AI work journal

Lee Pettigrew

X20730039

I did most of my work / testing locally and only really pushed working code to the github.

10/12/24

Team got started on project, Notion (Trello clone) page was set up from the outset, Cem and I took primary responsibility for the coding efforts. However, I quickly noticed that the code structure felt overly complex and difficult to navigate. To avoid unnecessary friction or a potential clash of leadership styles, I decided to let Cem take the lead on the project.

By aligning myself with the tasks Cem has outlined, I believe we can streamline the workflow and achieve a cohesive, polished final product more efficiently. This approach not only fosters a collaborative environment but also ensures that we stay focused on our shared goal of completing the project successfully.

19/12/24

I encountered a bug in the codebase. Cem was attempting to read data files and connect to MongoDB, but the file path for the data files was not set relative to the scripts, causing the implementation to fail. Additionally, there were issues with the database connection. Observing these challenges, I decided to restructure the codebase, which at that point consisted mostly of boilerplate and some exploratory work.

I implemented two new Python files to consolidate the smaller files Cem had previously introduced. As I am not particularly fond of using Jupyter notebooks (ipynb), I opted for raw Python files, which I find more efficient for my workflow. To enhance interactivity, I integrated a Streamlit dashboard, enabling interactive data manipulation—a feature I personally prefer.

The updated codebase now includes functionality for data exploration, cleaning, merging, and sending data to MongoDB. It also supports pulling data from MongoDB, performing analyses, and generating visualizations. I prefaced this update with a note indicating that some fine-tuning is still required, as certain plots and interactive elements are currently displaying unusual or unexpected numbers.

By aligning myself with the tasks Cem has outlined, and taking proactive steps to address technical challenges, I believe we can streamline the workflow and achieve a cohesive, polished final product more efficiently. This approach not only fosters a collaborative environment but also ensures that we stay focused on our shared goal of completing the project successfully.

I probably spent around 4-5 hours implementing this solution. (see revision 630e4a2bfcebdd041a10f2beb125708160df0b34)

20/12/24

After discussing with Cem, I learned that he wasn't particularly fond of the initial setup, where everything was contained within a single file. He expressed a preference for working with separate Jupyter Notebook ("ipynb") files for better modularity and readability. Taking his feedback into account, I restructured the project entirely, splitting it into multiple ipynb files, each focusing on a distinct aspect of the workflow. Additionally, I removed the dashboard and interactive elements to simplify the overall approach.

The transition to this modular structure went smoothly, with no major errors encountered during implementation. However, a few minor issues cropped up, such as unusual numbers appearing within some plots. These anomalies are likely tied to certain nuances in the data that need further investigation. I also identified areas in the data cleaning process where some irrelevant or low-quality data should potentially be dropped to enhance the overall reliability of the analysis.

To further streamline the process, I implemented detailed logging for every step of data cleaning and the subsequent process of sending data to MongoDB. This logging system is designed with user-friendly output, making it effortless to follow each step. The logs not only document the progress but also provide clarity, ensuring transparency and ease of debugging should any issues arise.

Overall, these enhancements have significantly improved the workflow's clarity, modularity, and usability, setting the stage for more effective collaboration and analysis moving forward.

More detailed walkthrough of what I created from scratch.

Step 1: Explore

I initiated the foundational phase of data analysis by systematically processing raw CSV datasets into Pandas DataFrames. The datasets included:

- covid-vaccinations-vs-covid-death-rate
- covid-vaccine-doses-by-manufacturer
- OECD_health_expenditure
- united-states-rates-of-covid-19-deaths-by-vaccination-status

Methodological Exploration

Each dataset was meticulously examined using a structured approach:

- Preliminary data inspection through the .head() function to assess format and content.
- Detailed metadata analysis via .info() to evaluate data types, null entries, and potential structural inconsistencies.
- Statistical profiling using .describe() to derive quantitative summaries, including distributions, central tendencies, and variance across variables.

Dataset Insights

COVID Vaccinations vs COVID Death Rate

- Conducted an initial review of the interplay between vaccination coverage and COVID-19 mortality rates.

COVID Vaccine Doses by Manufacturer

- Examined the composition and global dissemination of vaccine doses by manufacturing entities.

OECD Health Expenditure

- Analyzed temporal and regional variations in health expenditure across OECD countries.

US Death Rates by Vaccination Status

- Investigated mortality trends stratified by vaccination status within the United States.

Analytical Rigor

- **Optimized Data Accessibility**: Implemented robust file path corrections to streamline dataset ingestion.
- **Systematic Profiling**: Deployed uniform exploratory techniques to ascertain data readiness and infer potential areas for deeper investigation.

Step 2: Clean

I executed a comprehensive data cleaning process to prepare four datasets for analysis. The datasets included:

- covid-vaccinations-vs-covid-death-rate
- covid-vaccine-doses-by-manufacturer
- OECD_health_expenditure
- united-states-rates-of-covid-19-deaths-by-vaccination-status

Cleaning Methodology

I implemented a structured approach to ensure the data was clean, consistent, and analysis-ready. The steps included:

- **File Validation**: Verified that each dataset existed before attempting to load it, logging missing files as errors.
- **Standardization**: Renamed columns to snake_case for uniformity and removed problematic characters.
- **Date Parsing**: Converted date columns into proper datetime objects to enable chronological sorting and time-based analysis.

- **Handling Missing Values**: Filled missing numeric values with 0 or appropriate defaults and removed rows with critical null values.
- **Duplicate Removal**: Dropped duplicate rows to eliminate redundant information.
- **Outlier Handling**: Defined methods to identify and remove extreme outliers using the IQR method (optional, not yet applied).
- **Optimization**: Downcast numeric columns to reduce memory usage and categorized appropriate text columns for better performance.
- **Sorting**: Organized time series data chronologically for meaningful analysis.

Dataset-Specific Actions

COVID Vaccinations vs COVID Death Rate

- Standardized columns, parsed dates, and handled missing and duplicate data.
- Optimized numeric types and identified categorical columns for efficient processing.

COVID Vaccine Doses by Manufacturer

- Followed the same cleaning steps as above, ensuring uniformity across records.

OECD Health Expenditure

- Converted numeric columns like obs_value and time_period to appropriate types.
- Addressed duplicate column names and missing values effectively.

US Death Rates by Vaccination Status

- Processed columns like entity (representing age groups) to ensure completeness.
- Removed unnecessary columns (e.g., entirely null columns) to streamline the dataset.

Logging and Outputs

All cleaning steps were logged in a detailed cleaning.log file for traceability and debugging. Cleaned datasets were saved in the ../data/processed/ directory for subsequent analysis.

Highlights of My Approach

- **Robust Logging**: Documented every step, from initial loading to final outputs, ensuring transparency.
- **Efficiency**: Optimized memory usage while maintaining data integrity.
- **Scalability**: Designed modular cleaning functions to apply across diverse datasets seamlessly.

Step 3: Merge to MongoDB

I completed the process of storing cleaned datasets into MongoDB for analysis. The datasets included:

- covid-vaccinations-vs-covid-death-rate
- covid-vaccine-doses-by-manufacturer
- OECD_health_expenditure
- united-states-rates-of-covid-19-deaths-by-vaccination-status

Storage Methodology

I employed a systematic approach to ensure seamless data insertion into MongoDB. The steps included:

- **File Validation**: Verified the existence of each cleaned dataset before loading. Skipped non-existent files with appropriate logs.
- **Data Preparation**: Converted each CSV file into a Pandas DataFrame and then to a list of dictionaries for MongoDB insertion.
- **Collection Management**:
    - Cleared existing data from each MongoDB collection to avoid redundancy or stale records.
    - Ensured collections were consistently named and properly organized.
- **Batch Processing**: Implemented batch insertion to handle large datasets efficiently, avoiding timeouts and enhancing performance. Batches of 2000 records were used.
- **Error Handling**: Captured and logged any issues during insertion, ensuring traceability and quick resolution.

Dataset-Specific Actions

COVID Vaccinations vs COVID Death Rate

- Inserted all cleaned records into the corresponding MongoDB collection after clearing old data.

COVID Vaccine Doses by Manufacturer

- Processed and uploaded in a similar manner, ensuring uniformity in handling.

OECD Health Expenditure

- Addressed large numeric data volumes by leveraging optimized batch processing.

US Death Rates by Vaccination Status

- Ensured complete data insertion while maintaining the integrity of critical columns like entity.

Highlights of My Approach

- **Efficiency**: Optimized storage operations through batch processing, significantly reducing the risk of timeouts.
- **Data Integrity**: Cleared old collections before insertion to ensure clean and accurate datasets.
- **Error Resilience**: Designed robust error handling and logging mechanisms for transparency and debugging.

Step 4: Analysis

I conducted an in-depth analysis of the cleaned datasets stored in MongoDB to extract key insights and identify trends. The datasets analyzed included:

- covid-vaccinations-vs-covid-death-rate
- covid-vaccine-doses-by-manufacturer
- OECD_health_expenditure
- united-states-rates-of-covid-19-deaths-by-vaccination-status

Analysis Workflow

Data Loading

- Connected to MongoDB and loaded the datasets into Pandas DataFrames for analysis.
- Ensured a clean import by excluding MongoDB's _id fields from the DataFrames.

Initial Exploration

- Reviewed the structure of each dataset using .info() and .describe() to understand distributions, data types, and potential issues.
- Checked for null values and identified columns requiring further cleaning or transformation, such as date or numeric formats.

Dataset-Specific Analyses

COVID Vaccinations vs COVID Death Rate

- Calculated average vaccination rates per hundred people by country, identifying top performers.

- Examined correlations between vaccination rates and COVID-19 death rates (7-day rolling averages), uncovering key relationships.

COVID Vaccine Doses by Manufacturer

- Aggregated cumulative doses by manufacturer across all entities to determine global distribution leaders.

OECD Health Expenditure

- Computed average health expenditures by country and ranked top reference areas for healthcare spending.

US Death Rates by Vaccination Status

- Analyzed weekly death rates for the 80+ age group, highlighting temporal trends in unvaccinated populations.

Correlation and Integration

- Generated correlation matrices for numerical columns to explore relationships within each dataset.
- Investigated opportunities to merge datasets based on shared fields like country or time periods to enhance insights.

Key Outcomes

- **Insightful Trends**: Extracted meaningful patterns in vaccination rates, healthcare spending, and mortality.
- **Correlation Highlights**: Identified significant relationships within and across datasets, paving the way for deeper analysis.
- **Integration Opportunities**: Explored preliminary steps for merging datasets, aligning them for unified insights.

Step 5: Visualization

I developed a series of visualizations to illustrate key insights derived from the cleaned datasets stored in MongoDB. The datasets included:

- covid-vaccinations-vs-covid-death-rate
- covid-vaccine-doses-by-manufacturer
- OECD_health_expenditure
- united-states-rates-of-covid-19-deaths-by-vaccination-status

Visualization Workflow

Setup and Loading Data

- Connected to MongoDB and loaded datasets into Pandas DataFrames for visualization.
- Converted date columns to datetime for accurate time-series plots.

Visualizations Created

1. **Vaccination Progress Over Time**
    - Line plot showing cumulative doses per hundred over time for a specific country (e.g., Australia).
    - Highlights vaccination trends and milestones.
2. **Distribution of Daily Death Rates**
    - Histogram with a KDE overlay to visualize the distribution of daily COVID-19 death rates.
    - Provides insights into the frequency and range of mortality rates.
3. **Vaccine Manufacturer Totals**
    - Bar chart ranking vaccine manufacturers by total doses distributed globally.
    - Reveals leading contributors to global vaccination efforts.
4. **Health Expenditure Comparison**
    - Horizontal bar chart showing top 10 countries by average health expenditure.
    - Offers comparative insights into healthcare investments.
5. **US Death Rates by Age Group**
    - Multi-line time-series plot visualizing weekly unvaccinated death rates by age group.
    - Highlights temporal trends and age-related risks.
6. **Correlation Heatmap for Vaccine Manufacturers**
    - Heatmap illustrating correlations between vaccine manufacturers' dose distributions.
    - Uncovers relationships and patterns in production data.
7. **Facet Grid of US Death Rates**
    - Facet grid histogram showing the distribution of unvaccinated death rates by age group.
    - Provides detailed subgroup analysis.
8. **Ridgeline Plot of Unvaccinated Death Rates**
    - Ridgeline plot illustrating unvaccinated death rates across age groups.
    - Creates a visually engaging summary of data distributions.

Highlights of My Approach

- **Comprehensive Coverage**: Addressed diverse aspects of the datasets using multiple chart types.
- **Advanced Visuals**: Integrated advanced techniques like ridgeline plots and facet grids for richer insights.
- **Clarity and Style**: Ensured visualizations were clear, informative, and aesthetically consistent.

This took around 8 or so hours to rework into ipynb and enhance.

31/12/24

On the same day that Cem had restructured some elements, I explored the possibility of creating a new branch to experiment with alternative approaches. However, these changes did not yield the desired results, and ultimately, we decided to retain the structure of the main branch.

During this process, I discovered that certain paths in the code were hardcoded to Cem's desktop, which limited its usability across different systems. To resolve this issue, I updated the paths to be relative, ensuring the code could run seamlessly for everyone involved.

This extra branch took around 6 hours to test and play with. Mostly wasted but some insight gained.

05/01/24

All team members assembled to collaboratively finalize the project report, journals, and complete the video presentation. As part of this effort, I spearheaded the creation of a comprehensive PowerPoint presentation, meticulously crafting detailed speaker notes to guide the group during the presentation process. Additionally, I took charge of producing the video presentation—a crucial deliverable for the project. This involved not only recording the video but also editing it to a professional standard using Adobe Premiere Pro, ensuring a polished and engaging final product that effectively conveyed our work.

This collaborative session marked the culmination of our efforts, with every team member contributing to refining and perfecting the project outputs. My role in coordinating the presentation elements ensured cohesion across the various components, while the team's collective input helped address any last-minute improvements.

7 hours spent finalizing everything.

Non-date recorded

Throughout the project's timeline, numerous minor issues and errors emerged. These were often identified during team discussions and brainstorming sessions, where solutions were collaboratively devised. I adopted a methodical approach to implementing fixes, preferring to address challenges in larger, consolidated updates rather than through piecemeal corrections. This approach, which I find both efficient and effective, allowed for more substantial improvements to the overall structure and functionality of the project. By tackling issues in larger blocks, we minimized potential disruptions and maintained a consistent workflow.

The combination of collaborative problem-solving and my structured approach to implementing fixes ensured the project stayed on track while maintaining a high standard of quality. Each step of the process reflected a strong commitment to teamwork, precision, and the shared goal of delivering an excellent final product.

Probably around 15 hours spent dealing with such things.