Conducting a Sentiment Analysis on Taylor Swift to Understand Public Opinion and Perception

Mikayla Peterson (200425538)

ENSE 480

April 11, 2023

Table of Contents

List of Figures

1.0 Introduction

The question that this project aims to answer is '*How has public perception of Taylor Swift changed since the announcement and opening shows of the Eras Tour"*? My hypothesis is I think that for the brief period before the official announcement of the Eras Tour on November 1, 2022, the overall sentiment is positive. The Ticketmaster Verified Fan Presale for the Eras Tour began November 15, 2022; due to major complications with the presale, I think public sentiment shifted negatively. The opening shows of the tour were in Glendale, AZ on March 17 and 18 and the tour has received glowing reviews. As a result, I think public sentiment became more positive. The objective of the project is to answer the question and prove or disprove my hypothesis. The significance of this question is that Ms. Swift's level of fame has reached new heights since the October 21, 2022 release of her tenth studio album, *Midnights*, and subsequent tour announcement, but the most recent sentiment analysis about her using tweets that I have seen occurred back in July 2022 only using tweets mentioning her name (Sharma, 2022) (Daityari, 2019). The scope of this project was to download tweets made on or after October 21, 2022, containing Ms. Swift's Twitter account, *@taylorswift13*, and tweets that mentioned her account, then analyze those tweets using natural language processing (NLP) techniques and Valence Aware Dictionary for sEntiment Reasoning (VADER), then finally plotting the analyzed data to prove or disprove my hypothesis.

2.0 Knowledge and Data Representation

When the SNScrape library is used to query Twitter for a list of tweets, the resulting output is stored in a list which can be put into a pandas dataframe to be downloaded as a CSV file. I wanted to ensure that the data I used in the project could be saved outside the program for a few reasons. I knew that it would take multiple hours for the tweets to be downloaded so

instead of forcing a user to download a fresh data set every time they run the program, they should be able to use the dataset that I downloaded myself. I also wanted to make sure that my findings were repeatable, so the input tweets should be the same. Also, the time that the tweets are gathered is relevant to my project's main question. All the CSV files I made will be provided when I hand in the system code. These CSV files can then be loaded into dataframes so that I can use efficient, vectorized methods to manipulate them for data analysis purposes. The initial structure of tweet CSV files and the dataframes resulting from them before any manipulation is as follows:

ID	Row ID
URL	Exact URL to the original tweet.
Date	Date the tweet was published
Username	The user who made the tweet
Content	The text the user wrote
Retweets	The number of retweets attained
Likes	The number of likes attained
Replies	The number of tweets responding to this one.
Quotes	The number of quote tweets attained.

I did not end up using every single one of these columns in the final program. The columns that are of most interest are the Date and the Content; the other columns were gathered in case I planned to do other forms of analysis. I used dataframes over other data structures like lists or arrays since dataframes can be manipulated with commands similar to SQL (i.e. group by statements and other aggregate functions) which was extremely valuable when manipulating the

data for graphing, and dataframes can be manipulated with operators such as multiplication and addition across rows or across columns which was useful when averaging rows together. Being able to use aggregate functions on data was extremely valuable when I needed to find the number of tweets in the sentiment category on specific dates.

3.0 Approach, Techniques, and Algorithms

My approach can be split into four sections: data collection, classification with Naïve Bayes, classification with VADER, and graphical displays.

3.1 Data Collection

As stated in the previous section, all the data used in this project was scraped from Twitter using SNScrape and then saved to CSV files so that they could be reuploaded into dataframes. I have already stated my justifications for using dataframes in the previous section, so I will not be repeating them here. The reason why I used the SMS scrape library to scrape the data from Twitter instead of using the official Twitter API to download tweets like what people usually do with projects like this is that I had issues creating a Twitter developer account. Since the reasons why I had difficulties creating a developer account are not relevant to the project I will not be discussing that here. I also did not want to use the official Twitter API since there seems to be a limitation on the number of tweets you can download at one time as well as how far back you can download tweets from. By avoiding the Twitter API and scraping the data instead, I was able to avoid the download limit as well as the historical date range limit. The dataset I downloaded is unbalanced; however, in the context of my project, it does not make sense to have an equal number of positive negative and neutral tweets either altogether or on every single day since there would be no point in analyzing how sentiment changed over time. Approximately 400 of the tweets I downloaded had their sentiment (positive, negative or neutral)

classified by me for the Naïve Bayes Classifier. My tweets to dataframe function appears to have an approximate runtime of O(n).

Algorithm Pseudo-Code:

for each tweet returned by the scraper query:

    if we have not reached the maximum number of tweets:

        add the tweet and the information about it we want to download to a list

return the list

3.2 Naïve Bayes Classifier with Traditional NLP Techniques

One of my personal goals for the project outside of the actual project objective was to learn how natural language processing is done without a tool such as VADER. I chose to use the Naïve Bayes classifier along with traditional NLP techniques such as lemmatization, removing stop words, tokenization, and removing noise in order to learn and experience these traditional techniques. I used a small slice of my data set of about 400 tweets to use with this method. As stated previously I had to classify these 400 tweets myself into one of the three sentiment categories, which may have introduced some form of bias or error with respect to this approach. While using this approach I heavily referenced Digital Ocean's sentiment analysis tutorial so a lot of the code in this section is very similar to the tutorial (Daityari, 2019). I chose to use the Naïve Bayes classifier since I learned about it while taking ENSE 412 this semester and it is a good multi-class classifier that does not need a lot of data to be adequately trained due to the usage of Bayes' Classifier. Since I could not assume that my data was linearly separable, I was not able to use Support Vector Machines. Nearest Neighbors with clustering might have worked, but in order to get it to work the way that I needed it to, I would have had to have made it work in three dimensions since I would need to be able to take into account positive, negative, and

neutral sentiment. The runtime of functions such as remove_noise() and lemmatize_sentence() appears to be O(n).

Lemmatize Sentence Algorithm Pseudo-Code:

for each word or speech type tag:

    Determine if the word is a noun, action, or something else

    Lemmatize the word and append it to a list

return the list

3.3 VADER Classification

I chose to use VADER to classify the entire data set since according to the documentation the natural language toolkit implementation of VADER does not require any form of preprocessing to be done to the text. Preprocessing the text looks like it would tamper with VADER sentiment scores since the non-root form of words and emoticons affect the sentiment score (NLTK Team, 2023). Not having to preprocess the text made it faster to get the overall sentiment for every tweet. I chose to use VADER since other sentiment analyzers like Textblob do not compute separate scores for positive, negative, and neutral sentiments (ES, 2023). The algorithm that I use to get the VADER sentiment scores for every tweet appears to have a runtime of O(n).

Algorithm Pseudo-Code:

for every tweet that needs to be analyzed:

    get the VADER polarity score for each category

    determine which polarity score was largest to get the overall sentiment category

    append the raw polarity scores as well as the overall sentiment to the dataframe of tweets

3.4 Graphical Displays

To create all the graphs seen in the notebook, I used Matplotlib. Matplotlib is a very standard plotting library for Python that can be used with NumPy. I have had to use Matplotlib in labs for ENSE 412, so I was comfortable using this library for this project. I was planning on making an interactive dashboard; unfortunately, I ran out of time and was unable to finish the dashboard. I did not use Seaborn since I did not need any of the extended functionality it provides; the standard Matplotlib library was enough for this project.

4.0 Structural Diagram and Explanation

As stated previously my project can be split into four sections or modules. The first module is the ‘Data Collection’ module which scrapes Twitter for the most recent 300,000 containing Ms. Swift’s username or are a reply to one of her tweets. The second module is the ‘Naïve Bayes Classification’ module which uses a 400-tweet slice of the dataset and uses traditional NLP processes such as preprocessing the tweets using lemmatization and noise removal then classifies the tweets using a Naïve Bayes Classifier. The third module is the ‘VADER’ module which classifies all 600,000 tweets in the data set without doing any preprocessing. The ‘Graphs’ module uses the VADER analysis results and plots various graphs to analyze and interpret.

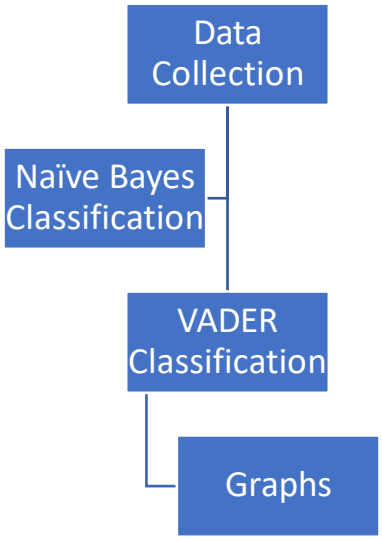


Figure 1 Structural Module Diagram

All the modules operate independently; however, most of the modules rely on data made by the previous module. That being said, the Naïve Bayes module could be removed from the final system without hampering the operation of the final two modules.

5.0 User Manual

If a user simply wants to view the project and my results, they can open the project file and view all the code and outputs without needing to run the code at all. If they chose to run it, this project can be run using Jupyter. I would also recommend downloading Anaconda to download a lot of common data science libraries such as numpy, pandas, and matplotlib which are required for this project. Before attempting to run the notebook, the user needs to install Python, Pip, and a couple of Python. Specifically, the user needs to install SNScrape (if they plan to download their own dataset), the Natural Language ToolKit, TextBlob, and Twython which can be directly installed via Jupyter by running the following commands:

- !pip install snscrap
- !pip install nltk
- !pip install Textblob

- !pip install twython

I strongly suggest that users download the datasets that I have already downloaded to save time. To ensure the project works, the Python Notebook containing the project code should be in the same folder as the CSV files containing all the data. The user can then run through the notebook one cell at a time skipping the cells under the 'Data Collection' header if they plan to download their own data. One thing to note is the graphs after the 'All Tweets Graphs' section might not show anything if they do not download any data in those date ranges.

6.0 Sample Sessions

The output of the Naïve Bayes Classifier including the most used tweets, accuracy, and most informative features can be seen below:

```
most common words [('taylorswift13', 178), ('http', 32), ('love', 27), ('’', 24), ('please', 22), ("'s", 18), ('wou
ld', 16), ('taylor', 16), ('tour', 15), ('time', 13)]
328

Accuracy is: 69.6969696969697
Most Informative Features
                  people = True           Negati : Positi =     11.7 : 1.0
                     n't = True           Negati : Positi =     11.1 : 1.0
                  please = True           Positi : Negati =      9.7 : 1.0
                      .. = True           Negati : Positi =      9.7 : 1.0
                    tour = True           Positi : Negati =      8.6 : 1.0
                     ... = True           Negati : Positi =      8.5 : 1.0
                   thing = True           Negati : Positi =      7.2 : 1.0
                     say = True           Negati : Positi =      6.4 : 1.0
                    know = True           Negati : Positi =      6.4 : 1.0
                    drug = True           Negati : Positi =      6.4 : 1.0
```

Figure 2 Naive Bayes Classifier Output

The total number of positive, negative, and neutral tweets as classified by VADER can be seen below:

```
total number of positive tweets:  308790
total number of negative tweets:  73893
total number of neutral tweets:  217317
```

Figure 3 VADER Classification Output

Various graphs that came from the VADER output such as a graph of all the tweets in each sentiment category and a graph of the sentiment percent of each category per day can be seen below:

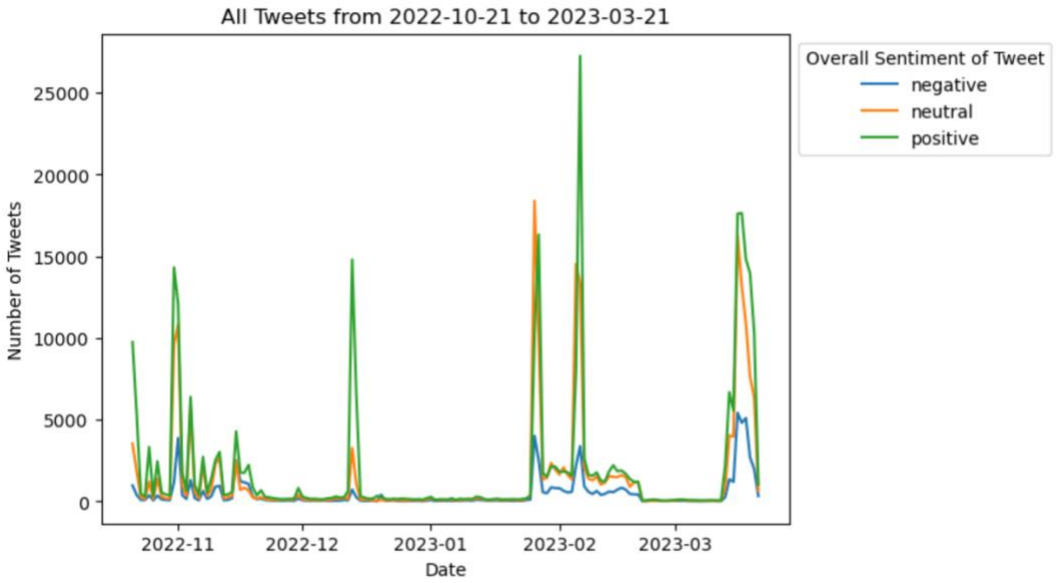


Figure 4 All Tweets in the Sentiment Categories Graph

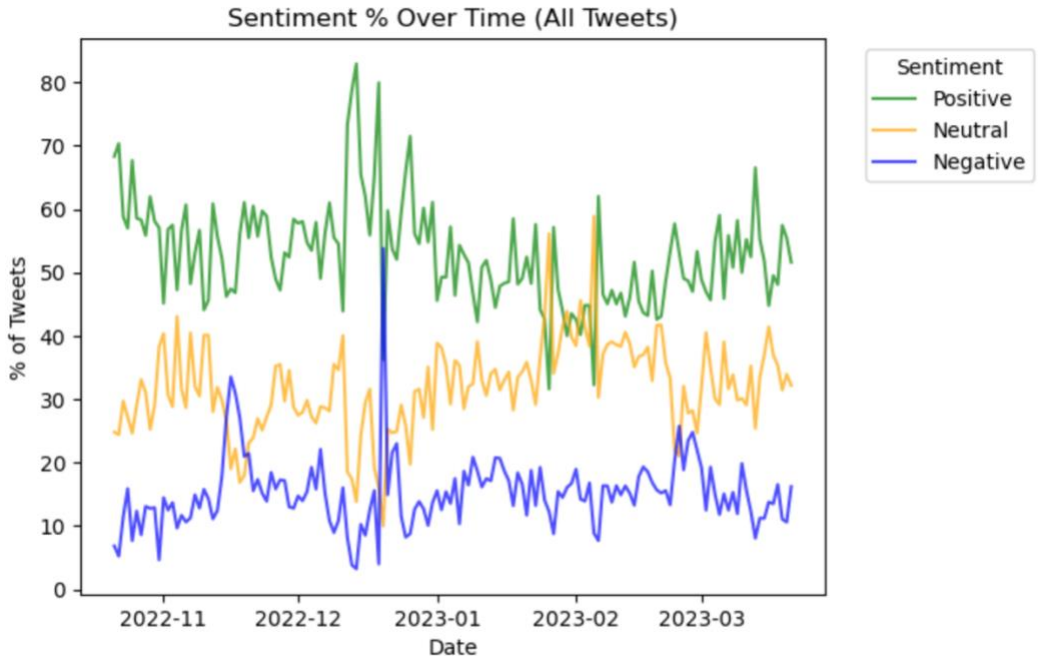


Figure 5 Sentiment Percentage per Category per Day Graph

7.0 Sample Listing

An example of well-documented code is the VADER classification code:

Function Name: (Technically not a function) VADER Classification Code

Input Parameters: tweets_list (a list of all the tweet content in the dataframe), tweets (the dataset dataframe)

Output Parameters: positive_list, negative_list, and neutral_list (lists to contain tweets in their respective sentiment categories), scores (a list to hold the polarity scores), overall (a list to hold the overall tweet sentiments), the tweets dataframe will be updated with the scores and overall sentiment appended

Description:

1. Takes in a list of all the tweet content in the dataset.
2. Runs through each tweet individually to determine polarity scores for each category and the overall sentiment for the tweet content and appends those to the dataset dataframe.
3. Prints the percentage of tweets in each sentiment category.

```
positive_list = []
negative_list = []
neutral_list = []

tweets = df_merged
tweets_list = tweets['Tweet']
numTweets = len(tweets_list)
```

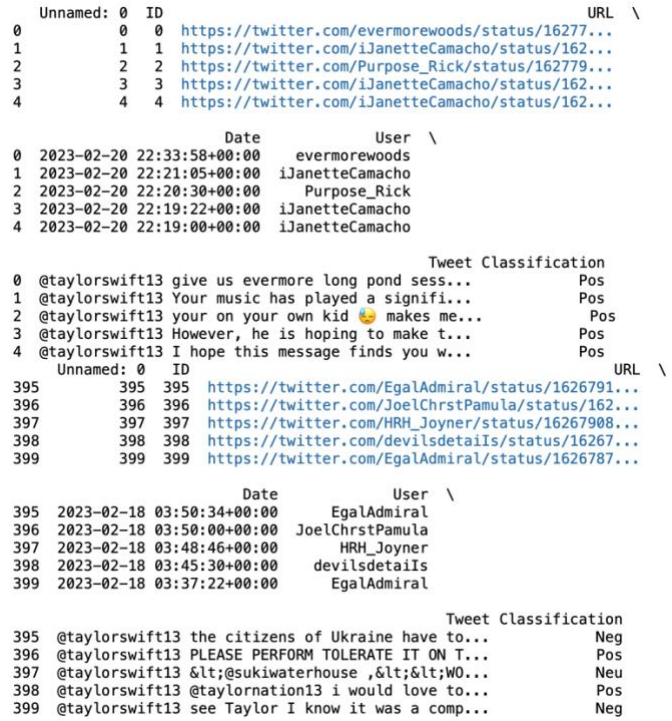
```
# takes ~1 hour to run with 600,000 tweets
scores = []
overall = []
for tweet in tweets_list:
    score = SentimentIntensityAnalyzer().polarity_scores(tweet)
    scores.append(score)

    # get the score for each category
    pos = score['pos']
    neg = score['neg']
    neu = score['neu']

    # determine how to categorize the overall tweet sentiment
    if neg > pos:
        negative_list.append(tweet)
        overall.append('negative')
    elif pos > neg:
        positive_list.append(tweet)
        overall.append('positive')
    else:
        neutral_list.append(tweet)
        overall.append('neutral')

# append the sentiment scores, overall sentiment, and fix the dates for graphing purposes
tweets['scores'] = scores
tweets['overall_sentiment'] = overall
tweets['Date'] = pd.to_datetime(tweets['Date']).dt.date

# print the percentage of positive, negative, and neutral tweets in the dataset
print('positive', len(positive_list) / numTweets * 100)
print('negative', len(negative_list) / numTweets * 100)
print('neutral', len(neutral_list) / numTweets * 100)
```

Figure 6 VADER Classification Code

8.0 Discussion

I have not accomplished anything new with respect to machine learning and lexicon-based analysis; the purpose of my project was to apply already known techniques to answer a research question. Something that I noticed I did differently compared to the July 2022 sentiment analysis project was that I analyzed tweets that mention Ms. Swift's Twitter account or are direct replies to her instead of only analyzing tweets containing her full name (Sharma, 2022). The biggest pro with my project approach is that sentiment analysis is a well-established form of sentiment analysis so there are multiple online tutorials and papers I was able to reference when I was stuck on part of my project. The VADER method was a huge pro since it allowed me to classify all 600,000 tweets in under an hour and then I was able to move on to making graphs

and interpreting them. The biggest con with my project was that I had to gather the data myself. Due to the specificity of the time-period and the nature of tweets I was looking to collect, I had to collect the tweets on my own. This was extremely time intensive and even classifying 400 tweets for Naïve Bayes took a long time since I was trying to do my best to objectively categorize each tweet. Despite doing my best to be objective, I still introduced some form of bias/error since the accuracy of the Naïve Bayes classifier was 69% which is quite poor. I also feel like I did not do the best job of trying to filter out noise tweets in my dataset. As we can see in the figure below:



	Unnamed: 0	ID	URL	Date	User	Tweet	Classification
0	0	0	https://twitter.com/evermorewoods/status/16277...	2023-02-20 22:33:58+00:00	evermorewoods	@taylorswift13 give us evermore long pond sess...	Pos
1	1	1	https://twitter.com/iJanetteCamacho/status/162...	2023-02-20 22:21:05+00:00	iJanetteCamacho	@taylorswift13 Your music has played a signifi...	Pos
2	2	2	https://twitter.com/Purpose_Rick/status/162779...	2023-02-20 22:20:30+00:00	Purpose_Rick	@taylorswift13 your on your own kid 😩 makes me...	Pos
3	3	3	https://twitter.com/iJanetteCamacho/status/162...	2023-02-20 22:19:22+00:00	iJanetteCamacho	@taylorswift13 However, he is hoping to make t...	Pos
4	4	4	https://twitter.com/iJanetteCamacho/status/162...	2023-02-20 22:19:00+00:00	iJanetteCamacho	@taylorswift13 I hope this message finds you w...	Pos

	Unnamed: 0	ID	URL	Date	User	Tweet	Classification
395	395	395	https://twitter.com/EgalAdmiral/status/1626791...	2023-02-18 03:50:34+00:00	EgalAdmiral	@taylorswift13 the citizens of Ukraine have to...	Neg
396	396	396	https://twitter.com/JoelChrstPamula/status/162...	2023-02-18 03:50:00+00:00	JoelChrstPamula	@taylorswift13 PLEASE PERFORM TOLERATE IT ON T...	Pos
397	397	397	https://twitter.com/HRH_Joyner/status/16267908...	2023-02-18 03:48:46+00:00	HRH_Joyner	@taylorswift13 <@sukiwaterhouse ,<<WO...	Neu
398	398	398	https://twitter.com/devilsdetaiIs/status/16267...	2023-02-18 03:45:30+00:00	devilsdetaiIs	@taylorswift13 @taylornation13 i would love to...	Pos
399	399	399	https://twitter.com/EgalAdmiral/status/1626787...	2023-02-18 03:37:22+00:00	EgalAdmiral	@taylorswift13 see Taylor I know it was a comp...	Neg

Figure 7 Some of the Tweets Collected

Some of the tweets I collected have nothing to do with Ms. Swift. When I was manually classifying some of the data, I noticed there were tweets focused on religion, the Ukraine crisis, and the Toronto opioid crisis that got included since they mention her Twitter account. If I had time, I would have liked to have done a topic analysis and removed all the irrelevant tweets.

I would say that my biggest accomplishment was that I was able to complete this project at all. This was my first project involving AI/Machine Learning and although it is a little rough around the edges, I am proud of it. The usage of the VADER sentiment analyzer to classify all the tweets in a timely manner greatly helped so that I could spend more time trying to visually analyze the graphs. My initial hypothesis did not directly correspond to the massive peaks in the data; however, I was still right about the sentiment being positive before the pre-sale, then becoming more negative during the presale, then becoming a bit more positive during the opening shows. So to answer my initial question, her sentiment was overall positive, then shifted negatively during the presale, and started to become more positive during the opening shows. I think my biggest weakness was not filtering out some of the noisy tweets in the dataset.

According to the previous discussion, I feel like I somewhat proved my hypothesis; however, the dates in my hypothesis do not directly correspond with peaks in the positive sentiment in the data. After brainstorming for a bit, I was able to come up with a list of events that may have caused the positive number of tweets to spike over 10k:

- October 2022: Taylor Swift released her 10th studio album Midnights
- November 2022: Taylor announces the Eras Tour and ticket presale date occurred
- December 2022: Taylor's birthday is December 13th
- Late January 2023: LiveNation/Ticketmaster USA senate hearing resulting from the Eras ticket presale fiasco
- Early February 2023: Record Store Day Exclusive of a live performance of her Folklore album was announced
- March 2023: Eras tour begins March 17th in Glendale, Arizona

I still have no idea what caused that massive spike in negative sentiment in December 2022.

9.0 Conclusion

I went into the project knowing nothing about natural language processing and was able to come out with a working project! I learned a lot about and was able to apply traditional NLP techniques such as lemmatization and removing stop words and was able to use a lexicon-based sentiment analyzer to categorize all the data I collected. I was then able to plot graphs of that data and analyze it in order to answer my initial question. The answer to the question is that her overall sentiment was largely positive after releasing her album, *Midnights*, and became a bit more negative/neutral after the presale of the Eras Tour tickets, then it started to become more positive leading up to the opening tour shows. I did not collect quite enough data to answer how the opening shows actually affected her public sentiment. Overall, I would say this project was successful and I am happy with how it turned out. If given the opportunity, I would love to take the time to work on filtering the data better and redo some parts of the project.

10.0 Future Work

Something that could have benefited this project would be collecting tweets about topics related to Taylor Swift, but that would require an in-depth topic analysis which I did not have time for. In the future, I would recommend filtering the dataset to remove unrelated tweets so that some of the noise is removed. If I had the time, I would have really liked to have made the interactive dashboard since it would have brought the project to life with user interactivity. Something that would be cool would be making a more general version of this project that can take a topic as an input, maybe even determine other relevant topics, gather tweets about that topic, and do a sentiment analysis on that topic. A general version of this project that functions like that would be cool to monitor current events happening in the world. I feel like the overall

approach I took to this project worked out well and I would recommend using it again in future projects.

Annotated References

Daityari, S. (2019, September 26). *How To Perform Sentiment Analysis in Python 3 Using the Natural Language Toolkit (NLTK)*. Retrieved March 2023, from Digital Ocean: https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk

This was the tutorial I followed to perform the sentiment analysis using Naïve Bayes and built-in natural language toolkit functionality.

ES, S. (2023, January 30). *Sentiment Analysis in Python: TextBlob vs Vader Sentiment vs Flair vs Building It From Scratch*. Retrieved March 2023, from neptune.ai: https://neptune.ai/blog/sentiment-analysis-python-textblob-vs-vader-vs-flair

This was the reference that I used to figure out if I should use VADER or another lexicon such as TextBlob and it led me to tinker with TextBlob and VADER before ultimately settling on using VADER.

NLTK Team. (2023, January 2). *Sample usage for sentiment*. Retrieved April 2023, from NLTK: https://www.nltk.org/howto/sentiment.html#vader%5D

This is a link to a section in the Natural Language Toolkit documentation that is about sentiment analysis. I referenced this in order to find out how to use VADER when I decided to use it.

Sharma, A. (2022, June 29). *Analyze Twitter's Reaction to Taylor Swift with HarperDB*. Retrieved April 2023, from Medium: https://medium.com/@aakriti.sharma18/analyze-twitters-reaction-to-taylor-swift-with-harperdb-6207a50aee5d

When I was initially researching Taylor Swift sentiment analyses, this is one of many that popped up. Despite sounding like similar projects, our projects are not the same, but it did inspire me to do my project and attempt to analyze the Eras Tour's impact on Ms. Swift's public sentiment.