

Meteorite Study

The workflow shown below shows the basic path the meteorite data takes to reach the final dashboard.

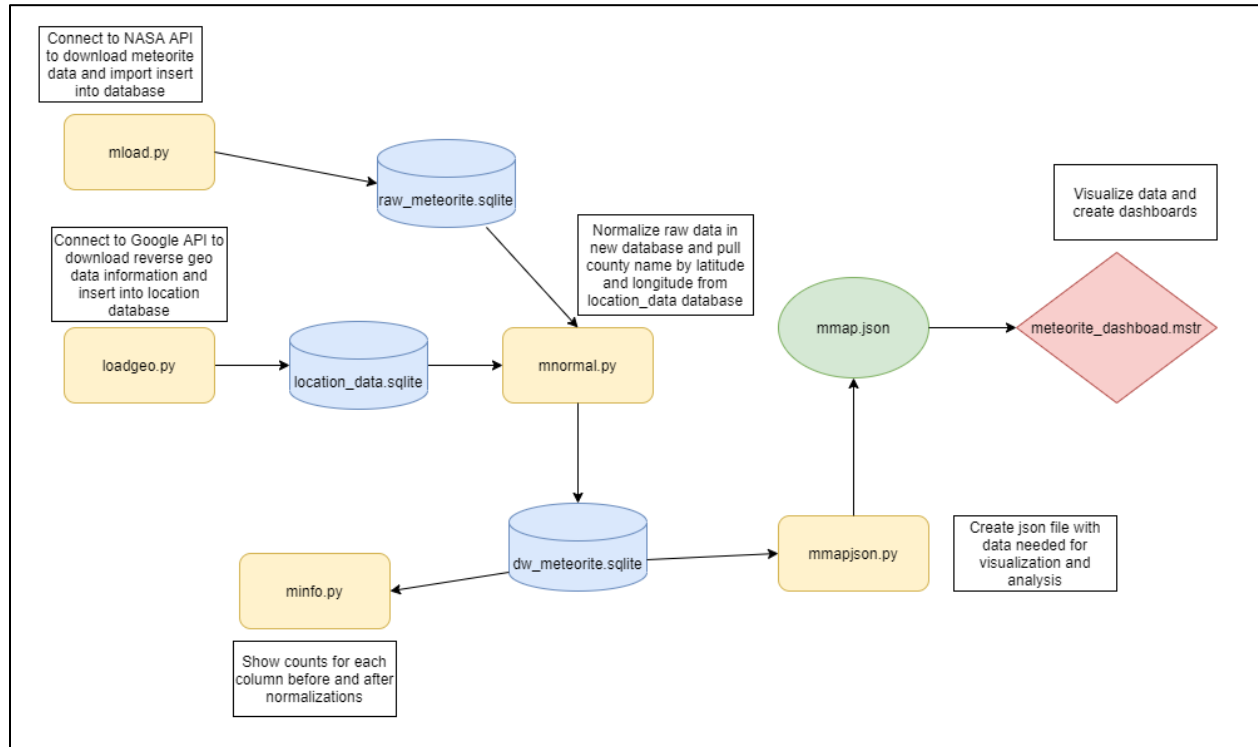


Figure 1: meteorite data flow

The Python file `mload.py` connects to a NASA JSON API, creates a single table database `raw_meteorite.sqlite`, and fills it with all available data. There are over 45,000 data rows of data.

The file `loadgeo.py` connects to a Google maps API that returns information pertaining to the given latitude and longitude. County is the information needed for this project but the information is stored in `location_data.sqlite` as the complete json file returned, thus preserving all location data for any future needs.

`mnormal.py` normalizes the data by creating 5 tables, Landings, Nametype, Class, Fallstatus, and Country. Then filling them with the appropriate data and assigning the necessary ids.

Normalized ERD shown below, Figure 2.

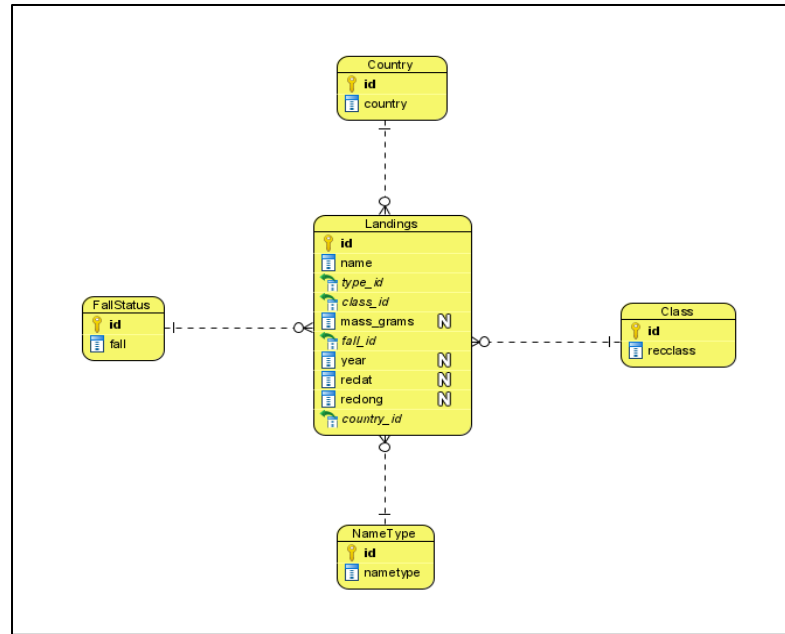


Figure 2: Normalized meteorite database

The file minfo.py pulls from dw_meteorite.sqlite and raw_meteorite.sqlite and displays the count of all none null data points in each column for each database. This will give visibility on how successful the normalization was.

Mmapjson.py converts the data need for our dashboard into a json file to be used by meteorite_dashboard, shown below.

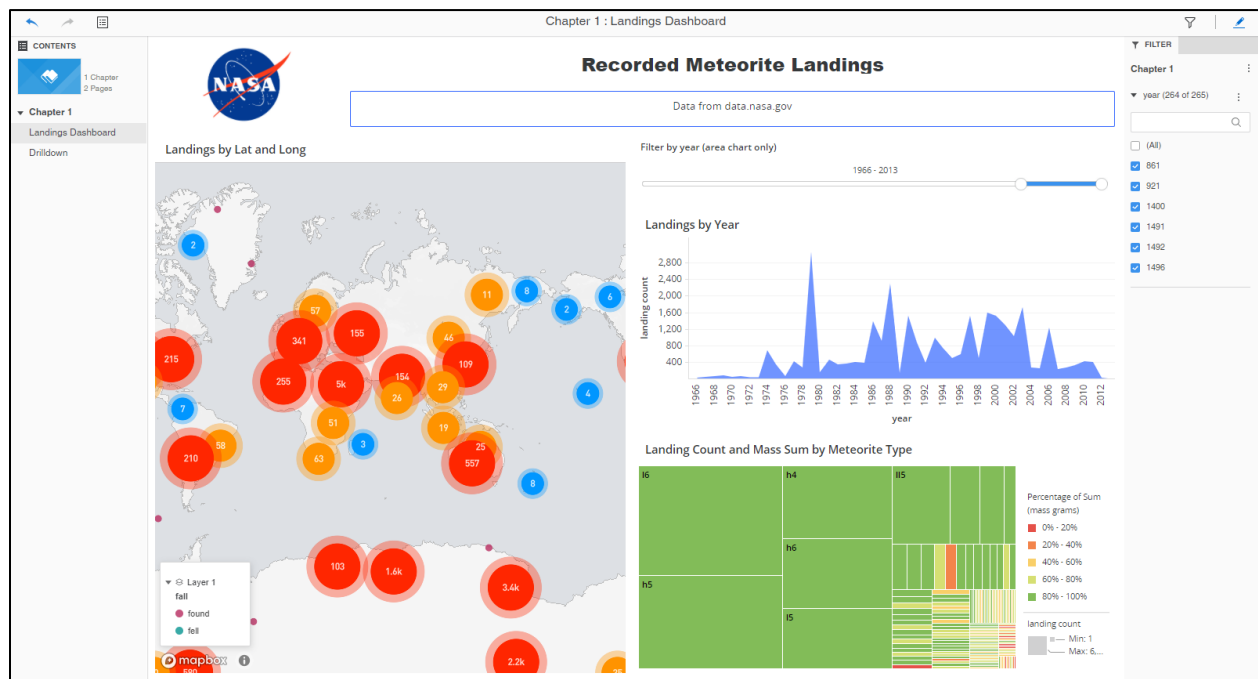


Figure 3: Meteorite Dashboard

Figure 3, Left: Meteorite landing count shown by latitude and longitude. Right top: Landing count by year. Right bottom: Landing count and mass sum by meteorite type.

Figure 4, Bubble chart showing meteorite landing count by country. It is helpful to remove Antarctica and compare the resto of the countries.

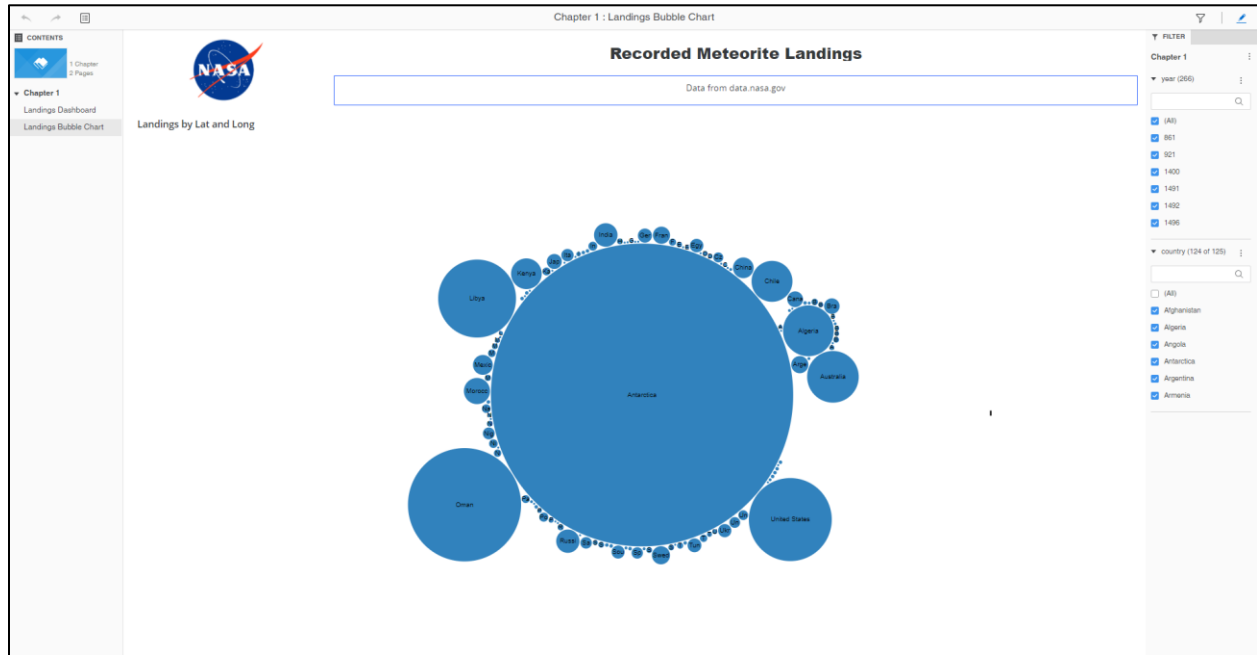


Figure 4: Bubble chart showing number of landings per country