LNCT℠
GROUP OF COLLEGES
"WORKING TOWARDS BEING THE BEST"

LNCT GROUP
1993 25 2018
EXCELLENCE IN EDUCATION

# Unit -5

## Decision Tree:

**Decision Tree :** Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**Construction of Decision Tree :**
A tree can be *"learned"* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called *recursive partitioning*. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

**Decision Tree Representation :**
Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree,testing the attribute specified by this node,then moving down the tree branch corresponding to the value of the attribute as shown in the above figure.This process is then repeated for the subtree rooted at the new node.
The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf.(in this case Yes or No).
For example,the instance

*(Outlook = Rain, Temperature = Hot, Humidity = High, Wind = Strong )*

would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance.

In other words we can say that decision tree represent a disjunction of conjunctions of constraints on the attribute values of instances.

*(Outlook = Sunny ^ Humidity = Normal) v (Outllok = Overcast) v (Outlook = Rain ^ Wind = Weak)*

**Strengths and Weakness of Decision Tree approach**
The strengths of decision tree methods are:

- Decision trees are able to generate understandable rules.
- Decision trees perform classification without requiring much computation.
- Decision trees are able to handle both continuous and categorical variables.
- Decision trees provide a clear indication of which fields are most important for prediction or classification.

The weaknesses of decision tree methods :

- Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
- Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.
- Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.

**Decision Tree Introduction with example:**

- Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.
- Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.
- We can represent any boolean function on discrete attributes using the decision tree.

**Below are some assumptions that we made while using decision tree:**

- At the beginning, we consider the whole training set as the root.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- On the basis of attribute values records are distributed recursively.
- We use statistical methods for ordering attributes as root or the internal node.

In Decision Tree the major challenge is to identification of the attribute for the root node in each level.

process is known as attribute selection. We have two popular attribute selection measures:

1. Information Gain
2. Gini Index

**1. Information Gain**

When we use a node in a decision tree to partition the training instances into smaller subsets the entropy changes. Information gain is a measure of this change in entropy.

***Definition***: Suppose S is a set of instances, A is an attribute, $S_v$ is the subset of S with A = v, and Values (A) is the set of all possible values of A, then

**Entropy**

Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content.

***Definition***: Suppose S is a set of instances, A is an attribute, $S_v$ is the subset of S with A = v, and Values (A) is the set of all possible values of A, then

Example:

For the set X = {a,a,a,b,b,b,b,b}

Total intances: 8

Instances of b: 5

Instances of a: 3

= -[0.375 * (-1.415) + 0.625 * (-0.678)]

=-(-0.53-0.424)

= 0.954

**Building Decision Tree using Information Gain**

**The essentials:**

- Start with all training instances associated with the root node
- Use info gain to choose which attribute to label each node with
- *Note:* No root-to-leaf path should contain the same discrete attribute twice
- Recursively construct each subtree on the subset of training instances that would be classified down that path in the tree.

**The border cases:**

- If all positive or all negative training instances remain, label that node "yes" or "no" accordingly
- If no attributes remain, label with a majority vote of training instances left at that node
- If no instances remain, label with a majority vote of the parent's training instances

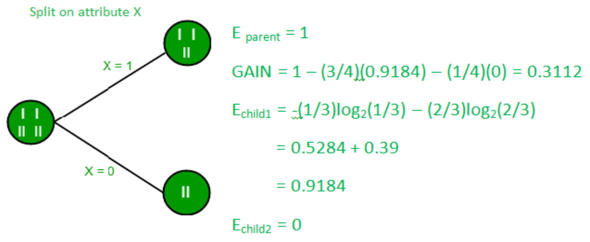
**Example:**

Now, lets draw a Decision Tree for the following data using Information gain.

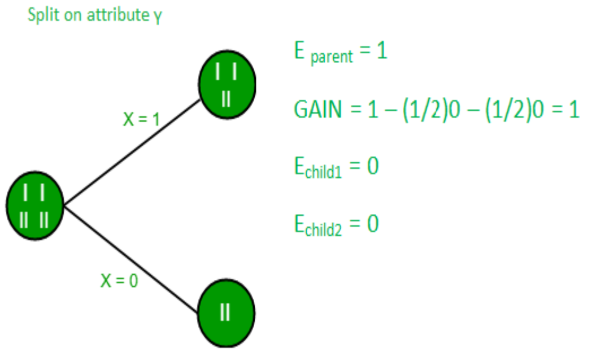
**Training set: 3 features and 2 classes**

| X | Y | Z | C |
|---|---|---|---|
| 1 | 1 | 1 | I |
| 1 | 1 | 0 | I |
| 0 | 0 | 1 | II |
| 1 | 0 | 0 | II |

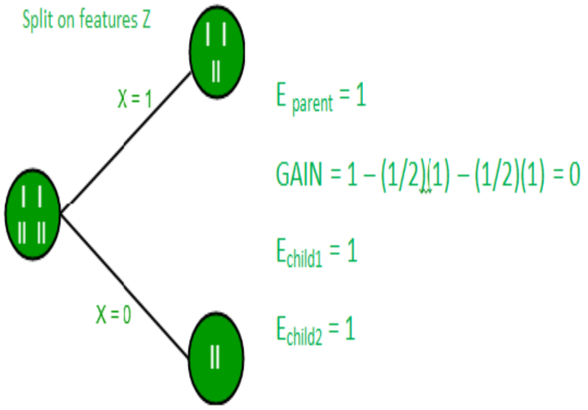
Here, we have 3 features and 2 output classes.
To build a decision tree using Information gain. We will take each of the feature and calculate the information for each feature.


Split on attribute X
X = 1
X = 0
E parent = 1
GAIN = 1 – (3/4)(0.9184) – (1/4)(0) = 0.3112
Echild1 = -(1/3)log2(1/3) – (2/3)log2(2/3)
= 0.5284 + 0.39
= 0.9184
Echild2 = 0

**Split on feature X**



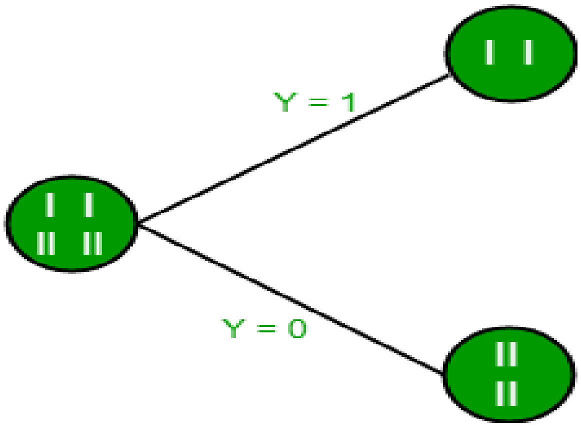
**Split on feature Y**



**Split on feature Z**

From the above images we can see that the information gain is maximum when we make a split on feature Y. So, for the root node best suited feature is feature Y. Now we can see that while splitting the dataset by feature Y, the child contains pure subset of the target variable. So we don't need to further split the dataset.

The final tree for the above dataset would be look like this:



**2. Gini Index**

- Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified.
- It means an attribute with lower Gini index should be preferred.
- Sklearn supports "Gini" criteria for Gini Index and by default, it takes "gini" value.
- The Formula for the calculation of the of the Gini Index is given below.

**Example:**

Lets consider the dataset in the image below and draw a decision tree using gini index.

| INDEX | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 4.8 | 3.4 | 1.9 | 0.2 | positive |
| 2 | 5 | 3 | 1.6 | 1.2 | positive |
| 3 | 5 | 3.4 | 1.6 | 0.2 | positive |
| 4 | 5.2 | 3.5 | 1.5 | 0.2 | positive |
| 5 | 5.2 | 3.4 | 1.4 | 0.2 | positive |
| 6 | 4.7 | 3.2 | 1.6 | 0.2 | positive |
| 7 | 4.8 | 3.1 | 1.6 | 0.2 | positive |

| INDEX | A | B | C | D | E |
|---|---|---|---|---|---|
| 8 | 5.4 | 3.4 | 1.5 | 0.4 | positive |
| 9 | 7 | 3.2 | 4.7 | 1.4 | negative |
| 10 | 6.4 | 3.2 | 4.7 | 1.5 | negative |
| 11 | 6.9 | 3.1 | 4.9 | 1.5 | negative |
| 12 | 5.5 | 2.3 | 4 | 1.3 | negative |
| 13 | 6.5 | 2.8 | 4.6 | 1.5 | negative |
| 14 | 5.7 | 2.8 | 4.5 | 1.3 | negative |
| 15 | 6.3 | 3.3 | 4.7 | 1.6 | negative |
| 16 | 4.9 | 2.4 | 3.3 | 1 | negative |

In the dataset above there are 5 attributes from which attribute E is the predicting feature which contains 2(Positive & Negative) classes. We have an equal proportion for both the classes.
In Gini Index, we have to choose some random values to categorize each attribute. These values for this dataset are:

| A | B | C | D |
|---|---|---|---|
| >= 5 | >= 3.0 | >= 4.2 | >= 1.4 |
| < 5 | < 3.0 | < 4.2 | < 1.4 |

**Calculating Gini Index for Var A:**
**Value >= 5: 12**

Attribute A >= 5 & class = positive:

Attribute A >= 5 & class = negative:

Gini(5, 7) = 1 –
**Value < 5: 4**

Attribute A < 5 & class = positive:

Attribute A < 5 & class = negative:

Gini(3, 1) = 1 –
By adding weight and sum each of the gini indices:

**Calculating Gini Index for Var B:**
**Value >= 3: 12**

Attribute B >= 3 & class = positive:

Attribute B >= 5 & class = negative:

Gini(5, 7) = 1 –
**Value < 3: 4**

Attribute A < 3 & class = positive:

Attribute A < 3 & class = negative:

Gini(3, 1) = 1 –
By adding weight and sum each of the gini indices:

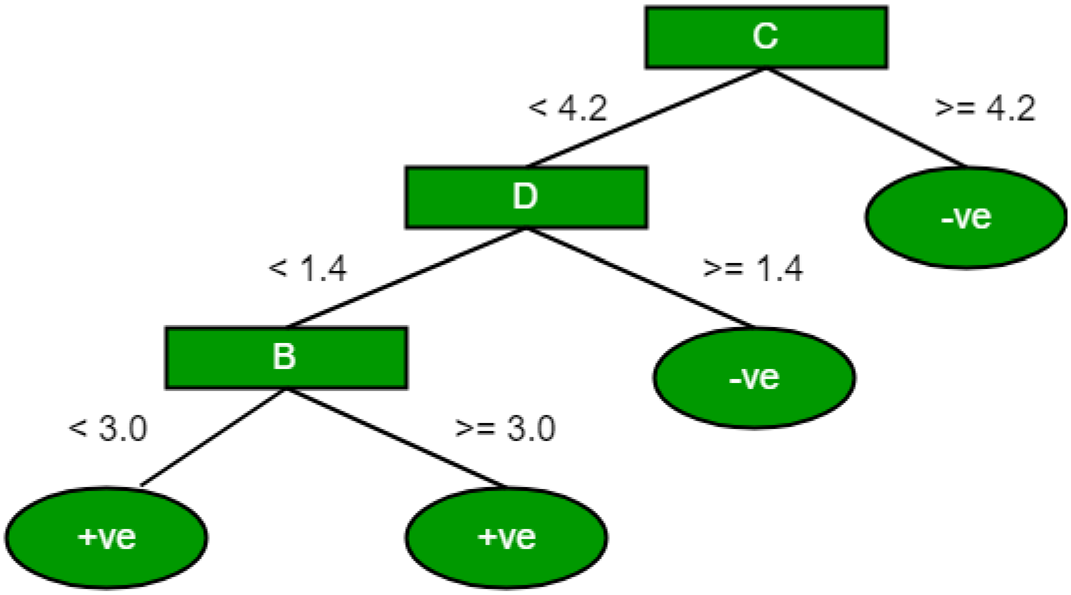
Using the same approach we can calculate the Gini index for C and D attributes.

```
            Positive    Negative
For A|>= 5.0    5       7
    |<5    3       1
Ginin Index of A = 0.45825
            Positive    Negative
For B|>= 3.0    8       4
    |< 3.0    0       4
Gini Index of B= 0.3345
            Positive    Negative
For C|>= 4.2    0       6
    |< 4.2    8       2
Gini Index of C= 0.2
```

| | | Positive | Negative |
|---|---|---|---|
| For D | >= 1.4 | 0 | 5 |
| | < 1.4 | 8 | 3 |

Gini Index of D= 0.273

Decision tree for above dataset



The most notable types of decision tree algorithms are:-

1. **Iterative Dichotomiser 3 (ID3):** This algorithm uses Information Gain to decide which attribute is to be used classify the current subset of the data. For each level of the tree, information gain is calculated for the remaining data recursively.

2. **C4.5:** This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifying attribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.

3. **Classification and Regression Tree(CART):** It is a dynamic learning algorithm which can produce a regression tree as well as a classification tree depending upon the dependent variable.

**Naive Bayes Classifiers:**

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

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

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

Here is a tabular representation of our dataset.

| | OUTLOOK | TEMPERATURE | HUMIDITY | WINDY | PLAY GOLF |
|---|---|---|---|---|---|
| 0 | Rainy | Hot | High | False | No |
| 1 | Rainy | Hot | High | True | No |
| 2 | Overcast | Hot | High | False | Yes |
| 3 | Sunny | Mild | High | False | Yes |
| 4 | Sunny | Cool | Normal | False | Yes |
| 5 | Sunny | Cool | Normal | True | No |
| 6 | Overcast | Cool | Normal | True | Yes |
| 7 | Rainy | Mild | High | False | No |
| 8 | Rainy | Cool | Normal | False | Yes |
| 9 | Sunny | Mild | Normal | False | Yes |
| 10 | Rainy | Mild | Normal | True | Yes |

| | OUTLOOK | TEMPERATURE | HUMIDITY | WINDY | PLAY GOLF |
|---|---|---|---|---|---|
| 11 | Overcast | Mild | High | True | Yes |
| 12 | Overcast | Hot | Normal | False | Yes |
| 13 | Sunny | Mild | High | True | No |

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

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

### Assumption:

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

- independent
- equal

contribution to the outcome.

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

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

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

## Bayes’ Theorem

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

where A and B are events and P(B) ? 0.

- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as **evidence**.
- P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.
  Now, with regards to our dataset, we can apply Bayes’ theorem in following way:

where, y is class variable and X is a dependent feature vector (of size *n*) where:

Just to clear, an example of a feature vector and corresponding class variable can be: (refer 1st row of dataset)

```
X = (Rainy, Hot, High, False)

y = No
```

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

**Naive assumption**:

Now, its time to put a naive assumption to the Bayes' theorem, which is, **independence** among the features. So now, we split **evidence** into the independent parts.

Now, if any two events A and B are independent, then,

```
P(A,B) = P(A)P(B)
```

Hence, we reach to the result:

which can be expressed as:

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

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

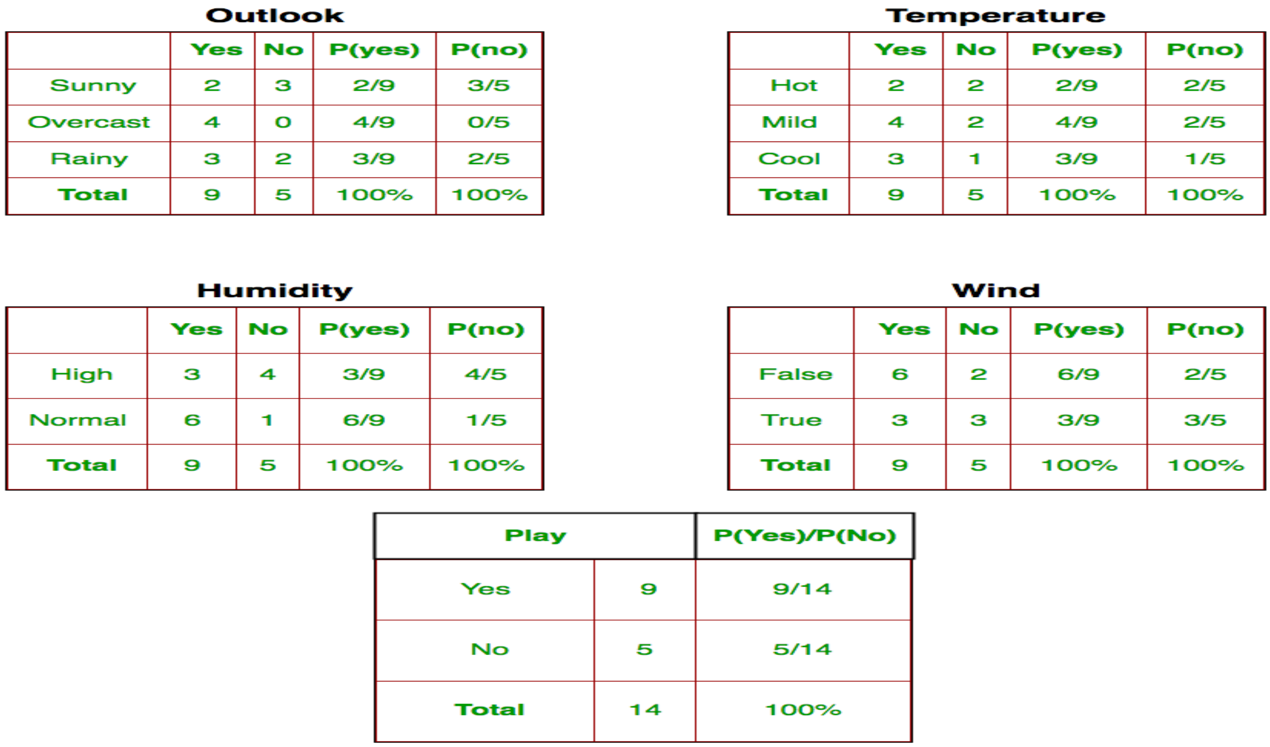
So, finally, we are left with the task of calculating P(y) and $P(x_i \mid y)$.

Please note that P(y) is also called **class probability** and $P(x_i \mid y)$ is called **conditional probability**.

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i \mid y)$.

Let us try to apply the above formula manually on our weather dataset. For this, we need to do some precomputations on our dataset.

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



Outlook

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

Temperature

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

Humidity

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

Wind

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

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

So, in the figure above, we have calculated P($x_i$ | $y_j$) for each $x_i$ in X and $y_j$ in y manually in the tables 1-4. For example, probability of playing golf given that the temperature is cool, i.e P(temp. = cool | play golf = Yes) = 3/9.

Also, we need to find class probabilities (P(y)) which has been calculated in the table 5. For example, P(play golf = Yes) = 9/14.

**Cluster Analysis :**

It is a class of techniques that are used to classify objects or cases into relative groups called clusters. Cluster analysis is also called classification analysis or numerical taxonomy. In cluster analysis, there is no prior information about the group or cluster membership for any of the objects.

Cluster Analysis has been used in marketing for various purposes. Segmentation of consumers in cluster analysis is used on the basis of benefits sought from the purchase of the product. It can be used to identify homogeneous groups of buyers.

Cluster analysis involves formulating a problem, selecting a distance measure, selecting a clustering procedure, deciding the number of clusters, interpreting the profile clusters and finally, assessing the validity of clustering.

The variables on which the cluster analysis is to be done should be selected by keeping past research in mind. It should also be selected by theory, the hypotheses being tested, and the judgment of the researcher. An appropriate measure of distance or similarity should be selected; the most commonly used measure is the Euclidean distance or its square.

Clustering procedures in cluster analysis may be hierarchical, non-hierarchical, or a two-step procedure. A hierarchical procedure in cluster analysis is characterized by the development of a tree like structure. A hierarchical procedure can be agglomerative or divisive. Agglomerative methods in cluster analysis consist of linkage methods, variance methods, and centroid methods. Linkage methods in cluster analysis are comprised of single linkage, complete linkage, and average linkage.

The non-hierarchical methods in cluster analysis are frequently referred to as K means clustering. The two-step procedure can automatically determine the optimal number of clusters by comparing the values of model choice criteria across different clustering solutions. The choice of clustering procedure and the choice of distance measure are interrelated. The relative sizes of clusters in cluster analysis should be meaningful. The clusters should be interpreted in terms of cluster centroids.

## There are certain concepts and statistics associated with cluster analysis:

- Agglomeration schedule in cluster analysis gives information on the objects or cases being combined at each stage of the hierarchical clustering process.
- Cluster Centroid is the mean value of a variable for all the cases or objects in a particular cluster.
- A dendrogram is a graphical device for displaying cluster results.

- Distances between cluster centers in cluster analysis indicate how separated the individual pairs of clusters are. The clusters that are widely separated are distinct and therefore desirable.
- Similarity/distance coefficient matrix in cluster analysis is a lower triangle matrix containing pairwise distances between objects or cases.

**4 Types of Cluster Analysis Techniques**

Cluster analysis is the task of grouping a set of data points in such a way that they can be characterized by their relevancy to one another. These techniques create clusters that allow us to understand how our data is related. The most common applications of cluster analysis in a business setting is to segment customers or activities.

**Centroid Clustering:**

This is one of the more common methodologies used in cluster analysis. In centroid cluster analysis you choose the number of clusters that you want to classify. For example, if you're a pet store owner you may choose to segment your customer list by people who bought dog and/or cat products.

The algorithm will start by randomly selecting centroids (cluster centers) to group the data points into the two pre-defined clusters. A line is then drawn separating the data points into the two clusters based on their proximity to the centroids. The algorithm will then reposition the centroid relative to all the points within each cluster. The centroids and points in a cluster will adjust through all iteratations, resulting in optimized clusters. The result of this analysis is the segmentation of your data into the two clusters. In this example, the data set will be segmented into customers who are own dogs and cats.

**Density Clustering:**

Density clustering groups data points by how densely populated they are. To group closely related data points, this algorithm leverages the understanding that the more dense the data points...the more related they are. To determine this, the algorithm will select a random point then start measuring the distance between each point around it. For most density algorithms a predetermined distance between data points is selected to benchmark how closely points need to be to one another to be considered related.. Then, the algorithm will identify all other points that are within the allowed distance of relevance. This process will continue to iterate by selecting different random data points to start with until the best clusters can be identified.

**Distribution Clustering:**

Distribution clustering identifies the probability that a point belongs to a cluster. Around each possible centroid The algorithm defines the density distributions for each cluster, quantifying the probability of belonging based on those distributions The algorithm optimizes the characteristics of the distributions to best represent the data.

These maps look a lot like targets at an archery range. In the event that a data point hits the bulls eye on the map, then the probability of that person/object belonging to that cluster is 100%. Each ring around the bulls eye represents lessening percentage or certainty.

Distribution clustering is a great technique to assign outliers to clusters, where as density clustering will not assign an outlier to acluster.

**Connectivity Clustering:**

Unlike the other three techniques of clustering analysis reviewed above, connectivity clustering initially recognizes each data point as its own cluster. The primary premise of this technique is that points closer to each other are more related. The iterative process of this algorithm is to continually incorporate a data point or group of data points with other data points and/or groups until all points are engulfed into one big cluster. The critical input for this type of algorithm is determining where to stop the grouping from getting bigger.

**Applications of Cluster Analysis**

- Clustering analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.
- Clustering can also help marketers discover distinct groups in their customer base. And they can characterize their customer groups based on the purchasing patterns.
- In the field of biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionalities and gain insight into structures inherent to populations.
- Clustering also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according to house type, value, and geographic location.
- Clustering also helps in classifying documents on the web for information discovery.
- Clustering is also used in outlier detection applications such as detection of credit card fraud.

## Requirements of Clustering in Data Mining

The following points throw light on why clustering is required in data mining −

- **Scalability** − We need highly scalable clustering algorithms to deal with large databases
- **Ability to deal with different kinds of attributes** − Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.
- **Discovery of clusters with attribute shape** − The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.
- **High dimensionality** − The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data** − Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability** − The clustering results should be interpretable, comprehensible, and usable.

## Clustering Methods

Clustering methods can be classified into the following categories −

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

### Partitioning Method:

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

- Each group contains at least one object.
- Each object must belong to exactly one group.

**Points to remember −**

- For a given number of partitions (say k), the partitioning method will create an initial partitioning.
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

### Hierarchical Methods:

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

- Agglomerative Approach
- Divisive Approach

### Agglomerative Approach:

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

### Divisive Approach:

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

**Approaches to Improve Quality of Hierarchical Clustering:**

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

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

**Density-based Method:**

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

**Grid-based Method:**

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

**Advantages:**

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

**Model-based methods:**

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

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