Wager Group Twitter Dashboard

Tools for Data Analysis - Final Project Write Up

Alec Wick

The Wagner Group is a private military organization that is allied with Russia. This group is notoriously known for its harsh fighting tactics and horrendous war crimes. Wagner fighters are currently invading Ukraine along with the Russian military. During this conflict, there have been multiple reports of the Wagner Group participating in war crimes. This project aims to provide a Ploty dashboard for the Department of Homeland Security (DHS) to view tweets and additional metrics about the Wagner Group. In order to accomplish the goal, I used Twitter's developer API and several Python packages such as Pandas, Tweepy, Plotly, and Datetime.

The first step in developing this dashboard involved collecting tweets that contain the hashtag #WagnerGroup. I also added filter conditions such as removing retweets and only displaying tweets that are written in English. I installed the Tweepy Python package and purchased a standard Twitter developer license that allowed me to extract 10,000 tweets. I then used the search_recent_tweets() function to pull in the most recent tweets regarding the Wagner Group and stored the results within a Twitter object. I created a separate list for messages, users, locations, and dates to store the Twitter object attributes. A For Loop then iterated through the Twitter object and stored the information in each list that I generated. I used the DataFrame function and converted each list to a data frame and merged the tables together using the concat() function. The final result is a table that displays the most recent tweets that contain #WagnerGroup with metrics such as username, location, date, and time.

The next step in the data-collecting process involved calculating the number of times #WagnerGroup was used in Twitter posts within the past 7 days. To accomplish this, I used the get_recent_tweets_count() function with the granularity parameter set to 'day'. This resulted in an array that displayed the count frequency by date. The resulting dates were not in the correct form so I used the Datetime package and used the strftime() function to get the date in m/d/y

format. I used the data to generate a line graph using the Plotly package. From the tweet count line chart, we can see that there was a large spike on May 6, 2023. Around this date, the Wagner Group announced that they were planning to pull their troops out of Bakhmut, Ukraine. This may be a reason for the large spike in users tweeting about the Wagner Group.

The next key component for developing this dashboard involved generating visuals that reflect which countries the users are located in. From the tweet table that I generated earlier, I used the location column to calculate country frequency. Unfortunately, the user location on Twitter is a free-form field that a user can enter. This means that the locations are not consistent in terms of formatting and may be inaccurate. For example, some users will have a location in New York, New York City, or New York USA. In addition, some users will use inaccurate locations such as Neverland, Atlantis, or the Pentagon. As you can see, there are discrepancies in the ways in which location is displayed. In order to properly display the correct country, I had to manually enter the country from the listed location. I then used the table function and calculated frequency and generated a choropleth plot by country. Within the choropleth function, I had to include additional parameters to determine the globe size and color density. As you can see from the map, the United States had the highest frequency of Wagner Group tweets. I included a bar chart tab as well to help distinguish the frequencies of each country.

The next phase of development includes building the Ploty dash dashboard and inserting visuals and the tweet table. In order to create this dashboard, I installed the dash Python package along with dash_bootstrap_components. The majority of my dashboard development took place within an application called PyCharm. PyCharm is a Python IDE that allows for easy dashboard development. I pulled my Twitter data into this dashboard by using public links to my GitHub repository. I had a separate source for my tweet table, country frequency, and past 7 days' frequency. I used Dash's app.layout feature which uses html divs, dbc columns and dbc rows to format my dashboard. Within the dash package, I used the dash_table function to generate my tweet table. This is a useful function because it allowed me

to add alternative cell formatting, column heading formatting, and scrollable functionality. The dash bootstrap function allowed me to format the whole dashboard by selecting a predefined theme. For this dashboard, I decided to use the Cosmo theme. Overall, the dash package was an essential part of creating the Twitter dashboard.

Within this project, there were several hurdles that I had to overcome in order to properly create a Twitter dashboard. The first obstacle involved Twitter's API license. Twitter recently had some changes with their terms of API licenses. There are some features that are free but the majority of the useful ones come at a cost. I decided to purchase the developer license so that I can get the information I need for this project. I didn't realize this at first, but Twitter's API had a major change between version 1 and version 2. The way in which the API has to be authenticated changed and the way in which objects were stored in a twitter object changed. Several of the Youtube videos I watched and articles that I read were focused on version 1. Once I was able to discover the differences between the two versions, I had more success with pulling the Twitter information. Another problem that I ran into was that I only had 10,000 tweets to pull. When I wrote my code, I had my function use the API and pull in the 100 most recent Wagner Tweets. Unfortunately, each time I ran my code, I was using 100 of my 10,000 tweets. It didn't take long for me to almost run out of tweets. With the tweets I had left, I gathered the Twitter information and stored the results in CSV files in my public GitHub repository. I then used this file to pull data into my dashboard. Another issue that I encountered is that my Plotly dashboard kept stacking my visuals and tables on top of each other. After several searches on stack overflow, I was able to determine that I had to include an external_stylesheet command in order to use my bootstrap theme and dbc formatting elements. Even though I had several obstacles, I believe I was able to create a high-quality dashboard for the Department of Homeland Security.

In conclusion, the Wagner Group is a military organization that is accused of committing horrible war crimes. The Department of Homeland Security asked our class to use our data

analysis skills and provide insightful information to their team. I believe that the Twitter information I was able to use will provide helpful information to DHS. During the project, I learned several important concepts such as creating dashboards using Plotly, bootstrap themes, and how to use APIs in Python. My recommendation to the DHS is to get an enterprise license and analyze thousands of tweets over time. I believe this would help provide a more complete picture of what users are saying about the Wagner Group. Overall, this was a great exercise to expand my analysis skills using Python and an opportunity to provide insight to the Department of Homeland Security.

Dashboard Screenshot