

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')

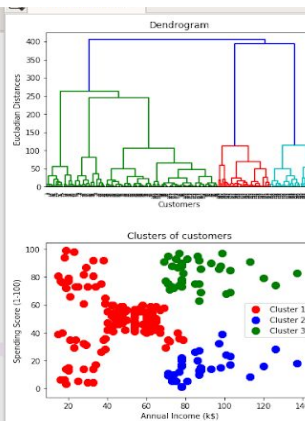
plt.xlabel('Sepal Length')

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]:

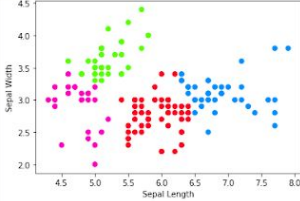
K-MEANS CLUSTERING:

```
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 x

In [28]:

IPython consoleHistory log

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