Data and Info Visualization - Twitter Word Cloud

Andrew Struthers, Gihane Ndjeuha, Kollin Trujillo, Nathan Chapman, Nick Haviland

February 2023

1 Introduction

In this project, we were tasked with generating visualizations regarding the top words Tweeted by Elon Musk over the past handful of years. We were supposed to make a word cloud and bar chart of the top 10, 30, and 50 words used, as well as just a word cloud for the top 100, 150, 200, and 250 words used. This data came from nearly 12.5k Tweets. As part of this project, we learned about "stop words", which are common words that don't necessarily add value on their own. These "stop words" are used in many natural language processing and language filtering models. Some common stop words are words like "a", "the", "it", "and", etc. We ended up filtering out nearly 200 of the most common stop words, as well as some emojis and punctuation, to get a clean visualization. That being said, some words that we ended up filtering out were words like "will", "just", and "lot", which do have multiple meanings. We felt that filtering out words like this wouldn't negatively impact the accuracy of our visualization because, even though words like this do have multiple meanings, their most common uses fall under the "stop words" category.

2 Programming Language

We used Python for the word processing and generation of this visualization. The packages we used included Numpy, Pandas, Matplotlib, CSV, itertools, RE, and WordCloud. We used Numpy and Pandas to read the .csv file containing all the Tweet information, loading it into a Pandas DataFrame. We then filtered out more than 100 common "stop words", a common filter in many natural language processing models, and loaded the filtered data into another Pandas DataFrame. We then used the RE package to perform some RegEx to clean up the Tweets even more. After building the word dictionaries and sorting the dictionaries so that the largest counts were closer to the beginning of the structure, we used itertools to slice out the specified number of words, and get smaller dictionaries of specified length to pass to the visualization libraries. We used Matplotlib to generate the horizontal bar charts, and the WordCloud library to generate the word clouds, using the given dictionary key, value pairs as the frequencies for each word.

3 Output

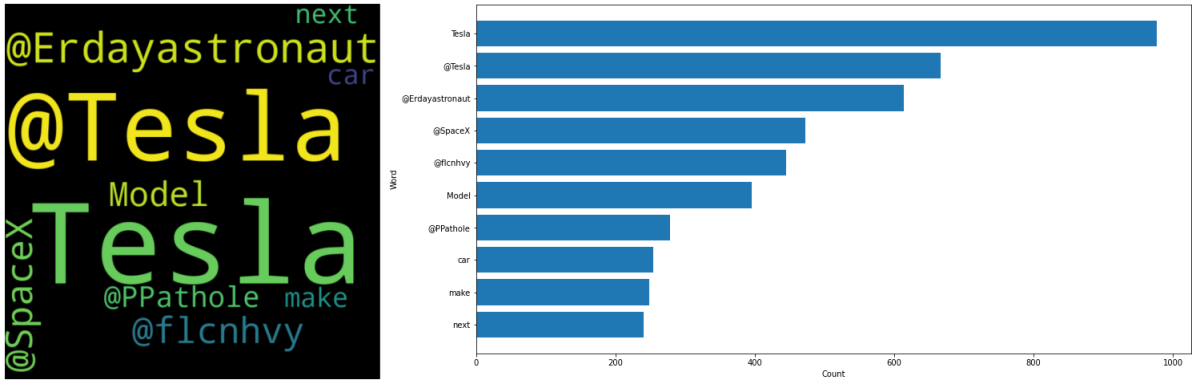

Figure 1: Word cloud and bar chart of the top 10 words after heavy filtering

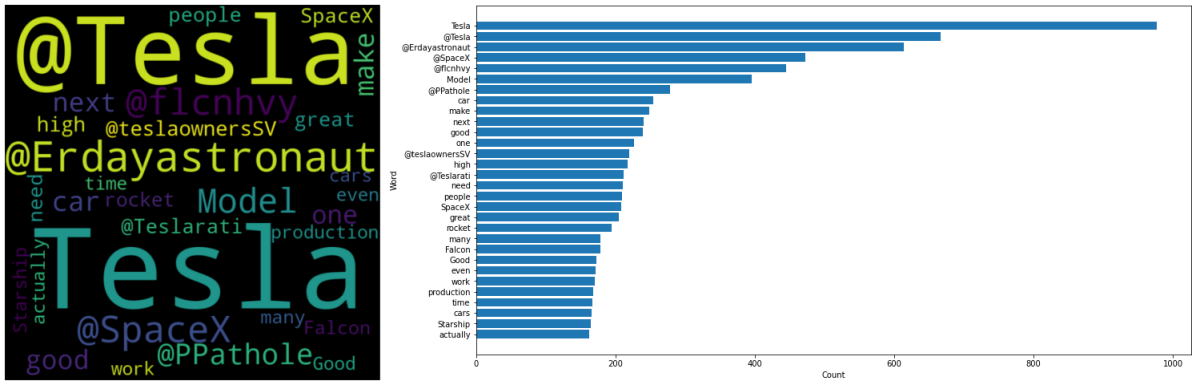

Figure 2: Word cloud and bar chart of the top 30 words after heavy filtering

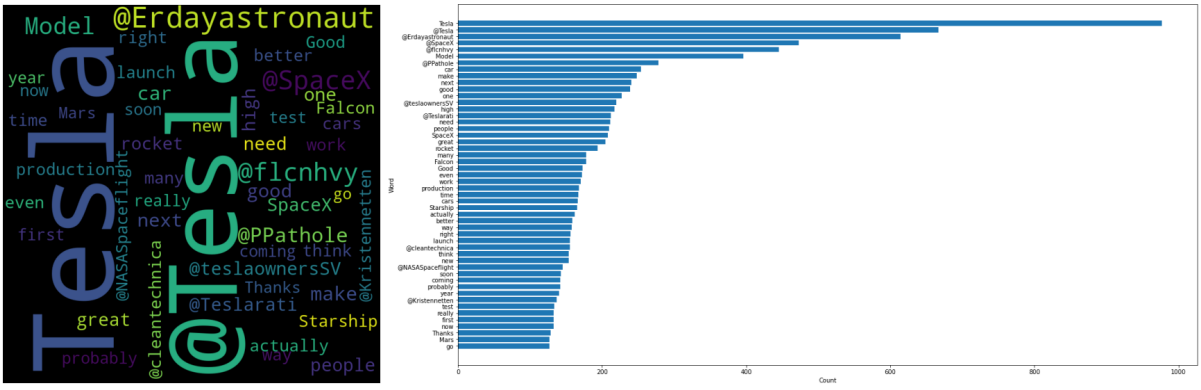

Figure 3: Word cloud and bar chart of the top 50 words after heavy filtering

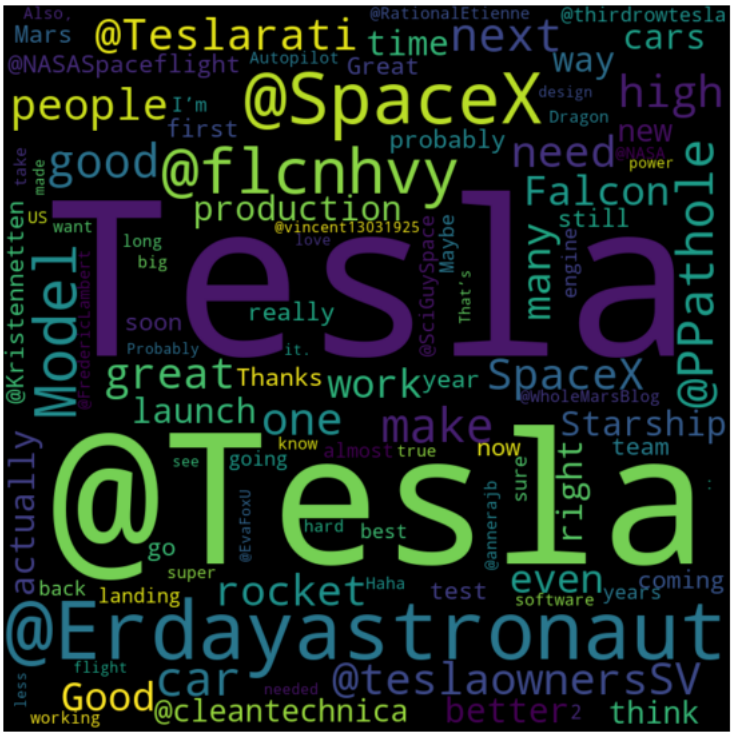

Figure 4: Word cloud of the top 100 words after heavy filtering

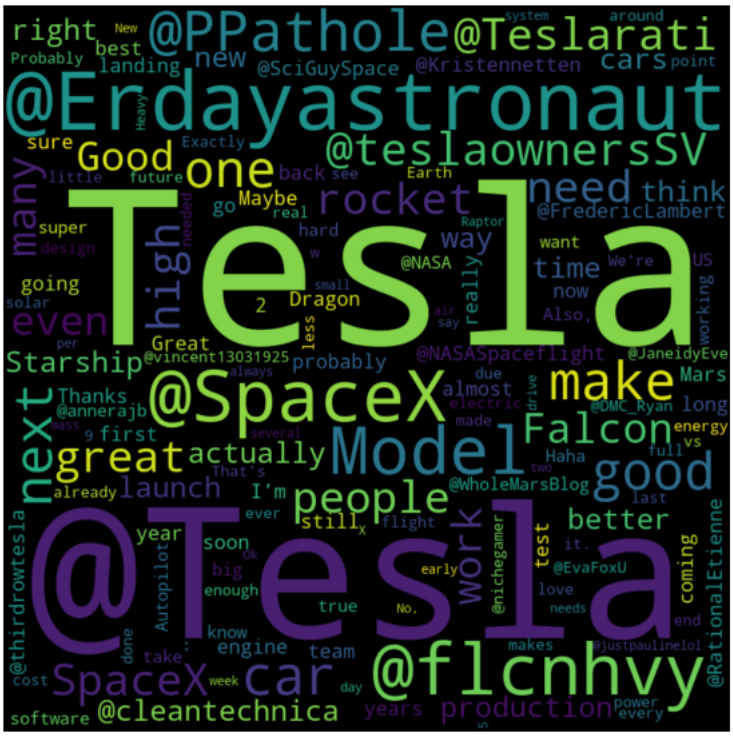

Figure 5: Word cloud of the top 150 words after heavy filtering

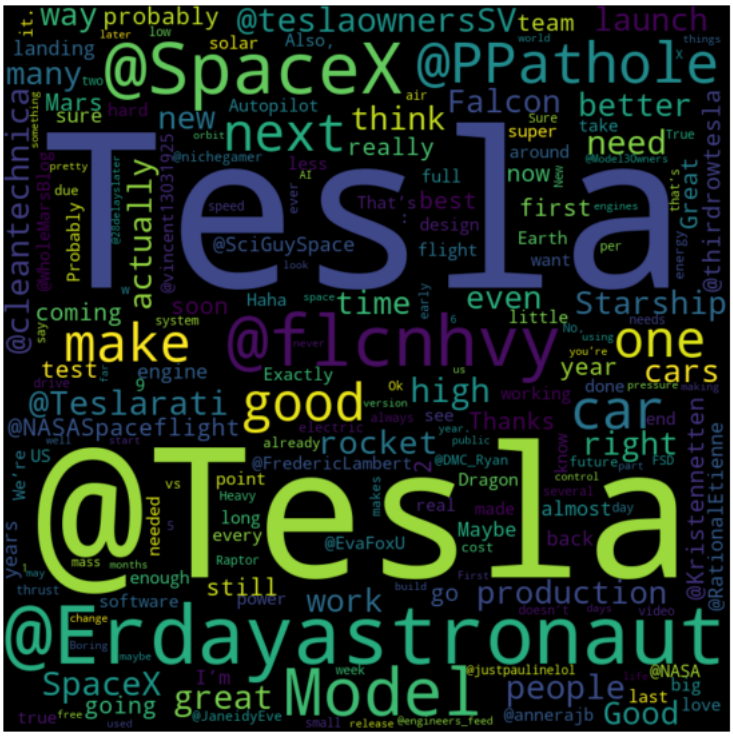

Figure 6: Word cloud of the top 200 words after heavy filtering

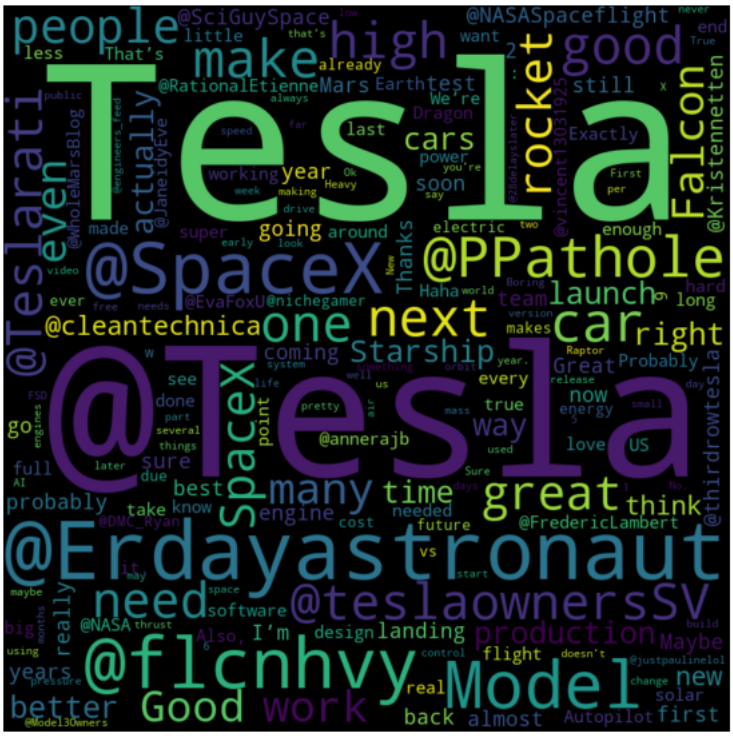

Figure 7: Word cloud of the top 250 words after heavy filtering

4 Interesting Findings

The most obvious conclusion is that Elon really likes to mention "Tesla". The word appears almost double the amount as the fourth-most frequently used word (that being "@SpaceX"). Next, the overall frequency distribution of the words follows a generally exponentially decreasing trend (shifted up). Another interesting note is that, out of the top ten most frequent "character runs", 50% are actually tags to other Twitter users. 60% of those tags are to individuals and the others are to the Twitter accounts for companies like "@Tesla" and "@SpaceX".

These conclusions are reflected by the word clouds, both "Tesla" and "@Tesla" using the overwhelming majority of the space. The word clouds offer a significant reduction in cognitive load since it's almost compulsory to immediately recognize the biggest objects in our vision.

Looking closer at the frequently mentioned Twitter handles in the word clouds, there are not many mentions of extremely popular twitter accounts. Certain accounts like @erdayastronaut, @ppathole, and @kristennetten are not media outlets or large accounts, but have a very high response rate from Elon. In addition to this, there is a high word count on topics related to Tesla and SpaceX, but none of the top 50 directly pertain to Neuralink or the Boring Company, even though they were founded in 2016 and 2017 respectively.

Additionally, we are curious to see how these visuals would update with a more recent data set. Since his acquisition of twitter, we wonder if there has been a high enough volume of tweets to place topics directly related to twitter and ownership of it within the top 50.

5 Lessons Learned

Some hardships from writing this code for this homework is that the bulk of the columns are worthless. There are some things we could have done with the other data, such as plot Tweets at time of day and visualize things such as that. We utilized Python for this visualization which made a lot of the data analysis and processing pretty straight forward. Some issues dealt with which words to remove to make the visualization more pertinent along with remembering how to iterate over the rows of the DataFrame as the Tweet to eliminate those words. Luckily Python made sorting the dictionary by value easy along with taking the first n values of the dictionary to graph with the WordCloud visualization.

One of the lessons we learned is just how many "stop words" there are. At first, most of our visualizations looked like the figure below. This was with minimal filtering, which we thought would have been sufficient. As we can see from the figure, there are some "useful" words, like "Tesla" and "Model", but the data is mostly filled with generic words that don't really provide useful insights. We generated a list of the top 100 most common "stop words" in the English language and applied that as a filter, which provided much better results, but even with this filter there were still many words we wished weren't in our figures. We manually went through the visualizations and added these words in to our filter, which took quite some time because each new run with more words in the filter caused new words we hadn't yet thought of to pop up. Even though we had generated the top 100 words to use as a filter, we manually added 93 new words to that filter after watching the code run over multiple iterations. We had to also filter out a few different emojis, due to Elon Musk Tweeting with emojis frequently enough to show up in the visualizations.

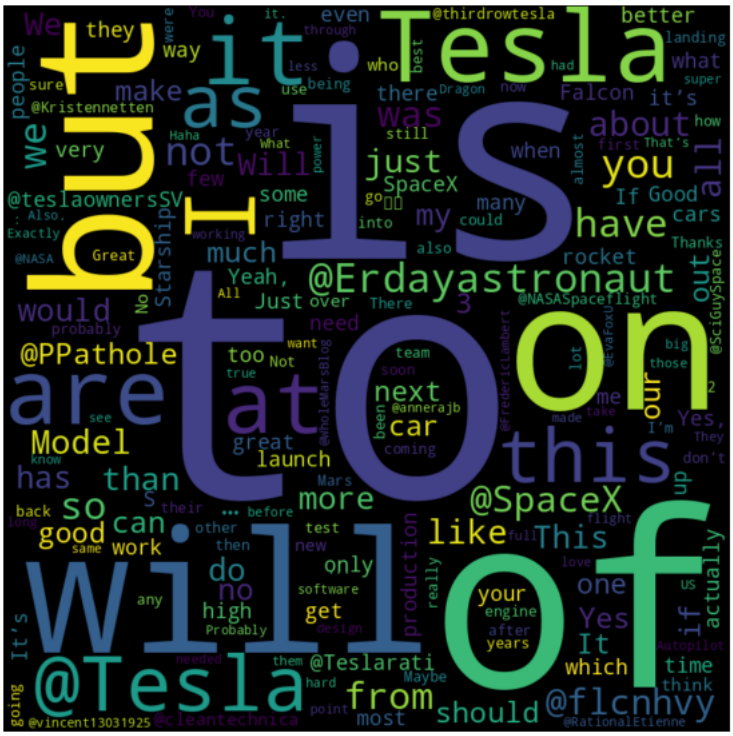

Figure 8: Very little filtering applied