Climate Analysis Graph Project Write-Up

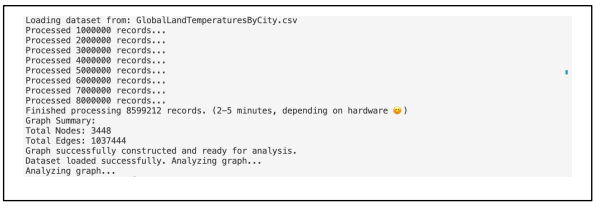
Mohammed Alhashmi

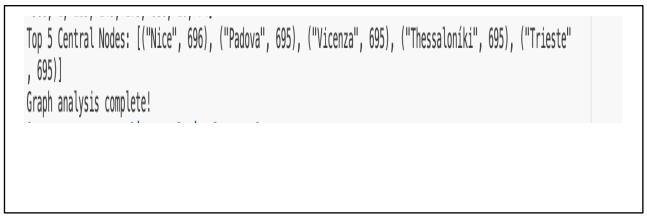
The goal of this project is to analyze a massive global climate dataset and output meaningful relationships between cities using a graph-based approach.

The dataset, provides global temperature data from 1743 to 2013. It includes temperature records for cities, countries, and continents.

Each city was represented as a node, and edges between them captured the similarities in their temperature patterns. By building this graph, I wanted to identify which cities were most connected to others based on climate similarity and to explore broader patterns in global temperature data.

Looking at the output, we can see the degree distribution reveals how connected the cities are, and the list of top central nodes diplays those cities that share strong temperature similarities with many others; this gives us a deeper understanding of the climate network and how different regions relate to one another.





The essential part of the project lies in the 'Graph' struct, implemented in the 'graph.rs'. This is where I defined the graph's core components: the nodes (cities), edges (connections between cities), and the weights (strength of those connections). Each function in this file plays a specific role in analyzing the graph.

The 'load_from_csv()' reads over 8.5 million rows of data using the 'csv::Reader' library, which handles large datasets by processing rows and doesn't load everything into memory at once. For every city, it looks at temperature records and calculates the similarity between them using Pearson's correlation. If two cities have a similarity score higher than 0.8, they are considered connected, and an edge is added to the graph.

A feature of this function is how it handles large dataset. It processes millions of rows while giving feedback every million rows so users know it's still running and ensures users expectation of wait time. This can only be run with 'cargo run –release'.

The 'add_edge()' connects two cities while ensuring no duplicate edges are added, which is important for maintaining the connection of the graph structure. I did this by using a 'HashSet' to store the neighboring cities for each node. A 'HashSet' guarantees that every connection is unique, which improves performance when processing a dataset as big as this one.

The edge weighs representing the Pearson correlation scores between cities are stored in a 'HashMap', so the program can retrieve the weight of any connection, which is useful for further analysis.
Once the graph is built, the 'analyze()' function takes over to extract data.
Two key operations:

- Degree Distribution: This calculates the number of connections (edges) each city has by iterating through the adjacency list. The degree distribution helps show how connected different cities are.
- Top Central Nodes: This identifies the most connected cities by sorting the degree distribution in descending order.

Main.rs

The main.rs loads the dataset, builds the graph, and calls the analysis functions.

In main.rs, the path to the dataset (GlobalLandTemperaturesByCity.csv) is passed to the 'Graph::load_from_csv()' function. This function, defined in 'graph.rs', is responsible for reading the file line by line using the 'csv::Reader' crate. The 'csv::Reader' handles the dataset of over 8.5 million rows, analyzing each line.

The tests in the main.rs file ensure that the graph implementation works correctly when loading data and analyzing the graph. The first test I did, 'test_graph_loading', verifies that the 'load_from_csv()' function successfully loads data from a CSV file (using a test file test_data.csv) and that the graph contains nodes afterward, showing the data was processed correctly. The second test I made, 'test_graph_analysis', checks that the graph analysis methods 'degree_distribution()' and 'centrality()' work as expected.

The output

By processing over 8.5 million rows of data, the program reveals climate-based relationships between cities that might not be geographically close, but, share similar weather trends. This connection is made through a graph where edges represent strong correlations between temperature records, with a threshold of 0.8 to ensure only meaningful links are considered.

Cities like Nice, Padova,Vicenza, Trieste, and Thessaloniki frequently appear among the top 5 central nodes, acting as climate hubs. These hubs are cities that share temperature correlations with many others. This insight shows cities that could be considered "climate mirrors" for other regions.

What's interesting about these hubs is their potential as climate benchmarks. If these cities experience shifts in temperature trends, it might signal broader changes in the regions they represent.

The slight changes in the top central cities each time we run the program make the results even more interesting. It shows the program's ability to identify cities that are statistically tied for centrality. While cities like Nice and Padova consistently rank near the top, others like Stara Zagora in Bulgaria make some appearances. This suggests that multiple cities can play equally central roles in the climate network. These subtle shifts highlight the depth of the data there isn't just one city dominating the climate patterns, but there is also a network of influence spread across various regions.
No single city dominates the network, and the changes reveal the diversity of shared patterns.

Insights

- Climate Similarity Across Borders: Cities like Thessaloniki in Greece and Trieste in Italy are linked through similar weather patterns, showing that climate isn't just a local or regional phenomenon. Instead, it reveals a global network of shared climate traits that goes beyond geographical boundaries

- Predicting Climate Trends: Since these central cities have connections to so many others, any significant temperature differences in them could hint at changes in connected regions. For instance, if Nice experiences consistent warming, cities with similar climates could soon experience the same trend.

- Sustainability and Urban Planning: For policymakers, this information could be valuable. Cities that share climate similarities might collaborate on solutions to climate-related challenges, such as managing rising temperatures, and avoiding drought.

- Cross-Regional Weather Modeling: From a scientific perspective, these findings can improve climate models. By focusing on climate hubs, researchers can create more accurate regional predictions using fewer data points while capturing a large part of the overall climate network's behavior.

In summary, the program turns millions of rows of raw temperature data into a meaningful graph of global climate relationships. It identifies key climate hubs like Nice, Padova, and Trieste, which are central to understanding how weather patterns connect cities worldwide. These insights have real-world applications in climate monitoring.
The program provides a new way to understand global climate behavior, showing the connectivity of cities and offering valuable insights for science, policy, planning and awareness.

Reference:

Earth, Berkeley. "Climate Change: Earth Surface Temperature Data." *Kaggle*, 1 May 2017, www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data.