Unit-4

Unsupervised learning techniques:

# Clustering :

Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset. It can be defined as **"A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group.*"***

It does it by finding some similar patterns in the unlabelled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.

It is an [unsupervised learning](https://www.javatpoint.com/unsupervised-machine-learning) method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.

After applying this clustering technique, each cluster or group is provided with a cluster- ID. ML system can use this id to simplify the processing of large and complex datasets.

The clustering technique is commonly used for **statistical data analysis.**

Note: Clustering is somewhere similar to the [classification algorithm](https://www.javatpoint.com/classification-algorithm-in-machine-learning), but the difference is the type of dataset that we are using. In classification, we work with the labeled data set, whereas in clustering, we work with the unlabelled dataset.

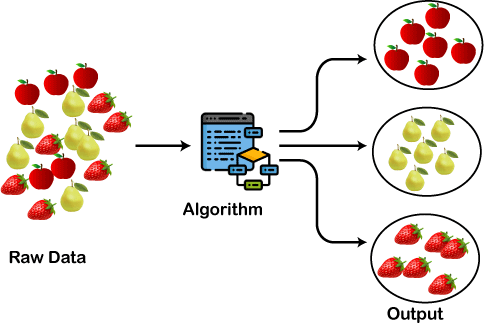
**Example**: Let's understand the clustering technique with the real-world example of Mall: When we visit any shopping mall, we can observe that the things with similar usage are grouped together. Such as the t-shirts are grouped in one section, and trousers are at other sections, similarly, at vegetable sections, apples, bananas, Mangoes, etc., are grouped in separate sections, so that we can easily find out the things. The clustering technique also works in the same way. Other examples of clustering are grouping documents according to the topic.

The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

* Market Segmentation
* Statistical data analysis
* Social network analysis
* Image segmentation
* Anomaly detection, etc.

Apart from these general usages, it is used by the **Amazon** in its recommendation system to provide the recommendations as per the past search of products. **Netflix** also uses this technique to recommend the movies and web-series to its users as per the watch history.

The below diagram explains the working of the clustering algorithm. We can see the different fruits are divided into several groups with similar properties.



## Types of Clustering Methods

The clustering methods are broadly divided into **Hard clustering** (datapoint belongs to only one group) and **Soft Clustering** (data points can belong to another group also). But there are also other various approaches of Clustering exist. Below are the main clustering methods used in Machine learning:

1. **Partitioning Clustering**
2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

## Applications of Clustering

Below are some commonly known applications of clustering technique in Machine Learning:

* **In Identification of Cancer Cells:** The clustering algorithms are widely used for the identification of cancerous cells. It divides the cancerous and non-cancerous data sets into different groups.
* **In Search Engines:** Search engines also work on the clustering technique. The search result appears based on the closest object to the search query. It does it by grouping similar data objects in one group that is far from the other dissimilar objects. The accurate result of a query depends on the quality of the clustering algorithm used.
* **Customer Segmentation:** It is used in market research to segment the customers based ontheir choice and preferences
* **In Biology:** It is used in the biology stream to classify different species of plants and animals using the image recognition technique.
* **In Land Use:** The clustering technique is used in identifying the area of similar lands use in the GIS database. This can be very useful to find that for what purpose the particular land should be used, that means for which purpose it is more suitable.

# 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,](https://www.javatpoint.com/unsupervised-machine-learning) 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](https://www.javatpoint.com/clustering-in-machine-learning) 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?K-Means Clustering Algorithm

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.

limits of k means:

* 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

**Image Segmentation**

* A digital image is made up of various components that need to be “analysed”, let’s use that word for simplicity sake and the “analysis” performed on such components can reveal a lot of hidden information from them. This information can help us address a plethora of business problems – which is one of the many end goals that are linked with image processing.
* Image Segmentation is the process by which a digital image is partitioned into various subgroups (of pixels) called Image Objects, which can reduce the complexity of the image, and thus analysing the image becomes simpler.
* We use various image segmentation algorithms to split and group a certain set of pixels together from the image. By doing so, we are actually assigning labels to pixels and the pixels with the same label fall under a category where they have some or the other thing common in them.
* Using these labels, we can specify boundaries, draw lines, and separate the most required objects in an image from the rest of the not-so-important ones. In the below example, from a main image on the left, we try to get the major components, e.g. chair, table etc. and hence all the chairs are colored uniformly. In the next tab, we have detected instances, which talk about individual objects, and hence the all the chairs have different colors.
* This is how different methods of segmentation of images work in varying degrees of complexity and yield different levels of outputs.

# Data Preprocessing in Machine learning

* Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.
* When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

## Why

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Semi-Supervised Learning:

Semi-Supervised learning is a type of Machine Learning algorithm that represents the intermediate ground between Supervis and Unsupervised learning algorithms. It uses the combination of labeled and unlabeled datasets during the training period.

## Assumptions followed by Semi-Supervised Learning

To work with the unlabeled dataset, there must be a relationship between the objects. To understand this, semi-supervised learning uses any of the following assumptions:

* **Continuity**
* As per the continuity assumption, the objects near each other tend to share the same group or label. This assumption is also used in supervised learning, and the datasets are separated by the decision boundaries. But in semi-supervised, the decision boundaries are added with the smoothness assumption in low-density boundaries.
* **Cluster assumptions-** In this assumption, data are divided into different discrete clusters. Further, the points in the same cluster share the output label.
* **Manifold assumptions-** This assumption helps to use distances and densities, and this data lie on a manifold of fewer dimensions than input space.
* The dimensional data are created by a process that has less degree of freedom and may be hard to model directly. **(This assumption becomes practical if high).**

## Working of Semi-Supervised Learning

Semi-supervised learning uses pseudo labeling to train the model with less labeled training data than supervised learning. The process can combine various neural network models and training ways. The whole working of semi-supervised learning is explained in the below points:

* Firstly, it trains the model with less amount of training data similar to the supervised learning models. The training continues until the model gives accurate results.
* The algorithms use the unlabeled dataset with pseudo labels in the next step, and now the result may not be accurate.
* Now, the labels from labeled training data and pseudo labels data are linked together.
* The input data in labeled training data and unlabeled training data are also linked.
* In the end, again train the model with the new combined input as did in the first step. It will reduce errors and improve the accuracy of the model.

# DBSCAN

Density-Based Clustering refers to one of the most popular unsupervised learning methodologies used in model building and machine learning algorithms. The data points in the region separated by two clusters of low point density are considered as noise. The surroundings with a radius ε of a given object are known as the ε neighborhood of the object. If the ε neighborhood of the object comprises at least a minimum number, MinPts of objects, then it is called a core object.

There are two different parameters to calculate the density-based clustering EPS: It is considered as the maximum radius of the neighborhood.

MinPts: MinPts refers to the minimum number of points in an Eps neighborhood of that point.

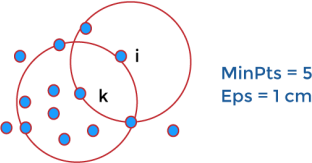
NEps (i) : { k belongs to D and dist (i,k) < = Eps} Directly density reachable:

A point i is considered as the directly density reachable from a point k with respect to Eps,

MinPts if

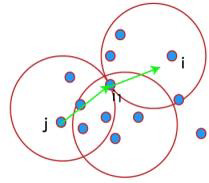
i belongs to NEps(k) Core point condition:

NEps (k) >= MinPts



#### Density reachable:

A point denoted by i is a density reachable from a point j with respect to Eps, MinPts if there is a sequence chain of a point i1,…., in, i1 = j, pn = i such that ii + 1 is directly density reachable from ii.



#### Density connected:

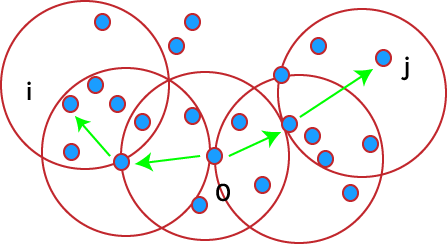
A point i refers to density connected to a point j with respect to Eps, MinPts if there is a point o such that both i and j are considered as density reachable from o with respect to Eps and MinPts.

# Gaussian Discriminant Analysis

There are two types of Supervised Learning algorithms are used in Machine Learning for classification.

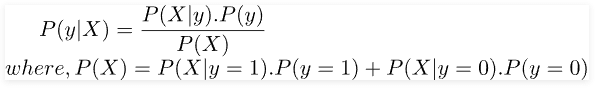
1. Discriminative Learning Algorithms
2. Generative Learning Algorithms

Logistic Regression, Perceptron, and other Discriminative Learning Algorithms are examples of discriminative learning algorithms. These algorithms attempt to determine a boundary between classes in the learning process. A Discriminative Learning Algorithm might be used to solve a classification problem that will determine if a patient has malaria. The boundary is then checked to see if the new example falls on the boundary, **P(y|X)**, i.e., Given a feature set X, what is its probability of belonging to the class "y".



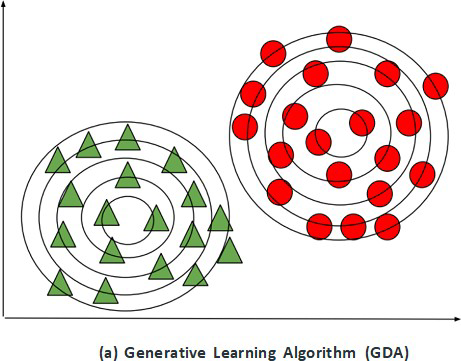
Generative Learning Algorithms, on the other hand, take a different approach. They try to capture each class distribution separately rather than finding a boundary between classes. A Generative Learning Algorithm, as mentioned, will examine the distribution of infected and healthy patients separately. It will then attempt to learn each distribution's features individually. When a new example is presented, it will be compared to both distributions, and the class that it most closely resembles will be assigned, **P(X|y)** for a given **P(y)** here, P(y) is known as a class prior.

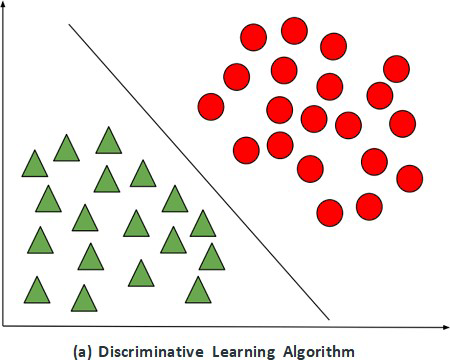
These Bayes Theory predictions are used to predict generative learning algorithms



By analysing only, the numbers of **P(X|y)** as well as **P(y)** in the specific class, we can determine P(y), i.e., considering the characteristics of a sample, how likely is it that it belongs to class "y".

Gaussian Discriminant Analysis is a Generative Learning Algorithm that aims to determine the distribution of every class. It attempts to create the Gaussian distribution to each category of data in a separate way. The likelihood of an outcome in the case using an algorithm known as the Generative learning algorithm is very high if it is close to the centre of the contour, which corresponds to its class. It diminishes when we move away from the middle of the contour. Below are images that illustrate the differences between Discriminative as well as Generative Learning Algorithms.



Curse of dimensionality : Handling the high-dimensional data is very difficult in practice, commonly known as the *curse of dimensionality.* If the dimensionality of the input dataset increases, any machine learning algorithm and model becomes more complex. As the number of features increases, the number of samples also gets increased

proportionally, and the chance of overfitting also increases. If the machine learning model is trained on high-dimensional data, it becomes overfitted and results in poor performance.

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

Dimensionality reduction & Methods:

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

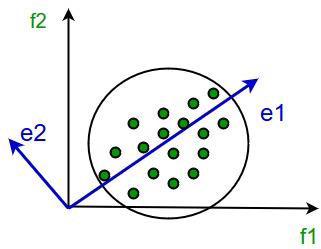
**Methods of Dimensionality Reduction**

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 or 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 a 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 eigen values 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.

**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.

Principal Component Analysis

* + Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in [machine learning.](https://www.javatpoint.com/machine-learning) It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.
  + PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.
  + PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are ***image processing, movie recommendation system, optimizing the power allocation in various communication channels.*** It is a feature extraction technique, so it contains the important variables and drops the least important variable.
  + The PCA algorithm is based on some mathematical concepts such as:
* Variance and Covariance
* Eigenvalues and Eigen factors Some common terms used in PCA algorithm:
* **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
* **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
* **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
* **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
* **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

# What is Scikit-Learn (Sklearn)

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon **NumPy, SciPy** and **Matplotlib**.

### Origin of Scikit-Learn

It was originally called ***scikits.learn*** and was initially developed by David Cournapeau as a Google summer of code project in 2007. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project at another level and made the first public release (v0.1 beta) on 1st Feb. 2010.

Let’s have a look at its version history −

* May 2019: scikit-learn 0.21.0
* March 2019: scikit-learn 0.20.3
* December 2018: scikit-learn 0.20.2
* November 2018: scikit-learn 0.20.1
* September 2018: scikit-learn 0.20.0
* July 2018: scikit-learn 0.19.2

### Features

Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modeling the data. Some of the most popular groups of models provided by Sklearn are as follows −

**Supervised Learning algorithms** − Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.

**Unsupervised Learning algorithms** − On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

**Clustering** − This model is used for grouping unlabeled data.

**Cross Validation** − It is used to check the accuracy of supervised models on unseen data.

**Dimensionality Reduction** − It is used for reducing the number of attributes in data which can be further used for summarisation, visualisation and feature selection.

**Ensemble methods** − As name suggest, it is used for combining the predictions of multiple supervised models. **Feature extraction** − It is used to extract the features from

data to define the attributes in image and text data. **Feature selection** − It is used to identify useful attributes to create supervised mode

data to define the attributes in image and text data. **Feature selection** − It is used to identify useful attributes to create supervised mode

**KERNEL PCA:**

PCA is a linear method. That is it can only be applied to datasets which are linearly separable. It does an excellent job for datasets, which are linearly separable. But, if we use it to non-linear datasets, we might get a result which may not be the optimal dimensionality reduction. Kernel PCA uses a kernel function to project dataset into a higher dimensional feature space, where it is linearly separable. It is similar to the idea of Support Vector Machines.

There are various kernel methods like linear, polynomial, and gaussian.

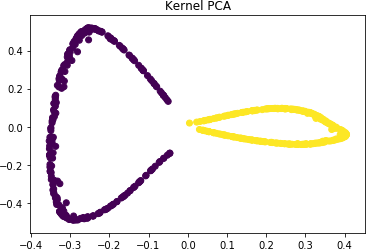
**Example:** Applying kernel PCA on this dataset with RBF kernel with a gamma value of 15. from sklearn.decomposition import KernelPCA

kpca = KernelPCA(kernel ='rbf', gamma = 15) X\_kpca = kpca.fit\_transform(X)

plt.title("Kernel PCA")

plt.scatter(X\_kpca[:, 0], X\_kpca[:, 1], c = y)

plt.show()



In the kernel space the two classes are linearly separable. Kernel PCA uses a kernel function to project the dataset into a higher-dimensional space, where it is linearly separable.

**Randomized PCA algorithm**

Both SVD and NIPALS are not very efficient when number of rows in dataset is very large (e.g. hundreds of thousands values or even more). Such datasets can be easily obtained in case of for example hyperspectral images. Direct use of the traditional algorithms with such datasets often leads to a lack of memory and long computational time.

One of the solution here is to use probabilistic algorithms, which allow to reduce the number of values needed fo r estimation of principal components. Starting from *0.9.0* one of the probabilistic approach is also implemented

in *mdatools*. The original idea can be found in [this paper](https://epubs.siam.org/doi/10.1137/090771806) and some examples on using the approach for PCA analysis of hyperspectral

A Randomized Algorithm for PCA 2

• Form Y = AΩ. •

QR decompose Y and discard R.

The main theoretical result is:

E||A − QQ∗A|| ≤ 1 + 4 ∗ √ k + p p − 1 p min(m, n) σk+1(A).

Proof Sketch.

Apply the triangle inequality many times in order to split the error into a part that involves optimizing over a space of dimension k and a separate high dimensional part.

Let Ω ∈ R n×(k+12) , W ∈ R (k+12)×n and Z ∈ R k×(k+12)

||A − QQ∗A|| ≤ 2||A − AΩW|| + 2||AΩ − QZ||||W||.

we want to choose W and Z to show

||A − QQ∗A|| ≤ Cσk+1(A).

The algorithm forms Q’s columns from singular vectors corresponding to the k + p greatest singular values of AΩ. This lets us choose Z such that

||A − QQ∗A|| ≤ σk+1(AΩ) ≤ ||Ω||σk+1(A)

where we understand the second inequality by recalling that we are working with the spectral norm in this note.

The existence of a (k +p)×n matrix W such that ||A−AΩW|| ≤ Cσk+1(A) is tedious and shown in the appendix of using results from [1]. A few notes about the result:

• A few iterations of the power method in our computation of Y can improve the accuracy of our method.

• We expect the bound in to involve a factor of σk+1(A) as σk+1(A) is the theoretical best bound we can find.

• Notice that increasing p greatly improves accuracy.