Sentiment Analysis of Moderna by Scraping Twitter API

George Washington University

Student Research Specialist

By: Rakshith Reddy Eleti

Goal:

The main moto of this project is to get to a conclusion by scraping the tweets from twitter API about the most predominant covid-19 vaccine Moderna. Twitter is nothing but it is one of the most popular social media platforms, for many businesses and individuals, having a strong Twitter presence is critical to keeping their audience interested.
To achieve the moto, we start by scraping the data of 1500 tweets worldwide. We then do some pre-processing for the extracted tweets.

Tools Used:

The tools used here are:

Tweepy:
It is an open-source Python program that makes it extremely easy to use Python to access the Twitter API. Tweepy comes with a collection of classes and methods that reflect Twitter's models and API endpoints, as well as the ability to handle different implementation details transparently. The Python package manager pip may be used to install Tweepy using the following command "!pip install tweepy".

Plotly Express:
Plotly Express or PX is the plotly.express module (typically imported as px) that provides functions that may build whole figures at once. Plotly Express is a portion of the plotly library that comes pre-installed and is the suggested starting point for most common figures. Plotly Express has around 30 functions for producing various sorts of graphs. Throughout a data exploration session, the API for these functions was deliberately intended to be as consistent and straightforward to understand as possible, making it simple to transition from a scatter plot to a bar chart to a histogram to a sunburst chart.

TextBlob:
Python provides a number of packages that make conducting Natural Language Processing jobs as simple as feasible. TextBlob is one of the most well-known and simple-to-use libraries. TextBlob is a text processing package for Python 2 and 3. It provides a straightforward API for doing standard natural language processing (NLP) activities including part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. For installation, we use "!pip install TextBlob".

Pandas:
Pandas is a Python data analysis package. Pandas grew into one of the most popular Python libraries as a result of a requirement for a robust and flexible quantitative analysis tool. It has an incredibly active contributor community. For installation, we use “!pip install pandas”.

Seaborn:
Seaborn is a Python module for creating statistical visuals. It is based on matplotlib and tightly interacts with pandas data structures. Seaborn assists you in exploring and comprehending your data. Its charting functions work with data frames and arrays containing whole datasets, doing the necessary semantic mapping and statistical aggregation internally to generate useful graphs. Its dataset-oriented, declarative API allows you to concentrate on the meaning of your charts rather than the mechanics of drawing them. For installation, we use “!pip install Seaborn”.

Matplotlib:
One of the most widely used Python libraries for data visualization is Matplotlib. It's a cross-platform library that generates 2D charts from array data. It provides an object-oriented API for embedding plots in programs written in Python utilizing GUI toolkits like PyQt and WxPythonotTkinter. It's also compatible with Python and IPython shells, as well as Jupyter notebooks and web application server. For installation, we use “!pip install Matplotlib”.

NLTK:
NLTK is a popular Python programming language for working with language processing data. It includes a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum, as well as easy-to-use interfaces to over 50 corpora and lexical resources like WordNet. NLTK is ideal for linguists, engineers, students, educators, academics, and industry users alike, thanks to a hands-on approach that introduces programming foundations alongside subjects in computational linguistics, as well as rich API documentation. "A superb tool for teaching and working in computational linguistics using Python," as well as "an outstanding library to play with natural language," have been said of NLTK.

Numpy:
NumPy is the most important Python module for scientific computing. It's a Python library that includes a multidimensional array object, derived objects (such as masked arrays and matrices), and a variety of routines for performing fast array operations, such as mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and more.

Working:

We start by importing all of the necessary libraries, and then we enter the necessary Twitter credentials for scraping the data: the consumer key, consumer secret key, access key, and access secret key. The keys may be found on the official Twitter account. The authentication handler is then used to connect to the API by obtaining the keys.

We begin extracting the tweets after supplying all the appropriate keys for access, starting with the quantity of tweets you wish to extract. Then make a blank list with the names of tweets, likes, and time. We then use the cursor to iterate across the API, using the command search tweets to enter the data we wish to retrieve. The extracted data is then appended to the previously constructed lists.

The data should be transformed to a dataframe after the tweets have been extracted. Then we go on to learning about the data. We need to understand the datatypes that are involved. The dataframe's form is defined by the number of columns and rows it contains. We used the command describe() to examine the statistics, which returns the number of likes, the mean, the standard deviation, and so on. We construct a histogram using Plotly.Express for EDA, which provides us the amount of tweets about Moderna per day, which gives us the knowledge of how many tweets are being tweeted about Moderna by a user every day. We next go to Data Pre-processing, where we check for null values in our data. If there are any, we add them all up. The Re-tweets are then removed from the data. We want original tweets, not the same old ones. Using the same tweets would not get the desired results. As a result, we exclude all Retweets from the data. Reset the index after that. We keep monitoring to see which tweet receives the most likes.

Then, using regular expressions, clean the data by removing any mentions, URLs, stopwords, extra space, and, ultimately, all emoticons found in the tweets. Tokenize the data after that. Do the lemmatization once the data has been tokenized. We then make a tree map of the most prevalent terms in the tweets; I opted to extract the top 50.

We then remove the columns Time, Likes, Tokenized, and Unprocessed Tweets from the sentiment analysis and replace them with the lemmatized column. Then, for further sentiment analysis, convert the lemmatized to a string.

We can get the subjectivity and polarity of tweets by using TextBlob. Following the extraction of the text's subjectivity and polarity ratings. We design a function for text analysis that assigns a bad tweet to texts with a score less than zero, a neutral tweet to texts with a score equal to zero, and a positive twitter to texts with a score greater than zero. The scores are then converted to percentages, and the result for the tweets is obtained.

Result:

After processing 1500 tweets, we get:

- 44.00715563506261 % of positive tweets
- 36.314847942754916 % of neutral tweets
- 19.67799642218247 % of negative tweets

Conclusion:

We may deduce from people's tweets on Moderna that 44 percent feel the vaccine is effective and safe. According to 36% of them, they are neither favorable nor negative about the vaccination. The vaccination does not completely satisfy 19% of them.

References:

- https://realpython.com/twitter-bot-python-tweepy/
- https://plotly.com/python/plotly-express/
- https://www.analyticsvidhya.com/blog/2021/10/making-natural-language-processing-easy-with-textblob/
- https://mode.com/python-tutorial/libraries/pandas/
- https://seaborn.pydata.org/introduction.html
- https://www.tutorialspoint.com/matplotlib/index.htm
- https://www.nltk.org
- https://numpy.org/doc/stable/user/whatisnumpy.html