Cluster Analysis

Introduction

For this lab, we are going to use the Agglomerative analysis technique to cluster the given data.

Objective

We going to cluster the cities of US based on the crime of that cities.

The Data

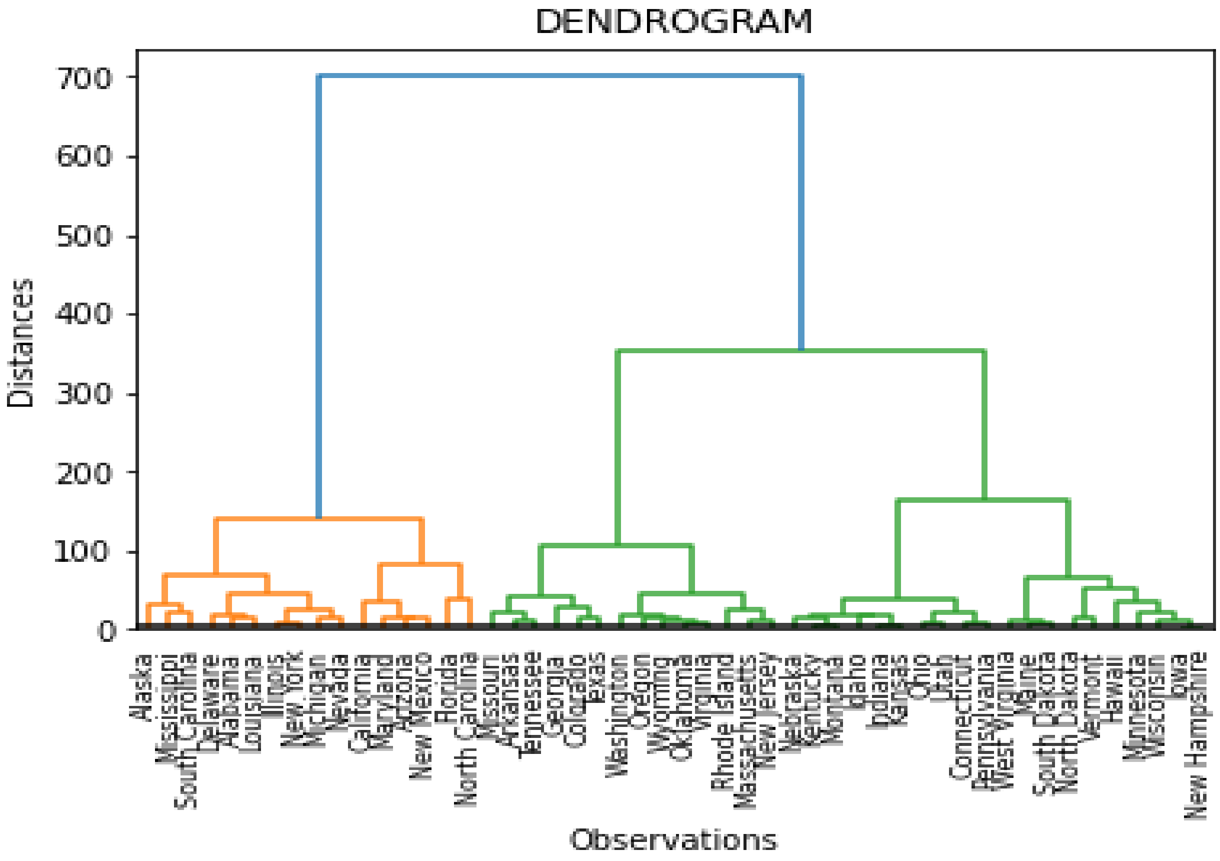
This data set consists of number of crime (murder, assault, urban pop, rape) in different cities of US country.

```
import pandas as pd
df= pd.read_csv(r"C:\Users\my pc\Desktop\MBA - BA II\lab/usarrest.csv")
# Using set_index() method on 'unnamed' column
df = df.set_index('City_Name')
df.head()



Out[2]:
            Murder  Assault  UrbanPop  Rape
City_Name
Alabama       13.2      236        58  21.2
Alaska        10.0      263        48  44.5
Arizona        8.1      294        80  31.0
Arkansas       8.8      190        50  19.5
California     9.0      276        91  40.6
```

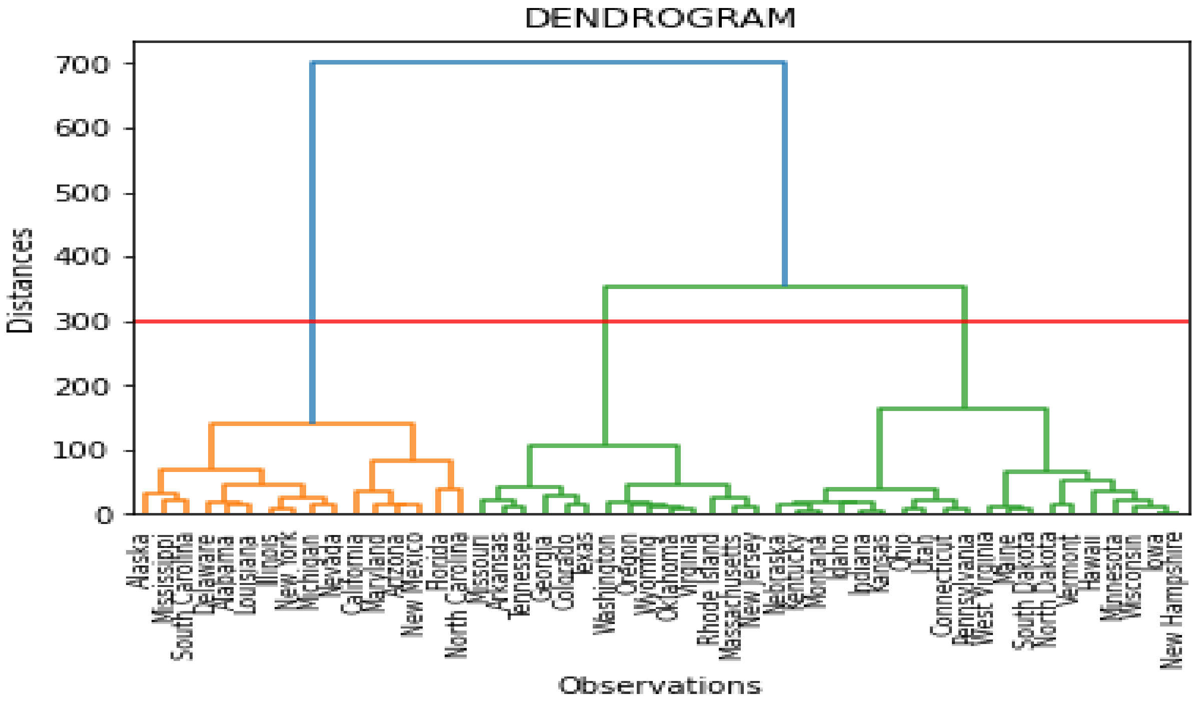
Dendrogram

Next, we need to know the clusters that we want our data to be split to. We will use the scipy library to create the dendrograms for our dataset. Execute the following script to do so:

```
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, ward
result=ward(df)
dendrogram(result,leaf_rotation=90, leaf_font_size=8, labels=df.index)
plt.title("DENDROGRAM")
plt.xlabel('Observations')
plt.ylabel('Distances')
plt.show()
```



By this dendogram we can able to see the number of cluster based on the distance between the point we can make the threshold of 300 as max distance we get 3 clusters as shown in the following figure:



Agglomerative Clustering

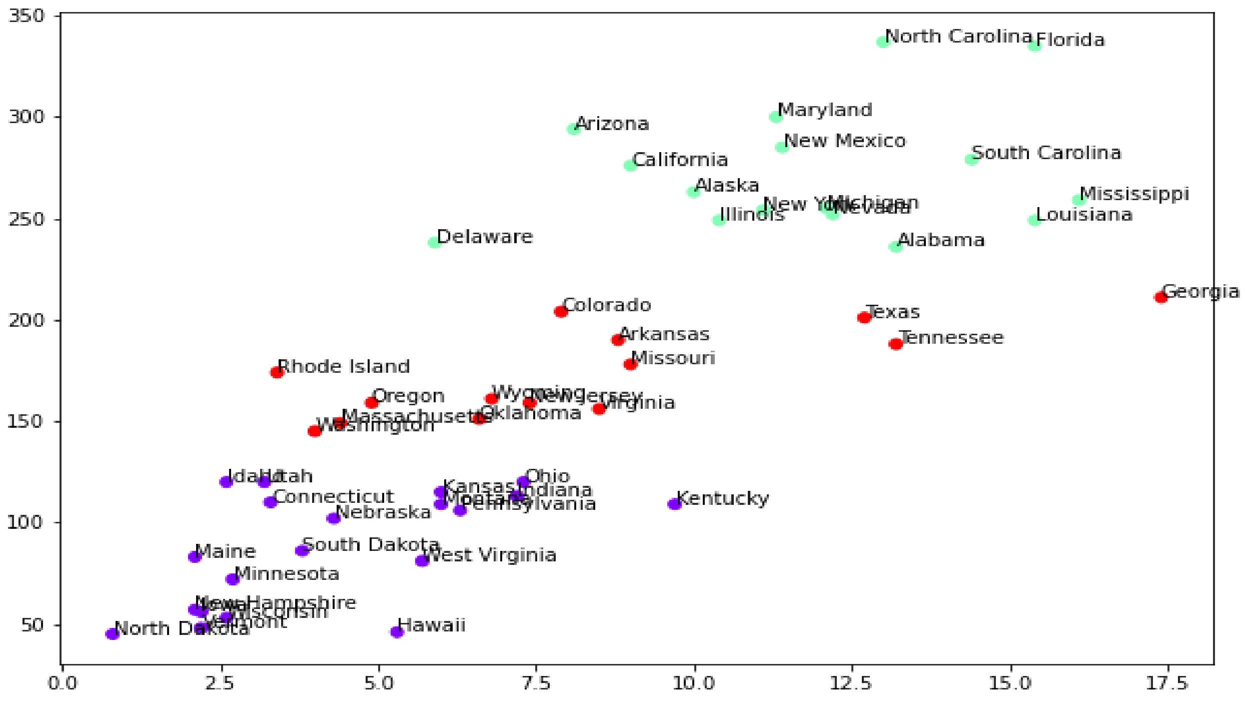
Now we know the number of clusters for our dataset, the next step is to group the data points into these three clusters. To do so we will again use the Agglomerative Clustering class of the sklearn.cluster library. Take a look at the following script:

```
from sklearn.cluster import AgglomerativeClustering

agg=AgglomerativeClustering(n_clusters=3,affinity='euclidean', linkage='ward')
agg.fit_predict(df)
x=agg.labels_
print(x)

output:
[1 1 1 2 1 2 0 1 1 2 0 0 1 0 0 0 0 1 0 1 2 1 0 1 2 0 0 1 0 2 1 1 1 0 0 2 2
 0 2 1 0 2 2 0 0 2 2 0 0 2]
```

You can see the cluster labels from all of your data points. Since we had three clusters, we have five labels in the output i.e. 0 to 2.

Plot the cluster data

```
plt.figure(figsize=(10, 7))
plt.scatter(df['Murder'], df['Assault'], c=agg.labels_ , cmap='rainbow')
for idx, row in enumerate(df.index):
   plt.annotate(row, (df['Murder'][idx],df['Assault'][idx]) )
plt.show()
```



Conclusion

The clustering technique can be very handy when it comes to unlabelled data. Since most of the data in the real-world is unlabelled .Here we can cluster cities of US based on the crime of rate of that city, finally we arrive into three group of cities.

Python code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jan 19 12:41:19 2022

@author: harigran
"""

import pandas as pd
df= pd.read_csv(r"C:\Users\my pc\Desktop\MBA - BA II\lab/usarrest.csv")
# Using set_index() method on 'unnamed' column
df = df.set_index('City_Name')
df.head()

#Creating dendrogram
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, ward
result=ward(df)
dendrogram(result,leaf_rotation=90, leaf_font_size=8, labels=df.index)
plt.title("DENDROGRAM")
plt.xlabel('Observations')
plt.ylabel('Distances')
plt.show()

#Perform Clustering
from sklearn.cluster import AgglomerativeClustering

agg=AgglomerativeClustering(n_clusters=3,affinity='euclidean', linkage='ward')
agg.fit_predict(df)
x=agg.labels_
print(x)

plt.figure(figsize=(10, 7))
plt.scatter(df['Murder'], df['Assault'], c=agg.labels_ , cmap='rainbow')
# annotate points in axis
for idx, row in enumerate(df.index):
    plt.annotate(row, (df['Murder'][idx],df['Assault'][idx]) )
# force matplotlib to draw the graph
plt.show()
```