Assignment 3c:

Cluster Time Series

Anupam Kumar (11940160) Mohit Verma(11940690)

1. Pre-processing: Collection and Cleaning of Data

- The dataset that we are using depicts the temperature change all over the world as an impact of climate change from the beginning of 1743 till 2014. This is a time-series data so, it contains the data for each day for each of the known countries.
- The data cleaning consists of some brief processes as follows:

  i. Removal of repeated data or duplicates.
  ii. Formatting of the year attribute.
  iii. Removal of useless columns from the dataset such as Country Code and Series Code.
  iv. After this, the NaN values are removed.
  v. Now, the dataset is good to go.

2. Clustering using K-Means

- The clustering algorithm that we are using is K-Means.
- The steps of the algorithm are:

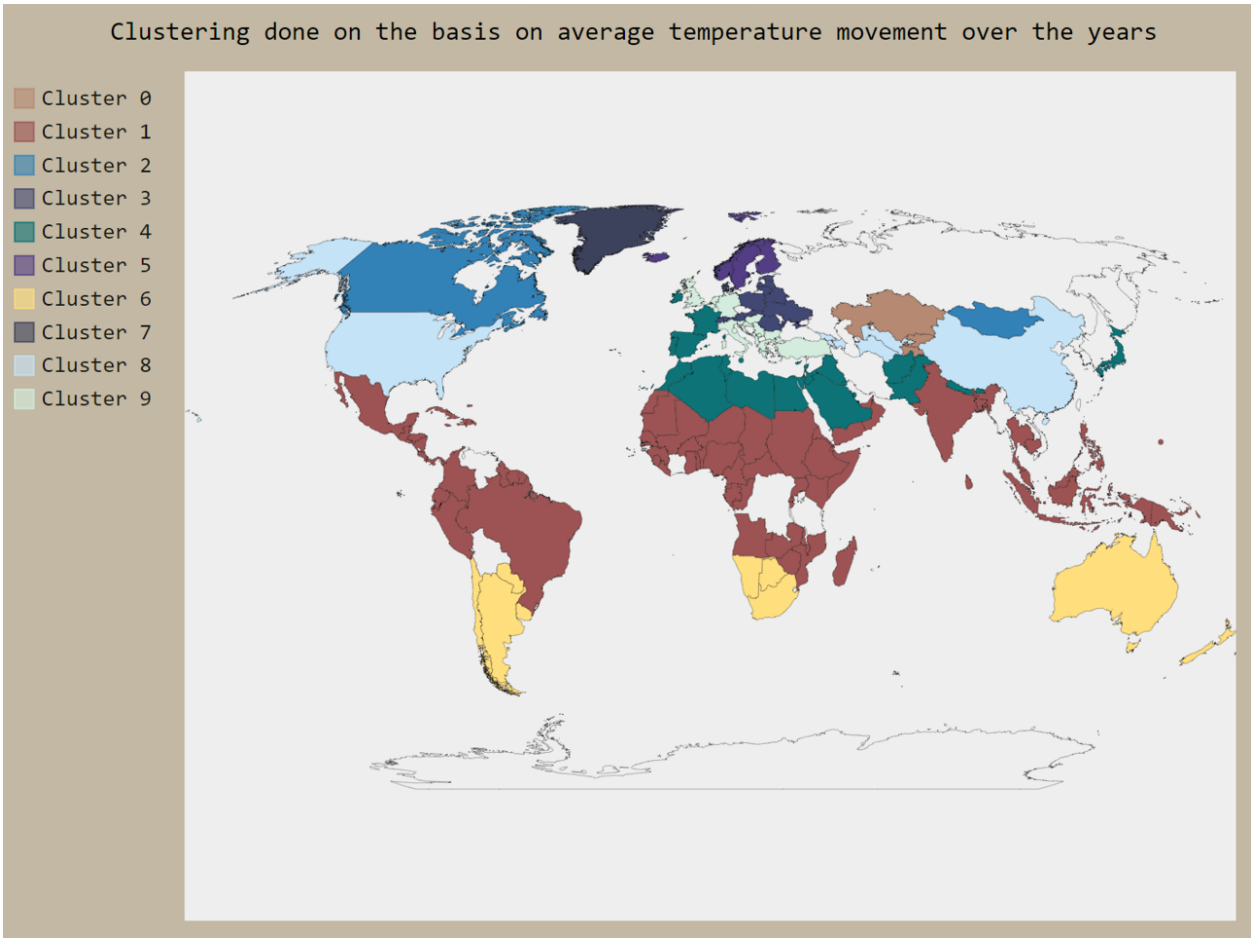
  i. Select the value of K, to decide the number of clusters to be formed.
  ii. Select random K points which will act as centroids.

iii. Assign each data point, based on their distance from the randomly selected points (Centroid), to the nearest/closest centroid which will form the predefined clusters.
iv. place a new centroid of each cluster.
v. Repeat step no. iii, which reassigns each datapoint to the new closest centroid of each cluster.
vi. If any reassignment occurs, then go to step-4 else go to Step 7.
vii. End the algorithm

- We have used the sklearn library for implementing this.
- First, a pipeline is created which normalizes the data, and then, K-Means clustering is applied to it.
- We have chosen the value of K as 10 (i.e., 10 clusters will be formed).
- The model is fit and the labels are predicted.

3. Plotting the result obtained on the dashboard

- The clustering algorithm returns the labels for each country which denotes the cluster they belong to.
- Therefore, to create a visualization of the countries, we have used the pygal library to get the countries and map the labels obtained corresponding to each of the countries, and plot it on the dashboard.
- We have used 10 different colors to represent 10 clusters obtained.
- The world map attained at the end is an SVG file that is interactive. The hover of each country displays the name of the country and the cluster it belongs to.
- We have obtained the SVG file and converted it to a plotly graph and then, used the plotly graph to display on the dashboard.
- The world map obtained after clustering is shown below:



- The clusters are clearly visible in the plot shown above.

4. Hierarchical Clustering

- We also implemented a Hierarchial clustering algorithm on the same dataset.
- The implementation part has been shown completely in the code but, some part of the code is running properly.
- DTW has been performed on the time series dataset.

5. Steps and Instructions to run the code

- The main directory contains two files 3c.py and asg_3c.ipynb, the main file is asg_3c.ipynb. The other file is a python file comprising some part of the code that was made to run the Dash application specifically.
- We have used a virtual environment to initialize and do the project so, that can be done using the following command in cmd:
    - **`virtualenv env`**
- Now that the environment is made, the requirements.txt file can be run to install the required dependencies at once. So, run :
    - **`pip install -r requirements.txt`**
- Now, the main file is ready to run. So, execute the ipynb file in the environment just created and we get an output as shown below:

```
Dash is running on http://127.0.0.1:8051/

 * Serving Flask app '__main__' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
```

- The URL where the dash application is running will be shown after that. So, head over to that link and access the webpage.