Introduction

Hierarchical clustering is one of the most famous clustering techniques used in unsupervised machine learning. K-means and hierarchical clustering are the two most popular and effective clustering algorithms. The working mechanism they apply in the backend allows them to provide such a high level of performance.



## Overview

Hierarchical clustering is an unsupervised machine-learning clustering strategy. Unlike [K-means clustering](https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/), tree-like morphologies are used to bunch the dataset, and dendrograms are used to create the hierarchy of the clusters.

Here, dendrograms are the tree-like morphologies of the dataset, in which the X axis of the dendrogram represents the features or columns of the dataset, and the Y axis of the dendrogram represents the Euclidian distance between data observations.

[Agglomerative Clustering](https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/) Agglomerative Clustering is one of the most common hierarchical clustering techniques. Dataset – Credit Card Dataset. **Assumption:** The clustering technique assumes that each data point is similar enough to the other data points that the data at the starting can be assumed to be clustered in 1 cluster

Note:

What is a LabelEncoder in Python?

Label encoding is a simple and effective way to convert categorical variables into numerical form.

What is the dendrogram method?

A dendrogram is a type of diagram used to represent hierarchical clustering of objects or data points

Plot the hierarchical clustering as a dendrogram.

The dendrogram illustrates how each cluster is composed by drawing a U-shaped link between a non-singleton cluster and its children. The top of the U-link indicates a cluster merge. The two legs of the U-link indicate which clusters were merged. The length of the two legs of the U-link represents the distance between the child clusters. It is also the cophenetic distance between original observations in the two children clusters.

Ward's method means calculating the incremental sum of squares. Half square Euclidean distance is the only distance measure that can be used with this clustering method.

**PROJECT**

**STEP1: import pandas as pd**

**#from sklearn.cluster import AgglomerativeClustering**

**#cluster = AgglomerativeClustering(n\_clusters=5,affinity = 'l1',linkage='single')**

**df=pd.read\_csv(r"C:\Users\GT-499\Downloads\Mall\_Customers.csv")**

**df**

**STEP2:**

**from sklearn import preprocessing**

**import scipy.cluster.hierarchy as sch**

**from sklearn.cluster import AgglomerativeClustering**

**step3:**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**plt.figure(1 , figsize = (15 , 6))**

**n = 0**

**for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:**

**n += 1**

**plt.subplot(1 , 3 , n)**

**plt.subplots\_adjust(hspace = 0.5 , wspace = 0.5)**

**sns.distplot(df[x] , bins = 15)**

**plt.title('Distplot of {}'.format(x))**

**plt.show()**

**# Sub(1,2,2),first row,secon column,first plot**

**Step4:**

**label\_encoder = preprocessing.LabelEncoder()**

**df['Gender'] = label\_encoder.fit\_transform(df['Gender'])**

**df.head()**

**step5:**

**plt.figure(1, figsize = (16 ,8))**

**dendrogram = sch.dendrogram(sch.linkage(df, method = "ward"))**

**plt.title('Dendrogram')**

**plt.xlabel('Customers')**

**plt.ylabel('Euclidean distances')**

**plt.show()**