Reflection on Building the ETL Data Processor

From developing our ETL data processor, we learned how data has a vast range of applicability from its different forms. This pipeline demonstrates a simple, yet effective streamlined process of how data might be converted and saved for different purposes. One of the primary challenges we faced was developing the convert_data function. Simultaneously handling both the CSV file and the API responses in a JSON format definitely was tricky as the preprocessed data was not formatted beforehand. While working with our data, I noticed that the conversion from CSV to JSON was a very simple and streamlined process since there were no null or missing values to correct within the file.

Additionally, working with APIs presented its own set of difficulties. We took a solid amount of time working with parsing JSON responses. This process required understanding the data structure as certain components such as the region were nested within the location while other data like temperature was located inside of current. However once we were able to print out the json response it was able to easily interpret and determine which variables were of importance.

On the other hand, the fetch processes for both the CSVs and APIs was a very simple process as it only really required the api,base_url, and the read_csv functions in order to be able to start the data processing. The error messages for these fetch functions were very easy to implement since they notified the user of such failures immediately before the rest of the pipeline was run.

Another aspect that we found easy was generating summaries at the preprocessing step and postprocessing step throughout the pipeline. We leveraged pandas' DataFrame properties for quick retrieval of the number of records and columns, which was essential for validating each step of the ETL process.

This utility has a lot of applications for future data projects in the sense that it allows for the ability to work with different formats and pull from any source available. For example, if you were given a CSV file you had the ability to streamline the process to interact directly with your database where we used SQL. Through the ingestion and processing of data, we were able to understand where data comes from and how it can be transformed for different analysis, making this pipeline very adaptable to different changes.

In conclusion, the ETL processor we built proved to be a valuable exercise that highlighted the importance of data handling and the practical challenges of working with diverse data sources. This project demonstrated the capabilities of how data can transform in a similar fashion to how technology transforms in the real world.