IRisk Lab Data Discovery and Consolidation - Task 2
Week 4 Report
Sep. 19
Rohit Valmeekam

<u>What was accomplished</u>

Expanded upon data pipeline and tested it on discovered data, including the following improvements:

- ❖ Created Support for Other File Extensions (e.g., .xlsx and .txt):
  - ➢ Implemented functionality to handle various file extensions such as .xlsx (Microsoft Excel) and .txt (plain text) in addition to the existing file formats.
  - ➢ Utilized appropriate libraries or modules (e.g., openpyxl for .xlsx and standard file I/O for .txt) to read and process these file types.
  - ➢ Ensured compatibility and data integrity when converting these file formats into the desired .csv format.
- ❖ Consolidated All Data Types into a Singular .csv Format:
  - ➢ Developed a data transformation process to consolidate data from different file types (e.g., .xlsx, .txt, and any others) into a common .csv (Comma-Separated Values) format.
  - ➢ Mapped or transformed data fields from various formats into the corresponding columns in the .csv file.
  - ➢ Handled cases where data structures or headers in different file types varied and ensured they were consistent in the resulting .csv file.
- ❖ Added Error Checking for File Types to Check for Invalid File Types:
  - ➢ Implemented robust error handling mechanisms to validate the file types before processing.
  - ➢ Checked file extensions, MIME types, or other metadata to verify the format's correctness.
  - ➢ Generated informative error messages or logs to alert users or administrators when an invalid file type is encountered.
- ❖ File Type Conversion Logging:
  - ➢ Maintained a log of all file type conversions, including successful conversions and those that encountered errors.
  - ➢ Recorded essential details such as the source file, target format, timestamp, and outcome (success/failure) for auditing and debugging purposes.
- ❖ Data Validation:

- Implemented data validation routines to identify and handle inconsistent or erroneous data within the input files.

Challenges

- Consolidating the data
- Structuring the data pipeline
- Deciding what processes the processing algorithm should cover
- Connecting the data pipeline to the database

Next Steps

- **Communicate with Team 1 about how to integrate their PDF conversion with the data pipeline**
- **Replace Dummy SQL Database:**
  - Replace the dummy SQLite database with the final database that you intend to use.
  - Ensure that the database schema is aligned with the structure of the output data.
  - Establish a connection to the final database within the script.
  - Modify the script to insert or update data in the final database as needed, based on the processed data.
- **Database Schema Design:**
  - Design the schema of the final database to accommodate the data you are processing.
  - Define tables, columns, and relationships to organize and store the data effectively.
  - Consider data types, constraints, and indexing for optimal database performance.

Example of the file structure after processing