7. Multi-Dimension Scaling and Bubble Plot

Prepared by Prajwal Singh

Goal

- Application of MDS (Multi-Dimensional Scaling) for dimension reduction.
- Create a 4D visualisation using a Bubble plot.

Details

- The MDS method uses a distance (dissimilarity) matrix to create a low-dimensional representation of high-dimensional data.
    - This method does not care how the distance (dissimilarity) is computed.
- In this assignment, we will use the distance between different Indian Cities and compute a 2D representation and use it to lay out the cities using a scatterplot.
    - Note: This exercise simply shows that the low dimensional representation is meaningful, not to real any dimension reduction.
- We will start with the Latitude and Longitude of cities and will later use this data for Geo-visualization.
- MDS does not have North-South or East-West direction knowledge. So, while it creates a meaningful layout, we will have to infer N-S and E-W orientations from the final plot.

Data Preparation

- Read the latitude-longitude of a few (150) Indian cities from the following database - https://simplemaps.com/data/in-cities.
    - The CSV data may be accessed directly from that site as "https://simplemaps.com/static/data/country-cities/in/in.csv"
- Compute the distance matrix (distance between every pair of cities) using distance.distance method available from the python package geopy.
    - https://geopy.readthedocs.io/en/stable/#module-geopy.distance
    - Example use:
        - import geopy import distance
        - D = distance.distance(latlon1, latlon2).km

    - D will return geodesic distance in km between two locations on earth, where latlon1 and latlon2 are the (latitude, longitude) pair of the two locations.
- Create a zone column in the data frame for each city based on its admin_name of the city.
    - Use the following mapping to assign a zone based on the admin_name.
        - zones = {
        - "North": ['Delhi', 'Himāchal Pradesh', 'Punjab', 'Uttar Pradesh', 'Haryāna', 'Jammu and Kashmīr', 'Chandīgarh'],
        - "East": ['Bihār', 'Odisha', 'Jhārkhand', 'West Bengal'],
        - "West": ['Rājasthān' , 'Gujarāt', 'Goa', 'Mahārāshtra'],
        - "South": ['Andhra Pradesh', 'Telangāna', 'Karnātaka', 'Kerala', 'Tamil Nādu', 'Puducherry'],
        - "Central": ['Madhya Pradesh', 'Chhattīsgarh'],
        - "North East": ['Assam']
        - }
    - Note: This mapping is specific to the data.
- Use MDS method available from Python package sklearn
    - see scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html
        - from sklearn.manifold import MDS
        - mds_model = MDS(random_state=0, dissimilarity='precomputed', normalized_stress='auto')
        - two_d_representation = mds_model.fit_transform(distanceMatrix)
    - By default, MDS creates a 2D representation for the data, so fit_data will contain an array of 2D coordinates.

Data Visualization

- Created a bubble plot for the city to visualization 4D data associated with each city whose
    - 2D positions, two quantitative dimensions, returned by mds.fit_transform.
    - Population, a quantitative dimension, available in the original dataframe.
    - zone info, a categorical dimension, generated by admin_name to zone mapping.
- Note: Unlike Matplotlib, Plotly has a direct way to create a bubble plot using scatter. See https://plotly.com/python/bubble-charts/

Sample Image