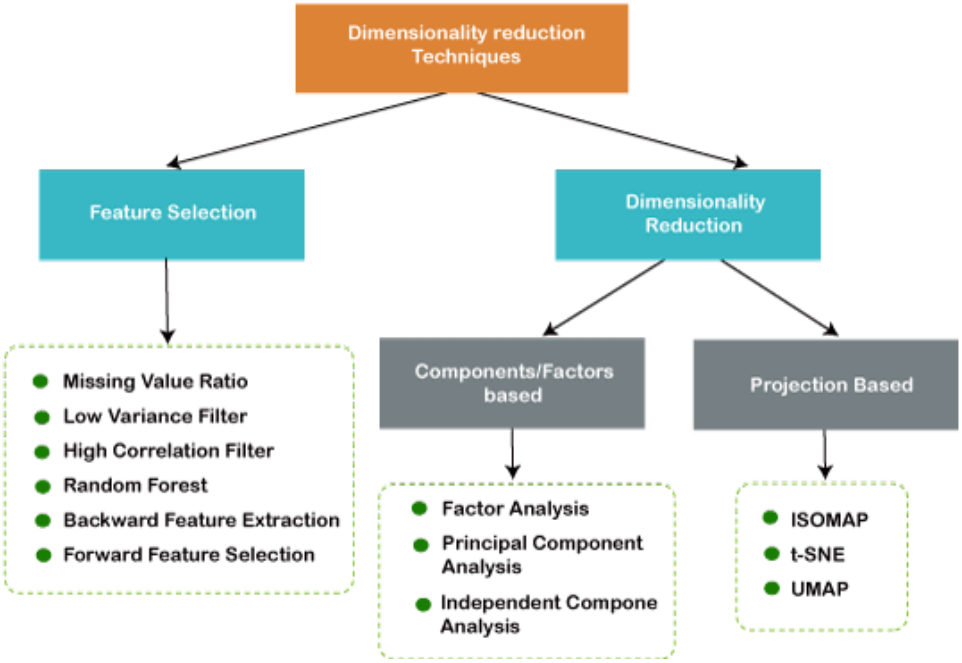
What is Dimensionality Reduction?

The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.

A dataset contains a huge number of input features in various cases, which makes the predictive modeling task more complicated. Because it is very difficult to visualize or make predictions for

the training dataset with a high number of features, for such cases, dimensionality reduction techniques are required to use.

Dimensionality reduction technique can be defined as, *"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information."* These techniques are widely used in machine learning for obtaining a better fit predictive model while solving the classification and regression problems.



The Curse of Dimensionality

Handling the high-dimensional data is very difficult in practice, commonly known as the *curse of dimensionality.* If the dimensionality of the input dataset increases, any machine learning algorithm and model becomes more complex. As the number of features increases, the number of samples also gets increased proportionally, and the chance of overfitting also increases. If the machine learning model is trained on high-dimensional data, it becomes overfitted and results in poor performance.

Hence, it is often required to reduce the number of features, which can be done with dimensionality reduction.

Benefits of applying Dimensionality Reduction

Some benefits of applying dimensionality reduction technique to the given dataset are given below:

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.
- Less Computation training time is required for reduced dimensions of features.
- Reduced dimensions of features of the dataset help in visualizing the data quickly.
- **I**t removes the redundant features (if present) by taking care of multicollinearity.

Disadvantages of dimensionality Reduction

There are also some disadvantages of applying the dimensionality reduction, which are given below:

- Some data may be lost due to dimensionality reduction.
- In the PCA dimensionality reduction technique, sometimes the principal components required to consider are unknown.

Approaches of Dimension Reduction

There are two ways to apply the dimension reduction technique, which are given below:

Feature Selection

Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy. In other words, it is a way of selecting the optimal features from the input dataset.

Three methods are used for the feature selection:

1. Filters Methods

In this method, the dataset is filtered, and a subset that contains only the relevant features is taken. Some common techniques of filters method are:

- **Correlation**
- **Chi-Square Test**
- **ANOVA**
- **Information Gain, etc.**

2. Wrappers Methods

The wrapper method has the same goal as the filter method, but it takes a machine learning model for its evaluation. In this method, some features are fed to the ML model, and evaluate the performance. The performance decides whether to add those features or remove to increase the accuracy of the model. This method is more accurate than the filtering method but complex to work. Some common techniques of wrapper methods are:

- Forward Selection
- Backward Selection
- Bi-directional Elimination

3. Embedded Methods: Embedded methods check the different training iterations of the machine learning model and evaluate the importance of each feature. Some common techniques of Embedded methods are:

- **LASSO**
- **Elastic Net**
- **Ridge Regression, etc.**

Feature Extraction:

Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions. This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

Some common feature extraction techniques are:

1. Principal Component Analysis
2. Linear Discriminant Analysis
3. Kernel PCA
4. Quadratic Discriminant Analysis

Common techniques of Dimensionality Reduction

1. **Principal Component Analysis**
2. **Backward Elimination**
3. **Forward Selection**
4. **Score comparison**
5. **Missing Value Ratio**
6. **Low Variance Filter**
7. **High Correlation Filter**
8. **Random Forest**
9. **Factor Analysis**
10. **Auto-Encoder**

Principal Component Analysis (PCA)

Principal Component Analysis is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are *image processing, movie recommendation system, optimizing the power allocation in various communication channels.*

Backward Feature Elimination

The backward feature elimination technique is mainly used while developing Linear Regression or Logistic Regression model. Below steps are performed in this technique to reduce the dimensionality or in feature selection:

- In this technique, firstly, all the n variables of the given dataset are taken to train the model.
- The performance of the model is checked.
- Now we will remove one feature each time and train the model on n-1 features for n times, and will compute the performance of the model.
- We will check the variable that has made the smallest or no change in the performance of the model, and then we will drop that variable or features; after that, we will be left with n-1 features.
- Repeat the complete process until no feature can be dropped.

In this technique, by selecting the optimum performance of the model and maximum tolerable error rate, we can define the optimal number of features require for the machine learning algorithms.

Forward Feature Selection

Forward feature selection follows the inverse process of the backward elimination process. It means, in this technique, we don't eliminate the feature; instead, we will find the best features that can produce the highest increase in the performance of the model. Below steps are performed in this technique:

- We start with a single feature only, and progressively we will add each feature at a time.
- Here we will train the model on each feature separately.
- The feature with the best performance is selected.
- The process will be repeated until we get a significant increase in the performance of the model.

Missing Value Ratio

If a dataset has too many missing values, then we drop those variables as they do not carry much useful information. To perform this, we can set a threshold level, and if a variable has missing values more than that threshold, we will drop that variable. The higher the threshold value, the more efficient the reduction.

Low Variance Filter

As same as missing value ratio technique, data columns with some changes in the data have less information. Therefore, we need to calculate the variance of each variable, and all data columns with variance lower than a given threshold are dropped because low variance features will not affect the target variable.

High Correlation Filter

High Correlation refers to the case when two variables carry approximately similar information. Due to this factor, the performance of the model can be degraded. This correlation between the independent numerical variable gives the calculated value of the correlation coefficient. If this value is higher than the threshold value, we can remove one of the variables from the dataset. We can consider those variables or features that show a high correlation with the target variable.

Random Forest

Random Forest is a popular and very useful feature selection algorithm in machine learning. This algorithm contains an in-built feature importance package, so we do not need to program it separately. In this technique, we need to generate a large set of trees against the target variable, and with the help of usage statistics of each attribute, we need to find the subset of features.

Random forest algorithm takes only numerical variables, so we need to convert the input data into numeric data using **hot encoding**.

Factor Analysis

Factor analysis is a technique in which each variable is kept within a group according to the correlation with other variables, it means variables within a group can have a high correlation between themselves, but they have a low correlation with variables of other groups.

We can understand it by an example, such as if we have two variables Income and spend. These two variables have a high correlation, which means people with high income spends more, and vice versa. So, such variables are put into a group, and that group is known as the **factor**. The number of these factors will be reduced as compared to the original dimension of the dataset.

Auto-encoders

One of the popular methods of dimensionality reduction is auto-encoder, which is a type of ANN or artificial neural network, and its main aim is to copy the inputs to their outputs. In this, the

input is compressed into latent-space representation, and output is occurred using this representation. It has mainly two parts:

- **Encoder:** The function of the encoder is to compress the input to form the latent-space representation.
- **Decoder:** The function of the decoder is to recreate the output from the latent-space representation.

Reference

[1] E. Malinowski *et al.*, “About the Tutorial Copyright & Disclaimer,” *Data Vault 2.0*, no. January 1999, pp. 1–15, 2019, doi: 10.1007/978-3-322-94873-1.

[2] Dataiku, “Machine learning Basics - An Illustrated Guide for Non-Technical Readers,” *Daiku*, p. 15, 2017, [Online]. Available: https://pages.dataiku.com/machine-learning-basics-thank-you?submissionGuid=80d21f82-ac46-45d0-969a-cd9914d06af9.

[3] A. L. Fradkov, “Early history of machine learning,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1385–1390, 2020, doi: 10.1016/j.ifacol.2020.12.1888.

[4] P. Dönmez, “Introduction to Machine Learning, 2nd ed., by Ethem Alpaydın. Cambridge, MA: The MIT Press2010. ISBN: 978-0-262-01243-0. $54/£ 39.95 + 584 pages.,” *Nat. Lang. Eng.*, vol. 19, no. 2, pp. 285–288, 2013, doi: 10.1017/s1351324912000290.

[5] D. Tree, “Contents 3,” *1964*, pp. XV–XVIII, 2021, doi: 10.1515/9783112500088-003.

[6] J. Han and M. Kamber, “Chapter 6 : classification and prediction,” *Data MiningConcepts Tech.*, pp. 380–394, 2000.

[7] A. Oken, “An Introduction To and Applications of Neural Networks,” pp. 1–30, 2017.

[8] N. Networks, S. Nns, and M. Nns, “Introduction To Neural Networks.”

[9] K. Gurney, *An introduction to neural networks An introduction to neural networks*. .