Meteorite Data Impact Application

A Full-stack Data Application Project

Project Team Members

Adriana Khul, Thomas Gartley, Valentina Sanhueza, Victor Rincon

Table of Contents

1. Introduction

The purpose of this project was to create a full-stack interactive data application that visualizes and analyzes meteorite impact data from around the world and within a range of years (between 0yrs and 2013) provided by a user. The dataset we used provided valuable insights into meteorite landings, including variables such as the location, mass, fall date, and classification of meteorites. By giving the user, a limited set of input variables (a start and end date), the user was presented with extensive and valuable data for analysis. We provided multiple visualizations for 'at a glance' data comparison, including maps with embedded data. These features assist in extracting meaningful patterns and trends from complex datasets. Through the use of interactive features like filters, maps, and charts, users can manipulate the data according to their interests and needs. Interactive dashboards enhance user experience by allowing them to explore data on their own terms, fostering deeper insights and more informed decision-making.

2. Project File Structure

The project/App directory structure is as follows: [Project 3 Meteorites GitHub]

1. Database/
   1.1. Create_Database.ipynb
   1.2. cleaned_meteorite_data.csv
   1.3. meteorites.sqlite
   1.4. test_queries.ipynb
   1.5. test_visualizations.ipynb
2. app/
   2.1. static/
      2.1.1. css/
         2.1.1.1. Styles.css

        2.1.2. Js/
            2.1.2.1. App.js
            2.1.2.2. Leaflet_heat.js
            2.1.2.3. Map.js
    2.2. templates/
        2.2.1. about_us.html
        2.2.2. Dashboard.html
        2.2.3.Index.html
        2.2.4.Map.html
        2.2.5.Works_cited.html
    2.3. app.py
    2.4. meteorites.sqlite
    2.5. sql_helper.py

3. Dataset

A. Dataset information and description

Our dataset was acquired from Kaggle.com, where it had already gone through several cleaning iterations. The original dataset is from the Meteoritical Society website. Their dataset was cleaned by NASA and posted as a dataset in Kaggles.com (NASA). That dataset was further processed by Jay Panderson and posted again in Kaggles.com

B. Data Processing

The dataset we used was from Jay Panderson/Kaggle. We further prepared it by preforming the following operation:

- The "year" column was converted from float to integer.
- Rows with missing (NaN) values were removed.
- Unnecessary or repetitive columns (e.g., name_type, location_type, fall, unnamed, geo_location) were dropped.

The resulting dataset used in this project includes the following columns:

- Id: The number id given to the meteorite.
- rec_class: The class type of the meteorite.
- mass: The size of the meteorite in (g).
- year: The year the meteorite landed on Earth.
- rec_lat: The latitude coordinates of the landing site.
- rec_long: The longitude coordinates of the landing site.

C. Bias and Data Limitations

Size: It is difficult to make visualize the size of the datasets due to the amount of data. i.e. the bar graph is capable of listing the top fifteen meteorite classifications but the miscellaneous category is not accurately visualized by the dashboard.

Time frame: The Kaggle dataset “meteorite strike data“ was from a year ago. It does not include meteorite recordings for the past year. Similarly, our map filter year is set to 2013. It does not account for the last twelve years of data. The dataset only covers up to 2013, the past 12 years of data is not accounted for in the dataset, this can cause errors in long- term trend analysis.

Data Cleaning Bias: In the process of data cleaning, removing columns such as fall, name_type, unnamed and removing rows with NaN values may have excluded valid data. This impacts the overall results of the dataset.

Human-Reported biases: Human reported biases occur due to population size of where meteorites occur. Specifically, areas with higher population, there are more people to come across a meteorite and have a higher reporting area. Vice versa a less populated area where meteorites have landed will be underreported.

Geographic Biases: Geographic biases occur due to the elevation and vegetation differences in meteorite landing areas. For example, if a meteorite lands in an open area, such as a desert, it is more visible than a meteorite landing in the rainforest.

4. Research Questions

We had 3 main questions we hoped to answer:

A. Impact Patterns

Is there a geographic pattern to location meteorites will impact? In understanding where the patterns of fall locations are, we can use the map dashboard to help understand the dataset. This can be shown by both the grouped/individual meteorite markers and heat map concentrations.

B. Strike Frequency

How many meteorites impact occur within specific time frame. Using the map dashboard filter a user can input a start and end year date. This filter will then change the map to include only meteorite landing to occur within that time frame. The user can then sum the meteorites that appear on the markers. For example, from 1950 to 2013 there were 15,251 recorded.

C. Meteor Classification

Which meteor class encounters Earth most frequently? In finding out the most common occurring classification of meteorites, we can use our interactive dashboard. When setting the start year to 1950 and the end year to 2013 in the dashboard filter; the bar graph will show that the L6 meteorite was the highest occurring meteorite class in that timeframe. For further insight into how this classification varies by year, the dashboard features a sunburst chart. Users can click on individual years, causing the chart to expand into a donut chart that offers an additional, detailed visual breakdown of meteorite classifications.

5. 'Meteorite Impact' Application

A. Overview

'Earth Impact Data App' webapp begins by loading the Home page (index.html) with the website description, purpose of site and links to other website sections (Dashboard, Maps, Works cited, About us). The interactive dashboard allows users to identify the number of meteorite landings in any given year and see the corresponding meteorite classifications. The visualizations of the sunburst chart help expand on the percentage of meteorite classification in a given year within the initial search range and the bar graph indicates the top fifteen classes of meteorites that had fallen in the given time frame. The map dashboard has a filter by start end year to search the dataset. On the map dashboard users can toggle between a heat map and markers to indicate where meteorites have landed throughout the world. With the dashboard we can gain insight on our research questions

The main app file (app.py), contains 9 app routes used for the queries to generate all components of the dashboard and map. Each calls a function from the sql_helper.py file and utilizes the SQL Alchemy library to call data from our SQLite database (meteorites.sqlite). All our queries accept two variables (min and max year) from user inputs. The first 5 routes render the template HTML files and remaining 4 routes manage the API queries, 3 for each of the visualizations on the page and one for the map.

The first query retrieves the data for the table on our dashboard. It groups meteorites by their type and year, then compute the average and total mass of meteorites for that year. We aggregated the data to improve the loading time on our dashboard and creating out table.

```
def query_data(self, min_year, max_year):
    """Query meteorite data filtered by year."""
    query = text("""
        SELECT
            Year as year,
            rec_class as class,
            COUNT(*) as count,
            SUM(mass) as sum_mass,
            AVG(mass) as avg_mass
        FROM meteorites
        WHERE Year >= :min_year AND Year <= :max_year
        GROUP BY class, Year
        ORDER BY Year ASC;
    """)
    return self._execute_query(query, {"min_year": min_year, "max_year": max_year})
```

For the filters, we allow the user to input a minimum and maximum year and then use the "Enter" button to refresh the data. This re-runs the initialization function that pulls the min and max year values by their IDs and then makes a call to one or more of our API routes depending on the page. For the dashboard page, the raw SQL for the table and bar chart is relatively simple.

B. Challenges

The size of the data loaded during the initial pre-loaded query would significantly impact webpage performance time. Limiting the classifications quantity in the bar chart section resolved this issue but limits the overall quantity of the data. After deploying the website and verifying proper operation of all components locally, the sunburst chart would not render completely. This was resolved by changing the query concatation function to its operator.

```
query1 = text("""
    WITH TopYears AS (
        SELECT Year
        FROM meteorites
        WHERE Year >= :min_year AND Year <= :max_year
        GROUP BY Year
        ORDER BY COUNT(*) DESC
        LIMIT 15
    ),
    TopClasses AS (
        SELECT rec_class_group, year, sum(count) as count
        FROM (
            SELECT CASE WHEN COUNT(*) < 100 THEN 'Misc.' Else rec_class END as rec_class_group, Year, COUNT(*) AS c
            FROM meteorites
            WHERE Year IN (SELECT Year FROM TopYears)
            GROUP BY rec_class, Year
        )
        GROUP BY rec_class_group, Year
    )
    SELECT rec_class_group AS label, count, Year AS parent, rec_class_group || "_" || Year as id
    FROM TopClasses
    ORDER BY Year, count DESC;
""")
```

Data formatting requirements of the sunburst chart required combining multiple queries into a single Data Frame and then jsonify it. One Data Frame with a year, classification, and the count for that classification and year and a second Data Frame with an empty column, year, and the count for that year. Combining the two gave us a Data Frame that had a column for the parent (either an empty string “” or the year), an ID (either the classification or the year), and then the count for that ID.

```python
query2 = text("""
    WITH TopYears AS (
        SELECT Year
        FROM meteorites
        WHERE Year >= :min_year AND Year <= :max_year
        GROUP BY Year
        ORDER BY COUNT(*) DESC
        LIMIT 15
    )
    SELECT Year AS label, COUNT(*) AS count, '' AS parent, Year as id
    FROM meteorites
    WHERE Year IN (SELECT Year FROM TopYears)
    GROUP BY Year
    ORDER BY count DESC;
""")
df2 = self._execute_query(query2, {"min_year": min_year,"max_year": max_year})

# Add parent column to df2 and combine
```

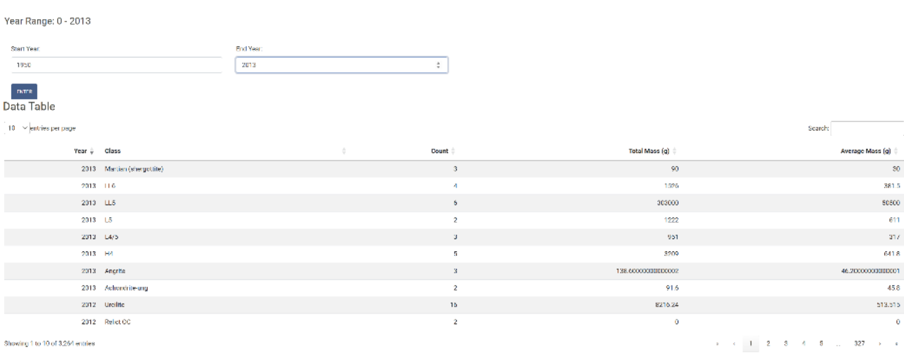
C. *Future Improvements*

- Incorporate additional visualizations (Bubble chart, regression plot) and more contextual information to enhance the overall user experience.
- A bubble chart that visualizes the average mass of meteorites, enabling users to quickly gauge the scale of each landing event.
- Reintroduce the meteorite name category from the original dataset, which will offer more detailed insights and identification of individual meteorites.
- More educational content about meteorite composition. For example, what makes up an L6 meteorite and exploring the origins of different classifications. This information can be shown through informational panels on our dashboard.
- Add additional filters to dashboard and map to filter by class and weight range and give immediate statistical data.

6. Visualizations

A. *Data Table*

Data set with a filter start and ending year from 0-2013. Allows users to view and sort all the categories from the dataset.


Year Range: 0 - 2013

Start Year: 1950

End Year: 2013

ENTER

Data Table

10 entries per page

Search:

Year	Class	Count	Total Mass (g)	Average Mass (g)
2013	Martian (shergottite)	3	90	30
2013	LL6	4	1526	381.5
2013	LL5	6	303000	50500
2013	L5	2	1222	611
2013	L4/5	3	951	317
2013	H4	5	3209	641.8
2013	Angrite	3	138.60000000000002	46.20000000000001
2013	Achondrite-ung	2	91.6	45.8
2012	Ureilite	16	8216.24	513.515
2012	Relict OC	2	0	0

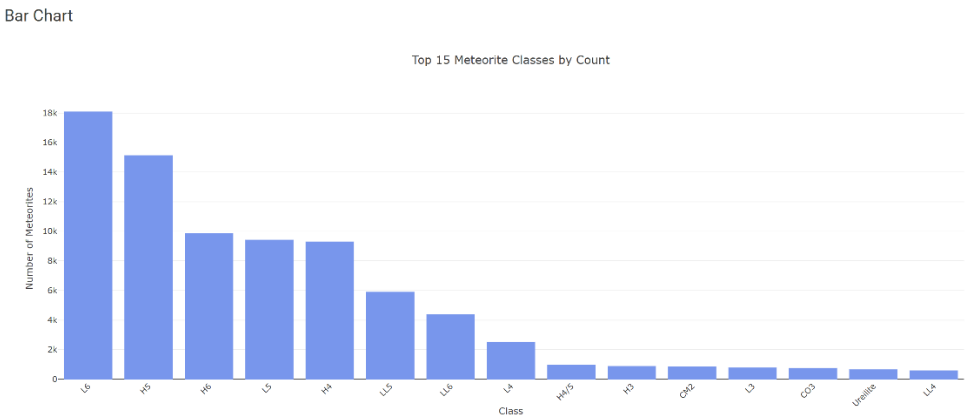
Showing 1 to 10 of 3,264 entries

1 2 3 4 5 … 327

Sunburst Chart

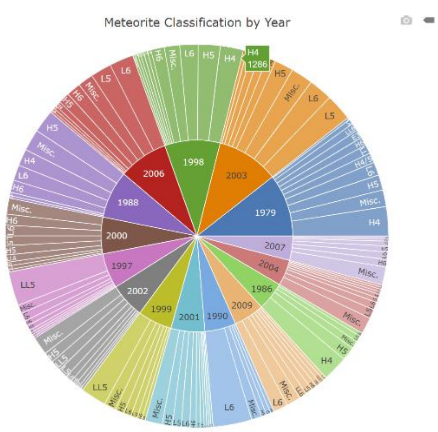
B. Bar Graph

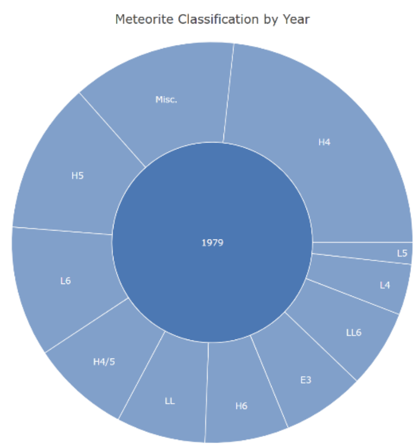
The bar graph shows the top 15 classes of meteorites that fell within the filtered time frame.



C. Sunburst Chart

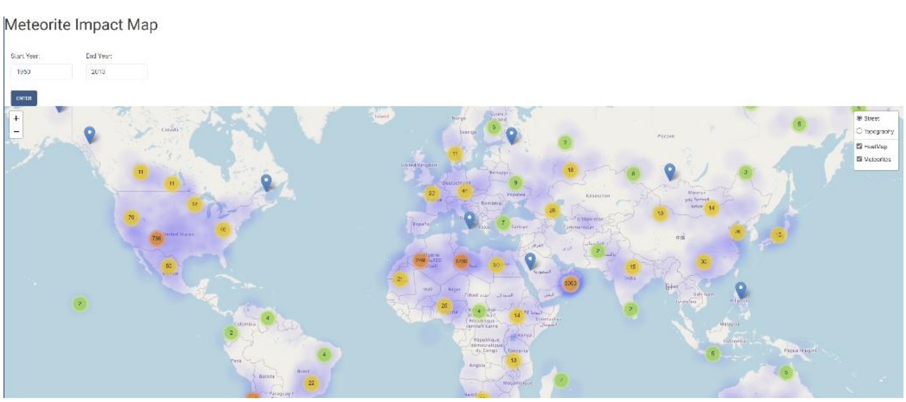
A sunburst chart that takes the year that the data was filtered by, and gives a visualization to see how many meteors of a specific class fell in that time frame.





D. Map

An interactive map where users can pan and zoom world-wide and visualize meteorite impact sites. With a heat map to quickly determine the highest impacts within the user query.



7. Conclusion

Key findings from our analysis fall into two main categories: geographical distribution and meteorite classifications. We observed that meteorite reports are more prevalent in the eastern hemisphere, with significant hotspots in Antarctica and the Middle East. These trends may reflect underlying biases. For example, the flat terrain in both Antarctica (glacial surfaces) and the Middle East (desert landscapes) makes meteorite sightings easier to report. Conversely, regions like Canada, which recorded fewer than 30 meteorite events from 1950 to 2013 despite its vast area, might suffer from underreporting due to lower population densities and more rural landscapes.

Building on these geographic insights, we also examined the classifications of meteorites that have impacted Earth. Utilizing our interactive dashboard, we can explore meteorite classes both by year with a sunburst chart and overall frequency with a bar graph. This analysis revealed that the L6 meteorite is the most commonly encountered class between 1950 and 2013.

A. Impact Patterns:

The dashboard indicates that the eastern hemisphere is more frequently impacted by meteorites, with consistent results even when filtering data over the past 100 years.

B. Strike Frequency:

Between 1950 and 2013, there were 15,251 recorded meteorite landings. Users can explore different time periods using the interactive filters.

C. Meteor Classification:

The most common meteorite class observed was L6 from the years of 1950 and 2013.

D. Data Limitations:

The dataset is not current (data only up to 2013), which limits the app’s relevance for recent meteorite activity. Visualization challenges arise with larger time frames due to data volume.

E. Reporting Biases:

Human-reported data tends to over-represent meteorite strikes in populated areas. Geographic factors make meteorites easier to spot in flat regions (e.g., deserts, Antarctica), possibly skewing the recorded data.

8. Works Cited

Meteorological Society Meteoritical Bulletin: Search the Database
https://www.lpi.usra.edu/meteor/metbull.php/

NASA Kaggle Dataset Meteorite Landings https://www.kaggle.com/datasets/nasa/meteorite-landings

Jay Panderson Kaggle Dataset cleaned_meteorite_data
https://www.kaggle.com/datasets/jaypanderson/cleaned-meteorite-data/data