SOCIAL MEDIA SENTIMENT ANALYSIS

Social Web Analytics – Final Project

JOSHUA BENZON (BS20DSY026)
OMKAR SHITOLE (BS20DMU021)
GAVRIEL OWENS VINCENTIO (BF21SDY004)
PRAKHAR MISHRA (BS20DMU19)
GARGI ASTHANA (BS20DMU002)

Contents

I. Abstract

The main purpose of this project is to analyse the sentiment of two given TV shows. Such an analysis can help potential marketers or producers gain an understanding of the overall impact of the show on its viewers (especially in comparison to others shows) and make improvements accordingly. It can also help in developing marketing strategies like targeting the right market segments, maintaining social trends with the most appeal etc. For the purposes of demonstrating this project, the process has been conducted on the basis of the following two shows: “Game of Thrones: House of the Dragon” and “Lord of the Rings: The Rings of Power”. The platform from which the sentiments were extracted is Twitter, and the tools used are all libraries of the programming language Python.

Introduction

This project aims to extract the data regarding the two shows “Game of Thrones: House of the Dragon” and “Lord of the Rings: The Rings of Power” from Twitter and use the extracted data to analyse the overall sentiment of each show, in order to draw meaningful comparisons and insights. The project achieves outcomes by implementing the following procedures:

1. Data Scraping: It is the process of using code or other software to extract useful information from online websites. It can have various applications, from data mining to market research etc. Here, its primary usage is for market research (in tandem with the following procedures). For the purposes of this project, data scraping was implemented in Python using the *‘snsrape’* library, as it allows an easy gateway into accessing large volumes of data from websites (here, Twitter).

2. Data Wrangling/Cleansing: This involves quality control and manipulation of data to gain meaningful insights. This can include establishing the right quality of data, feature extraction, filtering, aggregation, etc. In the context of this project, the Python library *‘pandas’* is primarily used, along with the library *‘re’*, in order to prepare the raw data extracted by *snscrape* for further sentiment analysis.

3. Sentiment Analysis: Also known as opinion mining, this process is for analysing the sentiment or emotion of a given piece of text. It does so by quantifying the emotion based on certain specified parameters, and it utilises natural language processing (NLP) for the procedure. Sentiment analysis is widely used for market research, as it is here, through the Python library ‘vader’.

4. EDA: EDA, or exploratory data analysis, refers to extensive research and study of given data, finding meaningful takeaways, developing insights, testing hypotheses, and reaching conclusions. It is an integral part of most research-related projects. Here, EDA is performed on the data by exploiting four Python libraries, namely, ‘geopandas’, ‘seaborn’, ‘matplotlib’ and ‘wordcloud’.

II. Problem Background

In the year 2022, two shows namely, Lord of the Rings: The Rings of Power and House of the Dragon were released. While the shows have reviews and audience satisfaction data readily available on websites such as Rotten Tomatoes and IMDb, we would like to further look into the emotions of the audience in a subjective manner and explore the sentiment expressed in the form of lexical using data from twitter. We also investigate the trend of emotions and activity one month prior to the show release, during the ongoing episodes and after the completion of the series.

III. Technologies Used

The social media platform we used for the project was Twitter. Twitter is one of the biggest social media websites to date, and a lot of pop culture discourse is conducted on this platform, making it a well-suited platform for sentiment analysis.

The Python programming language also proved to be ideal for this project. Apart from being the industry standard for tasks related to data mining, sentiment analysis, etc., its simplicity makes implementation very easy. Also, due to it being so widely used, support is readily available everywhere. Python also possesses a vast and varied range of libraries to suit most coding needs.

Within Python, the following libraries were used:

1. 'snscrape': Used for web scraping, i.e., extracting data from tweets from Twitter. 'snscrape' in particular has quite a few advantages relative to its competitors (such as 'tweepy' or 'twint'). For example, it avoids blocking by creating new random user agents. Empty pages fail safes and cursor pagination also come included within the library. However, above all, its most useful feature is its ability to scrape beyond the specified API limits.
2. 'pandas': This library is the standard for data wrangling, cleansing, and exploration. It possesses a robust set of features that make it easy to manipulate large volumes of data with relatively fast speeds.
3. 'numpy': Useful for performing fast operations on large quantities of data. It can also perform some more complex functions compared to Python's inbuilt ones.
4. 're': Used for passing in Regular Expression (RegEx) statements. Useful for using RegEx for string filtering, pattern matching etc.
5. 'datetime': Useful for converting separate date and time values into a single datetime feature.
6. 'vader': Also known as 'Valence Aware Dictionary for Sentiment Reasoning', this library is used for sentiment analysis. It is particularly useful in that it is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. Also, along with simple text, it also considers lexical features and emoticons. These additional accommodations provide 'vader' with an edge beyond other tools used for sentiment analysis (like 'nltk').

7. ‘geopandas’: A specialisation of the ‘pandas’ library that allows it to deal with geospatial data.
8. ‘seaborn’: A powerful tool used for plotting and graphing.
9. ‘matplotlib’: Another widely used library for creating plots and charts.
10. ‘wordcloud’: Used for creating word clouds.
11. ‘multiprocessing’:
12. ‘warnings’:

IV. User Manual

1. Data Scraping

It is the process of using code or other software to extract useful information from online websites. For the purposes of this project, data scraping was implemented in Python using the *‘snsrape’* library, as it allows an easy gateway into accessing large volumes of data from websites (here, Twitter). ‘snscrape’ has a variety of useful features: it avoids blocking by creating new random user agents, includes empty page fail safes and cursor pagination, and most importantly, it can scrape limitlessly (by bypassing API limits). It can even potentially scrape other social media platforms as well. Such features form the primary reason why this library was chosen for the project rather than ‘tweepy’ or ‘twint’ for scraping user tweets from Twitter.

The code performs data scraping in the following steps:

1. Imports necessary modules.
2. Defines a function for extracting a specified list of details from each tweet and storing it in a DataFrame.
3. Allocates a specified number of CPU Threads for faster processing.
4. Defines functions for setting a date (for querying in a certain timeframe), as well as a function for retrieving the date.
5. Uses the functions defined previously for establishing queries and storing the specified information in a Pandas DataFrame, before sorting it and saving it as a CSV.

In this way, tweets related to the query subject (i.e., the name of the desired TV show) will be extracted and stored as a CSV for further processing. Refer to the code for corresponding actions for each step.

2. Data Wrangling

This involves quality control and manipulation of data in order to gain meaningful insights. In the context of this project, this technique is used for preparing the raw data (obtained from scraping Twitter) for further sentiment analysis.

For this project, the libraries utilised include ‘pandas’ (for tabulating and manipulating the data), ‘numpy’ (useful for some basic numerical and tabular operations), ‘re’ (allows usage of RegEx), and ‘datetime’ (converts date and time values into a single datetime value).

Data wrangling proceeds in the following steps:

1. Dropping duplicates.
2. Resetting index.
3. Delete irrelevant bits of string in the tweets (e.g.: ‘https’,’www’ etc.).
4. Set up ‘day’, ‘week’, ‘month’, and ‘date’ columns.
5. Add columns storing negative polarity, neutral polarity, positive polarity, compound score, and sentiment (based on the compound score) into the DataFrame for each show (refer to ‘3. Sentiment Analysis’ for more details on polarity, compound scores and sentiment).
6. Calculate the adjusted compound score and add it into each DataFrame.

3. Sentiment Analysis

For Sentiment Analysis, we choose VADER (Valence Aware Dictionary for Sentiment Reasoning) which is modelled based on qualitative analysis and empirical validation using a human-centric approach. It assesses the polarity and intensity of the emotion expressed in the text. A major strength which determined our choice to use VADER is the emoticon and emoji integration which plays an integral part in emotions that are expressed online.

VADER first analyses words which are given a score between -4 and 4. These scores are then put together using 5 major heuristics that perform sentiment analysis on a lexical, lexical – punctuations, capitalization, degree modifiers, polarity shift due to “but” and examination of trigrams. Finally it outputs a sentiment score between -1 and 1 where -1 is most negative and 1 is most positive. A score nearing 0 represents a neutral emotion. While polarity simply tells us how positive or negative a word is by numerically representing it between -1 and 1, a compound score is an aggregate score or a sum of the valence scores of all terms in a lexicon.

The sentiment has been adjusted to incorporate public reactions and opinions to tweets. So, tweets with more likes and shares hold a higher weightage as compared to others. The weight of a tweet is depreciated with number or quotes as this suggests a controversial opinion. The following formulae have been used to obtain an Adjusted Polarity.

$$Adjusted\ Polarity = \frac{Polarity\ \times (Like\ Count + Retweet\ Count + 1)}{(Quote\ Count + Reply\ Count) * 0.1 + 1}$$

Finally, we take an adjusted polarity ratio between the Positive Adjusted Compound Score and the Total Compound score which represents a total percentage of positive scores.

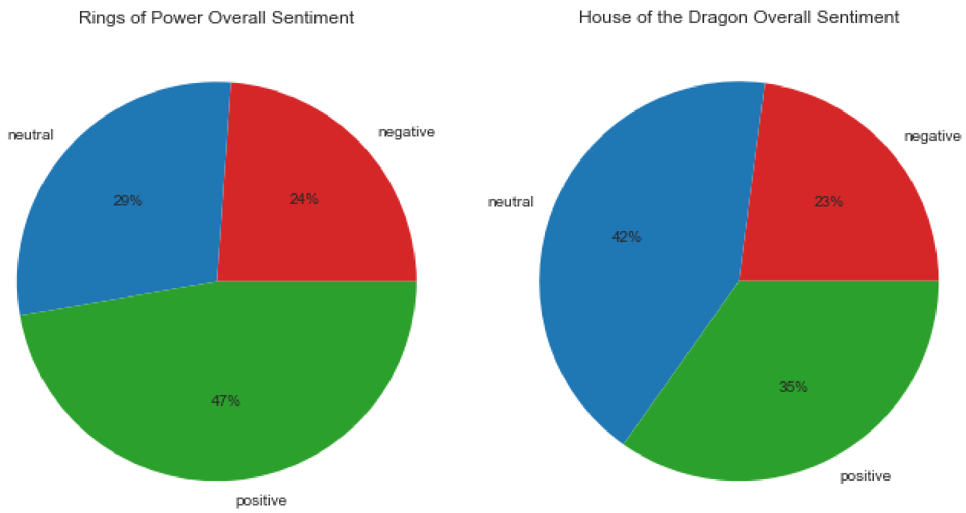
$$Adjusted\ Polarity\ Ratio = \frac{\sum |\ Positive\ Adjusted\ Compound\ Score\ |}{\sum |\ Adjusted\ Compound\ Score\ |}$$

4. Visualization

Now, let us investigate the results yielded and plot these insights to better understand the emotion of the audience. To visualise these results, we have used the following libraries:

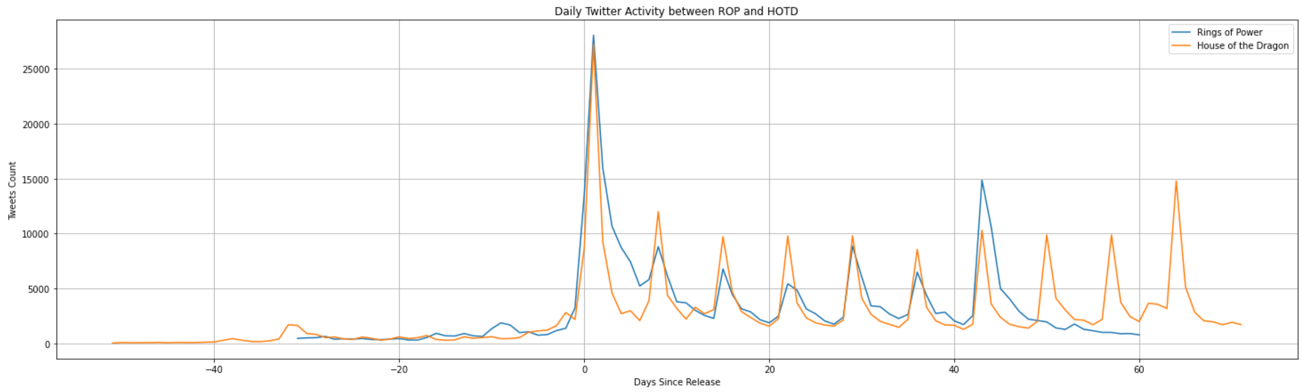
1. geopandas: this library is used to analyze geospatial data and gives us an accurate representation of the audience's location and region.
2. seaborn/matplotlib: these libraries were mainly used for making statistical graphs for comparative studies and trend analysis.
3. wordcloud: a wordcloud give us an overall picture of the words that are most popularly used

a) Overall Sentiment Ratio of both shows



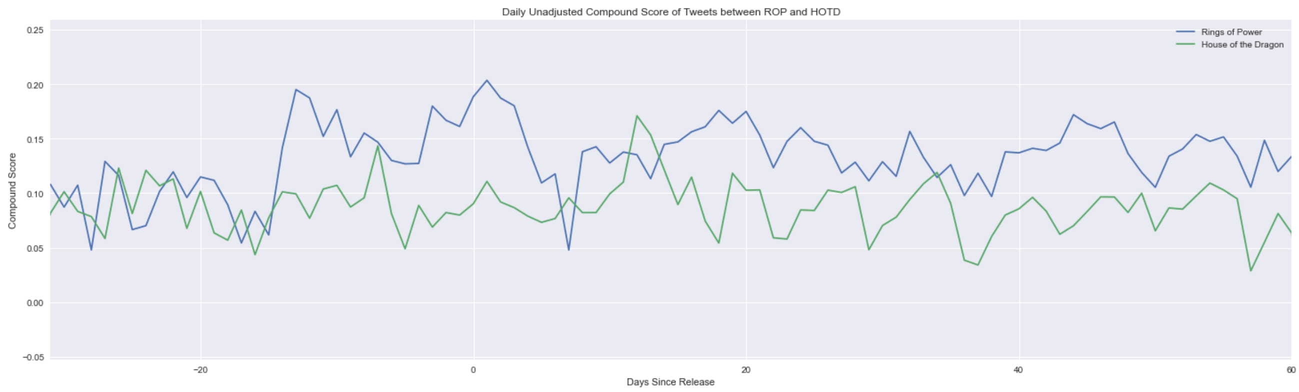
As we can see, Rings of Power has a more positive reaction as compared to the House of Dragon where reactions are balanced between positive and neutral.

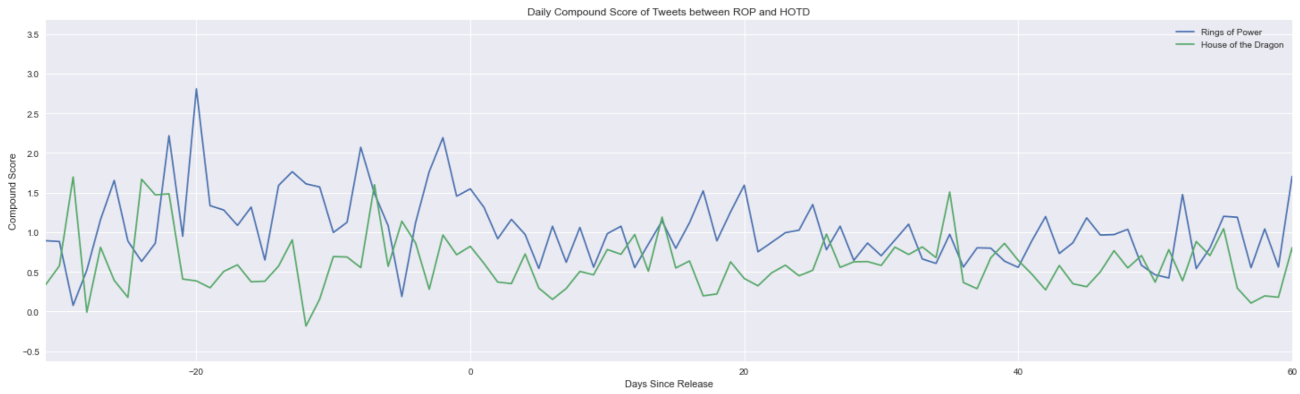
b) Twitter Activity (Counts) between Rings of Power and House of the Dragon



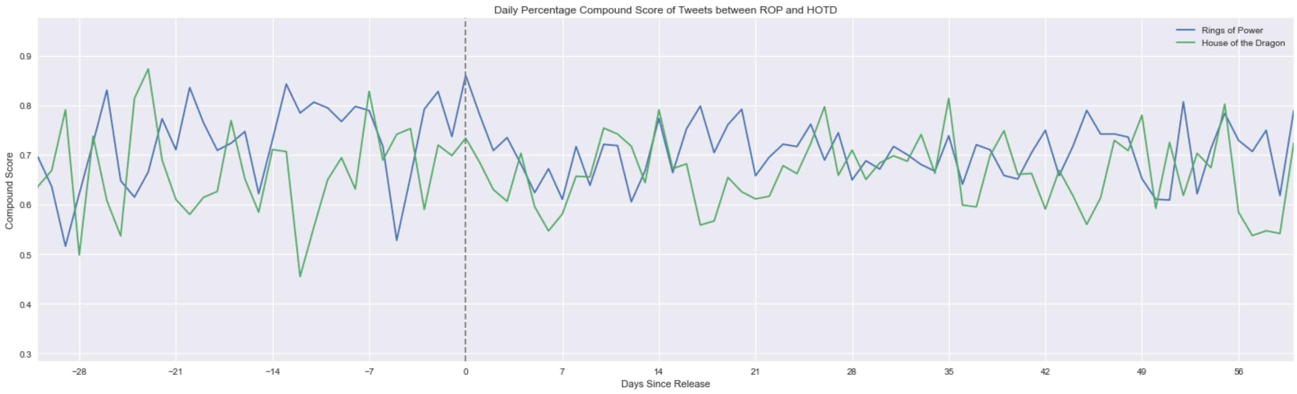
From the above graph we see that activity for both shows increases during release dates. A weekly spike indicates the weekly release of the episodes and as expected, we also see a rise in activity during season finales. House of Dragon activity is more constant while Rings of Power is less stable.

c) Before and After Adjustment of the Compound Score

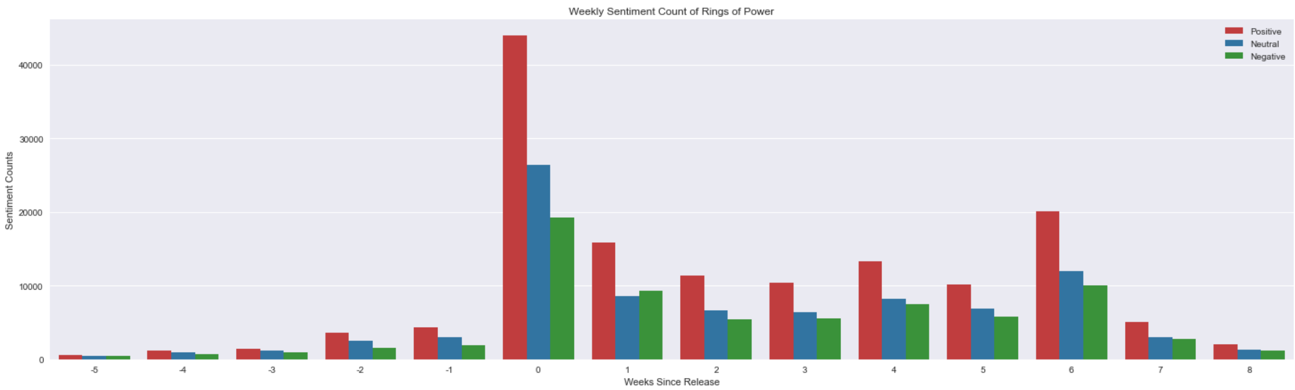


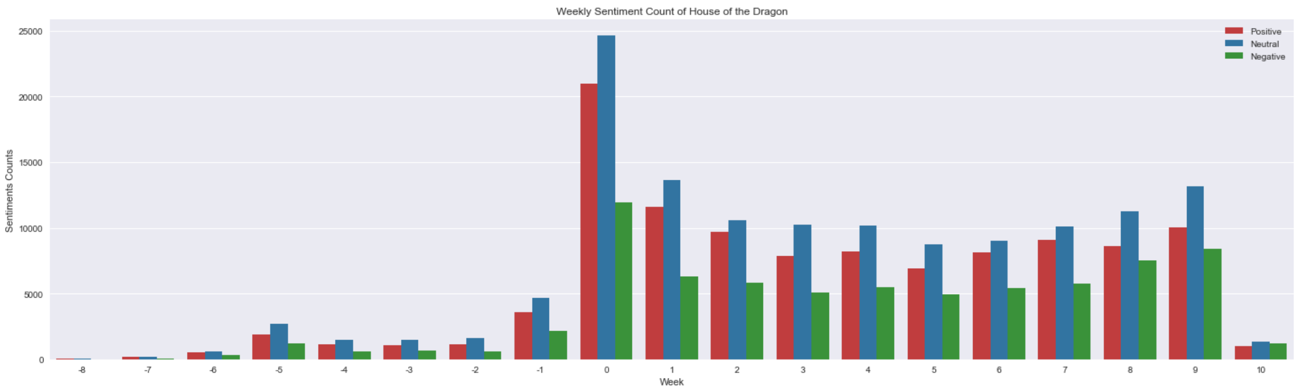


d) Positive Ratio of the Compound Score



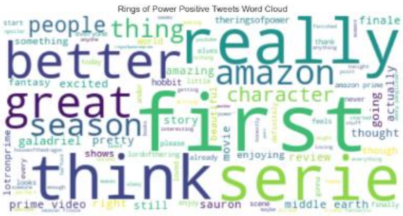
e) Weekly Sentiment Counts of Both Shows





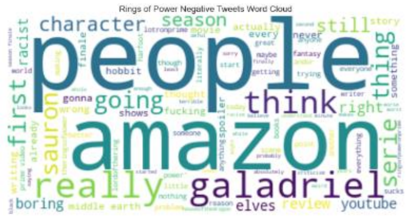
f) <u>Word Clouds</u>

Rings of Power



+

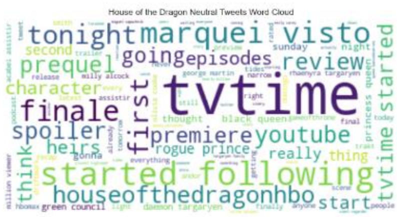
~



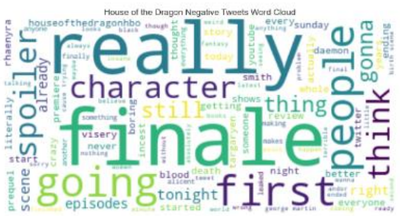
–

House of the Dragon

+



~

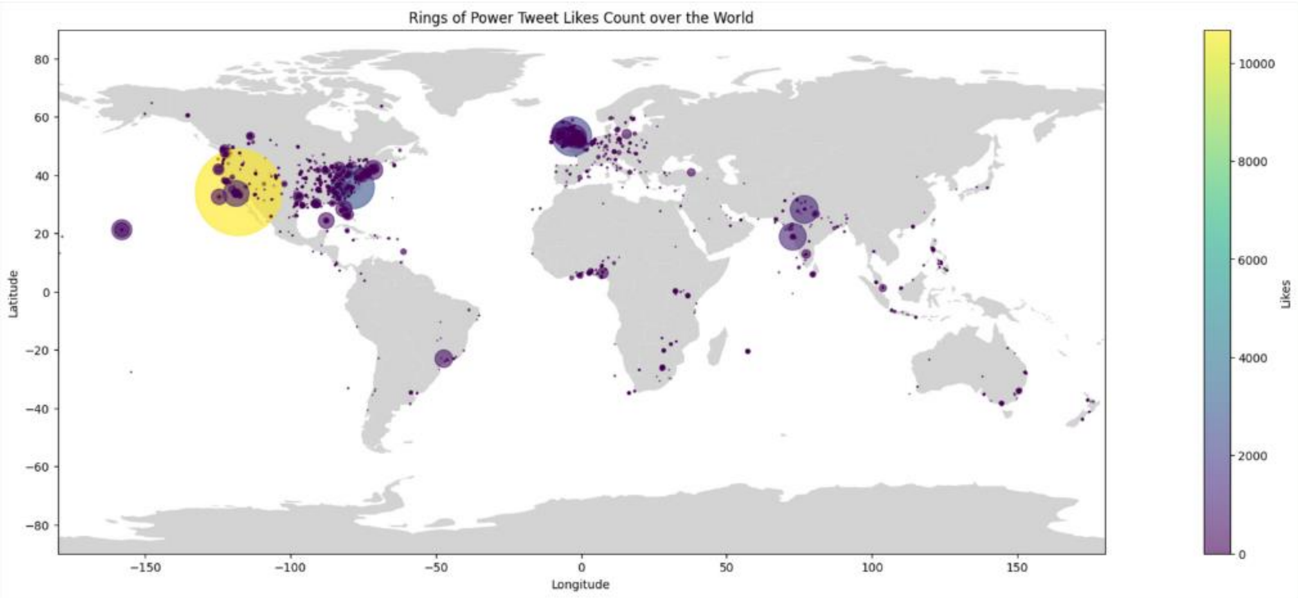


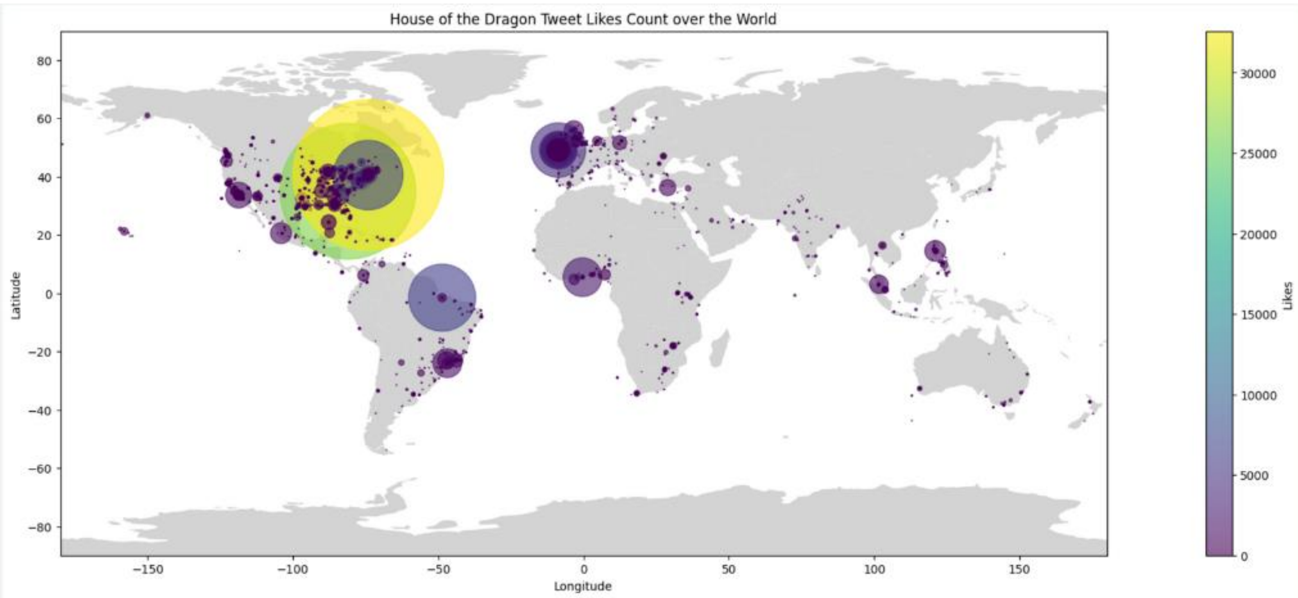
–

Here in the word clouds, we see a clear division of the words that correlate to the emotion. For Rings of Power we notice that a lot of the negative emotion stem towards terms such as Amazon and even characters such as Galadriel. Whereas in House of the Dragon, the negatives could be centric towards the finale.

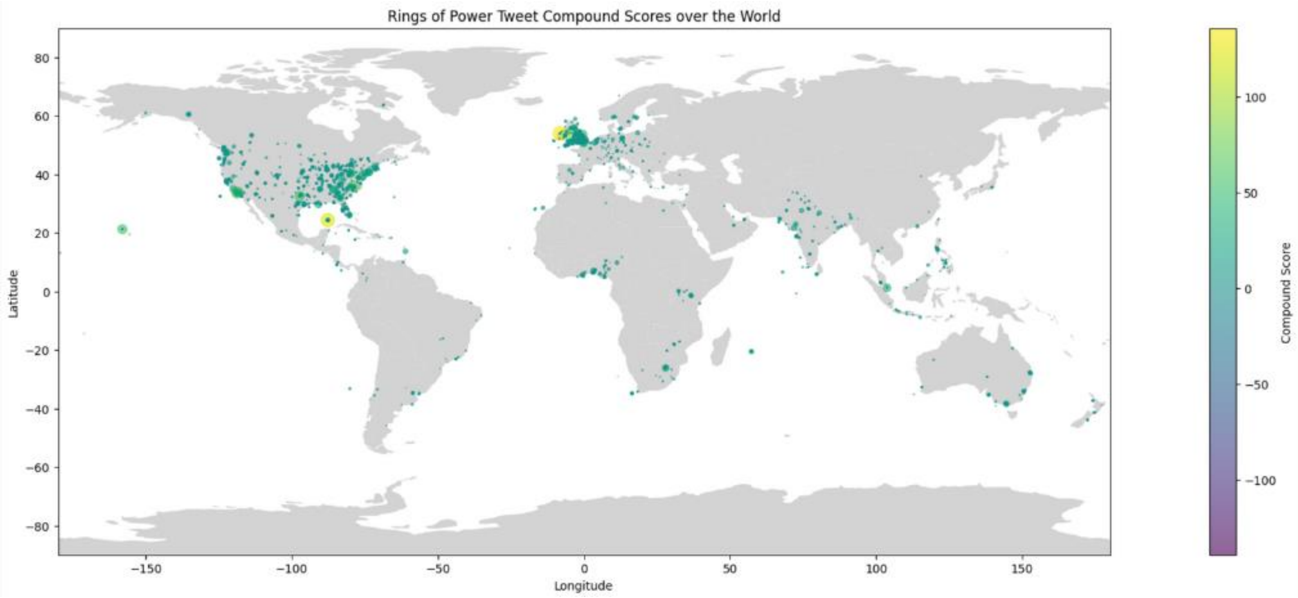
g) World Maps

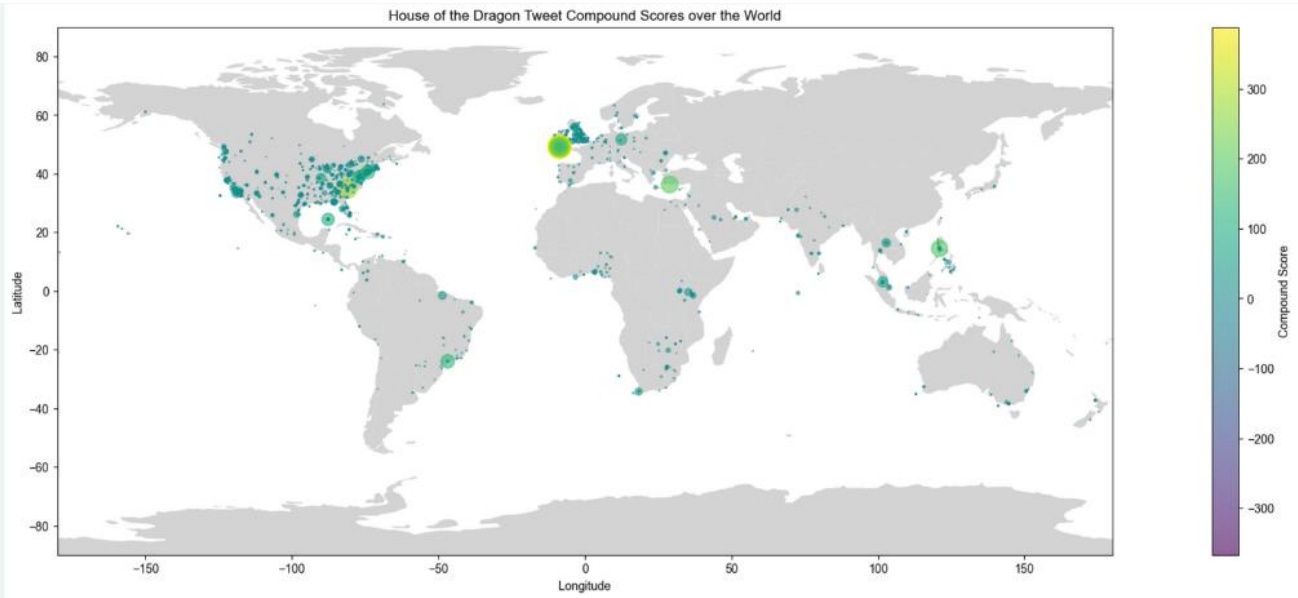
Worldwide Tweet Likes Count





Worldwide Compound Score





While the world maps represent most of the English speaking population due to our language filters, we do notice an activity trend between the east and

5. Multiprocessing

Web scraping involves retrieving extremely large volumes of data; as such, regardless of the performance or computational capabilities of a given computer, it can take a significant portion of time (up to an hour or even multiple hours) to finish retrieving it.

Multiprocessing is a technique that allows a machine with multiple CPUs or cores to fully exploit its computational power, by dividing a single task into several subtasks or subprocesses and running them simultaneously. It significantly cuts down processing time and increases overall performance and efficiency.

Here, the ‘multiprocessing’ library in Python enables multiprocessing by implementing it as follows:

1. First check for the number of CPU threads available. If it is less than 2, it allocates all the CPU threads available for scraping. If it is greater than 2, it will allocate 2 less than the total number of CPU threads.
2. Now, using the `Manager` from the library, it assigns the appropriate number of CPU threads and parallelises the process of scraping the tweets.

V. Testing

On testing to verify whether the DataFrames were created smoothly, we ran the following code:

Displaying Files

```
[4]: rop.head(5)
```

[4]:

	datetime	tweet_id	username	text	reply_count	retweet_count	like_count	quote_count	mentions	hashtags	...	month	dat
0	2022-08-31 23:59:56+00:00	1565127277850947585	philbertdy	i got sent the first two episodes of the rings...	1	2	46	5	NaN	NaN	...	8	2022-08-3
1	2022-08-31 23:59:47+00:00	1565127239141916675	SweetTalkBlog	the lord of the rings: the rings of power revi...	0	0	0	0	NaN	NaN	...	8	2022-08-3
2	2022-08-31 23:59:42+00:00	1565127218572959745	cinepacino	the rings of power tomorrow finally	0	0	1	0	NaN	NaN	...	8	2022-08-3
3	2022-08-31 23:59:32+00:00	1565127178710310912	johnettesnuggs	at a special fan screening of the first 2 ring...	2	0	5	0	NaN	NaN	...	8	2022-08-3

Data for “Lord of the Rings: The Rings of Power”

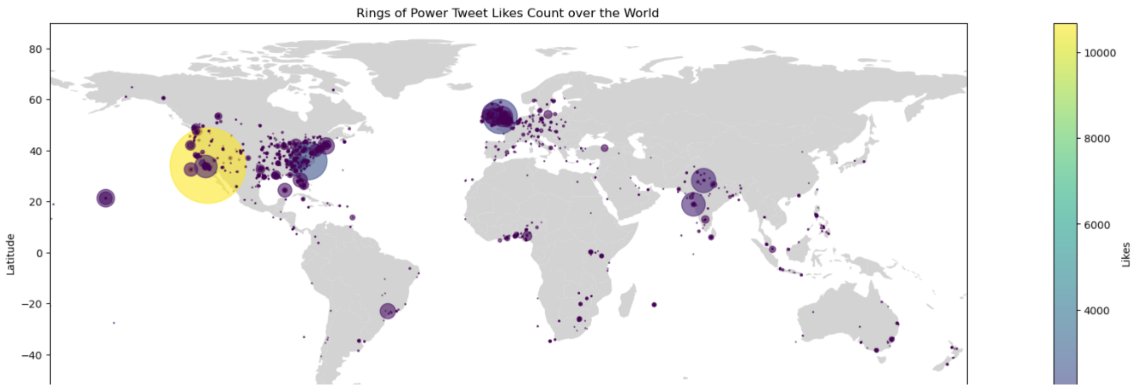
```
[5]:  hotd.head(5)
```

	datetime	tweet_id	username	text	reply_count	retweet_count	like_count	quote_count	mentions	hashtags	...	month	date
0	2022-08-31 23:59:59+00:00	1565127292040396801	nickhammy5	send alicent hightower to the wall (bonus: hou...	1	0	2	1	NaN	NaN	...	8	2022-08-31
1	2022-08-31 23:59:47+00:00	1565127239141916675	SweetTalkBlog	the lord of the rings: the rings of power revi...	0	0	0	0	NaN	NaN	...	8	2022-08-31
2	2022-08-31 23:59:19+00:00	1565127121558708229	GamingNews242	house of the dragon's miguel sapochnik leaving...	0	0	0	0	NaN	NaN	...	8	2022-08-31
3	2022-08-31 23:57:50+00:00	1565126749939179521	natalha______	finalmente vendo house of the dragon	0	0	0	0	NaN	NaN	...	8	2022-08-31

Data for "Game of Thrones: House of the Dragon"

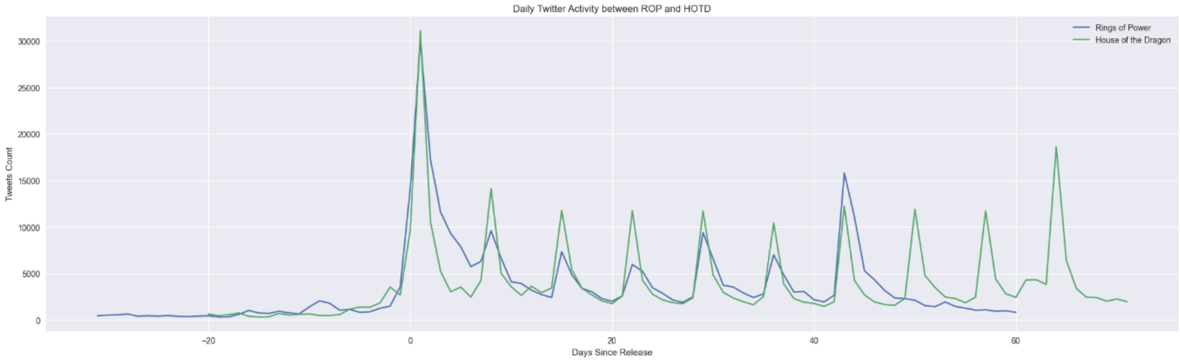
Beyond this, we also tested visualisation through different types of graphs and charts, including Geospatial graphs, line charts, bar graphs etc.; some of which are displayed below:

```
[6]:
rop_df = rop[rop['coordinates'].notna()==True]
rop_df.reset_index(inplace=True)

rop_df['longitude'] = [float(re.findall(r'[-+]?(?:\d{1,3}(?:,\d{3})+|\d+)(?:\.\d+)?', rop_df['coordinates'][i])[0]) for i in range(len(rop_df))]
rop_df['latitude'] = [float(re.findall(r'[-+]?(?:\d{1,3}(?:,\d{3})+|\d+)(?:\.\d+)?', rop_df['coordinates'][i])[1]) for i in range(len(rop_df))]

fig, ax = plt.subplots(figsize=(30,8))
countries = gpd.read_file(gpd.datasets.get_path("naturalearth_lowres"))
countries.plot(color="lightgrey", ax=ax)

plt.scatter(rop_df["longitude"], rop_df["latitude"], c=rop_df['like_count'], s=0.5*rop_df['like_count'], alpha=0.6)
plt.colorbar(label='Likes')

plt.xlim([-180, 180])
plt.ylim([-90, 90])

plt.title("Rings of Power Tweet Likes Count over the World")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.show()
```

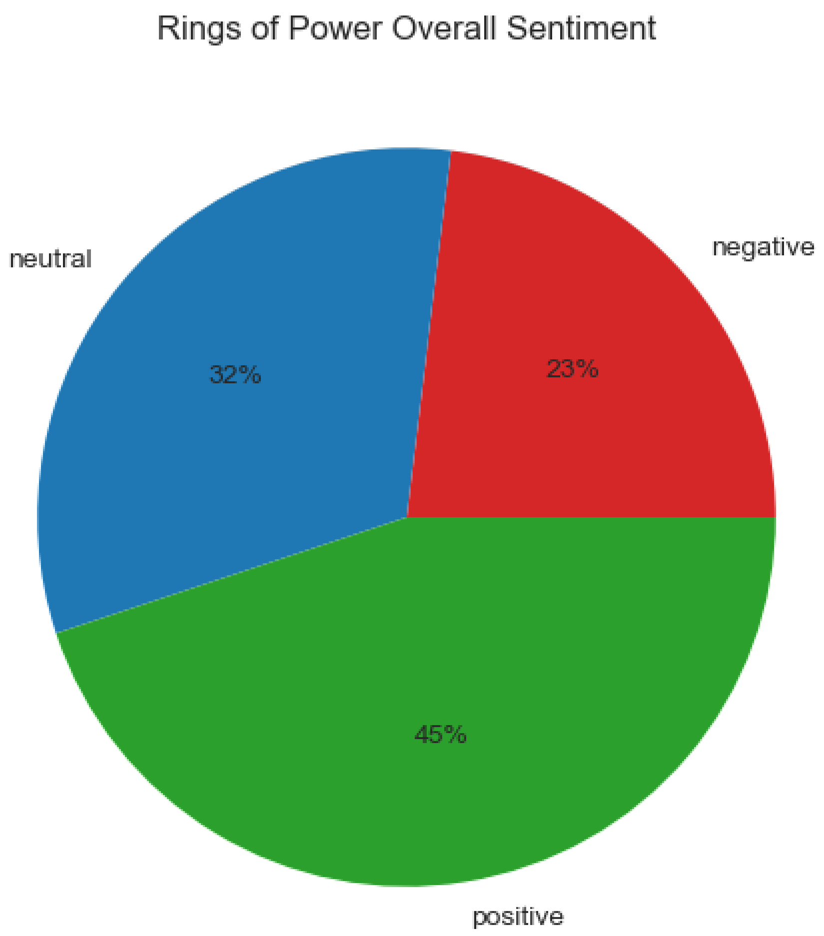


[10]:

```
rop_df = rop.groupby(['days_since_release'])['text'].count()
hotd_df = hotd.groupby(['days_since_release'])['text'].count()

fig, ax = plt.subplots(figsize=(25, 7))
rop_df.plot(x = 'days_since_release')
hotd_df.plot(x = 'days_since_release')
plt.grid(zorder=0)
plt.legend(['Rings of Power', 'House of the Dragon'])
plt.xlabel('Days Since Release')
plt.ylabel('Tweets Count')
plt.title('Daily Twitter Activity between ROP and HOTD')
plt.xticks()
plt.show()
```



```
[8]: plt.figure(figsize=(6,6))
plt.style.use('seaborn')
plt.pie(rop.groupby(['sentiment'])['sentiment'].count(),
        labels=['negative', 'neutral' ,'positive'],
        colors=['tab:red', 'tab:blue', 'tab:green'],
        autopct='%.0f%%')
plt.title('Rings of Power Overall Sentiment')
plt.show()

plt.figure(figsize=(6,6))
plt.style.use('seaborn')
plt.pie(hotd.groupby(['sentiment'])['sentiment'].count(),
        labels=['negative', 'neutral' ,'positive'],
        colors=['tab:red', 'tab:blue', 'tab:green'],
        autopct='%.0f%%')
plt.title('House of the Dragon Overall Sentiment')
plt.show()
```



Additionally, data from ‘The Last of Us’ was also scraped to test the generalisability of the code. This was retrieved successfully. The raw data is as follows:

	Datetime	Tweet ID	Username	Text	Reply Cou	Retweet C	Like Count	Quote Cou	Mentions	Hashtags	Coords	Link
2	2023-01-0	1.61E+18	KJFilmsTw	ðŸš¨The	0	1	2	0				https://twitter.com/KJFilmsTweet/status/1609338623853268993
3	2023-01-0	1.61E+18	1BadKitty8	@German	0	0	1	0	['GermanStrands']			https://twitter.com/1BadKitty88/status/1609338632678305792
4	2023-01-0	1.61E+18	Mystborn	Things I'm	0	0	6	0				https://twitter.com/MystbornGames/status/1609339387430461440
5	2023-01-0	1.61E+18	Noewatch	@IconicN	0	0	0	0	['IconicNephilim']			https://twitter.com/NoewatchesTV/status/1609339462709805059
6	2023-01-0	1.61E+18	LoneJedi_	My Most	1	0	0	0				https://twitter.com/LoneJedi_77/status/1609339483207467009
7	2023-01-0	1.61E+18	lokinceptic	Last of us	0	0	0	0				https://twitter.com/lokinception/status/1609339683200278529
8	2023-01-0	1.61E+18	DifonMD	@privlah t	0	0	1	0				https://twitter.com/DifonMD/status/1609339724644384769
9	2023-01-0	1.61E+18	harleysalic	Itâ€™s the	0	0	3	0				https://twitter.com/harleysalicent/status/1609339789148581889
10	2023-01-0	1.61E+18	AkumaGoj	@Counte	1	0	6	0	['CounterVince2', 'CraftyAD360']			https://twitter.com/AkumaGoji/status/1609339913878503426
11	2023-01-0	1.61E+18	Captainho	@MrKrabz	1	0	1	0	['MrKrabz04']			https://twitter.com/Captainhorizon7/status/1609339967599411200
12	2023-01-0	1.61E+18	tvsotherw	The Last o	0	0	0	0				https://twitter.com/tvsotherworlds/status/1609340141822435330
13	2023-01-0	1.61E+18	leia_roma	2 WEEKS U	0	0	0	0				https://twitter.com/leia_romanova/status/1609340192699355136
14	2023-01-0	1.61E+18	jefiyero	@meupla	0	0	4	0	['meuplaystation']			https://twitter.com/jefiyero/status/1609340210306859008
15	2023-01-0	1.61E+18	MrKrabz04	@Captain	0	0	1	0	['Captainhorizon7']			https://twitter.com/MrKrabz04/status/1609340231597305857
16	2023-01-0	1.61E+18	harristiel	happy new	0	0	0	0				https://twitter.com/harristiel/status/1609340381170130944
17	2023-01-0	1.61E+18	JasmineAlf	@mxxnso	0	0	4	0	['mxxnson']			https://twitter.com/JasmineAlfaro16/status/1609340662117445633
18	2023-01-0	1.61E+18	forestgree	@wizardja	0	0	1	0	['wizardjarin']			https://twitter.com/forestgreenpixy/status/1609340706065088513
19	2023-01-0	1.61E+18	vomitnum	Rockman	0	9	36	0				https://twitter.com/vomitnumber3/status/1609340940745076736
20	2023-01-0	1.61E+18	iAlexRusse	Saw the ne	0	0	0	0				https://twitter.com/iAlexRussell/status/1609341167329501187
21	2023-01-0	1.61E+18	mcrthajon	HAPPY THE	0	0	1	0				https://twitter.com/mcrthajones/status/1609341251702390787
22	2023-01-0	1.61E+18	stormwarr	Favourite	1	0	1	0		['PSshare', 'PSBlog', 'V		https://twitter.com/stormwarning/status/1609341368606031872
23	2023-01-0	1.61E+18	boiz_fn	@Noobye	1	0	0	0	['Noobyeeter691', 'SoaRGaming'			https://twitter.com/boiz_fn/status/1609341586449788928
24	2023-01-0	1.61E+18	doybins	@thegame	0	0	0	0	['thegameawards']		Coordinat	https://twitter.com/doybins/status/1609341905292361731
25	2023-01-0	1.61E+18	byelenak	Just a fant	0	0	0	0				https://twitter.com/byelenak/status/1609342004197982208
26	2023-01-0	1.61E+18	2791_lam	@tlouarch	0	0	0	0	['tlouarchive']			https://twitter.com/2791_lamg/status/1609342013547102208
27	2023-01-0	1.61E+18	Sonofkryp	@ShyVort	1	0	0	0	['ShyVortex']			https://twitter.com/Sonofkrypton92/status/1609342385871519744
28	2023-01-0	1.61E+18	starryquin	2023 is de	0	0	1	0				https://twitter.com/starryquin/status/1609342675127533568

VI. Use Cases and Generalization

While reviews and public opinion is readily available on the listed websites such as Rotten Tomatoes, IMDb and Google Reviews, we get a more accurate representation of what the audience actually thinks about the show. It encapsulates how the audience reacted to the show at a given time frame and gives us an idea about the performance of the before, after and all throughout the release of the episodes. The program can also be generalised to take the input of various shows whose performance we intend to assess. We can also set up multiple filters such as language, region, etc. for the purpose of analysis.

This method can also be used for more variety of shows and can be generalized to assess the performance of almost any show that has a recognisable fanbase on twitter. A tested example of this is another show released in 2023, *"The Last of Us".* This makes the project worthy of being developed into a dynamic and interactive application for the use of both the creators as well as the consumers of entertainment content.

The generalisability of the code written was tested by applying the same implementation code on a different query. The query in question refers to the TV show "The Last of Us", extracting information between the months of January 2023 to March 2023.

It was observed that there was a much larger volume of data for this show, in comparison to "Game of Thrones: House of the Dragon" or "Lord of the Rings: The Rings of Power". As such, it also necessitated more storage and most of all, more time. There were no changes made to the code to accommodate a different query. All functions were generic and were able to run smoothly to process the data for 'The Last of Us'. Refer to the code for more details.

VII. Conclusion

In conclusion, it can be observed that the sentiment of the majority on Twitter regarding "Lord of the Rings: The Rings of Power" is positive; while that of "Game of Thrones: House of the Dragon" is neutral. However, when comparing this to their critical reception, there is a notable disparity; in that audience reception for Rings of Power on the website 'Rotten Tomatoes' is largely negative, whereas that of House of the dragon is largely positive.
This could be because of the following reasons:

1. Rings of Power caters to an older audience, as it is part of a franchise much older than Game of Thrones. Such an audience might be more critical of the show, as opposed to Twitter users, who largely skew younger.
2. House of the Dragon also boasts a wider audience, as it was consistently more discussed than Rings of Power (aside from Rings of Power's finale).
3. Rings of Power having a more positive trend on Twitter, despite having a smaller audience, may imply that the show has a stronger, more passionate fanbase.

Another notable discovery was that tweets on Rings of Power from the west coast (California, etc.) are significantly more liked than the east coast. However, it is the opposite for House of the Dragon.

VIII. Note on Ethicality

When extracting and manipulating data retrieved from real people on the internet, ethics and legality are of paramount importance.

As such, all the data retrieved from Twitter for this project was in line with the guidelines, restrictions, terms and conditions of Twitter. Furthermore, the project was conducted in its entirety for personal use. Only the user data that was publicly available and present in each tweet was adopted for the project. None of the content of this project, in any capacity, is to be used publicly.

IX. References

1. Beri, A. (2020, May 27). *Sentimental analysis using vader*. Medium. Retrieved March 31, 2023, from https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664
2. JustAnotherArchivist. (n.d.). *Justanotherarchivist/snscrape: A social networking service scraper in Python*. GitHub. Retrieved March 31, 2023, from https://github.com/JustAnotherArchivist/snscrape
3. *Welcome to Vadersentiment's documentation!*¶. Welcome to VaderSentiment's documentation! - VaderSentiment 3.3.1 documentation. (n.d.). Retrieved March 31, 2023, from https://vadersentiment.readthedocs.io/en/latest/
4. *Documentation#*. Documentation - GeoPandas 0.12.2+0.gefcb367.dirty documentation. (n.d.). Retrieved March 31, 2023, from https://geopandas.org/en/stable/docs.html
5. *Statistical Data Visualization#*. seaborn. (n.d.). Retrieved March 31, 2023, from https://seaborn.pydata.org/