Data Project 2

Nishana Dahala and Bhavya Guduru

I. Data Set Selection

Inspired by the recent weather changes, we wanted to design a website that would return outfit recommendations for the day based on the current weather at a location. We easily found a live API that gave us the average temperature based on the city. We then needed a CSV file that would provide some data about outfit recommendations based on temperature or season. However, we could not find a relevant CSV for this. Then, we realized it is almost summer and people often go on vacation. Sometimes people have trouble picking places to go, and they often pick places with the opposite weather. So we decided and found a CSV with major cities and their average temperature.

II. Key Challenges

After we chose our data sets, we continued coding our project locally before pushing it to GitHub. Unfortunately, the CSV we had chosen was well over 25 MB. We attempted to update it from the terminal using the GitHub LFM library, but still faced issues with it. We decided to go back and find a new CSV file with the data we wanted. We then had to create a new extraction method as the file now had data from all 12 months of the year for a specific location, so we had to average it and use it in our algorithm.

Once we resolved this, we started deploying our website. However, we had an error where it would return “Sorry, we could not find the temperature.” We assumed it had to do with the new CSV file and the new method we created, so we continued to debug it and adjust our approach. We reviewed it from the beginning, and then we realized it was that we forgot to include our API key.

Another issue we had was an issue with the link between GitHub and PyCharm. We are used to pushing our files locally. Unfortunately, we could not resolve this and instead went in and copied and pasted our code directly into the new file option in GitHub.

III. Key Learnings and Discoveries

Our main takeaway from this project was how important data transformation is. Here we use Fahrenheit regularly, but in other parts of the world they use Celsius. When we were first coming up with our algorithm for choosing places, we had an issue with setting a threshold and returning locations that matched the opposite climate we were expecting. We went back to the API documentation and CSV files and realized they were in opposite units. This showed us how important it is to ensure data is consistent and properly transformed before applying any algorithm, which doesn't just apply to temperature but to any real-world dataset.

IV. Enhancements

If we had more time, there were a few enhancements we had in mind for our project. One option would be to add a prompt where we can ask for the user's preference of whether they want the same climate or opposite climate, and return a location based on this. We decided to go with the opposite climate in our code because there is a trend of people going somewhere warm if they are from a place that is typically cold. Another option is to ask the user if they want to stay in the same country as the location they asked for the average. This would require adding a parameter in the CSV describing what country the cities are located in. A similar enhancement is to ask whether the user would like to stay on the same continent. This would also require adding a parameter to the CSV, but include what continent the city is in.

All in all, this project taught us to be very meticulous in utilizing data, especially when transforming it. We saw how small details, such as measurement units or a simple error, can impact the overall page. This project helped us in both technical and problem-solving skills that we will carry on into the future.