# Clustering for Unsupervised Learning

This lesson will introduce clustering techniques in Python for unsupervised machine learning.

WE'LL COVER THE FOLLOWING

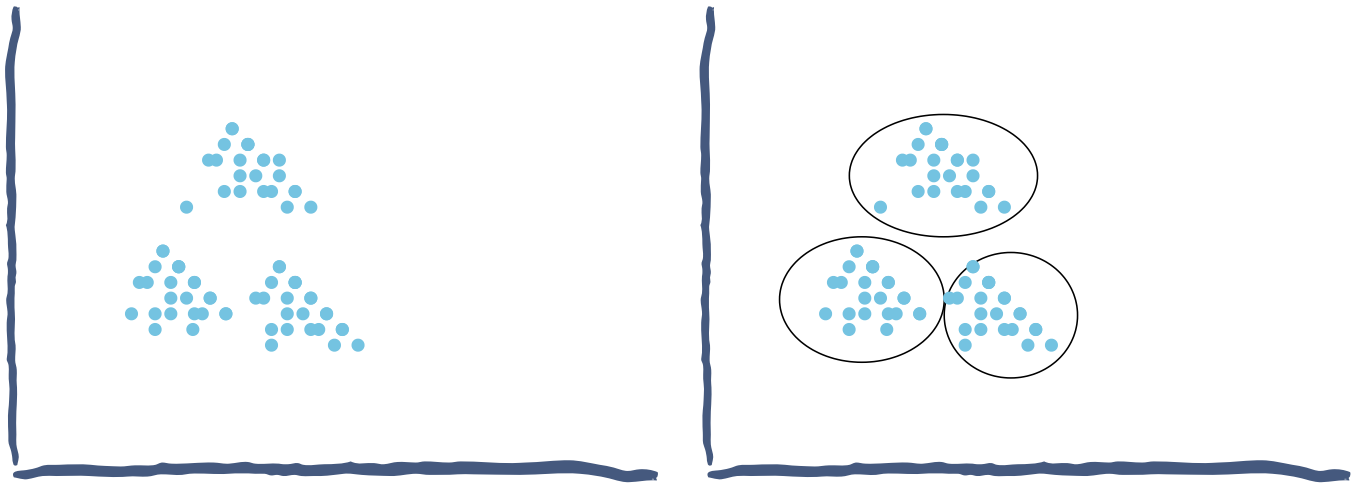


## Unsupervised learning #

**Unsupervised learning** is when we use unlabeled data to allow a model to learn relationships between data observations and pick up on underlying patterns. Most data in the world is unlabeled, which makes unsupervised learning a very useful method of machine learning. The most common algorithms for unsupervised learning are *clustering* algorithms. We will look at some of these later in this lesson.

## Clustering #

**Clustering** is the process of dividing the data points into groups such that the data points in a group are similar to each other, but they are dissimilar to other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.



Data points

Data points divided into clusters

In the above illustration, we can see the data was divided into three clusters based on the similarity of the data points.

## Applications of Clustering #

Clustering is widely used in different fields for different purposes such as in:

- **Marketing**: for discovering customer segments
- **Insurance**: categorizing potential customers or differentiating between potential fraudulent customers and safe customers
- **Biology**: for grouping together different species of animals or plants
- **Libraries:** for placing books on similar topics together
- **City Planning**: for grouping housing areas

## Common clustering algorithms #

Some of the common clustering algorithms are:

- K-means clustering
- Agglomerative hierarchical clustering
- Mean shift Clustering
- Density-based spectral

We will only look at K-means clustering algorithm.

# K-means Clustering #

The objective of a clustering algorithm is to group similar points and find underlying patterns in the data. K-means finds a fixed number (**k**) of clusters.

To understand this algorithm, we need to understand what a *centroid* is. A **centroid** is the location of the center of the cluster. Every data point is allocated to its nearest cluster. The nearest cluster is found by calculating the distance to the centroids of all clusters and then choosing the cluster whose centroid was the closest.

The centroid is found by averaging the data points in the cluster. That is why there is the term `means` in the name of the algorithm.

## How k-means works? #

The algorithm is as follows:

1. Choose k random centroids. Each centroid forms a cluster.
2. Assign every point to its closest cluster.
3. Compute the new centroid of all clusters.
4. Repeat steps 2 and 3 until the specified number of iterations has been reached or the centroids have stopped changing.

In the next lesson, we will implement K-Means in Python.