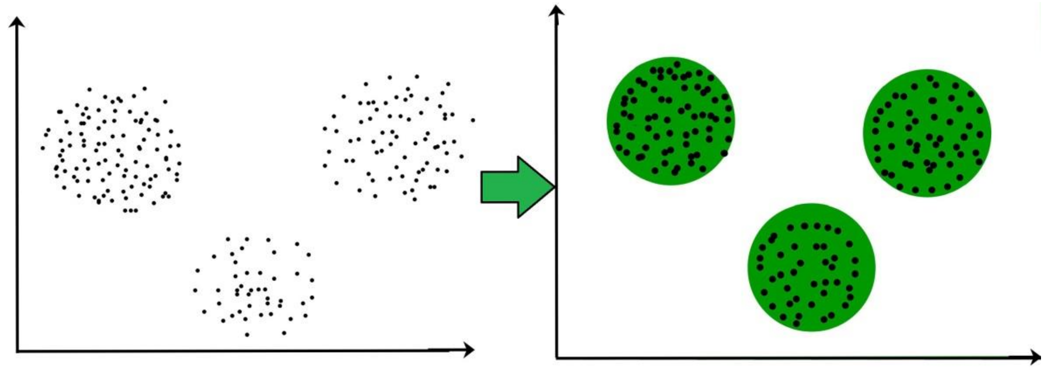
Clustering

The task of grouping data points based on their similarity with each other is called Clustering or Cluster Analysis. This method is defined under the branch of Unsupervised Learning, which aims at gaining insights from unlabelled data points, that is, unlike supervised learning we don’t have a target variable.

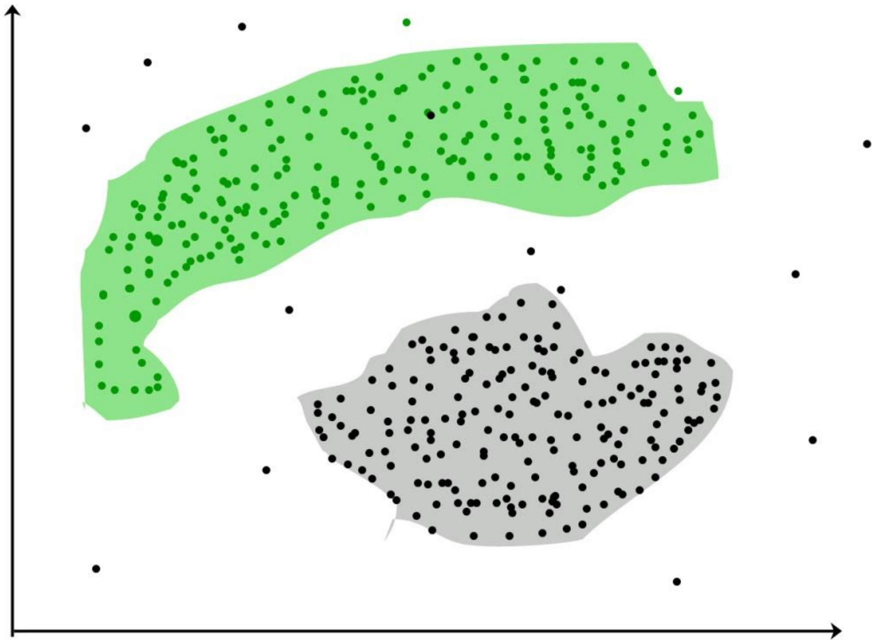
Clustering aims at forming groups of homogeneous data points from a heterogeneous dataset. It evaluates the similarity based on a metric like Euclidean distance, Cosine similarity, Manhattan distance, etc. and then group the points with highest similarity score together.

For Example, In the graph given below, we can clearly see that there are 3 circular clusters forming on the basis of distance.



It is not necessary that the clusters formed must be circular in shape. The shape of clusters can be arbitrary. There are many algortihms that work well with detecting arbitrary shaped clusters.

For example, In the below given graph we can see that the clusters formed are not circular in shape.



Types of Clustering

Broadly speaking, there are 2 types of clustering that can be performed to group similar data points:

- **Hard Clustering:** In this type of clustering, each data point belongs to a cluster completely or not. For example, Let's say there are 4 data point and we have to cluster them into 2 clusters. So each data point will either belong to cluster 1 or cluster 2.

Data Points	**Clusters**
A	C1
B	C2
C	C2
D	C1

- **Soft Clustering:** In this type of clustering, instead of assigning each data point into a separate cluster, a probability or likelihood of that point being that cluster is evaluated. For example, Let's say there are 4 data point and we have to cluster them into 2 clusters. So we will be evaluating a probability of a data point belonging to both clusters. This probability is calculated for all data points.

Data Points	Probability of C1	Probability of C2
A	0.91	0.09
B	0.3	0.7
C	0.17	0.83
D	1	0

Types of Clustering Algorithms

At the surface level, clustering helps in the analysis of unstructured data. Graphing, the shortest distance, and the density of the data points are a few of the elements that influence cluster formation. Clustering is the process of determining how related the objects are based on a metric called the similarity measure. Similarity metrics are easier to locate in smaller sets of features. It gets harder to create similarity measures as the number of features increases. Depending on the type of clustering algorithm being utilized in data mining, several techniques are employed to group the data from the datasets. In this part, the clustering techniques are described. Various types of clustering algorithms are:

1. Centroid-based Clustering (Partitioning methods)
2. Density-based Clustering (Model-based methods)
3. Connectivity-based Clustering (Hierarchical clustering)
4. Distribution-based Clustering

1. Centroid-based Clustering (Partitioning methods)

Partitioning methods are the most easiest clustering algorithms. They group data points on the basis of their closeness. Generally, the similarity measure chosen for these algorithms are Euclidian distance, Manhattan Distance or Minkowski Distance. The datasets are separated into a predetermined number of clusters, and each cluster is referenced by a vector of values. When compared to the vector value, the input data variable shows no difference and joins the cluster.

The primary drawback for these algorithms is the requirement that we establish the number of clusters, “k,” either intuitively or scientifically (using the Elbow Method) before any clustering machine learning system starts allocating the data points. Despite this, it is still the most popular type of clustering. K-means and K-medoids clustering are some examples of this type clustering.

2. Density-based Clustering (Model-based methods)

Density-based clustering, a model-based method, finds groups based on the density of data points. Contrary to centroid-based clustering, which requires that the number of clusters be predefined and is sensitive to initialization, density-based clustering determines the number of clusters automatically and is less susceptible to beginning positions. They are great at handling clusters of different sizes and forms, making them ideally suited for datasets with irregularly shaped or overlapping clusters. These methods manage both dense and sparse data regions by focusing on local density and can distinguish clusters with a variety of morphologies.

In contrast, centroid-based grouping, like k-means, has trouble finding arbitrary shaped clusters. Due to its preset number of cluster requirements and extreme sensitivity to the initial positioning of centroids, the outcomes can vary. Furthermore, the tendency of centroid-based approaches to produce spherical or convex clusters restricts their capacity to handle complicated or irregularly shaped clusters. In conclusion, density-based clustering overcomes the drawbacks of centroid-based techniques by autonomously choosing cluster sizes, being resilient to initialization, and successfully capturing clusters of various sizes and forms. The most popular density-based clustering algorithm is DBSCAN.

3. Connectivity-based Clustering (Hierarchical clustering)

A method for assembling related data points into hierarchical clusters is called hierarchical clustering. Each data point is initially taken into account as a separate cluster, which is subsequently combined with the clusters that are the most similar to form one large cluster that contains all of the data points.

Think about how you may arrange a collection of items based on how similar they are. Each object begins as its own cluster at the base of the tree when using hierarchical clustering, which creates a dendrogram, a tree-like structure. The closest pairings of clusters are then combined into larger clusters after the algorithm examines how similar the objects are to one another. When every object is in one cluster at the top of the tree, the merging process has finished. Exploring various granularity levels is one of the fun things about hierarchical clustering. To obtain a given number of clusters, you can select to cut the dendrogram at a particular height. The more similar two objects are within a cluster, the closer they are. It’s comparable to classifying items according to their family trees, where the nearest relatives are clustered together and the wider branches signify more general connections. There are 2 approaches for Hierarchical clustering:

- **Divisive Clustering:** It follows a top-down approach, here we consider all data points to be part one big cluster and then this cluster is divide into smaller groups.

- **Agglomerative Clustering:** It follows a bottom-up approach, here we consider all data points to be part of individual clusters and then these clusters are clubbed together to make one big cluster with all data points.

4. Distribution-based Clustering

Using distribution-based clustering, data points are generated and organized according to their propensity to fall into the same probability distribution (such as a Gaussian, binomial, or other) within the data. The data elements are grouped using a probability-based distribution that is based on statistical distributions. Included are data objects that have a higher likelihood of being in the cluster. A data point is less likely to be included in a cluster the further it is from the cluster's central point, which exists in every cluster.

A notable drawback of density and boundary-based approaches is the need to specify the clusters a priori for some algorithms, and primarily the definition of the cluster form for the bulk of algorithms. There must be at least one tuning or hyper-parameter selected, and while doing so should be simple, getting it wrong could have unanticipated repercussions. Distribution-based clustering has a definite advantage over proximity and centroid-based clustering approaches in terms of flexibility, accuracy, and cluster structure. The key issue is that, in order to avoid overfitting, many clustering methods only work with simulated or manufactured data, or when the bulk of the data points certainly belong to a preset distribution. The most popular distribution-based clustering algorithm is Gaussian Mixture Model.

Applications of Clustering in different fields:

1. **Marketing:** It can be used to characterize & discover customer segments for marketing purposes.
2. **Biology:** It can be used for classification among different species of plants and animals.
3. **Libraries:** It is used in clustering different books on the basis of topics and information.
4. **Insurance:** It is used to acknowledge the customers, their policies and identifying the frauds.
5. **City Planning:** It is used to make groups of houses and to study their values based on their geographical locations and other factors present.
6. **Earthquake studies:** By learning the earthquake-affected areas we can determine the dangerous zones.
7. **Image Processing**: Clustering can be used to group similar images together, classify images based on content, and identify patterns in image data.

8. **Genetics:** Clustering is used to group genes that have similar expression patterns and identify gene networks that work together in biological processes.
9. **Finance:** Clustering is used to identify market segments based on customer behavior, identify patterns in stock market data, and analyze risk in investment portfolios.
10. **Customer Service:** Clustering is used to group customer inquiries and complaints into categories, identify common issues, and develop targeted solutions.
11. **Manufacturing**: Clustering is used to group similar products together, optimize production processes, and identify defects in manufacturing processes.
12. **Medical diagnosis:** Clustering is used to group patients with similar symptoms or diseases, which helps in making accurate diagnoses and identifying effective treatments.
13. **Fraud detection:** Clustering is used to identify suspicious patterns or anomalies in financial transactions, which can help in detecting fraud or other financial crimes.
14. **Traffic analysis:** Clustering is used to group similar patterns of traffic data, such as peak hours, routes, and speeds, which can help in improving transportation planning and infrastructure.
15. **Social network analysis:** Clustering is used to identify communities or groups within social networks, which can help in understanding social behavior, influence, and trends.
16. **Cybersecurity:** Clustering is used to group similar patterns of network traffic or system behavior, which can help in detecting and preventing cyberattacks.
17. **Climate analysis:** Clustering is used to group similar patterns of climate data, such as temperature, precipitation, and wind, which can help in understanding climate change and its impact on the environment.
18. **Sports analysis:** Clustering is used to group similar patterns of player or team performance data, which can help in analyzing player or team strengths and weaknesses and making strategic decisions.
19. **Crime analysis:** Clustering is used to group similar patterns of crime data, such as location, time, and type, which can help in identifying crime hotspots, predicting future crime trends, and improving crime prevention strategies.

Introduction to Dimensionality Reduction

-

What is Predictive Modeling: Predictive modeling is a probabilistic process that allows us to forecast outcomes, on the basis of some predictors. These predictors are basically features that come into play when deciding the final result, i.e. the outcome of the model.
Dimensionality reduction is the process of reducing the number of features (or dimensions) in a dataset while retaining as much information as possible. This can be done for a variety of reasons, such as to reduce the complexity of a model, to improve the performance of a learning algorithm, or to make it easier to visualize the data. There are several techniques for dimensionality reduction, including principal component analysis (PCA), singular value decomposition (SVD), and linear discriminant analysis (LDA). Each technique uses a different method to project the data onto a lower-dimensional space while preserving important information.

What is Dimensionality Reduction?
Dimensionality reduction is a technique used to reduce the number of features in a dataset while retaining as much of the important information as possible. In other words, it is a process of transforming high-dimensional data into a lower-dimensional space that still preserves the essence of the original data.

In machine learning, high-dimensional data refers to data with a large number of features or variables. The curse of dimensionality is a common problem in machine learning, where the performance of the model deteriorates as the number of features increases. This is because the complexity of the model increases with the number of features, and it becomes more difficult to find a good solution. In addition, high-dimensional data can also lead to overfitting, where the model fits the training data too closely and does not generalize well to new data.

Dimensionality reduction can help to mitigate these problems by reducing the complexity of the model and improving its generalization performance. There are two main approaches to dimensionality reduction: feature selection and feature extraction.

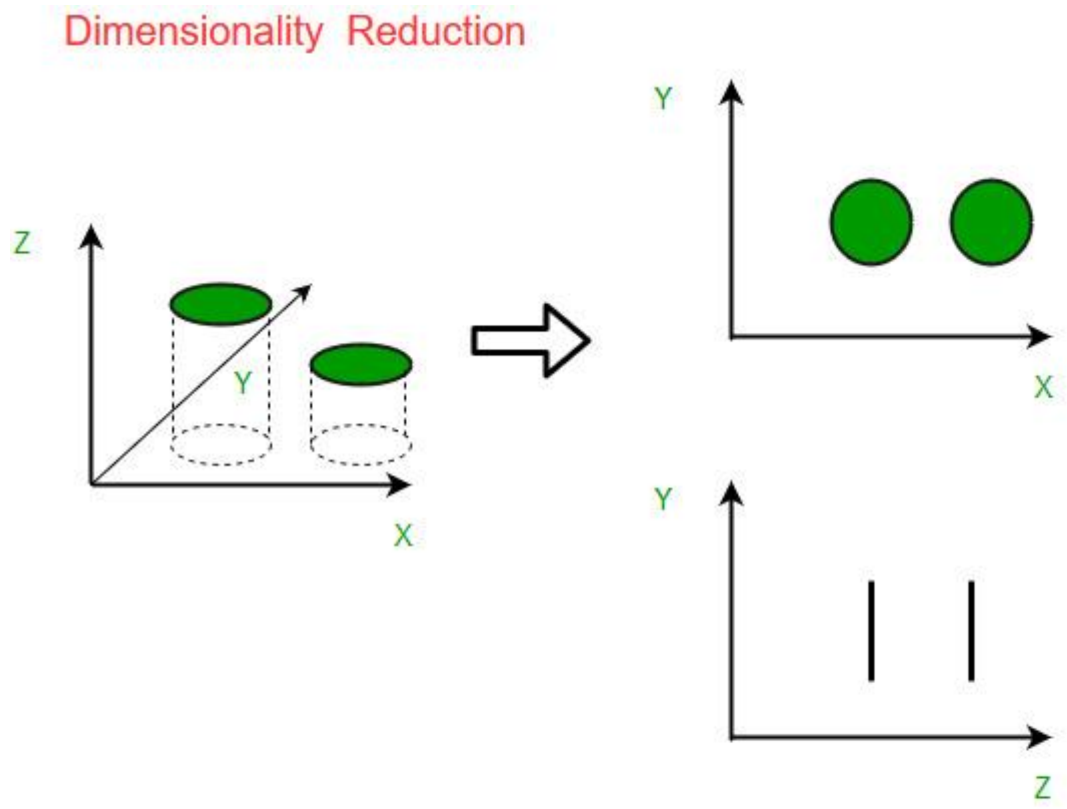
Feature Selection:
Feature selection involves selecting a subset of the original features that are most relevant to the problem at hand. The goal is to reduce the dimensionality of the dataset while retaining the most important features. There are several methods for feature selection, including filter methods, wrapper methods, and embedded methods. Filter methods rank the features based on their relevance to the target variable, wrapper methods use the model performance as the criteria for selecting features, and embedded methods combine feature selection with the model training process.

Feature Extraction:
Feature extraction involves creating new features by combining or transforming the original features. The goal is to create a set of features that captures the essence of the original data in a lower-dimensional space. There are several methods for feature extraction, including principal component analysis (PCA), linear discriminant analysis

(LDA), and t-distributed stochastic neighbor embedding (t-SNE). PCA is a popular technique that projects the original features onto a lower-dimensional space while preserving as much of the variance as possible.

Why is Dimensionality Reduction important in Machine Learning and Predictive Modeling?

An intuitive example of dimensionality reduction can be discussed through a simple e-mail classification problem, where we need to classify whether the e-mail is spam or not. This can involve a large number of features, such as whether or not the e-mail has a generic title, the content of the e-mail, whether the e-mail uses a template, etc. However, some of these features may overlap. In another condition, a classification problem that relies on both humidity and rainfall can be collapsed into just one underlying feature, since both of the aforementioned are correlated to a high degree. Hence, we can reduce the number of features in such problems. A 3-D classification problem can be hard to visualize, whereas a 2-D one can be mapped to a simple 2-dimensional space, and a 1-D problem to a simple line. The below figure illustrates this concept, where a 3-D feature space is split into two 2-D feature spaces, and later, if found to be correlated, the number of features can be reduced even further.



Components of Dimensionality Reduction

There are two components of dimensionality reduction:

- **Feature selection:** In this, we try to find a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem. It usually involves three ways:
  1. Filter
  2. Wrapper

3. Embedded

- **Feature extraction:** This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions.

Methods of Dimensionality Reduction

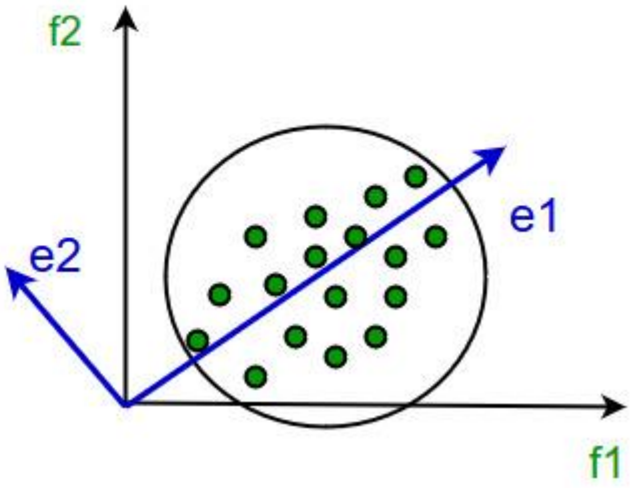
The various methods used for dimensionality reduction include:

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Generalized Discriminant Analysis (GDA)

Dimensionality reduction may be both linear and non-linear, depending upon the method used. The prime linear method, called Principal Component Analysis, or PCA, is discussed below.

Principal Component Analysis

This method was introduced by Karl Pearson. It works on the condition that while the data in a higher dimensional space is mapped to data in a lower dimension space, the variance of the data in the lower dimensional space should be maximum.



It involves the following steps:

- Construct the covariance matrix of the data.
- Compute the eigenvectors of this matrix.
- Eigenvectors corresponding to the largest eigenvalues are used to reconstruct a large fraction of variance of the original data.

Hence, we are left with a lesser number of eigenvectors, and there might have been some data loss in the process. But, the most important variances should be retained by the remaining eigenvectors.

Advantages of Dimensionality Reduction

- It helps in data compression, and hence reduced storage space.
- It reduces computation time.
- It also helps remove redundant features, if any.
- Improved Visualization: High dimensional data is difficult to visualize, and dimensionality reduction techniques can help in visualizing the data in 2D or 3D, which can help in better understanding and analysis.

- Overfitting Prevention: High dimensional data may lead to overfitting in machine learning models, which can lead to poor generalization performance. Dimensionality reduction can help in reducing the complexity of the data, and hence prevent overfitting.
- Feature Extraction: Dimensionality reduction can help in extracting important features from high dimensional data, which can be useful in feature selection for machine learning models.
- Data Preprocessing: Dimensionality reduction can be used as a preprocessing step before applying machine learning algorithms to reduce the dimensionality of the data and hence improve the performance of the model.
- Improved Performance: Dimensionality reduction can help in improving the performance of machine learning models by reducing the complexity of the data, and hence reducing the noise and irrelevant information in the data.

Disadvantages of Dimensionality Reduction

- It may lead to some amount of data loss.
- PCA tends to find linear correlations between variables, which is sometimes undesirable.
- PCA fails in cases where mean and covariance are not enough to define datasets.
- We may not know how many principal components to keep- in practice, some thumb rules are applied.
- Interpretability: The reduced dimensions may not be easily interpretable, and it may be difficult to understand the relationship between the original features and the reduced dimensions.
- Overfitting: In some cases, dimensionality reduction may lead to overfitting, especially when the number of components is chosen based on the training data.
- Sensitivity to outliers: Some dimensionality reduction techniques are sensitive to outliers, which can result in a biased representation of the data.
- Computational complexity: Some dimensionality reduction techniques, such as manifold learning, can be computationally intensive, especially when dealing with large datasets.