There are several Machine Learning algorithms, one such important algorithm of machine learning is Clustering.

Clustering is an unsupervised learning method in machine learning. It means that it is a machine learning algorithm that can draw inferences from a given dataset on its own, without any kind of human intervention.

Types of clustering method

There are five types of clustering methods in machine learning, these are as follows:

1.Partitioning Clustering

2.Density-Based Clustering

3.Distribution Model-Based Clustering

4.Hierarchical Clustering

5.Fuzzy Clustering

About Hierarchical Clustering

Hierarchical clustering, also known as hierarchical cluster analysis or HCA, is another unsupervised machine learning approach for grouping unlabeled datasets into clusters.

The hierarchy of clusters is developed in the form of a tree in this technique, and this tree-shaped structure is known as the dendrogram.

Simply speaking, Separating data into groups based on some measure of similarity, finding a technique to quantify how they're alike and different, and limiting down the data is what hierarchical clustering is all about.

Hierarchical clustering method functions in two approaches-

1.Agglomerative

2.Divisive

Approaches of Hierarchical Clustering

1.Agglomerative clustering:

Agglomerative Clustering is a bottom-up strategy in which each data point is originally a cluster of its own, and as one travels up the hierarchy, more pairs of clusters are combined. In it, two nearest clusters are taken and joined to form one single cluster.

2.Divisive clustering:

The divisive clustering algorithm is a top-down clustering strategy in which all points in the dataset are initially assigned to one cluster and then divided iteratively as one progresses down the hierarchy.

It partitions data points that are clustered together into one cluster based on the slightest difference. This process continues till the desired number of clusters is obtained.

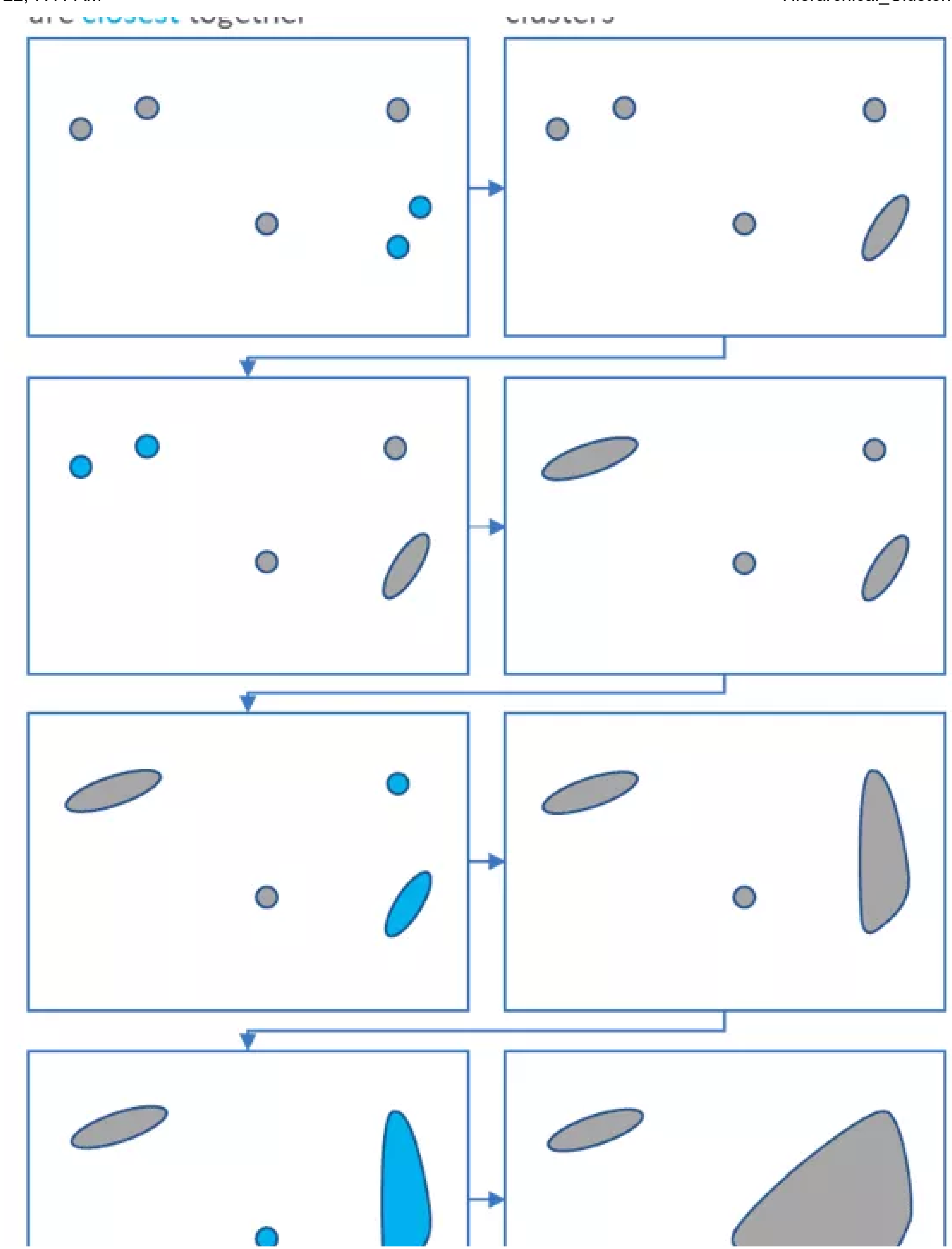
How hierarchical clustering works

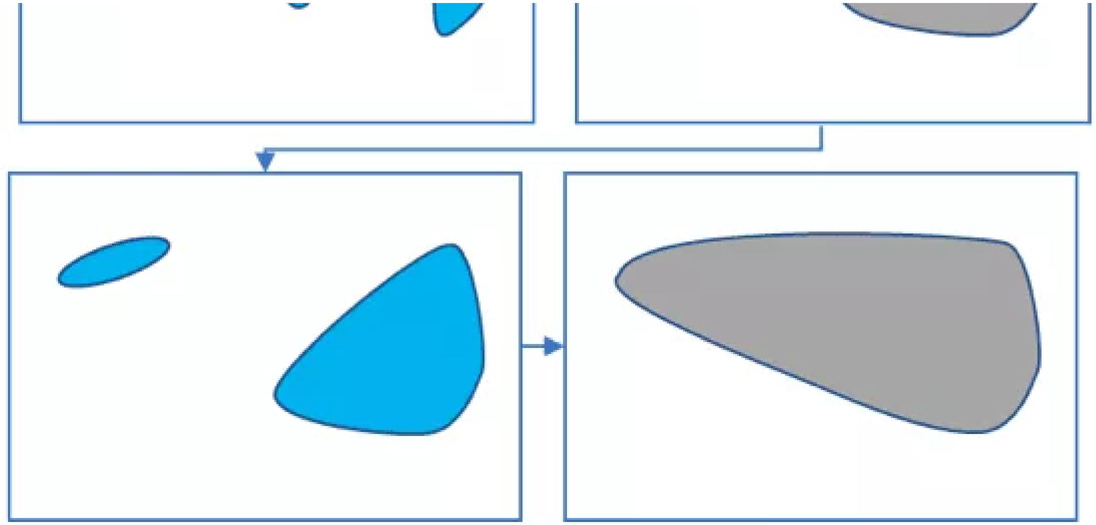
Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:

(1) identify the two clusters that are closest together,

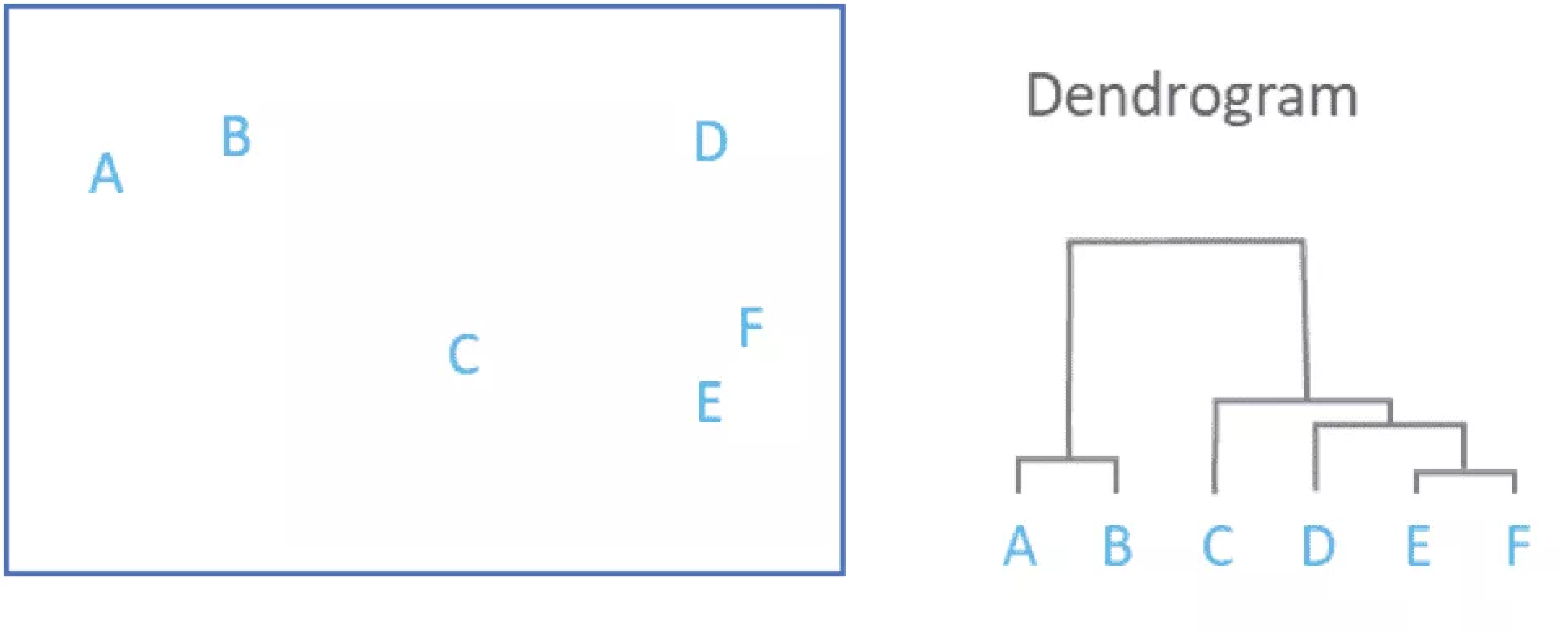
and (2) merge the two most similar clusters. This iterative process continues until all the clusters are merged together. This is illustrated in the diagrams below

Identify the two clusters that are closest together
Merge the two most similar clusters


are closest together
clusters



The main output of Hierarchical Clustering is a dendrogram, which shows the hierarchical relationship between the clusters:



Double-click (or enter) to edit

Double-click (or enter) to edit

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values
```

Training the Hierarchical Clustering model on the dataset

```
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage = 'ward')
y_hc = hc.fit_predict(X)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-96bcb8a45aad> in <module>
      1 from sklearn.cluster import AgglomerativeClustering
      2 hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage = 'ward')
----> 3 y_hc = hc.fit_predict(X)

NameError: name 'X' is not defined
```

SEARCH STACK OVERFLOW

Visualising the clusters

Double-click (or enter) to edit

```
plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

❗ 0s completed at 7:39 AM ● ✕