Samantha Ritchie and Lauren Medica

# Project 1 Reflection

For this project, we implemented an ETL pipeline in Python that successfully retrieves and processes raw data from remote and local sources. The pipeline also allows conversion between different formats like CSV, JSON, and SQL. The code is designed to allow users to choose their output format, whether it be CSV, JSON, or a SQLite database, and also provides flexibility to modify data columns. We used standard logic to drop unwanted columns and add new ones with preset data. This ensures the pipeline is adaptable for different data processing needs.

The original retrieval of the data was difficult as no previous examples involving Kaggle were done in lab or for homework. The original code was based off the homework involving the stock market however this failed to work and colab suggested the use of the os library which proved successful.

Once the data was retrieved, the process of converting between formats like CSV, JSON, and SQLite was straightforward because of the Pandas and SQLite libraries. However we faced a challenge when we added functions for the user to dynamically modify columns. We used logic to let users drop unnecessary columns and create new ones with user desired values, such as discovery year or chemical formulas in this case. This flexibility required us to handle edge cases, particularly ensuring that the user's input was correctly formatted for adding or dropping columns. Producing informative error messages in case of improper inputs required additional logic code to ensure the users input was valid.

In the final part of the project, we implemented a function to store the modified data in the user's preferred format and generate summaries of both the pre- and post-processing files. These summaries include the number of records and columns, allowing the user to quickly reference the data structure. The error-handling portion was designed to notify the user if operations like file writing or data transformation failed, making the pipeline usable for many different cases.

This utility is useful for any data science project requiring flexible data ingestion and transformation. It simplifies the often time-consuming task of data preprocessing, offering a customizable and automated way to handle diverse data sources and outputs. With its ability to modify data on-the-fly and support multiple file formats, the pipeline ensures users can seamlessly integrate and prepare datasets for analysis, regardless of the data's origin or intended usage.