UNIT 5

Clustering

What is Clustering?

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

- 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 data similarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.
- 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.

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.

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 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.

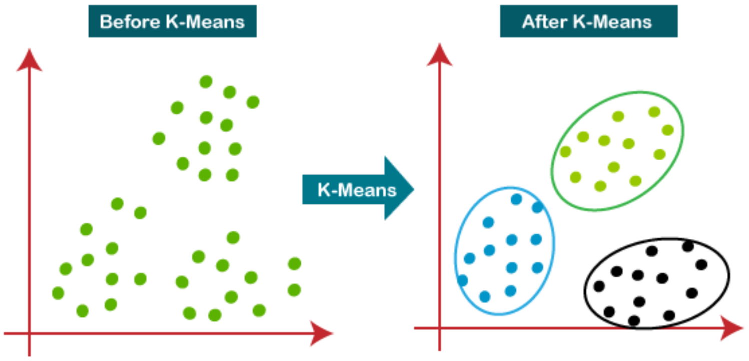
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

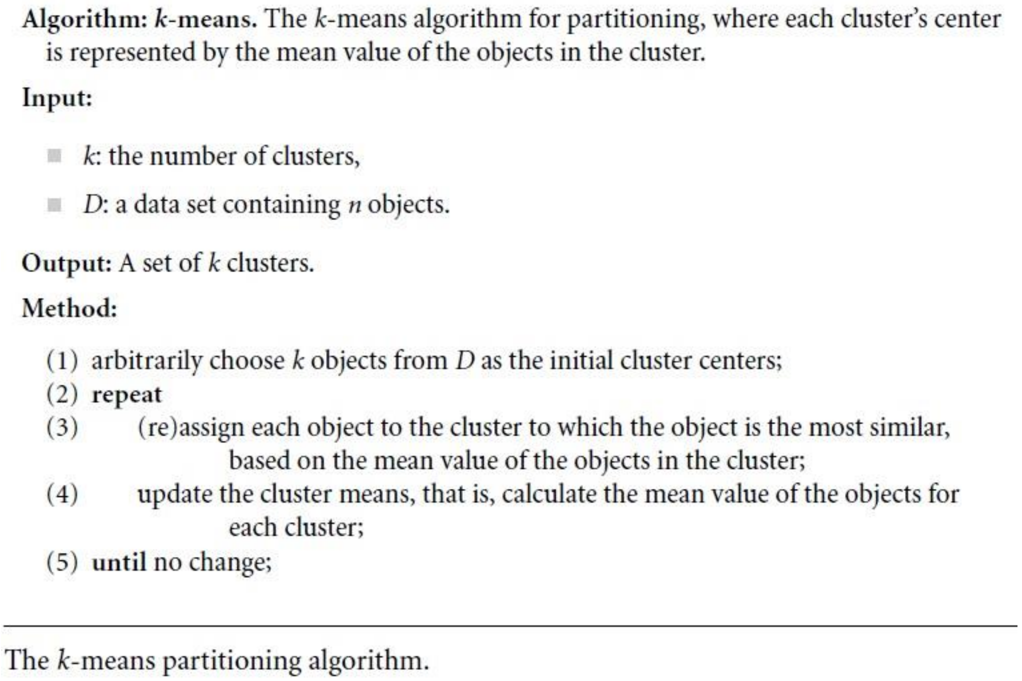
The k-means clustering algorithm mainly performs two tasks:

- 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:





Algorithm: ***k*-means.** The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

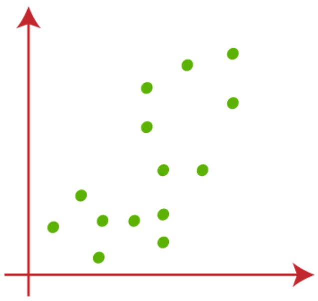
Method:

```
(1)  arbitrarily choose k objects from D as the initial cluster centers;
(2)  repeat
(3)        (re)assign each object to the cluster to which the object is the most similar,
                 based on the mean value of the objects in the cluster;
(4)        update the cluster means, that is, calculate the mean value of the objects for
                 each cluster;
(5)  until no change;
```

The *k*-means partitioning algorithm.

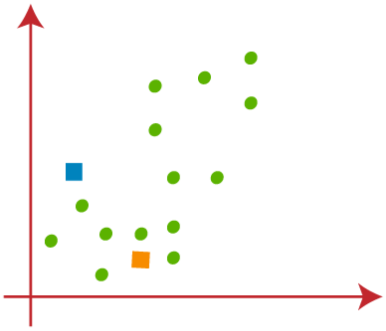
Let's understand the above steps by considering the visual plots:

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

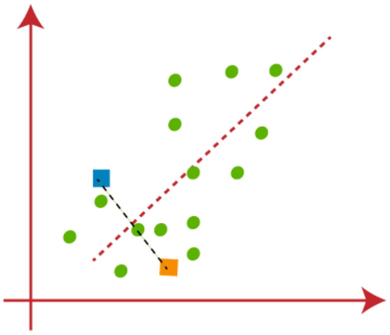


- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.

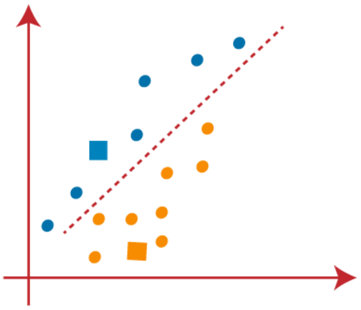
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:



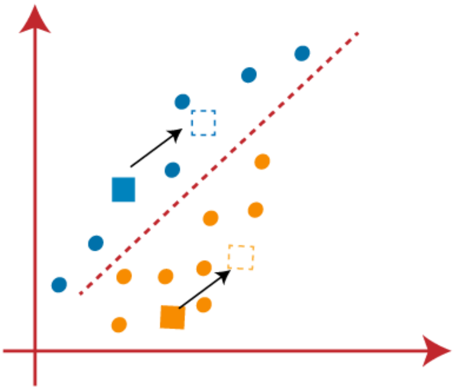
- Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



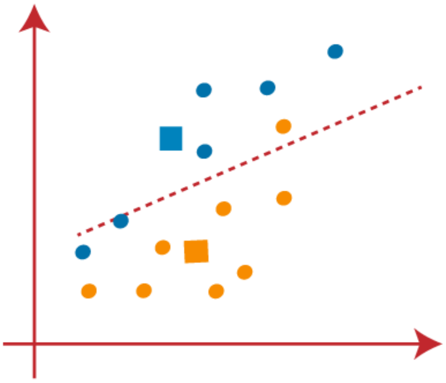
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



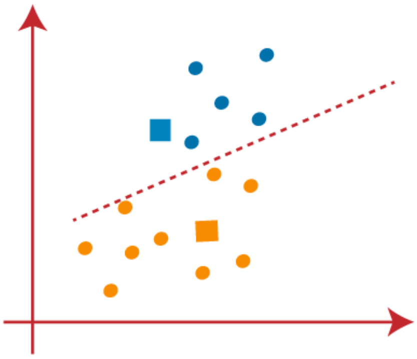
- ○ As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



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:

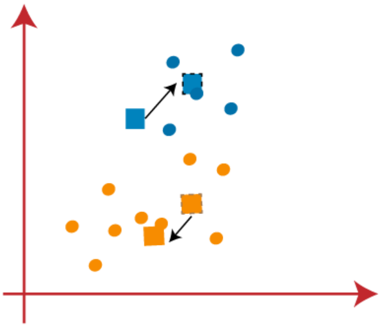


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.



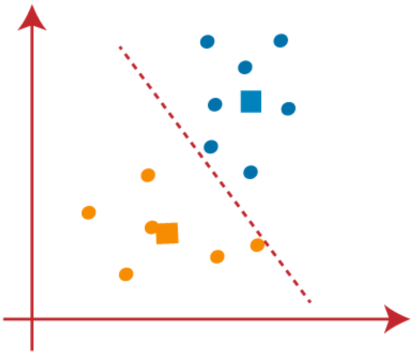
As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

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

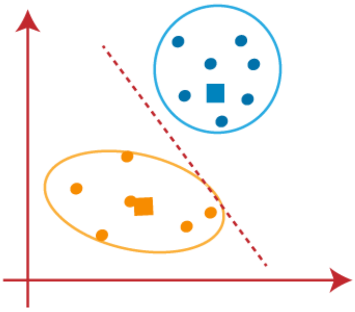


○

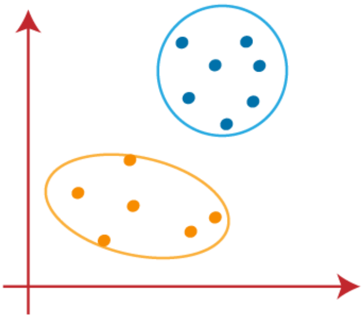
- As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



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

In the above formula of WCSS,

$\sum_{P_i \text{ in Cluster1}}$ **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

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 −

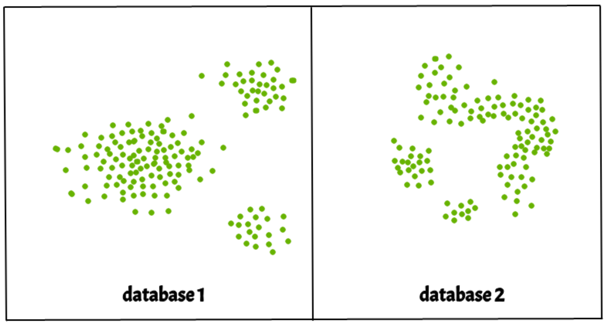
- Perform careful analysis of object linkages at each hierarchical partitioning.
- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

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.

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 neighbourhood of a given radius has to contain at least a minimum number of points.

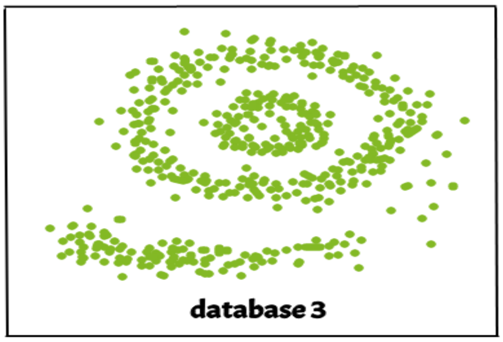


Why DBSCAN?

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:

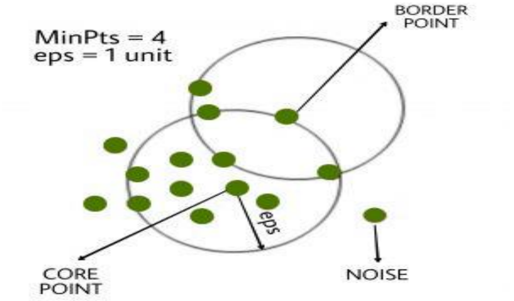
1. Clusters can be of arbitrary shape such as those shown in the figure below.
2. Data may contain noise.

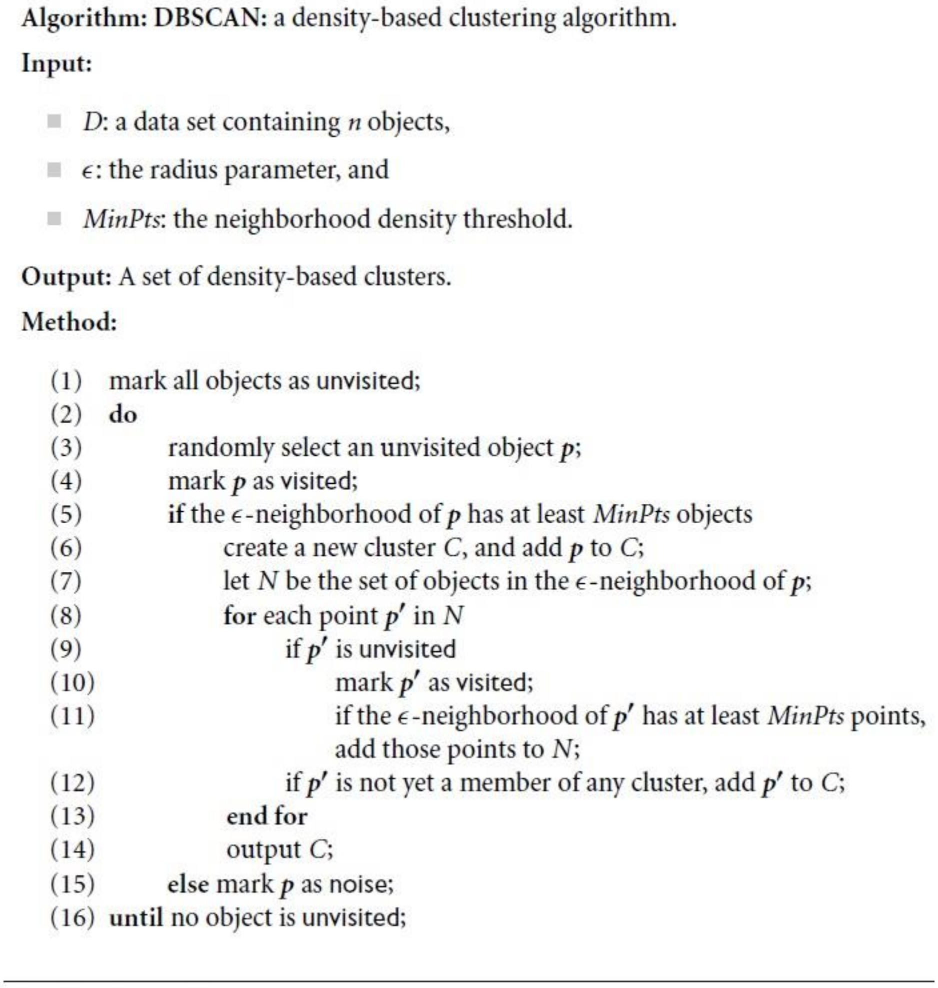


The figure below shows a data set containing nonconvex clusters and outliers/noises. Given such data, k-means algorithm has difficulties for identifying these clusters with arbitrary shapes.

DBSCAN algorithm requires two parameters:

1. **eps** : It defines the neighbourhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered as neighbours. 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 neighbours (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.




Algorithm: DBSCAN: a density-based clustering algorithm.

Input:

- D: a data set containing n objects,
- ϵ: the radius parameter, and
- $MinPts$: the neighborhood density threshold.

Output: A set of density-based clusters.

Method:

```
mark all objects as unvisited;
do
    randomly select an unvisited object p;
    mark p as visited;
    if the ε-neighborhood of p has at least MinPts objects
        create a new cluster C, and add p to C;
        let N be the set of objects in the ε-neighborhood of p;
        for each point p′ in N
            if p′ is unvisited
                mark p′ as visited;
                if the ε-neighborhood of p′ has at least MinPts points,
                add those points to N;
            if p′ is not yet a member of any cluster, add p′ to C;
        end for
        output C;
    else mark p as noise;
until no object is unvisited;
```

DBSCAN algorithm.

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.

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.

Outliter Analysis:

Outlier is a data object that deviates significantly from the rest of the data objects and behaves in a different manner. An outlier is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution errors. The analysis of outlier data is referred to as outlier analysis or outlier mining.

An outlier cannot be termed as a noise or error. Instead, they are suspected of not being generated by the same method as the rest of the data objects.

An outlier cannot be termed as a noise or error. Instead, they are suspected of not being generated by the same method as the rest of the data objects.

Outliers are of three types, namely –

1. Global (or Point) Outliers
2. Collective Outliers
3. Contextual (or Conditional) Outliers

1. Global Outliers

They are also known as *Point Outliers*. These are the simplest form of outliers. If, in a given dataset, a data point strongly deviates from all the rest of the data points, it is known as a global outlier. Mostly, all of the outlier detection methods are aimed at finding global outliers.

For example, In Intrusion Detection System, if a large number of packages are broadcast in a very short span of time, then this may be considered as a global outlier and we can say that that particular system has been potentially hacked.

Outlier Detection Methods

Models for Outlier Detection Analysis

There are several approaches to detecting Outliers. Outlier detection models may be classified into the following groups:

1. Extreme Value Analysis

Extreme Value Analysis is the most basic form of outlier detection and great for 1-dimension data. In this Outlier analysis approach, it is assumed that values which are too large or too small are outliers. Z-test and Student's t-test are classic examples.

These are good heuristics for initial analysis of data but they do not have much value in multivariate settings. Extreme Value Analysis is largely used as final step for interpreting outputs of other outlier detection methods.

2. Linear Models

In this approach, the data is modelled into a lower-dimensional sub-space with the use of linear correlations. Then the distance of each data point to a plane that fits the sub-space is being calculated. This distance is used to find outliers. PCA (Principal Component Analysis) is an example of linear models for anomaly detection.

3. Probabilistic and Statistical Models

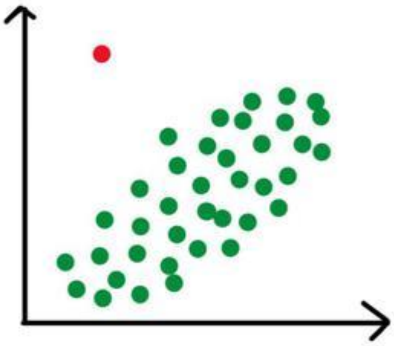
In this approach, Probabilistic and Statistical Models assume specific distributions for data. They make use of the expectation-maximization (EM) methods to estimate the parameters of the model. Finally, they calculate the probability of membership of each data point to calculated distribution. The points with a low probability of membership are marked as outliers.

4. Proximity-based Models

In this method, outliers are modelled as points isolated from the rest of the observations. Cluster analysis, density-based analysis, and nearest neighborhood are the principal approaches of this kind.

5. Information-Theoretic Models

In this method, the outliers increase the minimum code length to describe a data set.

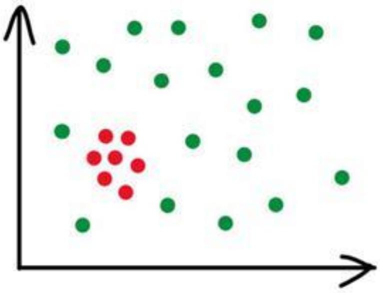


The red data point is a global outlier.

2. Collective Outliers

As the name suggests, if in a given dataset, some of the data points, as a whole, deviate significantly from the rest of the dataset, they may be termed as collective outliers. Here, the individual data objects may not be outliers, but when seen as a whole, they may behave as outliers. To detect these types of outliers, we might need background information about the relationship between those data objects showing the behaviour of outliers.

For example: In an Intrusion Detection System, a DOS (denial-of-service) package from one computer to another may be considered as normal behaviour. However, if this happens with several computers at the same time, then this may be considered as abnormal behaviour and as a whole they can be termed as collective outliers.

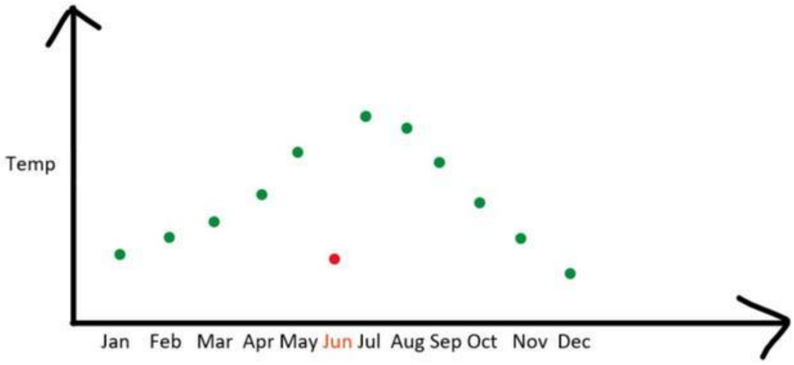


The red data points as a whole are collective outliers.

3. Contextual Outliers

They are also known as *Conditional Outliers.* Here, if in a given dataset, a data object deviates significantly from the other data points based on a specific context or condition only. A data point may be an outlier due to a certain condition and may show normal behaviour under another condition. Therefore, a context has to be specified as part of the problem statement in order to identify contextual outliers. Contextual outlier analysis provides flexibility for users where one can examine outliers in different contexts, which can be highly desirable in many applications. The attributes of the data point are decided on the basis of both contextual and behavioural attributes.

For example: A temperature reading of 40°C may behave as an outlier in the context of a “winter season” but will behave like a normal data point in the context of a “summer season”.



A low temperature value in June is a contextual outlier because the same value in December is not an outlier.

Outliers are generally defined as samples that are exceptionally far from the mainstream of data. There is no rigid mathematical definition of what constitutes an outlier; determining whether or not an observation is an outlier is ultimately a subjective exercise.

An outlier may also be explained as a piece of data or observation that deviates drastically from the given norm or average of the data set. An outlier may be caused simply by chance, but it may also indicate measurement error or that the given data set has a heavy-tailed distribution.

Therefore, Outlier Detection may be defined as the process of detecting and subsequently excluding outliers from a given set of data. There are no standardized Outlier identification methods as these are largely dependent upon the data set. Outlier Detection as a branch of data mining has many applications in data stream analysis.

Our discussion will also cover areas of standard applications of Outlier Detection, such as Fraud detection, public health, and sports and touch upon the various approaches like Proximity-based approaches and Angle-based approaches.