SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

UNIT – V - Data Mining and Data Warehousing - SIT1301

CLUSTERING, APPLICATIONS AND TRENDS IN DATA MINING

Cluster analysis - Types of data in Cluster Analysis - Categorization of major clustering methods -Partitioning methods - K Means - K Medoids - Hierarchical methods - Density-based methods - Grid-based methods - Model based clustering methods - Constraint Based cluster analysis - Outlier analysis - Data Mining Spatial Applications.

CLUSTER ANALYSIS

Cluster is a group of objects that belongs to the same class. In other words, similar objects aregrouped in one cluster and dissimilar objects are grouped in another cluster.

What is Clustering?

Clustering is the process of making a group of abstract objects into classes of similar objects.

Points to Remember

- A cluster of data objects can be treated as one group.
- While doing cluster analysis, we first partition the set of data into groups based on datasimilarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that, it is adaptable to changes andhelps single out useful features that distinguish different groups.

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.
- As a data mining function, cluster analysis serves as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

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.

Types of Data in Cluster Analysis

1. Interval - scaled variable
2. Binary Variables
3. Nominal (Categorical) Variables
4. Ordinal Variables
5. Ratio-Scaled Variables
6. Variables of Mixed Types

CATEGORIZATION OF MAJOR CLUSTERING METHODS

Clustering methods can be classified into the following categories −

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

Partitioning Method

Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partition of data. Each partition will represent a cluster and $k \leq n$. It means that it will classify thedata 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., oncea 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.

Advantage

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

Model-based methods

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

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

Constraint-based Method

In this method, the clustering is performed by the incorporation of user or application-oriented constraints. A constraint refers to the user expectation or the properties of desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. Constraints can be specified by the user or the application requirement.

PARTITIONING METHODS

K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

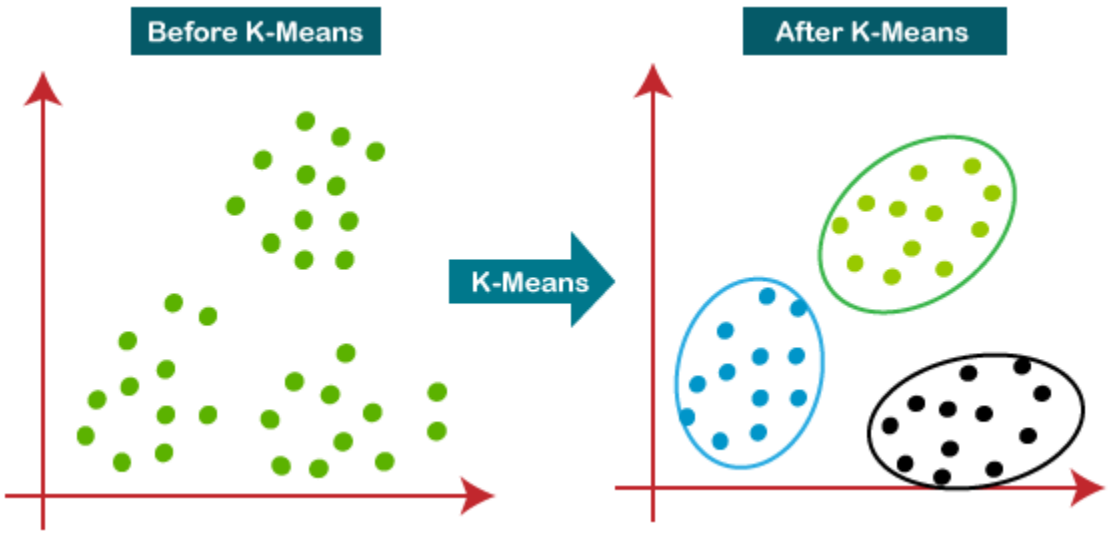
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

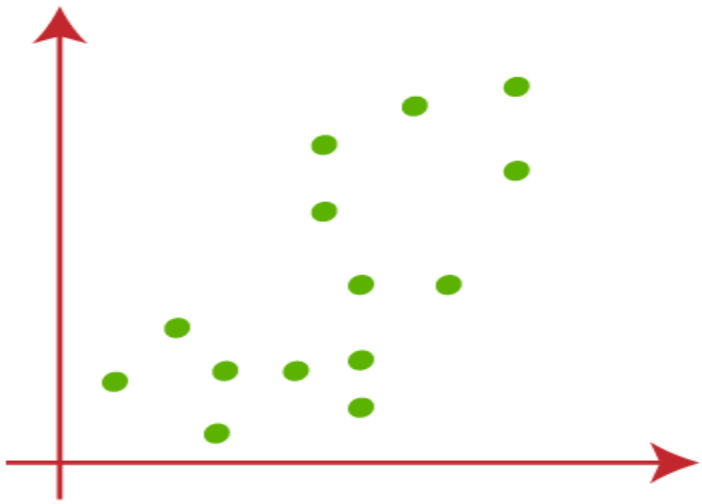
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

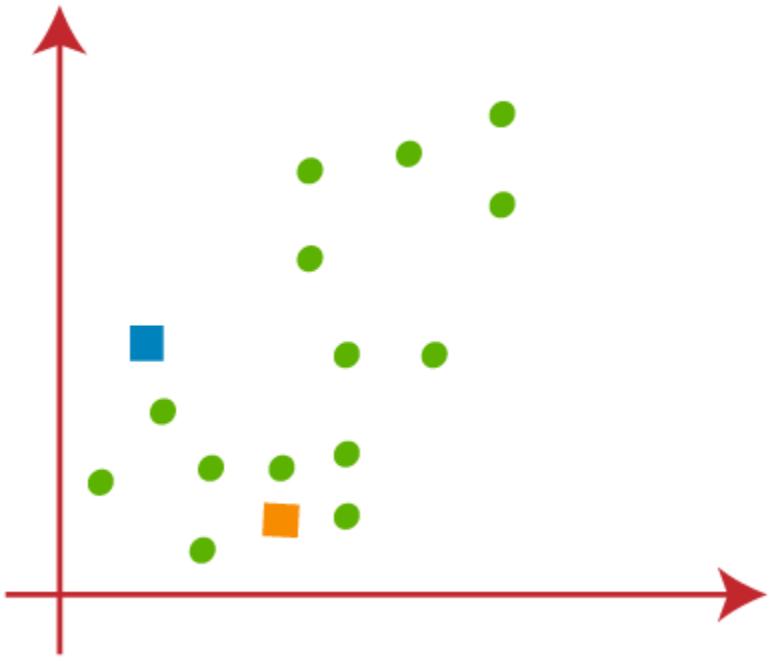
Let's understand the above steps by considering the visual plots:

Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:

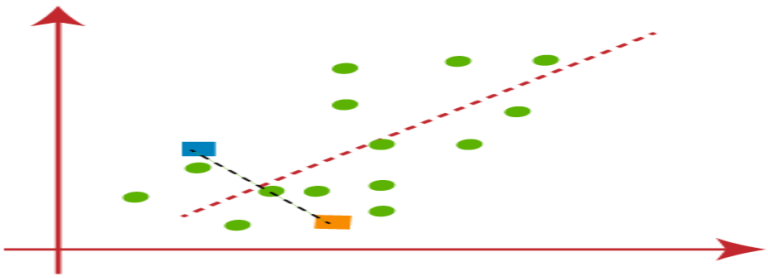


- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the

below image:

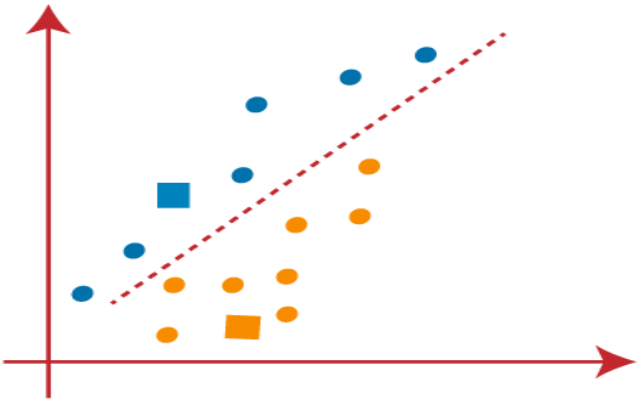


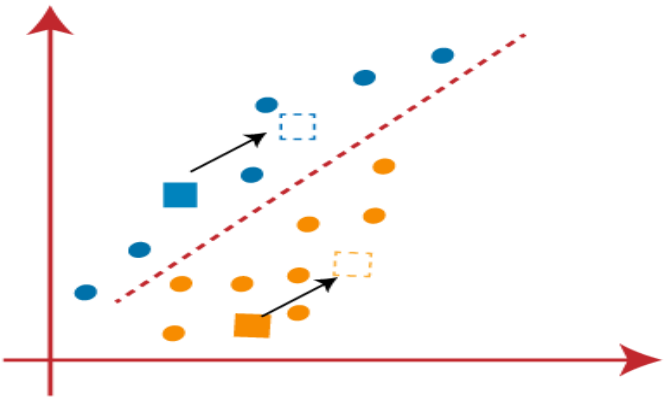
- Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



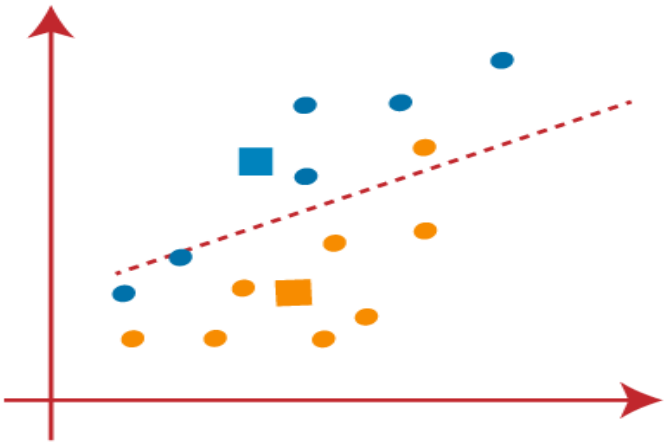
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.

- As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:

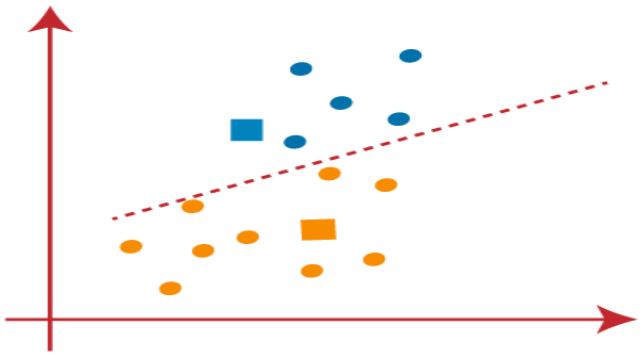




Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

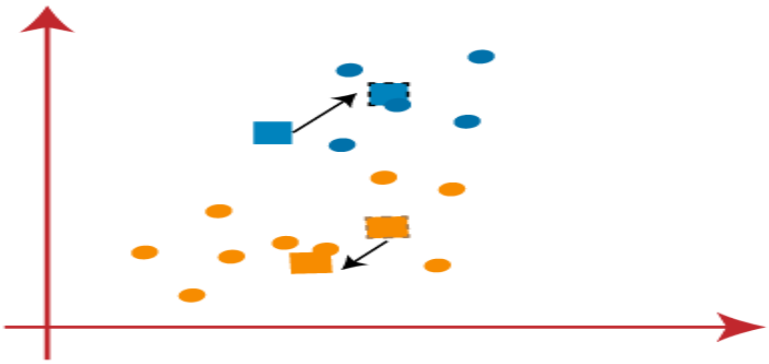


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

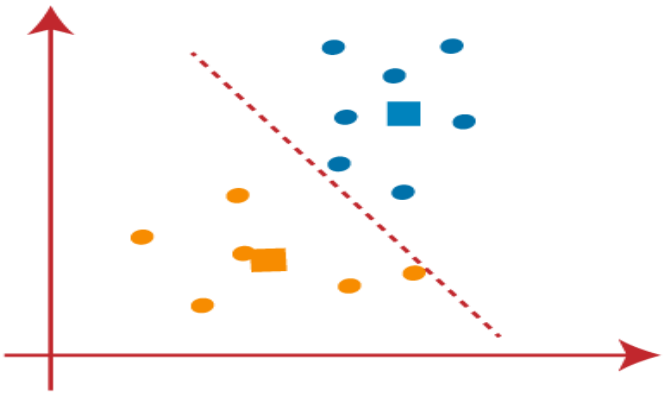


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

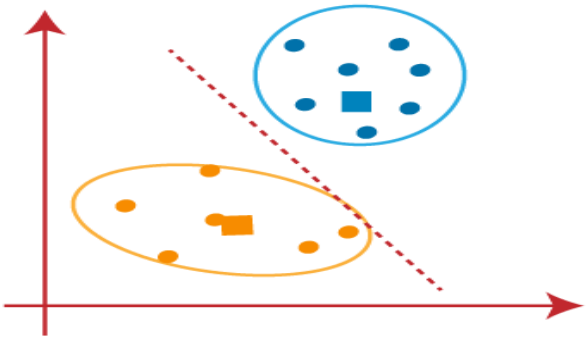
- We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



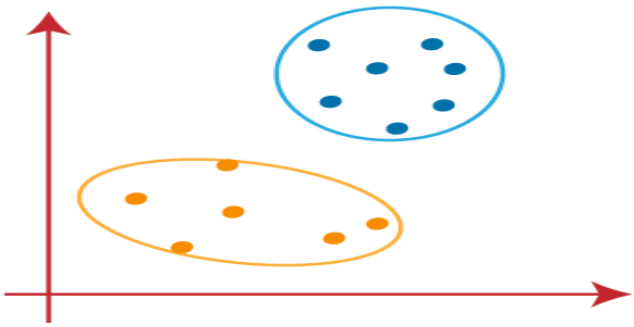
- As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



- We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:



As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



How to choose the value of "K number of clusters" in K-means Clustering?

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$WCSS= \sum_{Pi \text{ in Cluster1}} distance(P_i\ C_1)^2 + \sum_{Pi \text{ in Cluster2}} distance(P_i\ C_2)^2 + \sum_{Pi \text{ in CLuster3}} distance(P_i\ C_3)^2$$

In the above formula of WCSS,

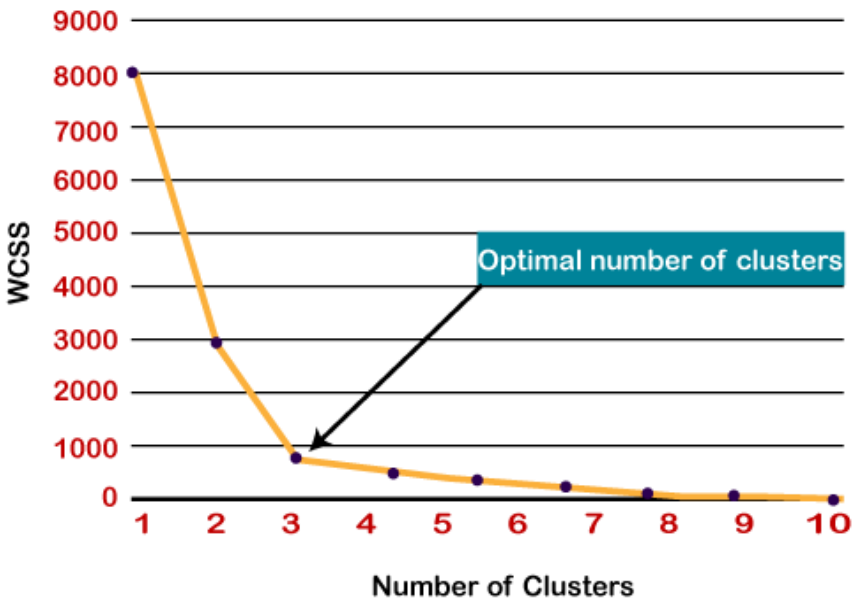
$\sum_{Pi \text{ in Cluster1}} distance(P_i\ C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



We can choose the number of clusters equal to the given data points. If we choose the number of clusters equal to the data points, then the value of WCSS becomes zero, and that will be the endpoint of the plot.

In the above section, we have discussed the K-means algorithm, now let's see how it can be implemented using Python.

Before implementation, let's understand what type of problem we will solve here. So, we have a dataset of **Mall_Customers**, which is the data of customers who visit the mall and spend there.

In the given dataset, we have **Customer_Id, Gender, Age, Annual Income ($), and Spending Score** (which is the calculated value of how much a customer has spent in the mall, the more the value, the more he has spent). From this dataset, we need to calculate some patterns, as it is an unsupervised method, so we don't know what to calculate exactly.

The steps to be followed for the implementation are given below:

- **Data Pre-processing**
- **Finding the optimal number of clusters using the elbow method**
- **Training the K-means algorithm on the training dataset**
- **Visualizing the clusters**

Step-1: Data pre-processing Step

The first step will be the data pre-processing, as we did in our earlier topics of Regression and Classification. But for the clustering problem, it will be different from other models. Let's discuss it:

- **Importing Libraries**

  As we did in previous topics, firstly, we will import the libraries for our model, which is part of data pre-processing. The code is given below:

```
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

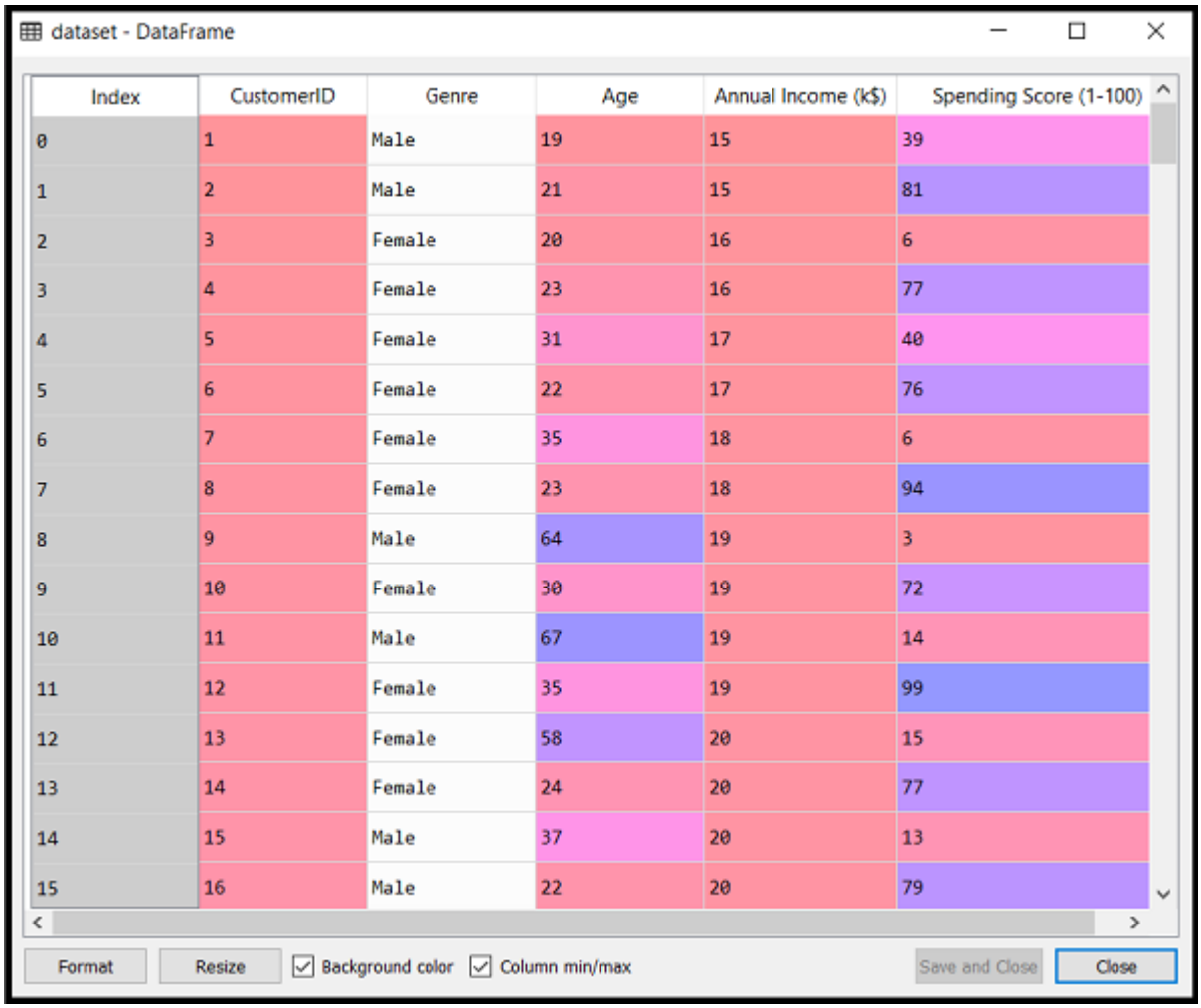
In the above code, the numpy we have imported for the performing mathematics calculation, **matplotlib** is for plotting the graph, and **pandas** are for managing the dataset.

- **Importing the Dataset:**

  Next, we will import the dataset that we need to use. So here, we are using the Mall_Customer_data.csv dataset. It can be imported using the below code:

```
# Importing the dataset
dataset = pd.read_csv('Mall_Customers_data.csv')
```

By executing the above lines of code, we will get our dataset in the Spyder IDE. The dataset looks like the below image:



Index	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72
10	11	Male	67	19	14
11	12	Female	35	19	99
12	13	Female	58	20	15
13	14	Female	24	20	77
14	15	Male	37	20	13
15	16	Male	22	20	79

From the above dataset, we need to find some patterns in it.

- **Extracting Independent Variables**

Here we don't need any dependent variable for data pre-processing step as it is a clustering problem, and we have no idea about what to determine. So we will just add a line of code for the matrix of features.

```
x = dataset.iloc[:, [3, 4]].values
```

As we can see, we are extracting only 3rd and 4th feature. It is because we need a 2d plot to visualize the model, and some features are not required, such as customer_id.

Step-2: Finding the optimal number of clusters using the elbow method

In the second step, we will try to find the optimal number of clusters for our clustering problem. So, as discussed above, here we are going to use the elbow method for this purpose.

As we know, the elbow method uses the WCSS concept to draw the plot by plotting WCSS values on the Y-axis and the number of clusters on the X-axis. So we are going to calculate the value for WCSS for different k values ranging from 1 to 10. Below is the code for it:

```
#finding optimal number of clusters using the elbow method
from sklearn.cluster import KMeans
wcss_list= []  #Initializing the list for the values of WCSS
#Using for loop for iterations from 1 to 10.
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1, 11), wcss_list)
mtp.title('The Elobw Method Graph')
mtp.xlabel('Number of clusters(k)')
mtp.ylabel('wcss_list')
mtp.show()
```

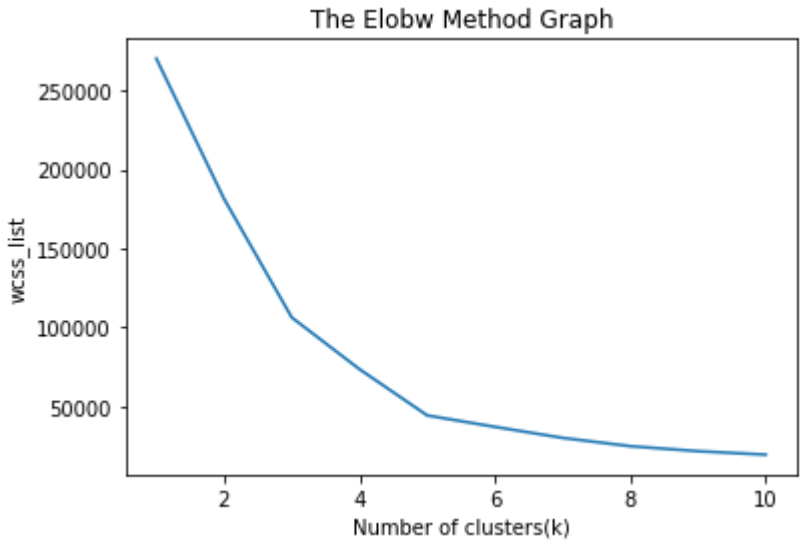
As we can see in the above code, we have used **the KMeans** class of sklearn. cluster library to form the clusters.

Next, we have created the **wcss_list** variable to initialize an empty list, which is used to contain the value of wcss computed for different values of k ranging from 1 to 10.

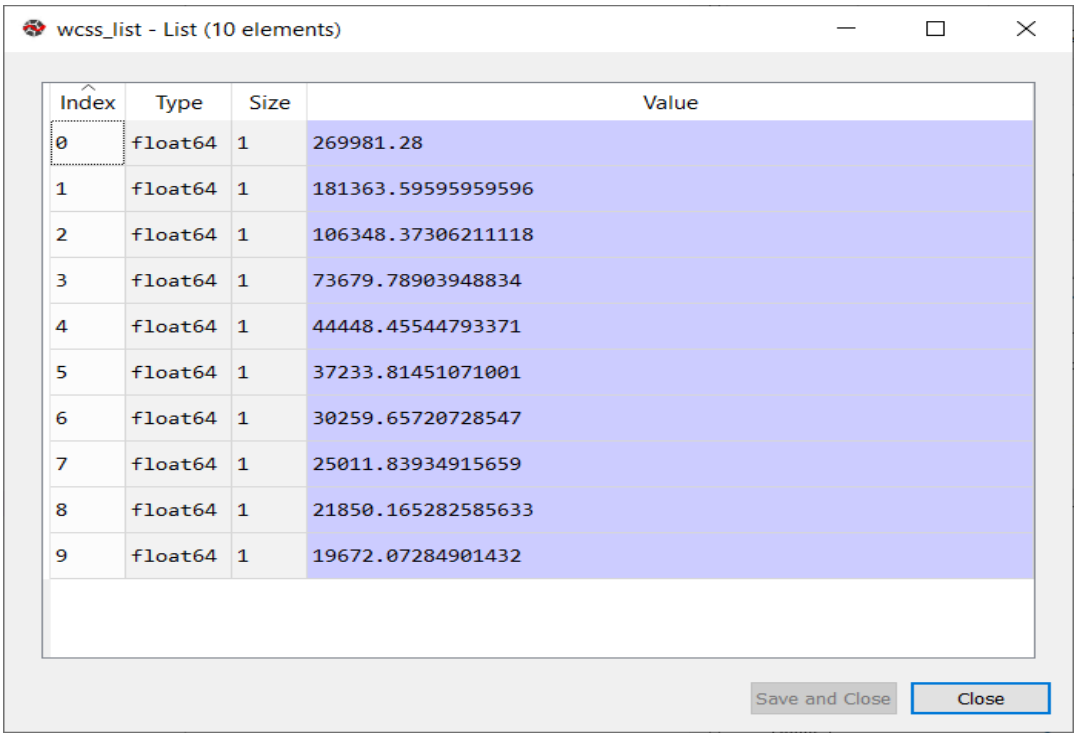
After that, we have initialized the for loop for the iteration on a different value of k ranging from 1 to 10; since for loop in Python, exclude the outbound limit, so it is taken as 11 to include 10^{th} value.

The rest part of the code is similar as we did in earlier topics, as we have fitted the model on a matrix of features and then plotted the graph between the number of clusters and WCSS.

Output: After executing the above code, we will get the below output:



From the above plot, we can see the elbow point is at **5. So the number of clusters here will be 5.**



Step- 3: Training the K-means algorithm on the training dataset

As we have got the number of clusters, so we can now train the model on the dataset.

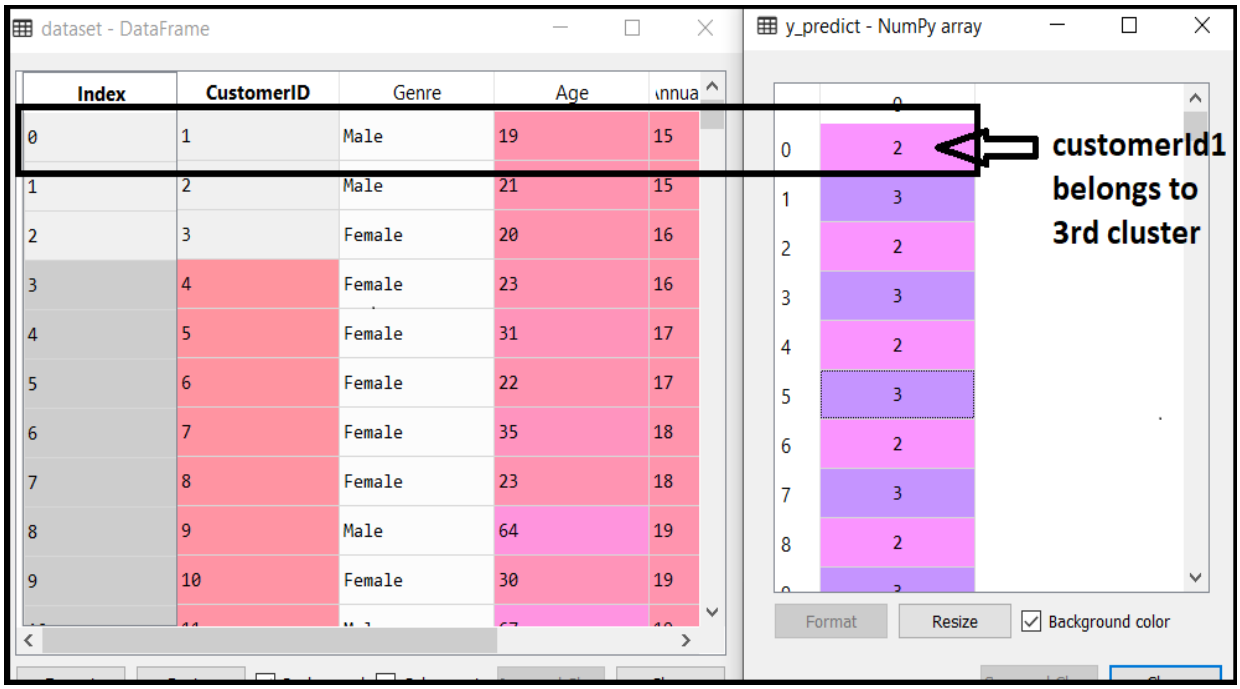
To train the model, we will use the same two lines of code as we have used in the above section, but here instead of using i, we will use 5, as we know there are 5 clusters that need to be formed. The code is given below:

```
#training the K-means model on a dataset
kmeans = KMeans(n_clusters=5, init='k-means++', random_state= 42)
y_predict= kmeans.fit_predict(x)
```

The first line is the same as above for creating the object of KMeans class.

In the second line of code, we have created the dependent variable **y_predict** to train the model.

By executing the above lines of code, we will get the y_predict variable. We can check it under **the variable explorer** option in the Spyder IDE. We can now compare the values of y_predict with our original dataset. Consider the below image:



From the above image, we can now relate that the CustomerID 1 belongs to a cluster

3(as index starts from 0, hence 2 will be considered as 3), and 2 belongs to cluster 4, and so on.

Step-4: Visualizing the Clusters

The last step is to visualize the clusters. As we have 5 clusters for our model, so we will visualize each cluster one by one.

To visualize the clusters will use scatter plot using mtp.scatter() function of matplotlib.

```
#visulaizing the clusters
mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c = 'blue', label = 'Cluster 1') #for first cluster
mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c = 'green', label = 'Cluster 2') #for second cluster
mtp.scatter(x[y_predict== 2, 0], x[y_predict == 2, 1], s = 100, c = 'red', label = 'Cluster 3') #for third cluster
mtp.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4') #for fourth cluster
mtp.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5') #for fifth cluster
mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroid')
mtp.title('Clusters of customers')
mtp.xlabel('Annual Income (k$)')
mtp.ylabel('Spending Score (1-100)')
mtp.legend()
mtp.show()
```

In above lines of code, we have written code for each clusters, ranging from 1 to 5. The first coordinate of the mtp.scatter, i.e., x[y_predict == 0, 0] containing the x value for the showing the matrix of features values, and the y_predict is ranging from 0 to 1.

Output:

The output image is clearly showing the five different clusters with different colors. The clusters are formed between two parameters of the dataset; Annual income of customer and Spending. We can change the colors and labels as per the requirement or choice. We can also observe some points from the above patterns, which are given below:

- **Cluster1** shows the customers with average salary and average spending so we can categorize these customers as
- Cluster2 shows the customer has a high income but low spending, so we can categorize them as **careful**.
- Cluster3 shows the low income and also low spending so they can be categorized as sensible.
- Cluster4 shows the customers with low income with very high spending so they can be categorized as **careless**.
- Cluster5 shows the customers with high income and high spending so they can be categorized as target, and these customers can be the most profitable customers for the mall owner.

K - MEDOIDS ALGORITHM

K-Medoids (also called as Partitioning Around Medoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw. A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.

The dissimilarity of the medoid(Ci) and object(Pi) is calculated by using E = |Pi - Ci|

The cost in K-Medoids algorithm is given as

Algorithm:

1. Initialize: select k random points out of the n data points as the medoids.

2. Associate each data point to the closest medoid by using any common distance metric methods.

3. While the cost decreases:

For each medoid m, for each data o point which is not a medoid:

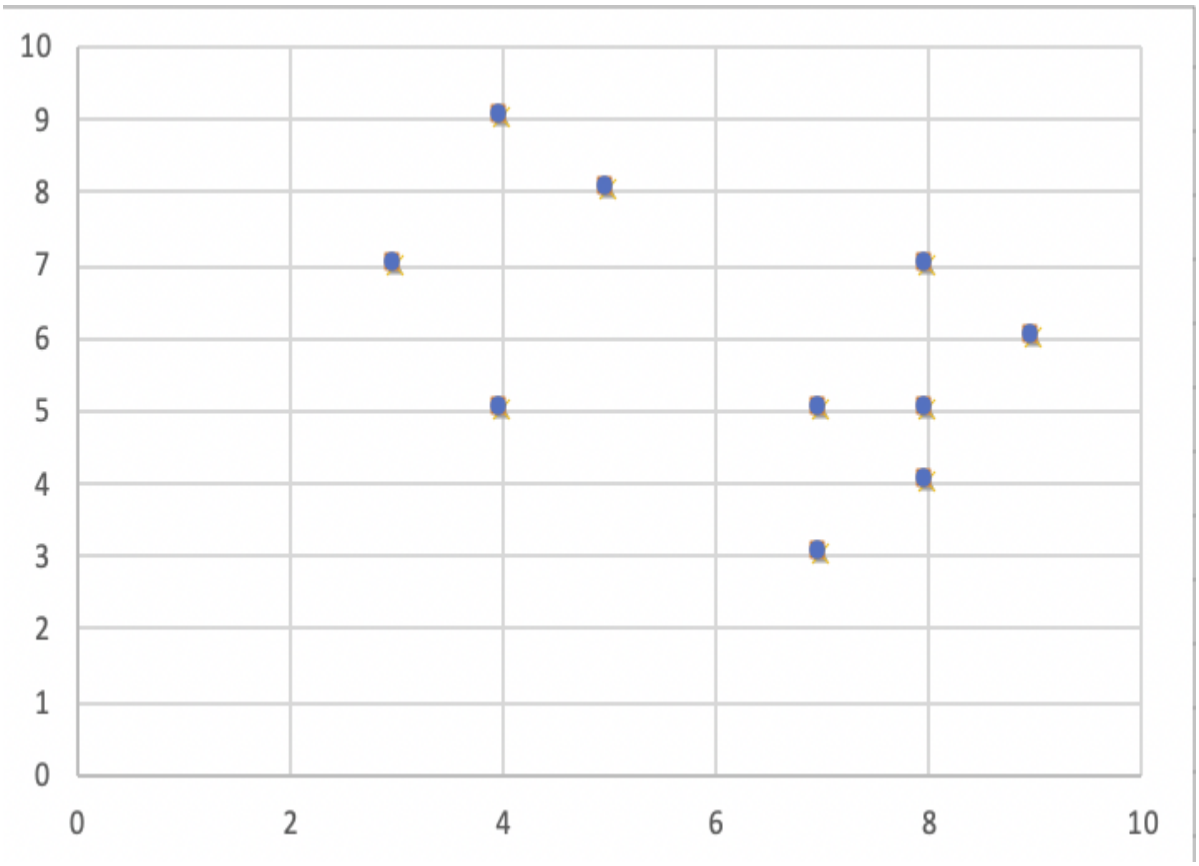
1. Swap m and o, associate each data point to the closest medoid, recompute the cost.

2. If the total cost is more than that in the previous step, undo the swap.

Let’s consider the following example:

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5

If a graph is drawn using the above data points, we obtain the following:



Step 1:

Let the randomly selected 2 medoids, so select k = 2 and let **C1 -(4, 5)** and **C2 -(8, 5)** are the two medoids.

Step 2: Calculating cost.

The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	8	5	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	4	5	-	-

Each point is assigned to the cluster of that medoid whose dissimilarity is less.

The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

Step 3: randomly select one non-medoid point and recalculate the cost.

Let the randomly selected point be (8, 4). The dissimilarity of each non-medoid point with the medoids – C1 (4, 5) and C2 (8, 4) is calculated and tabulated.

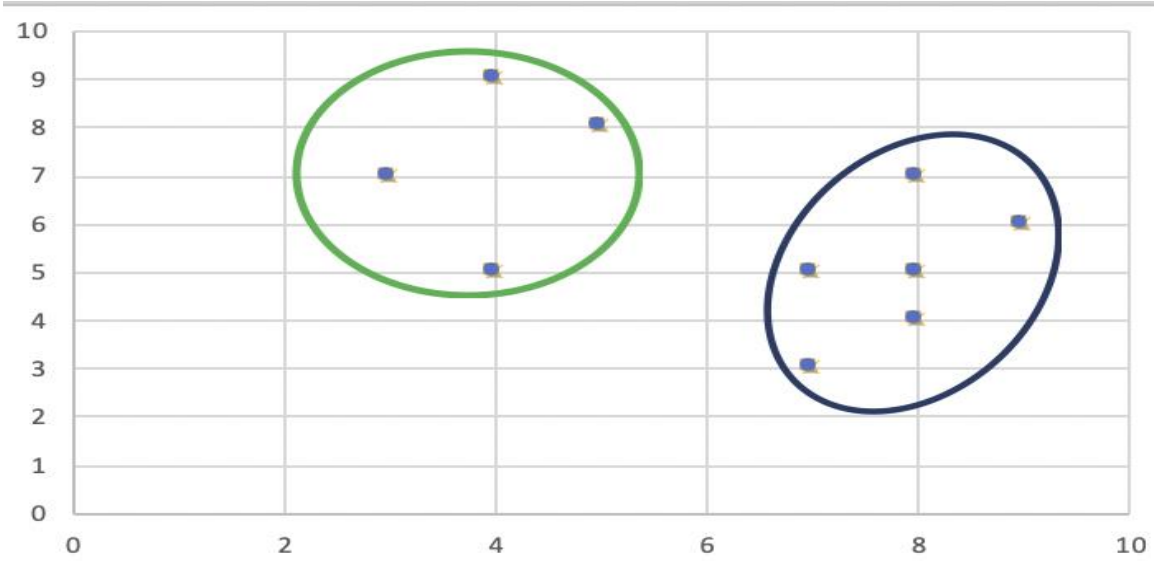
	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	8	4	-	-
8	7	5	3	2
9	4	5	-	-

Each point is assigned to that cluster whose dissimilarity is less. So, the points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

The New cost = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22

Swap Cost = New Cost – Previous Cost = 22 – 20 and **2 >0**

As the swap cost is not less than zero, we undo the swap. Hence (3, 4) and (7, 4) are the final medoids. The clustering would be in the following way



The **time complexity** is .

Advantages:

1. It is simple to understand and easy to implement.
2. K-Medoid Algorithm is fast and converges in a fixed number of steps.
3. PAM is less sensitive to outliers than other partitioning algorithms.

Disadvantages:

1. The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster centre) – briefly, it uses compactness as clustering criteria instead of connectivity.
2. It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.

Hierarchical Clustering in Data Mining

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

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

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

The basic method to generate hierarchical clustering are:

1. Agglomerative:

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

Algorithm for Agglomerative Hierarchical Clustering is:

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

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

Note:

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

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

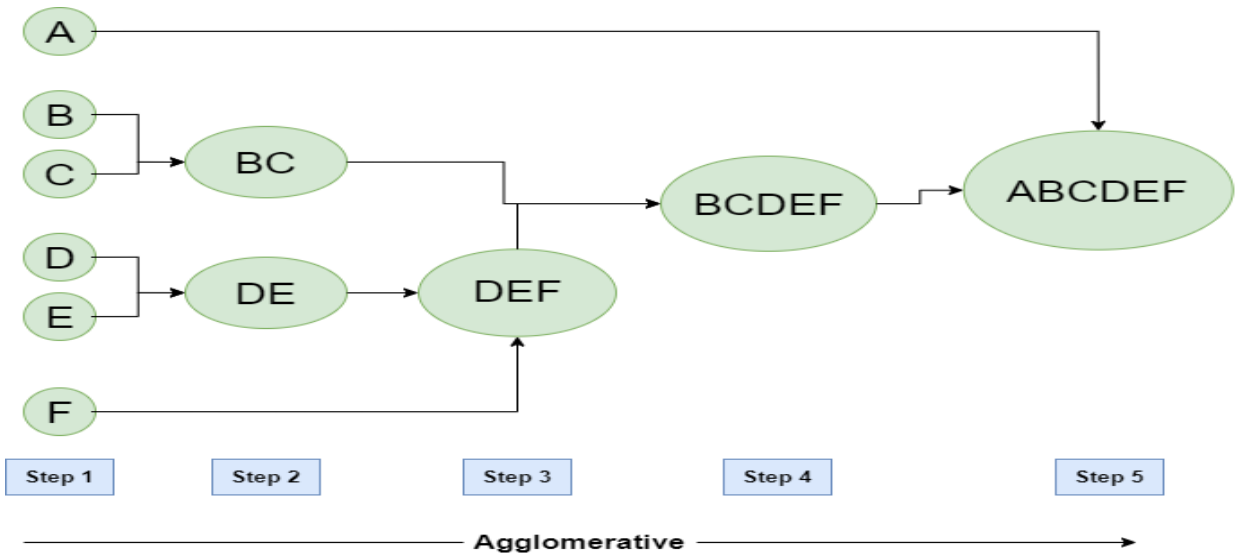


Figure – Agglomerative Hierarchical clustering

- **Step-1:**

  Consider each alphabet as a single cluster and calculate the distance of one cluster from all the other clusters.

- **Step-2:**

  In the second step comparable clusters are merged together to form a single cluster. Let’s say cluster (B) and cluster (C) are very similar to each other therefore we merge them in the second step similarly with cluster (D) and (E) and at last, we get the clusters

  [(A), (BC), (DE), (F)]

- **Step-3:**

  We recalculate the proximity according to the algorithm and merge the two nearest clusters([(DE), (F)]) together to form new clusters as [(A), (BC), (DEF)]

- **Step-4:**

  Repeating the same process; The clusters DEF and BC are comparable and merged together to form a new cluster. We're now left with clusters [(A), (BCDEF)].

- **Step-5:**

  At last the two remaining clusters are merged together to form a single cluster [(ABCDEF)].

2. Divisive:

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

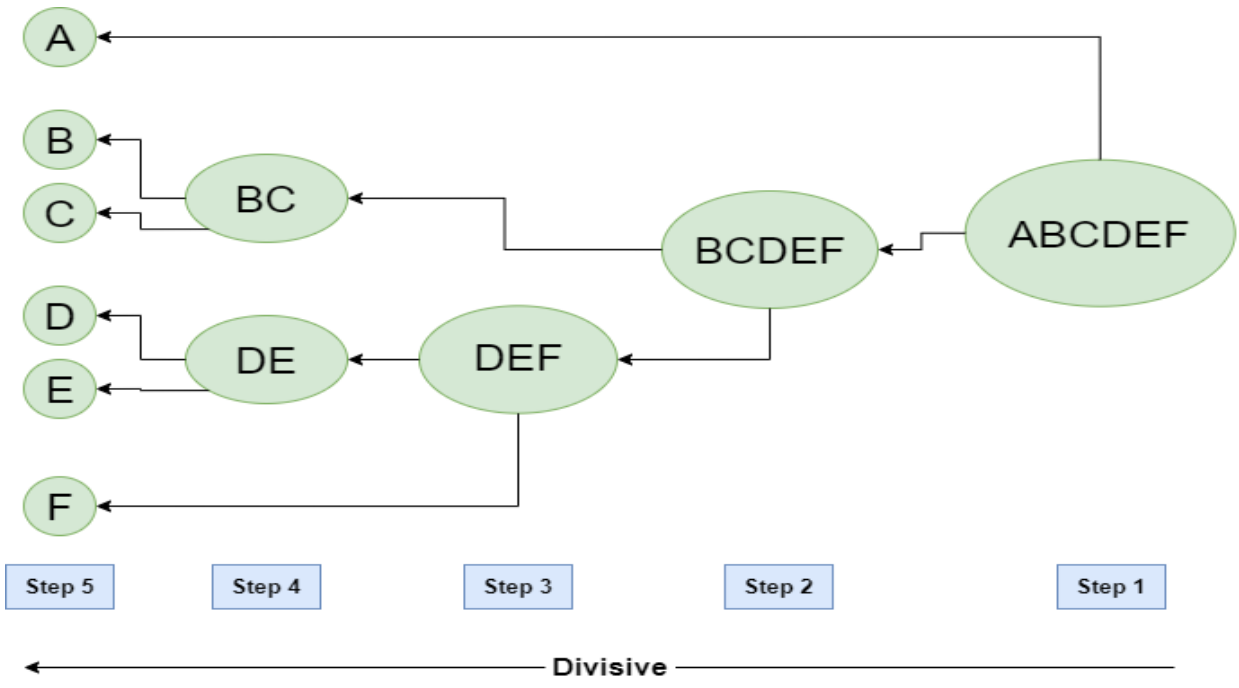


Figure – Divisive Hierarchical clustering

DENSITY BASED CLUSTERING METHOD - DB SCAN

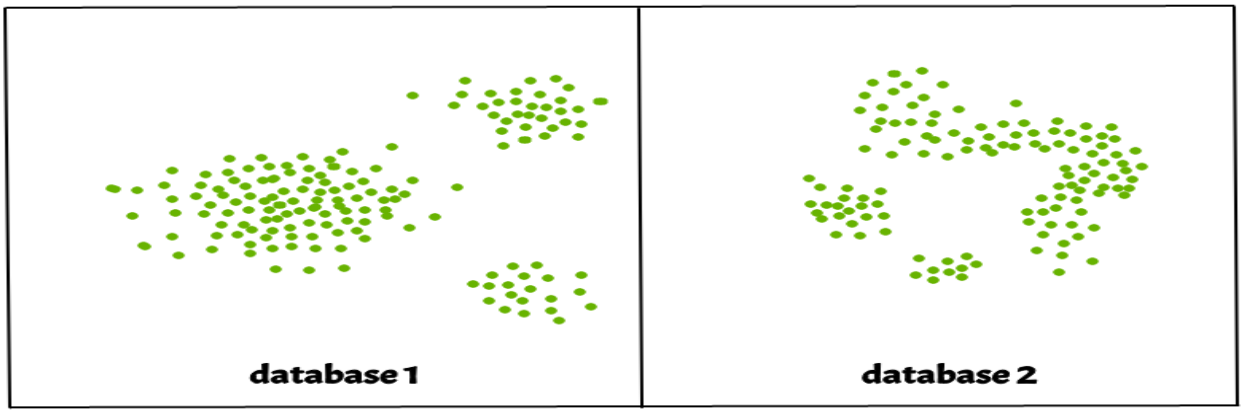
Clustering analysis or simply Clustering is basically an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups have different

properties in some sense. It comprises of many different methods based on different evolution.

E.g. K-Means (distance between points), Affinity propagation (graph distance), Mean-shift (distance between points), DBSCAN (distance between nearest points), Gaussian mixtures (Mahalanobis distance to centers), Spectral clustering (graph distance) etc.

Fundamentally, all clustering methods use the same approach i.e. first we calculate similarities and then we use it to cluster the data points into groups or batches. Here we will focus on **Density-based spatial clustering of applications with noise** (DBSCAN) clustering method.

Clusters are dense regions in the data space, separated by regions of the lower density of points. The **DBSCAN algorithm** is based on this intuitive notion of "clusters" and "noise". The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.



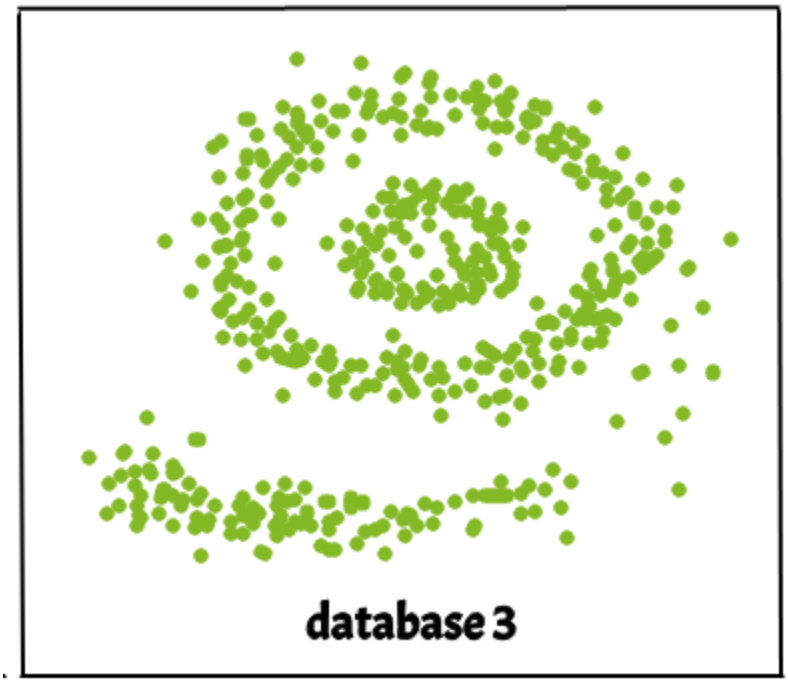
Why DBSCAN ?

Partitioning methods (K-means, PAM clustering) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.

Real life data may contain irregularities, like –

i) Clusters can be of arbitrary shape such as those shown in the figure below.

ii) Data may contain noise.



The figure below shows a data set containing nonconvex clusters and outliers/noises. Given such data, k-means algorithm has difficulties for identifying these clusters with arbitrary shapes.

DBSCAN algorithm requires two parameters –

1. **eps** : It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered as neighbors. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.
2. **MinPts**: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

   In this algorithm, we have 3 types of data points.

3.

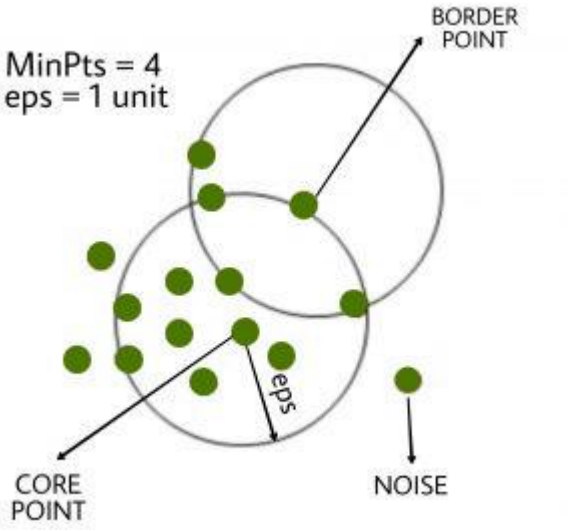
   Core Point*: A point is a core point if it has more than MinPts points within eps.*

   Border Point*: A point which has fewer than MinPts within eps but it is in the*

neighborhood of a core point.

***Noise or outlier**: A point which is not a core point or border point.*

4.



5.

DBSCAN algorithm can be abstracted in the following steps –

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density connected points and assign them to the same cluster as the core point.

   A point a and b are said to be density connected if there exist a point c which has a sufficient number of points in its neighbors and both the points a and b are within the eps distance. This is a chaining process. So, if b is neighbor of c, c is neighbor of d, d is neighbor of e, which in turn is neighbor of a implies that b is neighbor of a.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

Below is the DBSCAN clustering algorithm in pseudocode:

DBSCAN(dataset, eps, MinPts){

cluster index

```
C = 1
for each unvisited point p in dataset {
        mark p as visited
        # find neighbors
        Neighbors N = find the neighboring points of p

        if |N|>=MinPts:
           N = N U N'
           if p' is not a member of any cluster:
              add p' to cluster C
}
```

Implementation of above algorithm in Python :

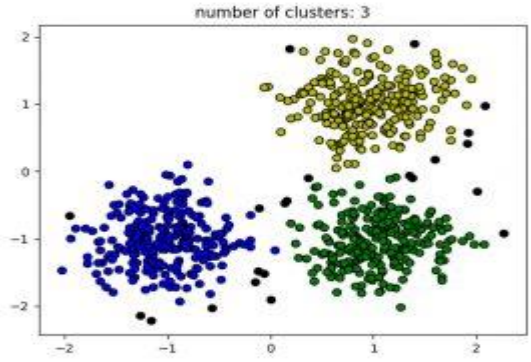
Here, we'll use the Python library sklearn to compute DBSCAN. We'll also use the matplotlib.pyplot library for visualizing clusters.

The dataset used can be found here.

```
import numpy as np
from sklearn.cluster import DBSCAN
from sklearn import metrics
from sklearn.datasets.samples_generator import make_blobs
from sklearn.preprocessing import StandardScaler
from sklearn import datasets

# Load data in X
db = DBSCAN(eps=0.3, min_samples=10).fit(X)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
```

```
# Number of clusters in labels, ignoring noise if present.
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)

print(labels)

# Plot result
import matplotlib.pyplot as plt

# Black removed and is used for noise instead.
unique_labels = set(labels)
colors = ['y', 'b', 'g', 'r']
print(colors)
for k, col in zip(unique_labels, colors):
    if k == -1:
        # Black used for noise.
        col = 'k'

    class_member_mask = (labels == k)

    xy = X[class_member_mask & core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=col,
                                 markeredgecolor='k',
                                 markersize=6)

    xy = X[class_member_mask & ~core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=col,
                                 markeredgecolor='k',
                                 markersize=6)

plt.title('number of clusters: %d' %n_clusters_)
plt.show()
```

Output:

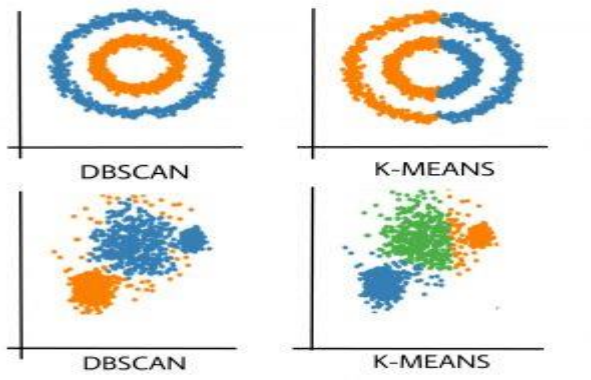


Black points represent outliers. By changing the eps and the MinPts , we can change the cluster configuration.

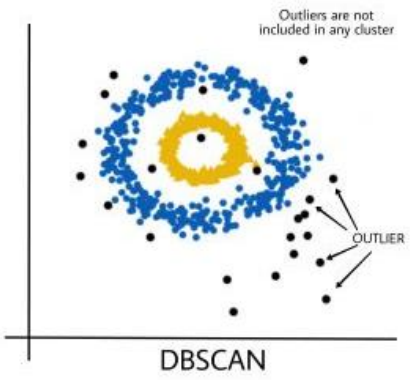
Now the question should be raised is – *Why should we use DBSCAN where K-Means is the widely used method in clustering analysis?*

Disadvantage Of K-MEANS:

1. K-Means forms spherical clusters only. This algorithm fails when data is not spherical (i.e. same variance in all directions).



2. K-Means algorithm is sensitive towards outlier. Outliers can skew the clusters in K-Means in very large extent.



3. K-Means algorithm requires one to specify the number of clusters a priory etc.

Basically, DBSCAN algorithm overcomes all the above-mentioned drawbacks of K-Means algorithm. DBSCAN algorithm identifies the dense region by grouping together data points that are closed to each other based on distance measurement.

OPTICS

OPTICS works like an extension of DBSCAN. The only difference is that it does not assign cluster memberships but stores the order in which the points are processed. So for each object stores: *Core distance* and Reachability distance. Order Seeds is called the record which constructs the output order.

Core Distance: The minimum value of ε which is present in the ε-neighborhood of a P is a core distance. Of course, it's needed to hold the minimum MinPts objects.

Reachability Distance: Reachability distance between p and q is defined as the least radius value that formulates p density reachable from q.

Algorithm

1. Randomly selects an unvisited point P
2. Selects all point's density reachable from P w.r.t Eps, MinPts.
3. Assign core distance & reachability distance = NULL
4. If P is not a core point
5. Move next point in the order Seeds list
6. If P is a core point
7. For each object q, in the ε — neighborhood of P
8. UPDATE reachability distance from P
9. If q is unvisited INSERT q into Order Seeds
10. Until no object is unvisited

Note: The algorithm above is taken from 'Review on Density-Based Clustering Algorithms for Big Data'.

Advantage

- It does not require density parameters.
- The clustering order is useful to extract the basic clustering information.

Disadvantage

- It only produces a cluster ordering.
- It can’t handle high dimensional data.

Grid-Based Clustering - STING, WaveCluster & CLIQUE

April 06, 2020

Grid-Based Clustering

Grid-Based Clustering method uses a multi-resolution grid data structure.

Several interesting methods

- **STING** (a **ST**atistical **IN**formation **G**rid approach)
- **WaveCluster** - A multi-resolution clustering approach using wavelet method
- **CLIQUE**

STING - A Statistical Information Grid Approach

STING was proposed by Wang, Yang, and Muntz.

In this method, the spatial area is divided into rectangular cells.

There are several levels of cells corresponding to different levels of resolution.For each cell, the high level is partitioned into several smaller cells in the next lower level.

The statistical info of each cell is calculated and stored beforehand and is used to answer queries.

The parameters of higher-level cells can be easily calculated from parameters of lower-level cell

- Count, mean, s, min, max
- Type of distribution—normal, uniform, etc.

Then using a top-down approach we need to answer spatial data queries. Then start from a pre-selected layer—typically with a small number of cells.

For each cell in the current level compute the confidence interval.
Now remove the irrelevant cells from further consideration.

When finishing examining the current layer, proceed to the next lower level.
Repeat this process until the bottom layer is reached.

Advantages:

It is Query-independent, easy to parallelize, incremental update.

O(K), where K is the number of grid cells at the lowest level.

Disadvantages:

All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected.

WaveCluster

It was proposed by Sheikholeslami, Chatterjee, and Zhang.

It is a multi-resolution clustering approach which applies wavelet transform to the feature space

- A wavelet transform is a signal processing technique that decomposes a signal into different frequency sub-band.
- It can be both grid-based and density-based method.

Input parameters:

- No of grid cells for each dimension
- The wavelet, and the no of applications of wavelet transform.

How to apply the wavelet transform to find clusters

- It summaries the data by imposing a multidimensional grid structure onto data space.
- These multidimensional spatial data objects are represented in an n-dimensional feature space.

- Now apply wavelet transform on feature space to find the dense regions in the feature space.
- Then apply wavelet transform multiple times which results in clusters at different scales from fine to coarse.

Why is wavelet transformation useful for clustering

- It uses hat-shape filters to emphasize region where points cluster, but simultaneously to suppress weaker information in their boundary.
- It is an effective removal method for outliers.
- It is of Multi-resolution method.
- It is cost-efficiency.

Major features:

- The time complexity of this method is O(N).
- It detects arbitrary shaped clusters at different scales.
- It is not sensitive to noise, not sensitive to input order.
- It only applicable to low dimensional data.

CLIQUE - Clustering In QUEst

It was proposed by Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).
It is based on automatically identifying the subspaces of high dimensional data space that allow better clustering than original space.

CLIQUE can be considered as both density-based and grid-based:

- It partitions each dimension into the same number of equal-length intervals.
- It partitions an m-dimensional data space into non-overlapping rectangular units.
- A unit is dense if the fraction of the total data points contained in the unit exceeds the input model parameter.

- A cluster is a maximal set of connected dense units within a subspace.

Partition the data space and find the number of points that lie inside each cell of the partition.

Identify the subspaces that contain clusters using the Apriori principle.

Identify clusters:

- Determine dense units in all subspaces of interests.
- Determine connected dense units in all subspaces of interests.

Generate minimal description for the clusters:

- Determine maximal regions that cover a cluster of connected dense units for each cluster.
- Determination of minimal cover for each cluster.

Advantages

It automatically finds subspaces of the highest dimensionality such that high-density clusters exist in those subspaces.

It is insensitive to the order of records in input and does not presume some canonical data distribution.

It scales linearly with the size of input and has good scalability as the number of dimensions in the data increases.

Disadvantages

The accuracy of the clustering result may be degraded at the expense of the simplicity of the method.

Summary

Grid-Based Clustering -> It is one of the methods of cluster analysis which uses a multi-resolution grid data structure.

MODEL BASED CLUSTERING

The traditional clustering methods, such as hierarchical clustering and k-means clustering, are heuristic and are not based on formal models. Furthermore, k-means algorithm is commonly randomnly initialized, so different runs of k-means will often yield different results. Additionally, k-means requires the user to specify the the optimal number of clusters.

An alternative is **model-based clustering**, which consider the data as coming from a distribution that is mixture of two or more clusters (Fraley and Raftery 2002, Fraley et al. (2012)). Unlike k-means, the model-based clustering uses a soft assignment, where each data point has a probability of belonging to each cluster.

Concept of model-based clustering

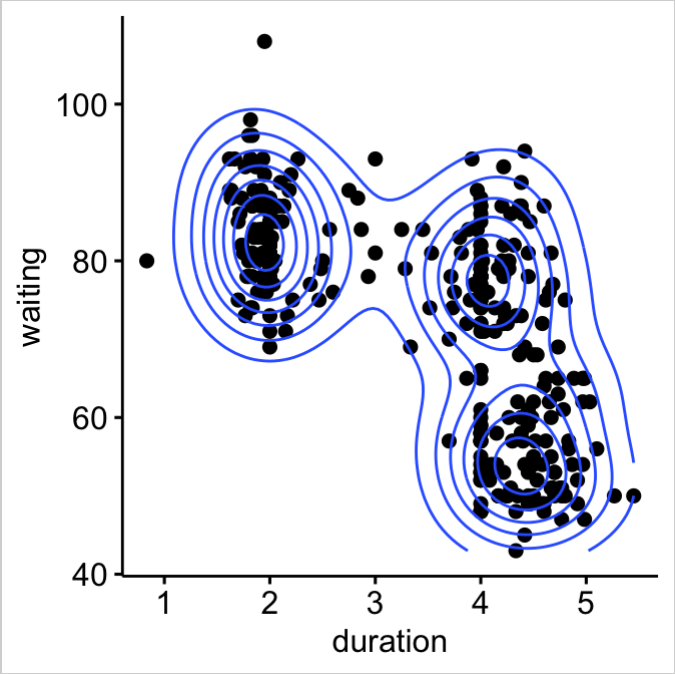
In model-based clustering, the data is considered as coming from a mixture of density.

Each component (i.e. cluster) k is modeled by the normal or Gaussian distribution which is characterized by the parameters:

- μkμk: mean vector,
- ∑k∑k: covariance matrix,
- An associated probability in the mixture. Each point has a probability of belonging to each cluster.

For example, consider the “old faithful geyser data” [in MASS R package], which can be illustrated as follow using the ggpubr R package:

```
# Load the data
library("MASS")

data("geyser")

# Scatter plot
library("ggpubr")

ggscatter(geyser, x = "duration", y = "waiting")+

 geom_density2d() # Add 2D density
```



The plot above suggests at least 3 clusters in the mixture. The shape of each of the 3 clusters appears to be approximately elliptical suggesting three bivariate normal distributions. As the 3 ellipses seems to be similar in terms of volume, shape and orientation, we might anticipate that the three components of this mixture might have homogeneous covariance matrices.

Estimating model parameters

The model parameters can be estimated using the *Expectation-Maximization* (EM) algorithm initialized by hierarchical model-based clustering. Each cluster k is centered at the means μkμk, with increased density for points near the mean.

Geometric features (shape, volume, orientation) of each cluster are determined by the covariance matrix ∑k∑k.

Different possible parameterizations of ∑k∑k are available in the R package *mclust* (see *?mclustModelNames*).

The available model options, in *mclust* package, are represented by identifiers including: EII, VII, EEI, VEI, EVI, VVI, EEE, EEV, VEV and VVV.

The first identifier refers to volume, the second to shape and the third to orientation. E stands for “equal”, V for “variable” and I for “coordinate axes”.

For example:

- EVI denotes a model in which the volumes of all clusters are equal (E), the shapes of the clusters may vary (V), and the orientation is the identity (I) or "coordinate axes.
- EEE means that the clusters have the same volume, shape and orientation in p-dimensional space.
- VEI means that the clusters have variable volume, the same shape and orientation equal to coordinate axes.

Choosing the best model

The *Mclust* package uses maximum likelihood to fit all these models, with different covariance matrix parameterizations, for a range of k components.

The best model is selected using the Bayesian Information Criterion or *BIC*. A large BIC score indicates strong evidence for the corresponding model.

Computing model-based clustering

We start by installing the *mclust* package as follow: *install.packages("mclust")*

Note that, model-based clustering can be applied on univariate or multivariate data.

Here, we illustrate model-based clustering on the diabetes data set [mclust package] giving three measurements and the diagnosis for 145 subjects described as follow:

```
library("mclust")

data("diabetes")

head(diabetes, 3)
```

```
##    class glucose insulin sspg

## 1 Normal      80     356  124

## 2 Normal      97     289  117

## 3 Normal     105     319  143
```

- class: the diagnosis: normal, chemically diabetic, and overtly diabetic. Excluded from the cluster analysis.
- glucose: plasma glucose response to oral glucose
- insulin: plasma insulin response to oral glucose
- sspg: steady-state plasma glucose (measures insulin resistance)

Model-based clustering can be computed using the function Mclust() as follow:

```
library(mclust)

df <- scale(diabetes[, -1]) # Standardize the data

mc <- Mclust(df)            # Model-based-clustering

summary(mc)                 # Print a summary
```

```
## ----------------------------------------------------

## Gaussian finite mixture model fitted by EM algorithm

## ----------------------------------------------------

##

## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 3 components:

##  log.likelihood   n df  BIC  ICL

##           -169 145 29 -483 -501

## Clustering table:

##  1  2  3

## 81 36 28
```

For this data, it can be seen that model-based clustering selected a model with three components (i.e. clusters). The optimal selected model name is VVV model. That is the three

components are ellipsoidal with varying volume, shape, and orientation. The summary contains also the clustering table specifying the number of observations in each clusters.

You can access to the results as follow:

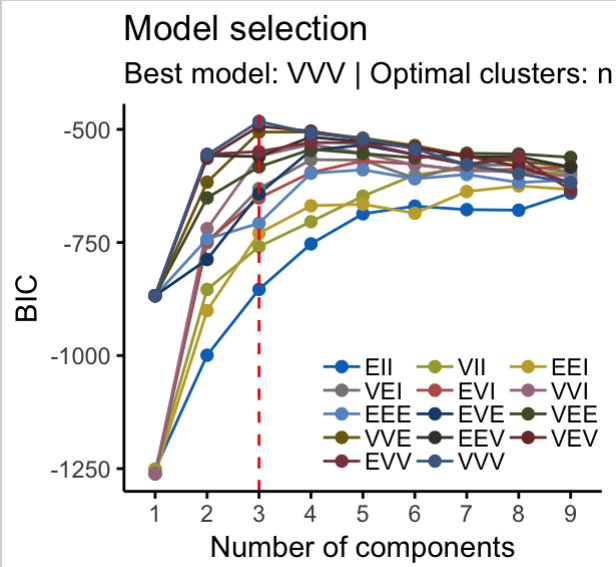
```
mc$modelName              # Optimal selected model ==> "VVV"

mc$G                      # Optimal number of cluster => 3

head(mc$z, 30)            # Probality to belong to a given cluster

head(mc$classification, 30) # Cluster assignement of each observation
```

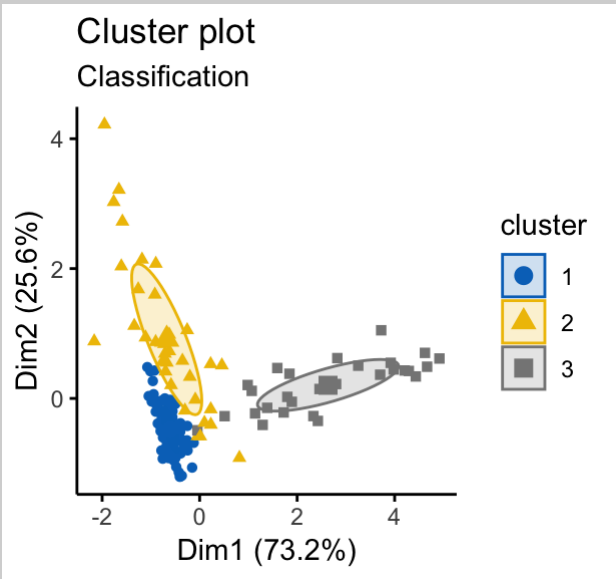
Visualizing model-based clustering

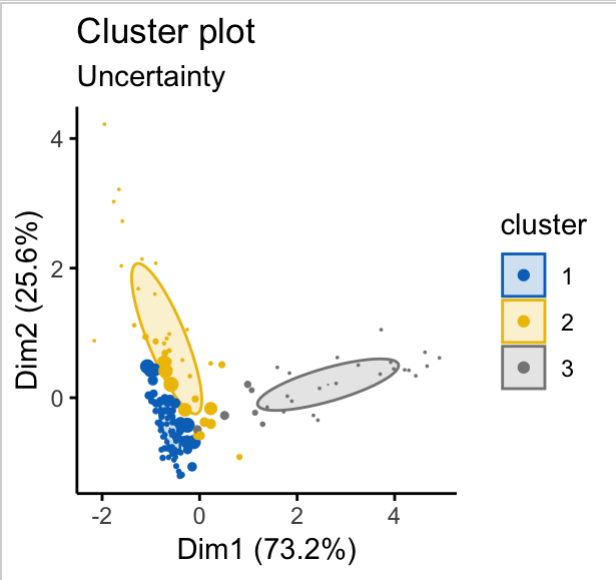
Model-based clustering results can be drawn using the base function plot.Mclust() [in mclust package]. Here we'll use the function *fviz_mclust*() [in *factoextra* package] to create beautiful plots based on ggplot2.

In the situation, where the data contain more than two variables, *fviz_mclust*() uses a principal component analysis to reduce the dimensionnality of the data. The first two principal components are used to produce a scatter plot of the data. However, if you want to plot the data using only two variables of interest, let say here c("insulin", "sspg"), you can specify that in the *fviz_mclust*() function using the argument *choose.vars = c("insulin", "sspg")*.

```
library(factoextra)# BIC values used for choosing the number of clusters

fviz_mclust(mc, "BIC", palette = "jco")# Classification: plot showing the clustering

fviz_mclust(mc, "classification", geom = "point",

        pointsize = 1.5, palette = "jco")# Classification uncertainty

fviz_mclust(mc, "uncertainty", palette = "jco")
```







Note that, in the uncertainty plot, larger symbols indicate the more uncertain observations.

CONTRAINED CLUSTERING

Constrained clustering is a class of semi-supervised learning algorithms. Typically, constrained clustering incorporates either a set of must-link constraints, cannot-link constraints, or both, with a Data clustering algorithm. Both a must-link and a cannot-link constraint define a relationship between two data instances. A must-link constraint is used to specify that the two instances in the must-link relation should be associated with the same cluster. A cannot-link constraint is used to specify that the two instances in the cannot-link relation should *not* be associated with the same cluster. These sets of constraints acts as a guide for which a constrained clustering algorithm will attempt to find clusters in a data set which satisfy the specified must-link and cannot-link constraints. Some constrained clustering algorithms will abort if no such clustering exists which satisfies the specified constraints. Others will try to minimize the amount of constraint violation should it be impossible to find a clustering which satisfies the constraints. Constraints could also be used to guide the selection of a clustering model among several possible solutions. A cluster in which the members conform to all must-link and cannot-link constraints is called a **chunklet**.

Examples of constrained clustering algorithms include:

COP K-means

PCKmeans (Pairwise Constrained K-means)

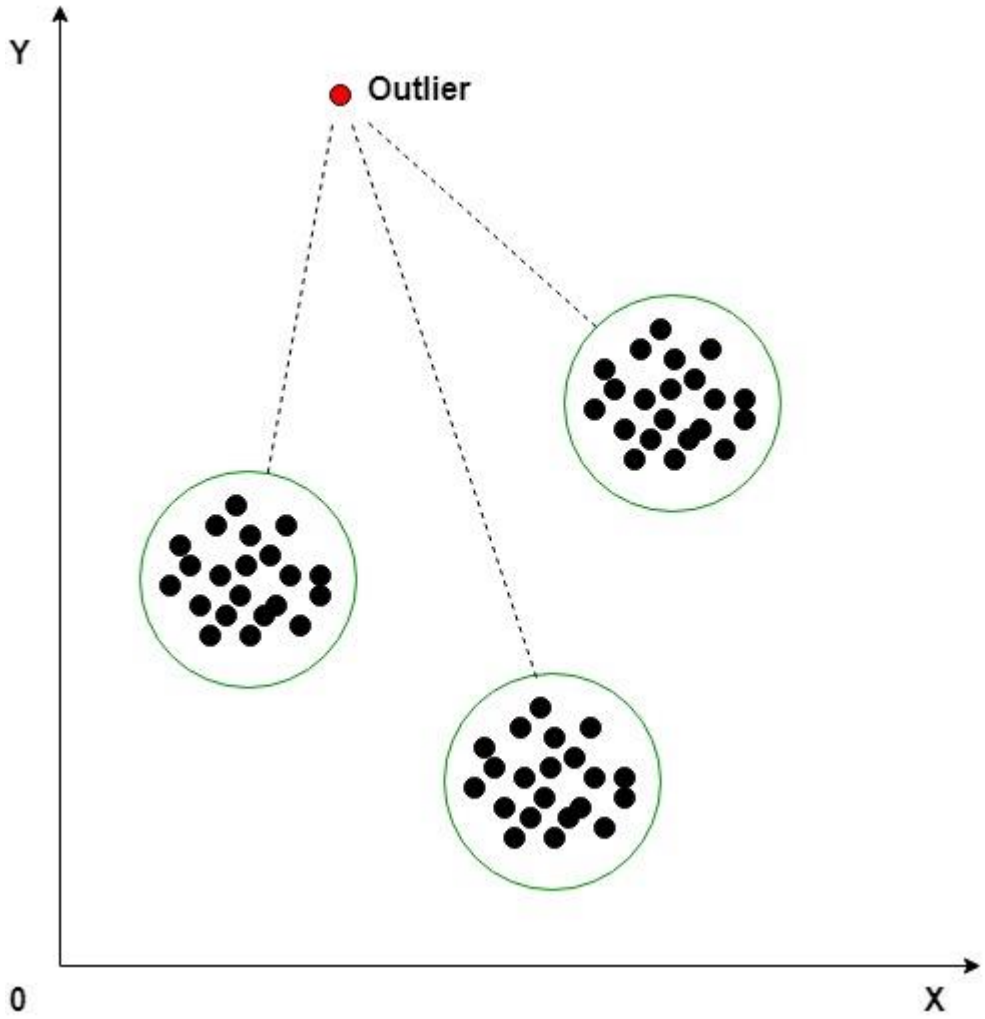
MWK-Means (Constrained Minkowski Weighted K-Means)

OUTLIER ANALYSIS

An **outlier** is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.

Why outlier analysis?

Most data mining methods discard outliers noise or exceptions, however, in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring one and hence, the outlier analysis becomes important in such case.



Detecting Outlier:

Clustering based outlier detection using distance to the closest cluster:

In the K-Means clustering technique, each cluster has a mean value. Objects belong to the cluster whose mean value is closest to it. In order to identify the Outlier, firstly we need to initialize the threshold value such that any distance of any data point greater than it from its nearest cluster identifies it as an outlier for our purpose. Then we need to find the distance of the test data to each cluster mean. Now, if the distance between the test data and the closest cluster to it is greater than the threshold value then we will classify the test data as an outlier.

Algorithm:

1. Calculate the mean of each cluster
2. Initialize the Threshold value

3. Calculate the distance of the test data from each cluster mean
4. Find the nearest cluster to the test data
5. If (Distance > Threshold) then, Outlier

Data Mining Applications

Data mining is widely used in diverse areas. There are a number of commercial data mining system available today and yet there are many challenges in this field. In this tutorial, we will discuss the applications and the trend of data mining.

Applications

Here is the list of areas where data mining is widely used −

- Financial Data Analysis
- Retail Industry
- Telecommunication Industry
- Biological Data Analysis
- Other Scientific Applications
- Intrusion Detection

1) Data Mining for Financial Data Analysis

- Design and construction of data warehouses for multidimensional data analysis and data mining
- Loan payment prediction and customer credit policy analysis
- Classification and clustering of customers for targeted marketing
- Detection of money laundering and other financial crimes
- Data Mining for the Retail Industry

2)A few examples of data mining in the retail industry

- Design and construction of data warehouses based on the benefits of data mining
- Multidimensional analysis of sales, customers, products, time, and region

- Analysis of the effectiveness of sales campaigns
- Customer retention—analysis of customer loyalty
- Product recommendation and cross-referencing of items

3) Data Mining for the Telecommunication Industry

- Multidimensional analysis of telecommunication data
- Fraudulent pattern analysis and the identification of unusual patterns
- Multidimensional association and sequential pattern analysis
- Mobile telecommunication services
- Use of visualization tools in telecommunication data analysis

4) Data Mining for Biological Data Analysis

- Semantic integration of heterogeneous, distributed genomic and proteomic databases
- Alignment, indexing, similarity search, and comparative analysis of multiple nucleotide , protein sequences.
- Discovery of structural patterns and analysis of genetic networks and protein pathways.
- Association and path analysis: identifying co-occurring gene sequences and linking genes to different stages of disease development.
- Visualization tools in genetic data analysis.

5) Data Mining in Scientific Applications

- Scientific data can be amassed at much higher speeds and lower costs.
- This has resulted in the accumulation of huge volumes of high-dimensional data, stream data, and heterogeneous data, containing rich spatial and temporal information.
- Scientific applications are shifting from the “hypothesize-and-test” paradigm toward a “collect and store data, mine for new hypotheses, confirm with data or experimentation” process.

6) Data Mining for Intrusion Detection

- Development of data mining algorithms for intrusion detection
- Association and correlation analysis, and aggregation to help select and build discriminating attributes
- Analysis of stream data
- Distributed data mining
- Visualization and querying tools

7) Trends in Data Mining

- Application exploration
- Scalable and interactive data mining methods
- Integration of data mining with database systems, data warehouse systems, and Webdatabase systems
- Standardization of data mining language
- Visual data mining
- Biological data mining
- Data mining and software engineering
- Web mining
- Distributed data mining
- Real-time or time-critical data mining
- Graph mining, link analysis, and social network analysis
- Multi relational and multi database data mining
- New methods for mining complex types of data
- Privacy protection and information security in data mining

8) Assessment of a Data mining System

1. Data types
2. System issues
3. Data sources
4. Data mining functions and methodologies
5. Coupling data mining with database and/or data warehouse systems.
6. Scalability
7. Visualization tools
8. Data mining query language and graphical user interface

9) Theoretical Foundations of Data Mining

- Data reduction
- Data compression
- Pattern discovery
- Probability theory
- Microeconomic view
- Inductive databases

10) Statistical Data Mining techniques

1. Regression
2. Generalized linear model
3. Analysis of variance
4. mixed effect model
5. Factor analysis
6. Discriminate analysis
7. Time series analysis
8. Survival analysis
9. Quality control

11) Visual and Audio Data Mining

- Visual data mining discovers implicit and useful knowledge from large data sets using data and/or knowledge visualization
- Data visualization and data mining can be integrated in the following ways:
- Data visualization
- Data mining result visualization
- Data mining process visualization
- Interactive visual data mining techniques

12) Security of Data Mining

- Data security enhancing techniques have been developed to help protect data
- Databases can employ a multilevel security model to classify and restrict data according to various security levels, with users permitted access to only their authorized level

- Privacy-sensitive data mining deals with obtaining valid data mining results without learning the underlying data values

Social Impacts of Data Mining

1 . Is Data Mining a Hype or Will It Be Persistent?

- Data mining is a technology
- Technological life cycle
    - Innovators
    - Early Adopters
    - Early Adopters
    - Chasm
    - Early Majority
    - Late Majority
    - Laggards

2. Data Mining: Managers' Business or Everyone's?

- Data mining will surely be an important tool for managers' decision making
    - Bill Gates: "Business @ the speed of thought"
- The amount of the available data is increasing, and data mining systems will be more affordable
- Multiple personal uses
    - Mine your family's medical history to identify genetically-related medical conditions
    - Mine the records of the companies you deal with
    - Mine data on stocks and company performance, etc.
- Invisible data mining
    - Build data mining functions into many intelligent tools

3. Social Impacts: Threat to Privacy and Data Security?

- Is data mining a threat to privacy and data security?
    - "Big Brother", "Big Banker", and "Big Business" are carefully watching you
- Profiling information is collected every time

• credit card, debit card, supermarket loyalty card, or frequent flyer card, or apply for any of the above

• You surf the Web, rent a video, fill out a contest entry form,

• You pay for prescription drugs, or present you medical care number when visiting the

doctor

• Collection of personal data may be beneficial for companies and consumers, there is also potential for misuse

• Medical Records, Employee Evaluations, etc.

4. Protect Privacy and Data Security

1. Fair information practices
   - International guidelines for data privacy protection
   - Cover aspects relating to data collection, purpose, use, quality, openness, individual participation, and accountability
   - Purpose specification and use limitation
   - Openness : Individuals have the right to know what information is collected about them, who has access to the data, and how the data are being used

2. Develop and use data security-enhancing techniques
   - Blind signatures
   - Biometric encryption
   - Anonymous databases

Examples Of Data Mining In Real Life

The importance of data mining and analysis is growing day by day in our real life. Today most organizations use data mining for analysis of Big Data.

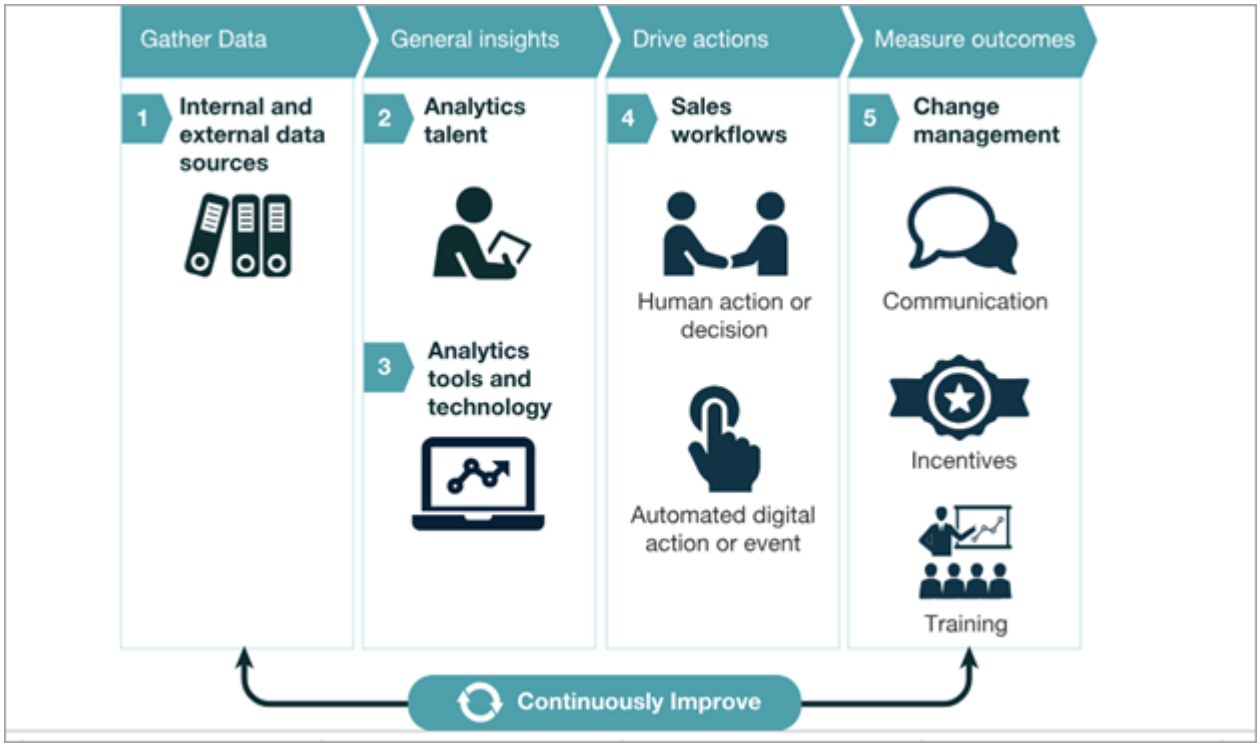
Let us see how these technologies benefit us.

#1) Mobile Service Providers

Mobile service providers use data mining to design their marketing campaigns and to retain customers from moving to other vendors.

From a large amount of data such as billing information, email, text messages, web data transmissions, and customer service, the data mining tools can predict “churn” that tells the customers who are looking to change the vendors.

With these results, a probability score is given. The mobile service providers are then able to provide incentives, offers to customers who are at higher risk of churning. This kind of mining is often used by major service providers such as broadband, phone, gas providers, etc.



#2) Retail Sector

Data Mining helps the supermarket and retail sector owners to know the choices of the customers. Looking at the purchase history of the customers, the data mining tools show the buying preferences of the customers.

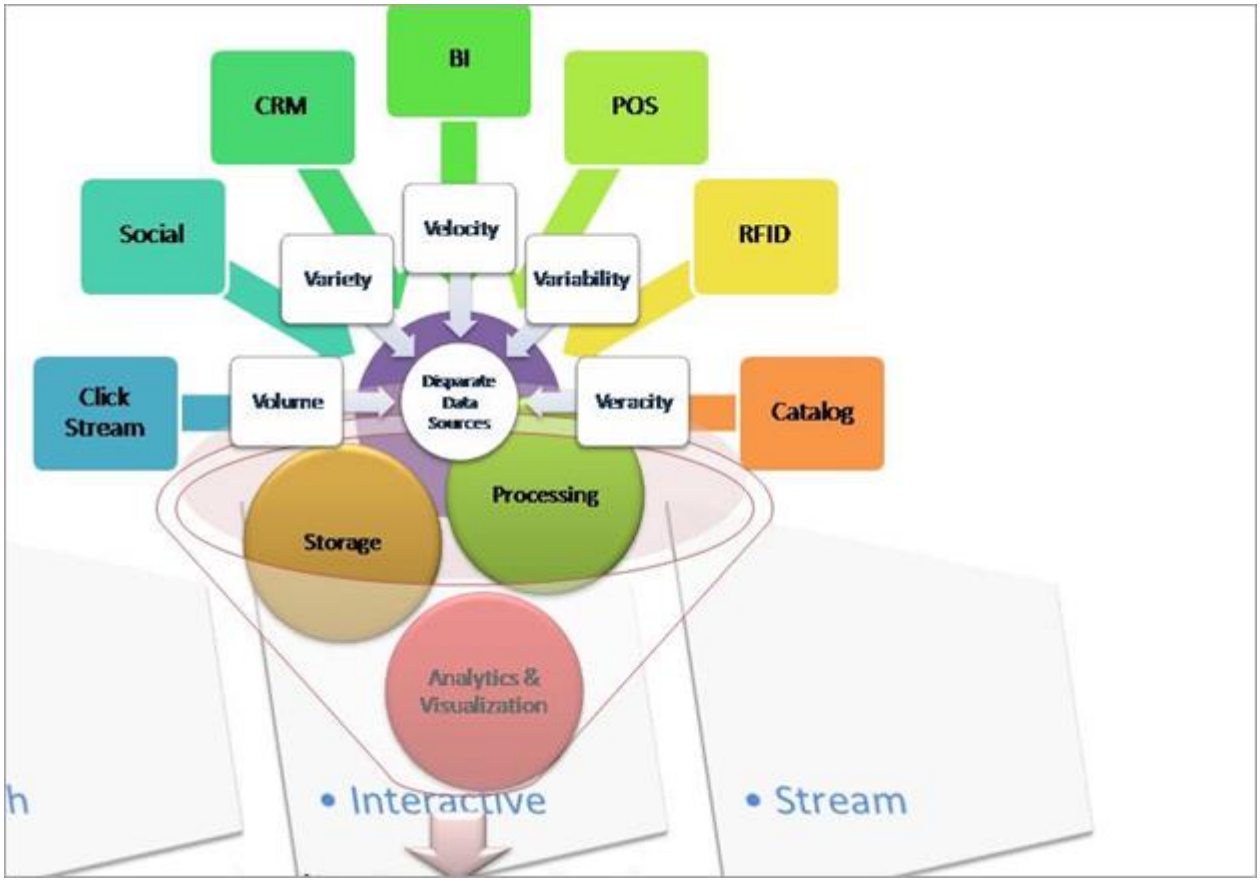
With the help of these results, the supermarkets design the placements of products on shelves and bring out offers on items such as coupons on matching products, and special discounts on some products.

These campaigns are based on RFM grouping. RFM stands for recency, frequency, and monetary grouping. The promotions and marketing campaigns are customized for these

segments. The customer who spends a lot but very less frequently will be treated differently from the customer who buys every 2-3 days but of less amount.

Data Mining can be used for product recommendation and cross-referencing of items.

Data Mining In Retail Sector From Different Data Sources.



#3) Artificial Intelligence

A system is made artificially intelligent by feeding it with relevant patterns. These patterns come from data mining outputs. The outputs of the artificially intelligent systems are also analyzed for their relevance using the data mining techniques.

The recommender systems use data mining techniques to make personalized recommendations when the customer is interacting with the machines. The artificial intelligence is used on mined data such as giving product recommendations based on the past purchasing history of the customer in Amazon.

#4) Ecommerce

Many E-commerce sites use data mining to offer cross-selling and upselling of their products. The shopping sites such as Amazon, Flipkart show “People also viewed”, “Frequently bought together” to the customers who are interacting with the site.

These recommendations are provided using data mining over the purchasing history of the customers of the website.

#5) Science And Engineering

With the advent of data mining, scientific applications are now moving from statistical techniques to using “collect and store data” techniques, and then perform mining on new data, output new results and experiment with the process. A large amount of data is collected from scientific domains such as astronomy, geology, satellite sensors, global positioning system, etc.

Data mining in computer science helps to monitor system status, improve its performance, find out software bugs, discover plagiarism and find out faults. Data mining also helps in analyzing the user feedback regarding products, articles to deduce opinions and sentiments of the views.

#6) Crime Prevention

Data Mining detects outliers across a vast amount of data. The criminal data includes all details of the crime that has happened. Data Mining will study the patterns and trends and predict future events with better accuracy.

The agencies can find out which area is more prone to crime, how much police personnel should be deployed, which age group should be targeted, vehicle numbers to be scrutinized, etc.

#7) Research

Researchers use Data Mining tools to explore the associations between the parameters under research such as environmental conditions like air pollution and the spread of diseases like asthma among people in targeted regions.

#8) Farming

Farmers use Data Mining to find out the yield of vegetables with the amount of water required by the plants.

#9) Automation

By using data mining, the computer systems learn to recognize patterns among the parameters which are under comparison. The system will store the patterns that will be useful in the future to achieve business goals. This learning is automation as it helps in meeting the targets through machine learning.

#10) Dynamic Pricing

Data mining helps the service providers such as cab services to dynamically charge the customers based on the demand and supply. It is one of the key factors for the success of companies.

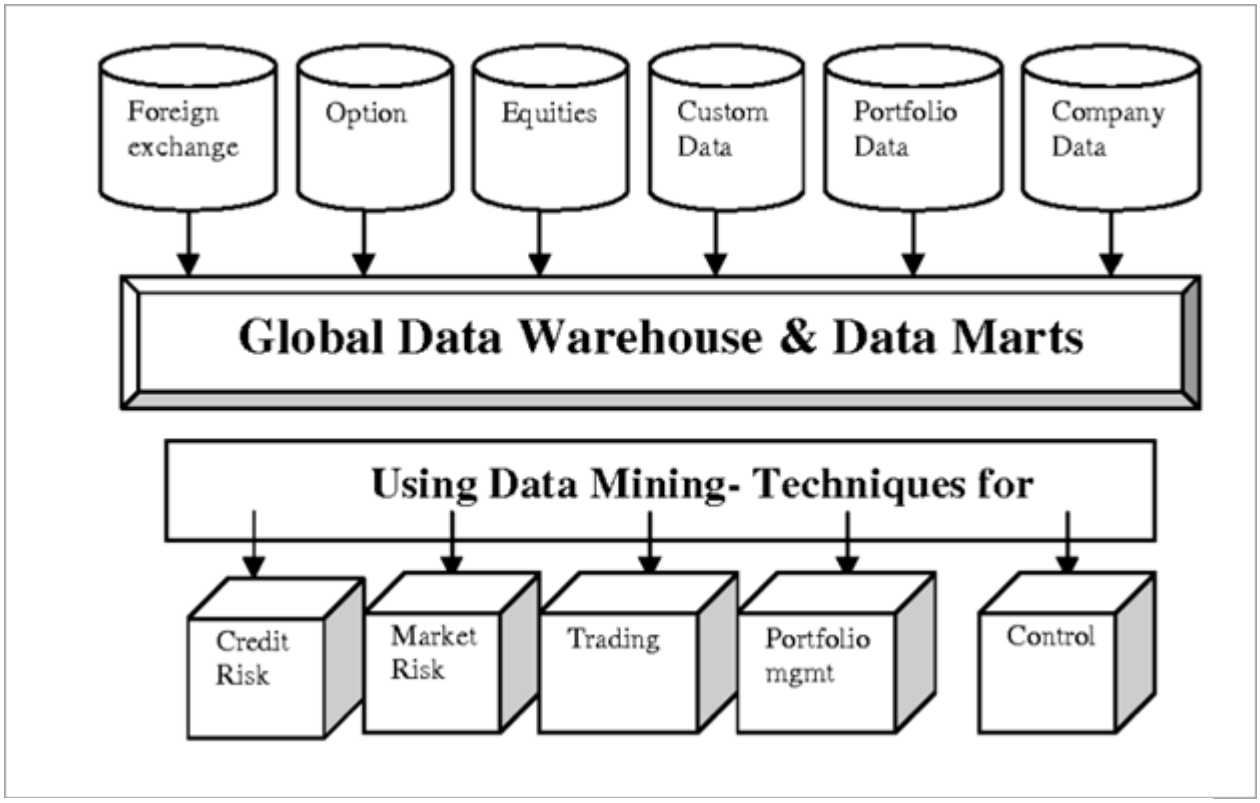
#11) Transportation

Data Mining helps in scheduling the moving of vehicles from warehouses to outlets and analyze the product loading patterns.

#12) Insurance

Data mining methods help in forecasting the customers who buy the policies, analyze the medical claims that are used together, find out fraudulent behaviors and risky customers.

Data Mining Examples In Finance



The finance sector includes banks, insurance companies, and investment companies. These institutions collect a huge amount of data. The data is often complete, reliable and of high quality and demands a systematic data analysis.

To store financial data, data warehouses that store data in the form of data cubes are constructed. To analyze this data, advanced data cube concepts are used. Data mining methods such as clustering and outlier analysis, characterization are used in financial data analysis and mining.

Some cases in finance where data mining is used are given below.

#1) Loan Payment Prediction

Data mining methods like attribute selection and attribute ranking will analyze the customer payment history and select important factors such as payment to income ratio, credit history, the term of the loan, etc. The results will help the banks decide its loan granting policy, and also grant loans to the customers as per factor analysis.

#2) Targeted Marketing

Clustering and classification data mining methods will help in finding the factors that influence the customer's decisions towards banking. Similar behavioral customers' identification will facilitate targeted marketing.

#3) Detect Financial Crimes

Banking data come from many different sources, various cities, and different bank locations. Multiple data analysis tools are deployed to study and to detect unusual trends like big value transactions. Data visualization tools, outlier analysis tools, clustering tools, etc are used to identify the relationships and patterns of action.

The figure below is a study from Infosys showing the customer's willingness to banking online system in different countries. Infosys used Big Data Analytics for this study.

Applications Of Data Mining In Marketing

Data mining boosts the company's marketing strategy and promotes business. It is one of the key factors for the success of companies. A huge amount of data is collected on sales, customer shopping, consumption, etc. This data is increasing day by day due to e-commerce.

Data mining helps to identify customer buying behavior, improve customer service, focus on customer retention, enhance sales, and reduce the cost of businesses.

Some examples of data mining in marketing are:

#1) Forecasting Market

To predict the market, the marketing professionals will use Data Mining techniques like regression to study customer behavior, changes, and habits, customer response and other factors like marketing budget, other incurring costs, etc. In the future, it will be easier for professionals to predict the customers in case of any factor changes.

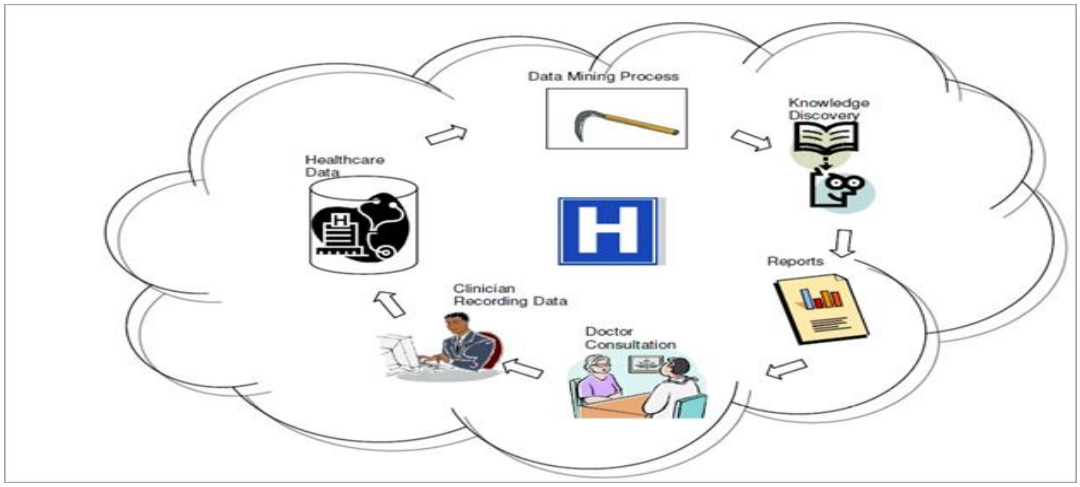
#2) Anomaly Detection

Data mining techniques are deployed to detect any abnormalities in data that may cause any kind of flaw in the system. The system will scan thousands of complex entries to perform this operation.

#3) System Security

Data Mining tools detect intrusions that may harm the database offering greater security to the entire system. These intrusions may be in the form of duplicate entries, viruses in the form of data by hackers, etc.

Examples Of Data Mining Applications In Healthcare



In healthcare, data mining is becoming increasingly popular and essential.

Data generated by healthcare is complex and voluminous. To avoid medical fraud and abuse, data mining tools are used to detect fraudulent items and thereby prevent loss.

Some data mining examples of the healthcare industry are given below for your reference.

#1) Healthcare Management

The data mining method is used to identify chronic diseases, track high-risk regions prone to the spread of disease, design programs to reduce the spread of disease. Healthcare professionals will analyze the diseases, regions of patients with maximum admissions to the hospital.

With this data, they will design the campaigns for the region to make people aware of the disease and see how to avoid it. This will reduce the number of patients admitted to hospitals.

#2) Effective Treatments

Using data mining, the treatments can be improved. By continuous comparison of symptoms, causes, and medicines, data analysis can be performed to make effective treatments. Data mining is also used for the treatment of specific diseases, and the association of side-effects of treatments.

#3) Fraudulent And Abusive Data

Data mining applications are used to find abnormal patterns such as laboratory, physician's results, inappropriate prescriptions, and fraudulent medical claims.

Data Mining And Recommender Systems

Recommender systems give customers with product recommendations that may be of interest to the users.

The recommended items are either similar to the items queried by the user in the past or by looking at the other customer preferences which have similar taste as the user. This approach is called a content-based approach and a collaborative approach appropriately.

Many techniques like information retrieval, statistics, machine learning, etc are used in recommender systems.

Recommender systems search for keywords, user profiles, user transactions, common features among items to estimate an item for the user. These systems also find the other users who have a similar history of buying and predict items that those users could buy.

There are many challenges in this approach. The recommendation system needs to search through millions of data in real-time.

There are two types of errors made by Recommender Systems:

False negatives and False positives.

False negatives are products that were not recommended by the system but the customer would want them. **False-positive** are products that were recommended by the system but not wanted by the customer. Another challenge is the recommendation for the users who are new without any purchasing history.

An intelligent query answering technique is used to analyze the query and provide generalized, associated information relevant to the query. **For Example:** Showing the review of restaurants instead of just the address and phone number of the restaurant searched for.

Data Mining For CRM (Customer Relationship Management)

Customer Relationship Management can be reinforced with data mining. Good customer Relations can be built by attracting more suitable customers, better cross-selling and up-selling, better retention.

Data Mining can enhance CRM by:

1. Data mining can help businesses create targeted programs for higher response and better ROI.
2. Businesses can offer more products and services as desired by the customers through up-selling and cross-selling thereby increasing customer satisfaction.
3. With data mining, a business can detect which customers are looking for other options. Using that information companies can build ideas to retain the customer from leaving.

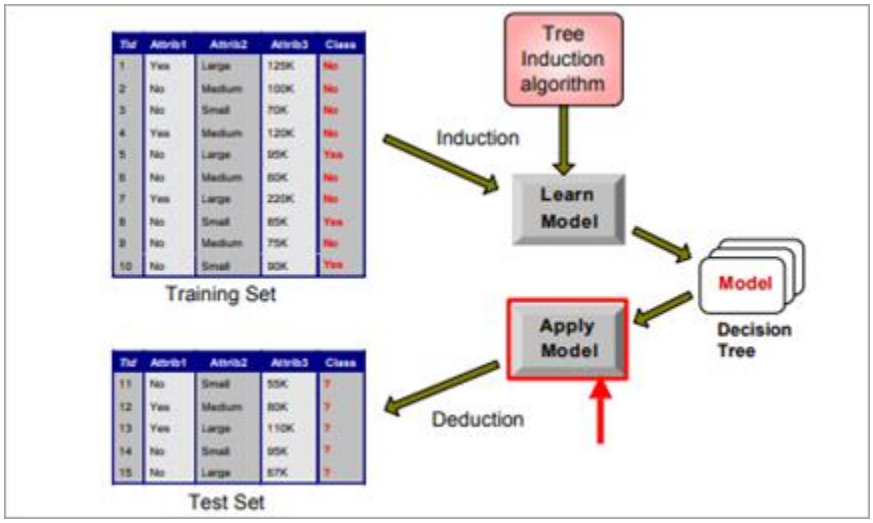
Data Mining helps CRM in:

1. **Database Marketing:** Marketing software enables companies to send messages and emails to customers. This tool along with data mining can do targeted marketing. With data mining, automation, and scheduling of jobs can be performed. It helps in better decision making. It will also help in technical decisions as to what kind of customers are interested in a new product, which market area is good for product launching.
2. **Customer Acquisition Campaign:** With data mining, the market professional will be able to identify potential customers who are unaware of the products or new buyers. They will be able to design the offers and initiatives for such customers.
3. **Campaign Optimization:** Companies use data mining for the effectiveness of the campaign. It can model customer responses to marketing offers.

Data Mining Using Decision Tree Example

Decision tree algorithms are called CART(Classification and Regression Trees). It is a supervised learning method. A tree structure is built on the features chosen, conditions for splitting and when to stop. Decision trees are used to predict the value of class variables based on learning from the previous training data.

The internal node represents an attribute and the leaf node represents a class label.



Following steps are used to build a Decision Tree Structure:

1. Place the best attribute at the top of the tree (root).
2. Subsets are created in such a way that each subset represents data with the same value for an attribute.
3. Repeat the same steps to find the leaf nodes of all branches.

To predict a class label, the record's attribute is compared with the root of the tree. On comparing, the next branch is chosen. The internal nodes are also compared in the same way until the leaf node reached predicts the class variable.

Some algorithms used for Decision Tree Induction include Hunt's Algorithm, CART, ID3, C4.5, SLIQ, and SPRINT.

Most Popular Example Of Data Mining: Marketing And Sales

Marketing and Sales are the domains in which companies have large volumes of data.

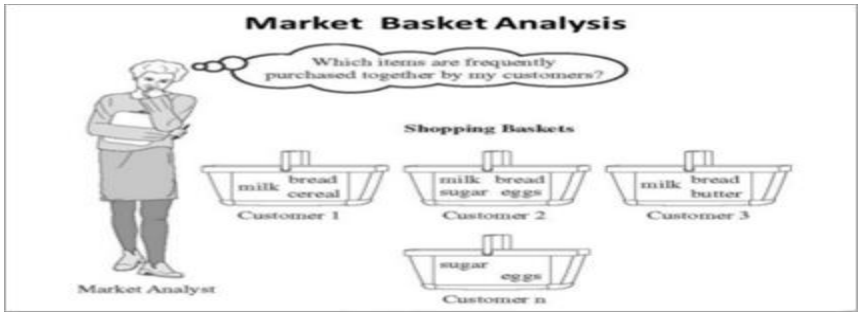
#1) Banks are the first users of data mining technology as it helps them with credit assessment. Data mining analyzes what services offered by banks are used by customers, what type of customers use ATM cards and what do they generally buy using their cards (for cross-selling).

Banks use data mining to analyze the transactions which the customer do before they decide to change the bank to reduce customer attrition. Also, some outliers in transactions are analyzed for fraud detection.

#2) Cellular Phone Companies use data mining techniques to avoid churning. Churning is a measure showing the number of customers leaving the services. It detects patterns that show how customers can benefit from the services to retain customers.

#3) Market Basket Analysis is the technique to find the groups of items that are bought together in stores. Analysis of the transactions show the patterns such as which things are bought together often like bread and butter, or which items have higher sales volume on certain days such as beer on Fridays.

This information helps in planning the store layouts, offering a special discount to the items that are less in demand, creating offers such as “buy 2 get 1 free” or “get 50% on second purchase” etc.



Big Companies Using Data Mining

Some online companies using data mining techniques are given below:

- **AMAZON:** Amazon uses Text Mining to find the lowest price of the product.
- **MC Donald’s:** McDonald’s uses big data mining to enhance its customer experience. It studies the ordering pattern of customers, waiting times, size of orders, etc.
- **NETFLIX:** Netflix finds out how to make a movie or a series popular among the customers using its data mining insights.

Conclusion

Data mining is used in diverse applications such as banking, marketing, healthcare, telecom industries, and many other areas.Data mining techniques help companies to gain knowledgeable information, increase their profitability by making adjustments in processes

and operations. It is a fast process which helps business in decision making through analysis of hidden patterns and trends.

QUESTIONS

PART A

1. What do you infer from the word cluser analysis? Highlight various types of Data in Cluster Analysis.
2. Compare and Contrast K- Means and K – Medoids.
3. Which is better? Agglomerative or Divisive Hierarchical Clustering. Illustrate.
4. Highlight the importance of Neural Network Approach in Clustering.
5. How do cluster high dimensional Data. Elaborate.
6. Comment on Constraint Based Cluster Analysis.
7. Justify the essence of outlier analysis.
8. Compare various clustering techniques.
9. How do you categorize Major Clustering Methods.
10. Illustrate Grid Based Clustering Methods.

PART B

1. Justify the need of Clustering Technique.
2. How can the data for a variable be standardized?
3. Compare Manhattan and Euclidean Distance.
4. Highlight various clustering techniques.
5. What do you infer from the term “Density based cluster”.
6. Comment on STING.

TEXT/REFERENCE BOOKS

1. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, 2nd Edition, Elsevier 2007.
2. Alex Berson and Stephen J. Smith, “Data Warehousing, Data Mining & OLAP”, Tata McGrew Hill, 2007.
3. www.tutorialspoint.com/dwh/dwh.overview.htm
4. http://study.com/academy/lesson/data-warehousing-and-data-minig-information-for-business-intelligence.html
5. http://www.dei.unpd_it/-capt/SVMATERIALE/DWDM0405.pdf
6. Data mining Concepts and Techniques, Book by Jewei Han and Kamber.