

LP2

## Assignment 2.

Date of Completion  
27.8.2020

Date of Submission:-  
30.9.2020

Title:- Visualize the clusters using suitable tool

Problem Statement :- Consider a suitable dataset. For clustering of data instances in different groups, apply different clustering techniques. Visualize the clusters using suitable tool

Learning Objective :- Understand clustering and different algorithms used for clustering data

Learning Outcomes :- Students will be able to understand different clustering methods and implement them.

Theor. Software / Hardware requirements :- Python, numpy, sklearn (packages)

Theory :-

Clustering Algorithms :-

1. Clustering is a machine Learning technique that involves the grouping of points.



2. It is used to analyse patterns and grouping.
3. It is a unsupervised learning algorithm.

Types of clustering algorithms:-

- 1) K Means clustering
- 2) Hierarchical clustering
- 3) Mean shift clustering
- 4) Fuzzy C clustering
- 5) Spectral Clustering etc.

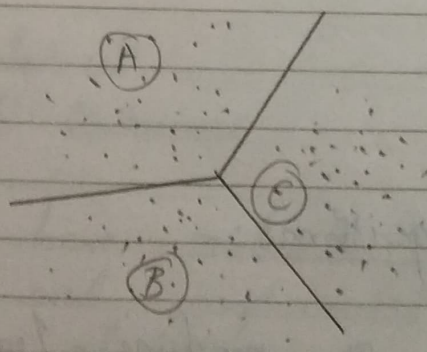
Clustering Methods:-

1. Density Based
2. Hierarchical Based
3. Partitioning Based
4. Grid-based

K-Means algorithm:-

1. Simplest unsupervised clustering algorithm.
2. It partitions  $n$  observation into  $k$  clusters where each observation belongs to clusters nearest mean serving as a prototype cluster.

A



Applications:-

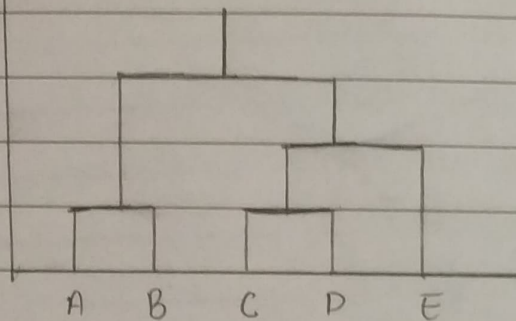
Marketing, Biology, Earthquake studies etc.

## Hierarchical clustering:-

- 1) It builds a hierarchy of clusters.
- 2) 2 types :-
  - a) Agglomerative :- "Bottom up" approach
  - b) Divisive :- "Top down" approach.
- 3) Results are usually presented in a dendrogram.
- 4) Slow.

## Linkage Criteria:-

To compute the distance between two similar clusters many linkage criteria have developed.



## Dataset Used:-

K-means :- iris dataset

Hierarchical :- Mall Customers.

Conclusion:- Thus I have understood different clustering algorithms and implemented K-means & hierarchical clustering algorithm.

## **CODE:**

### **Hierarchical Clustering:**

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

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

#using dendrogram for finding optimal number of clusterings
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Eucladian Distances')
plt.show()

#training the cluster Using HC
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters = 3, affinity='euclidean', linkage='ward')
y_hc = hc.fit_predict(X)

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.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

### **KMeans Clustering:**

```
# Importing the libraries
from sklearn import datasets
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#Dataset
iris = datasets.load_iris()

# Importing the dataset
x = iris.data[:, :2]
y = iris.data[:, :2]
```



# Using the elbow method to find the optimal number of clusters

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=4)

y\_kmeans = kmeans.fit\_predict(x)

# print(y\_kmeans)

kmeans.cluster\_centers\_

# Fitting K-Means to the dataset

plt.scatter(x[:,0], x[:,1], c=y\_kmeans, cmap='gist\_rainbow')

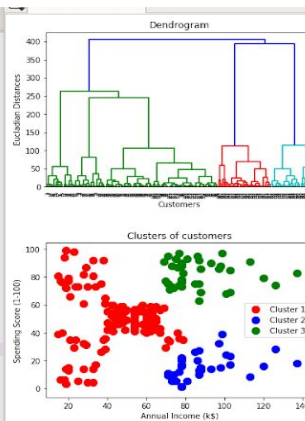
plt.xlabel('Sepal Length')

plt.ylabel('Sepal Width')

## OUTPUT:

## HIERARCHICAL CLUSTERING

```
home.py x kmeans.py x n.py x
4
5 import numpy as np
6 import pandas as pd
7 import matplotlib.pyplot as plt
8
9 dataset = pd.read_csv('Mall_Customers.csv')
10 X = dataset.iloc[:, [3, 4]].values
11
12 #using dendrogram for finding optimal number of clusterings
13 import scipy.cluster.hierarchy as sch
14 dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))
15 plt.title('Dendrogram')
16 plt.xlabel('Customers')
17 plt.ylabel('Eucladian Distances')
18 plt.show()
19
20 #training the cluster Using HC
21 from sklearn.cluster import AgglomerativeClustering
22 hc = AgglomerativeClustering(n_clusters = 3, affinity='euclidean', linkage='ward')
23 y_hc = hc.fit_predict(X)
24
25
26 plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
27 plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
28 plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
29 plt.title('Clusters of customers')
30 plt.xlabel('Annual Income (k$)')
31 plt.ylabel('Spending Score (1-100)')
32 plt.legend()
33 plt.show()
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
```



In [30]:

# K-MEANS CLUSTERING:

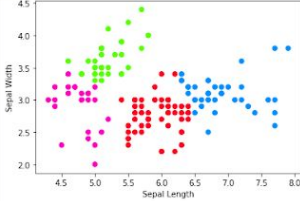
```
1#!/usr/bin/env python3
2# -*- coding: utf-8 -*-
3"""
4Created on Tue Sep 29 16:11:43 2020
5
6@author: srushti
7"""
8
9# K-Means Clustering
10
11# Importing the libraries
12from sklearn import datasets
13import numpy as np
14import matplotlib.pyplot as plt
15import pandas as pd
16
17#Dataset
18iris = datasets.load_iris()
19
20# Importing the dataset
21x = iris.data[:, :2]
22# print(x)
23y = iris.data[:, :2]
24# print(y)
25
26# Using the elbow method to find the optimal number of clusters
27from sklearn.cluster import KMeans
28kmeans = KMeans(n_clusters=4)
29y_kmeans = kmeans.fit_predict(x)
30
31# print(y_kmeans)
32kmeans.cluster_centers_
33
34
35# Fitting K-Means to the dataset
36plt.scatter(x[:,0], x[:,1], c=y_kmeans, cmap='gist_rainbow')
37plt.xlabel('Sepal Length')
38plt.ylabel('Sepal Width')
39
```

Name	Size	Type	Date Modified
n.py	3 KB	py File	27/09/20 9:48 PM
natural_language_processing.py	1 KB	py File	23/11/16 2:22 AM
natural_language_processing.R	1 KB	R File	23/11/16 2:24 AM
Restaurant_Reviews.tsv	59 KB	tsv File	15/11/16 8:08 AM

HelpVariable explorerFile explorer

IPython console

Console 1/A



In [28]:

IPython consoleHistory log

Permissions: RWEnd-of-lines: LFEncoding: UTF-8Line: 31Column: 18Memory: 79%