Technical Report: Satellite Exploration

Prepared by Kholiswa Tsotetsi, Azzy Caceres, Ravi Chinnamaidu and Michael Striffler

Rutgers University Data Science Bootcamp

December 10th, 2020

Abstract: In this report, data from real-time satellite tracking website, N2YO.com was extracted utilizing web scraping and API calls. From here, data was either stored in a PostgreSQL database and/or directly converted to a json file. Visualizations were created using Javascript D3, Plotly, Leaflet, and ThreeJS libraries and plugins. Ultimately, a dashboard was created to display this real-time data utilizing Flask and live server on the local server.

Introduction

Since the 1950s, the curiosity and fascination with space has led to the boom in its exploration. Though space exploration was initially limited to governmental programs, its expansion has allowed for non-governmental agencies to utilize space for their own benefit. In today's age, the use of rovers, spacecraft, satellites and other instruments has allowed us, as a society, to gain a deeper understanding of this universe.

The recent launches by SpaceX, a private space exploration company founded by Elon Musk, has sparked our interest in what man-made spacecrafts and satellites are truly orbiting above us. As these instruments are used for governments, climate tracking, cell phones, television, radio, and more, our group wanted to focus on these instruments, specifically satellites, by exploring the following questions:

1. What satellites might be over our heads at this moment?
2. Where were these satellites launched from? Which country?
3. How many satellites have been launched throughout the years?
4. What is their altitude?

To answer these questions, we searched for data pertaining to the different satellites that are currently in orbit. In doing so, we came across NY20.com as a satellite data source that tracks all satellites in real time. We chose this data source because it consolidates global satellite information all in one source. As you will find later in this report, there are thousands of satellites in orbit that are divided in many different categories. With that said, each one serves a purpose.

Data Acquisition

In order to access the information for our analysis, an API was required. Initially, we attempted to call the API directly in javascript however, the url for this API key required the input of each satellite ID. Thus, Python Jupyter notebook was utilized to loop through a select set of satellite IDs to receive the requested information. We compiled a list of satellite IDs from the Earth Resources category into a csv file which was uploaded into our notebook file. From here, we defined an open list and looped through this list which inputted each satellite ID into the API url for each variable of the list until the end. The returned data was then appended to our open list until completion. From here, we converted this list to a json object which was later used in our javascript for the Earth Resources visualizations. A similar procedure was performed to retrieve data for our satellite chart, which will be described below. In this case, a different API url was used to acquire all satellites, regardless of category, that are within a 90 degree radius from New Jersey. This API call resulted in close to 3,000 satellites which were converted into a json object for later use.

In addition to calling our API from a Python jupyter notebook and converting the response into a json object, we also utilized web scraping to obtain the real-time satellite data by satellite country and category. Here, we also utilized Python jupyter notebook and called SQLAlchemy among other toolkits to merge our data to a PostgreSQL database. Using the splinter toolkit, we called the N2YO.com satellite homepage which includes the satellite category, and number of satellites, debris, and rocket bodies currently in orbit. This information was scraped into an empty list which was then converted to a pandas dataframe. From here, the database was pushed to our SQL database. This process was repeated to extract data for the satellite country and additional satellite information. For the satellite country, we scraped the N2YO webpage for satellites by country and limited our search to satellites from the United States and the former states of the USSR which included over 3,000 satellites. The extracted data included the name, ID, international code, launch date, and period (in minutes) for each satellite. Again, this data was appended to a list, converted to a pandas dataframe, then sent to the corresponding table in our PostgreSQL database. Lastly, for additional satellite information, we scraped the N2YO.com for a list of countries along with the corresponding number of satellites launched. Once all of our extracted data was transferred to the PostgreSQL database, we then created a python file connected to the database and utilized a Flask app to run the data.

Data Analysis & Visualizations

Maps

Using our extracted data, we created five different styled maps to represent this data utilizing Leaflet and D3 JS libraries. In our first map which is present on the home page, we utilized the real-time data from the web scraping Python jupyter notebook files to populate the map with satellites based on the user's selection of country, and satellite quantity. Alternatively, the user can also make their selection based on the satellite ID. Using the navigation bar styled with Bootstrap framework, the other three maps along with the bar chart can be viewed.

The second map, which visualizes the data from the satellites within a 90 degree radius of New Jersey, displays these satellites based on their altitude, which the user can indicate.

The third map displays the satellite data from the Earth Resources category. By clicking each icon, the satellite name and launch date is displayed. Additionally, in the bottom right corner, the set view can be altered by the user by swiping the map in either direction.

For the fourth map, we visualized the data from the third map in a 3D format using the ThreeJS library. Here, we created a spherical globe which shows our satellite data points as 3D markers. These points were derived from the json object and the latitude and longitude were extracted using a for loop then appended to an empty list. From here, we created markers by looping through this list and transforming the coordinates to a globe object.

Our last map shows where each satellite is launched from and the country that owns the satellite as multiple countries launch their satellites from sites in other nations. To show this, we created a market cluster map. The tooltip allows a hover over which shows that information. There are surprisingly very few launch locations around the globe thus countries without a space center have to partner with other countries to launch their own satellite.

<u>Bar chart</u>

Going back to our initial questions, we next wanted to explore how long satellites have been in orbit. To get a better idea of this, we looped through our json file of satellites within a 90 degree radius of New Jersey to retrieve the launch dates. Next, we enacted a string slicing method to identify only the year instead of the full date (year-month-day). Once all the dates were identified, we totaled the launches by decade and visualized the results using the PlotlyJS library. It was interesting to find that there are still satellites in orbit that were launched decades ago. Our findings showed that more satellites were launched during the 1990s than any other decade. One could hypothesize that this particular boom is the result from the increase in commercial space exploration.

<u>Limitations</u>

Our main limitation was the sheer amount of objects in orbit, which, according to N2YO, is over 21,000 space objects. When using API calling, we were unable to call and visualize all satellites at once as it took too much time to retrieve this data. Because of this, we shifted our strategy on how to build on our data set to starting small and working our way to a larger scale to evade some of the errors we encountered. As an example, it takes only a few seconds to call real-time locations of just 10 satellites versus a couple of minutes to retrieve hundreds.

Initially, we planned on mapping live locations on the 3D globe though, we quickly realized that in order to utilize Leaflet JS, a 2D map would be better suited. There are multiple libraries that can be utilized to visualize data using a 3D object such as BabylonJS, GioJS, ThreeJS and WebGL. In our case, we found that the ThreeJS library suited our needs though, given more time, we would have liked to explore the other 3D visualization libraries.

Our last limitation was figuring out how to call our API. At first, we sought to call our API directly from our js script though after many unsuccessful attempts, we decided to utilize python for our API calls.

Conclusion

From this project, we have gained a deeper understanding of space, and in particular, the objects that orbit this planet. Using our app, users can navigate the site with ease to view satellites from various countries and categories and their launch sites. Additionally, if users are in the New Jersey area, they can view all satellites within a 90 degree radius of their location. Lastly, focusing on satellites for Earth Resources, which monitor and track the Earth's climate, users can view their current location in orbit.

Ultimately we were able to create an HTML webpage that runs through our Flask server that holds all our findings. Despite our limitations, we successfully produced interactive visualizations of our extracted results.