Team 8: Christina Barton, Brooke Beenhouwer, Corinna Keum, Maria Vitória Klein

Data Project 2: Documentation/Reflection

http://127.0.0.1:5000

Because our project focuses on creating a bot that would be prompted to give the current weather of a city that was asked for, we needed to find a dataset that would have real-time weather. Thus, we chose to use https://weatherstack.com/ to get a live API call because it uses real-time and historical world weather data. From weatherstack, we were able to get data on the temperature, condition, feels like, humidity, and moon phase on the location the user would like to know about. Additionally, we combined this with a weather_data.csv, which has the temperature, humidity, precipitation, and wind speed 982808 rows of data with historical weather data on cities specifically in the U.S. Because the CSV only uses data in the United States, it was important to use the live API, so our bot could be able to answer about any city around the world.

There were many challenges throughout creating and implementing our chat bot. One difficult task was making the GCP instance and making it accessible to others. Certain team members' computers had difficulty running the app, so we had to attempt this process on multiple computers before getting it to work. Additionally, it was challenging to decide what words would signal which data set to use and to incorporate the two datasets into our code. We had to create two different functions: one to pull data from the csv and convert it and one to pull data from the API. One final challenge we had was getting our chat bot to accept two words as a city name. There are many cities in the csv file we chose, such as San Jose, that are made up of two words. At first, our code was only accepting the last word in a query as a city, which was difficult to

change. However, in our function that found the city name in the user input, we went through a multistep process using regex to find the city name based off of capitalization in the user's input.

Throughout trials and challenges, we discovered many insights about how to utilize multiple data sources to implement a chat bot. Learning and understanding how to create a Flask app past the simple websites we have created in class was very valuable. Additionally, throughout this process and previous project processes, we have gained a stronger grasp on the importance of clean error messaging and using try blocks. This helps with debugging and making code clear and explainable. We also discovered that combining uses of live and static data is difficult, but utilizing functions rather than raw code makes the process faster and simpler. Understanding how to use data from multiple sources at the same time is very valuable for data cleaning and creating future data projects, as they will not likely be as simple as one dataset.

If we had more time, we would have wanted to add a variety of enhancements for our bot. For instance, we would have wanted to integrate our bot to discord. Embedding the weather bot directly into a discord server would promote accessibility to user interaction. Additionally, it would be beneficial to expand out location support, this means we would want to increase the number of supported cities and incorporate more precise geolocation data to have more precise to improve the accuracy for smaller towns, so it wouldn’t only be specifically major cities. This would require us to have a larger data set and more live API data. We would have also wanted to have more forecasting capabilities to the bot, this would mean the bot would be able to add short-term weather forecasts, not just current or historical weather conditions. We would have also wanted to improve the bot’s ability to be customizable and improve the language. For instance, adding user ability to choose between Celsius and Fahrenheit, languages, or to receive weather alerts from the bot.