SUNWAY UNIVERSITY

SCHOOL OF SCIENCE AND TECHNOLOGY

COURSEWORK FOR THE BSC (HONS) INFORMATION SYSTEMS (DATA ANALYTICS): YEAR 2

ACADEMIC SESSION MARCH 2021; SEMESTER 4 and 5

IST 2134: SOCIAL MEDIA ANALYTICS DEADLINE: 16 July 2021

STUDENT NAME:	STUDENT ID:
SHAMALAN RAJESVARAN	18042945
NEO JUI JIE	18025536
NG WEI XIANG	18033167
YAP ZI HAN	18046524
GAN TECK LIANG	18035253

INSTRUCTIONS TO CANDIDATES

This is a group assignment that covers 50% of your final coursework marks.

IMPORTANT

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

- Coursework submitted after the deadline but within 1 week will be accepted for a maximum mark of 50%.
- Work handed in following the extension of 1 week after the original deadline will be regarded as a non-submission and marked zero.

Lecturer's Remark (Use additional sheet if required)

SHAMALAN RAJESVARAN	18042945
NEO JUI JIE	18025536
NG WEI XIANG	18033167
YAP ZI HAN	18046524
GAN TECK LIANG	18035253

We .... (Name) ....std. ID received the assignment and read the comments Sham Jui Jie Xiang Zi Han Jeffrey (Signature/date) 15 July 2021

Academic Honesty Acknowledgement

SHAMALAN RAJESVARAN
NEO JUI JIE
NG WEI XIANG
YAP ZI HAN
GAN TECK LIANG

"We .... (student name) verify that this paper contains entirely our own work. We have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, we have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. We realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment."

Sham Jui Jie Xiang Zi Han Jeffrey 15 July 2021
.... (Student's signature / Date)

ABSTRACT

This assignment in general highlights the importance of social media analytics for all brands. We have taken a broad approach in understanding the various areas that we can address in order to provide a good data-driven analysis which can ultimately help the brand to come up with recommendations. We have elected to show the analysis for Twitter analysis on Disney Plus. We believe that we have covered a significant area whereby there are various different tools being used which can provide a good analysis into what we are doing. We have also done extensive research into the various platforms that Disney Plus is located on as well. This has allowed us to survey multiple different areas which can help us to understand our analysis better. We have exhausted Disney Plus' presence on Twitter, Facebook and Instagram. This has allowed us to obtain key insights and sufficient analysis into the various areas and then make our recommendations. Through this approach we are able to make credible improvements and recommendations so as to allow for a better structured analysis. We have managed to curate five distinct recommendations based on our analysis. We believe that we have sufficiently analysed the data and provided good quality explanation to justify our selection and recommendation.

1. Introduction

In this modern world, the competitiveness among businesses is increasing on a steep slope. More and more businesses are entering the market and according to Oberlo.com, the number of new business applications in the United States has doubled from 2.5 million in 2010 to 4.35 million in 2020. Starting from the year 1970, started the information era and everything was having a spike especially in the year 2000. A lot of businesses are having a boost and are growing faster than ever with the aid of technology during this information era. By now, 8 out of 10 of the world's largest companies by market cap were populated by technology companies like Facebook, Amazon, Google(FAANG) and more. Those companies that failed to adapt in the technology wave like Blockbuster were eliminated from this cruel competition and went broke in September 2010. This proves the power of technology and its ability to change the game in the business world is like a double edged sword and when used correctly, the business grows larger than ever like Disney and if used wrongly, the business ends up failing like Blockbuster.

Based on statistics, there are over 2.95 billion people that participated in at least one form of social media in 2019. Furthermore, the number is predicted to grow to 3.43 billion by 2023. The way that the users are interacting with social media keeps on improving and evolving, guiding to changes in the analytics. The future direction of social media and analytics continue to make a large impact on how the world will communicate and how the businesses will collect and analyse the data. Since social media enables the consumers and business to have an open conversation, marketers will have an opportunity to communicate directly to the target customers on a large-scale and they can use social media as the main method to engage with their potential customers and gain brand exposure. Hence, we can say that social media analytics will be an essential asset for marketing in the future. With the society furnishing their personal information and preferences in the social media platforms, marketers can easily perform social media analytics to gather information about their customers. Through social media analytics, the data gathered also allows the businesses to obtain crucial information that can help to improve their marketing strategy and deploy marketing campaigns to their target customers.

During this information era, the power of technology was looked at and remembered by everyone and anyone couldn't survive in this era without technology being involved in daily life. However there is a new thing emerging quietly in the IT world which was quoted as the new oil in the information eram, was none other 'Big Data'. Big Data being quoted as the new oil was not an over statement because till this point of the day, no one or not a single tool can process all the data that the world has. According to an article by the guardian from 2012, it is quoted that 1% of the world's data was analysed. However, we do not know how many % of data was analysed in the year 2021, but we do know that 2.5 quintillion bytes of data were created everyday in 2020. These figures make technology such a big thing and an important "sword" for your business. Using the example mentioned above with Disney being a successful adopter of the technology wave, it came to our attention to investigate and understand Disney's business and marketing strategy on its social media platforms.

The Walt Disney Company or commonly known as Disney was a successful American conglomerate in the mass media and entertainment industry. They established themselves as the leader in the American animation industry before diversifying into other industries such as live-action film production, television series, hotels and theme parks. However, Disney got its name and fame mainly through its film studio division which consists of The Walt Disney Studio, Pixar, Marvel Studios, Lucasfilm, 20th Century Studios and Searchlight Pictures. These film studios made multiple box office record breaking movies like Frozen, Avengers : End Game, and the Star Wars trilogy. At the same time, Disney had other business units such as Disney Channel, ESPN, and National Geographic under their television and broadcasting division. While in the direct-to-customer streaming services, Disney owns streaming services like Disney+, Hulu, ESPN+ and Hotstar as platforms to stream movies and films on a subscription basis. Last but not least, we cannot leave out the division that Disney was doing very well as a business unit which was the leisure division. They are the owners of "Disney Parks, Experiences, and Products" which consists of theme parks, resort hotels, and cruise lines from around the world. All these business units had contributed directly into making Disney being such a successful business till now with a market cap of 318 billion USD and getting a spot at the top 40 companies in market cap. Although Disney was originally founded in 1923 in the animation industry, we can see how far and big they have expanded across these years especially in the 1980s where they started acquiring corporate divisions and expanding their footprint around the globe when the information era had just started. Thus, with such success that Disney has gained over the years, it is exciting and motivating for us to discover the insights hidden in their social media and to see how we can elevate Disney's influence to another level based on our analytics.

2. Literature Review

Disney+ is a younger competitor in the movie streaming services industry, which makes them having some disadvantages at the start. However, Disney+ is slowly catching up to the competition year by year with more users on their platform. So it is interesting for us to research and review their marketing strategy for any insights. According to research on Disney+'s marketing strategy (Pereira, 2021), Disney+ is able to create nostalgic feelings with the audience through their brand name and movies such as Bambi, Jungle, The Lion King etc. Other than that, Disney+ also premiered the character "Baby Yoda" from the series The Mandalorian that in the end received much love which caused a hit across the Internet. Not only that, another research by Indigo Digital (Mckinnon, 2020) mentioned that Disney+ timed their app launch in an almost perfect timing with the movie "Frozen 2" released 10 days after launch to promote the fans to rewatch the movie using Disney+. An icing on the cake of their perfect timing was seen when their launch is close to the holiday season where many of their fans will utilise and subscribe their service to entertain themselves. At the same time, the website "Pathmatics" (Merchan, 2021) showed that Disney+ is active on 3 social media platforms which are Facebook, Instagram, and Twitter. The report shows that Disney+had heavily relied on Facebook to promote Disney+ with spendings of about $ 244 million on digital advertising. In the end, we can see the overall direction of Disney+ with their marketing strategy. Disney+ firstly utilized their brand name to create a friendly feeling with the audiences and release nostalgic movies to hook up their feelings. Then, they timed their official release date to be close to one of their hit movies "Frozen 2" and the holidays. When they are finally releasing it, they premiered the movie "The Mandalorian" with a cute character to spark conversations and netizens' love to viral it across the Internet. Lastly, they decided to invest heavily on Facebook as their primary social media platform for digital advertising.

Disney+ faces tight and difficult competition with top streaming movie platforms such as Netflix and Hulu over obtaining the rights to stream movies that are well-known. It is solely an effort to attract the attention of customers while keeping hold of their existing ones. However, Disney+ would be in an advantageous position with the existing library of content produced by the company over its history. In addition to that, Disney+ possesses the ability to stream titles from acquired properties such as Pixar, Marvel, Lucasfilm, and Fox (Havard, 2021). These franchises are referred to as stepping stones for the contents in Disney+ platform. These priceless contents are impossible for rivals to match and Disney+ continues to grow by expanding their own acquired exclusive contents such as the Marvel Cinematic or

Star Wars Universe as their focal point in retaining and attempting to win over more viewers or subscribers (Scott, 2020). The launch of Disney+ in 2019 enables the parent Disney+ Company, The Walt Disney Company to get their hands on some critical data from customers on the contents provided in Disney+ platform. Iger, the CEO of Disney, would be able to effectively allocate its funds in order to produce the next major blockbuster with the help of data analytics collected through Disney+ (S.O'Flynn, 2019). Disney+ also added a function that allows sharing with friends and families on social media nearly a year after its launch. They are free to choose any social media platform to share and the message includes links for Disney+ title page. This proves to be a good strategy for promoting the Disney+ platform by letting subscribers act as their brand ambassadors (Spangler, 2020).

Other than that, as Disney+ is constantly expanding its content library, it will continue to face more and more stiff competition from other large streaming platforms. In this case, we will make a direct comparison between Disney+ and Amazon Prime Video. First of all, we will compare the most fundamental point, which is the subscription fee for both of them. Disney+ costs 7.99USD a month while Amazon Prime Video costs either 8.99USD or 12.99USD. As for the promotion part, both of the streaming platforms actually do provide special discounts for students if they have validated student status. Next, we will compare the availability of the two streaming platforms. For Disney+, it is available on the majority of smart devices such as iOS, Android devices, any gaming devices, etc. On the other hand, Prime Video is available globally except for Mainland China, Iran, Syria and North Korea. In terms of diversity, Prime Video is currently more superior than Disney+, which means that Prime Video has more types of content than Disney+. The last element that we will compare is the possibility of the future content. As of now, Video Prime has a very consistent customer base and they are also updating their available content regularly to maintain their reputation. For Disney+, as they are still considered as a relatively new streaming platform, they are now growing dramatically and will soon become one of the top streaming platforms.

Disney entered the subscription Video-On-Demand (VOD) warfare with a narrower content selection than competitors, focusing on quality rather than quantity. Disney+, the SVOD service that launched in the United States in November for $7.99 a month, compared to $8.99 a month for Netflix's standard HD plan. According to a report by research firm Ampere Analysis, Disney+ would have less than one-fifth of the selections on Netflix in the United States in its first year. In the first year of Disney+, they offered 7,500 episodes of current and past TV shows, as well as 500 movies. According to Ampere Analysis, this represents only 16 percent of Netflix's US inventory of 47,000 TV episodes and 12.5 percent of the Netflix movie library of 4,000 movies. To begin with, Disney+ will not have a title count comparable to other SVOD players right out of the gate. More importantly, raw numbers don't tell the whole story: they don't indicate a subscription service's overall worth or popularity. Netflix, arguably the world's largest streaming service, is the go-to site for a wide range of old, new, and platform-exclusive television episodes and movies. In the first quarter of 2021, Netflix recorded more than 200 million subscribers (Stoll, 2021). Netflix has established itself as a viable alternative to traditional cable television, which it sees as its primary competition. Netflix acts to disguise the origins of material branded by other networks, instead positioning itself as the audience's primary point of identification. Netflix was the first streaming service to release exclusive content, igniting the current “original content” trend in the streaming market. Due to the fact that this trend has spread to all other competing streaming platforms, it is clear that "Netflix's impact appears to be both permanent and profound." In early 2019, Netflix announced a platform-wide price increase, raising the prices of all three of the platform's plans by up to 18%. This isn't the company's first price hike, but it is by far the most significant and the first to affect all customers at once (Siegal, 2019). Netflix made this move at an interesting time, because the competitive streaming industry is currently undergoing a fast transition, with consumers having more options than ever before.

Another comparison can be given to the establishment of iFlix, a free and subscription video on demand service that is based in Malaysia. Based on our analysis, iFlix is available on Facebook, Instagram, Twitter and Youtube. This has allowed them to effectively share their content to the general public and to look into areas which they can best promote and market their content. It is also indicated that the price of iFlix stands at RM10 per month. iFlix also offers a free trial of 30 Days to all of their customers who try out the platform. iFlix also posts original content which can best attract the customers to their platform. This allows them to access the various content and the videos. It is also accessible to 2 devices.

	Disney+	Prime Video	Netflix	iFlix
Monthly price (RM)	18.30	14 / 27	33 / 41 / 51	10
Free trial (days)	7	30	30	30
Original content	Yes	Yes	Yes	Yes
Number of device	4	3	1 / 2 / 4	2

3. Research Topic 1 : Disney+ with their social media account

3.1 Methodology

When we were searching for potential research topics in our discussion, we came into a common interest to investigate how Disney+ performs on their social media accounts internally. To know how, we had to perform analytics across 3 different platforms that Disney+ were active in which are Facebook, Instagram, and Twitter. It came to our knowledge after the literature review when Disney+ had advertisements on these platforms but half of the budgets were spent on Facebook alone and the remaining on Instagram and Twitter. Thus, we are interested to investigate whether their digital marketing on social media was accurate and successful by such budget allocation with Facebook being the higher priority. Other than that, the demographic of people in different platforms should also be under consideration because anyone would definitely want to market their product to their target audiences. So, what they really need to avoid is to market and apply advertisements to the wrong audiences which makes the decision of prioritizing one platform crucial. It is seen that the younger generations like Gen Y and Gen Z were more active and preferred Instagram than Facebook. However, for the boomers and Gen X, they tend to be more active and prefer Facebook as their primary social media platform.

To really understand how Disney+ performs across these social media platforms with different user demographics, we decided to approach this by looking into the comments under the posts as well as like, comment, share count. These aspects were helpful in a way that we can see how different people react on different social media platforms, like how active they are, their engagement rate, and their sentiments. Thus, our one biggest criterion is to source our data from identical posts that are posted on these 3 different platforms. Disney+ do not have consistent posts across the platforms, especially Twitter having some more tweets that are not available on Facebook and Instagram. Therefore, we aimed for identical posts that are present on 3 different platforms for the consistency of the analytics to be performed later. As mentioned earlier, we will be looking into 2 main aspects of the posts, with the first one being the comments on the posts, and the second one being the statistical counts ie: likes, shares/retweets, comment counts. We will be comparing these aspects later in analysis to see which platform performs better or more positively as compared to the other 2.

When looking into the comments of the posts, we wanted to investigate 2 questions. The first one being how often the users comment on the post, and the second one being how the users react to the post. Thus, we will be performing descriptive analytics to obtain the comment counts, and text sentiment analytics for user reaction. Comment count was used because it reflects the posts and the content's engagement rate with the audience. The more counts we get, reflects how engaged the audience are on the posts and how active they are in participating in the discussion by commenting below. User reaction being the second objective was to understand user's sentiment on the platform. We wanted to understand whether the platform's audience will comment their opinions that carry positive or negative sentiments towards the content. With that, we can understand the behavior of different groups of audience and provide feedback and constructive advice for better usage.

When looking into the statistical counts of the posts, we wanted to investigate whether which platform is performing better in contacting and hooking up with its audiences. “Count” is the most straight-forward yet effective method to figure out the amount of engagement the post on the platform is having with their audience. With that, the performance of accounts on different platforms is shown and then we can provide feedback and maybe even advice on repositioning their advertisement marketing.

A total of 3 different levels of posts will be scraped in each platform, which means that we will be analyzing the result of 9 posts in total. We chose 3 different levels with low, medium, and high engagement to have a fair and balanced result instead of being biased with focusing on “higher counts” posts only. Back to the research topic, what we really wanted to investigate is to see how Disney+ is doing with their social media accounts across different social media platforms. The investigation and analytics were carried out in a way that a lot of considerations and criteria were included to make sure the result in the end was fair and square for higher accuracy and best result.

3.2 Analysis

3.2.1 Scraping

Before we start with our analysis, we first picked 3 posts from different levels. After successfully identifying the 3 posts that were common across all platforms on Facebook, Instagram, and Twitter as shown below, we can finally start the analytics process.

i. Low engagement post :

Facebook link, here

Twitter link , here

Instagram link, here

ii. Medium engagement post :

Facebook link, here

Twitter link , here

Instagram link, here

iii. High engagement post :

Facebook link, here

Twitter link , here

Instagram link, here

```python
from facebook_scraper import get_posts
import pandas as pd
import numpy as np

listposts = []
for post in get_posts("disneyplus", pages=7, options={"comments": True}):
    print(post['text'][:50])
    listposts.append(post)
```

Figure 1: Code snippet for facebook scraping

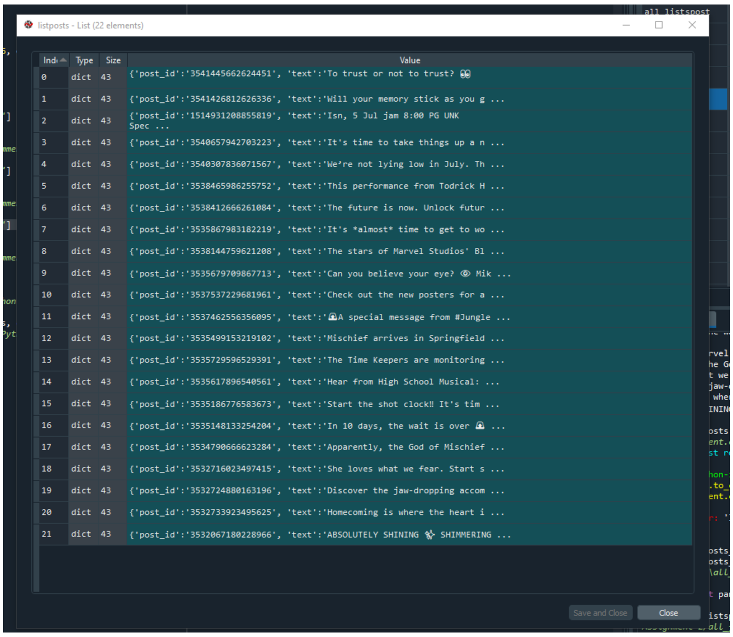
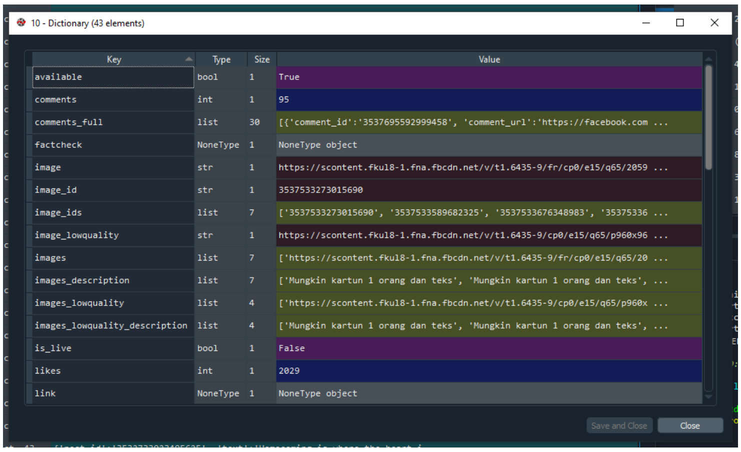


Figure 2: list of returned posts from scraper


10 - Dictionary (43 elements)

Key	Type	Size	Value
available	bool	1	True
comments	int	1	95
comments_full	list	30	[{'comment_id':'3537695592999458', 'comment_url':'https://facebook.com ...
factcheck	NoneType	1	NoneType object
image	str	1	https://scontent.fkul8-1.fna.fbcdn.net/v/t1.6435-9/fr/cp0/e15/q65/2059 ...
image_id	str	1	3537533273015690
image_ids	list	7	['3537533273015690', '3537533589682325', '3537533676348983', '35375336 ...
image_lowquality	str	1	https://scontent.fkul8-1.fna.fbcdn.net/v/t1.6435-9/cp0/e15/q65/p960x96 ...
images	list	7	['https://scontent.fkul8-1.fna.fbcdn.net/v/t1.6435-9/fr/cp0/e15/q65/20 ...
images_description	list	7	['Mungkin kartun 1 orang dan teks', 'Mungkin kartun 1 orang dan teks', ...
images_lowquality	list	4	['https://scontent.fkul8-1.fna.fbcdn.net/v/t1.6435-9/cp0/e15/q65/p960x ...
images_lowquality_description	list	4	['Mungkin kartun 1 orang dan teks', 'Mungkin kartun 1 orang dan teks', ...
is_live	bool	1	False
likes	int	1	2029
link	NoneType	1	NoneType object

Save and Close | Close

Figure 3: Data of each posts in list recorded in dictionary format

To scrape the posts from Facebook, we used a tool called "facebook_scraper" to perform the scraping without API as shown in figure 1. Due to the nature of this tool, it scrapes posts of a user/page by pages instead of the direct scraping by link. We can give a random number as long as it is larger than 5 to set a number of pages we want to scrape. The scraper then returns the information of the posts which was stored by us in a list. Then as shown in figure 2 which is the list of posts, we look for the index number of the post we stated earlier and extract the number of comments, likes, and shares accordingly from the dictionary shown in figure 3 into a new list. We repeat the process of extracting numbers for the counts of the other 2 posts and then store them into a single pandas dataframe for easy comparison. To scrape the comments, we once again use the index number of the posts from figure 2 and use a for loop to loop through each of the items in the "comments_full" list to scrape for the content of the comment only. Similarly, we repeat the process for 2 other posts and store them in separate lists. After successfully scraping all required data, we export the combined count dataframe, and lists of comments of the 3 posts to their respective Comma Separated Value(CSV) file.

```
from instascrape import Post
import pandas as pd

''' example 1 '''
# Instantiate the scraper objects
igpost1 = Post('https://www.instagram.com/p/CQwEyiVsgLa/?utm_source=ig_web_copy_link')
# Scrape their respective data
igpost1.scrape()

igpost2 = Post('https://www.instagram.com/p/CQvzQz3sUr3/?utm_source=ig_web_copy_link')
igpost2.scrape()

igpost3 = Post('https://www.instagram.com/p/CQtcM3MMLUY/?utm_source=ig_web_copy_link')
igpost3.scrape()
```

Figure 4: Code snippet for Instagram scrapping

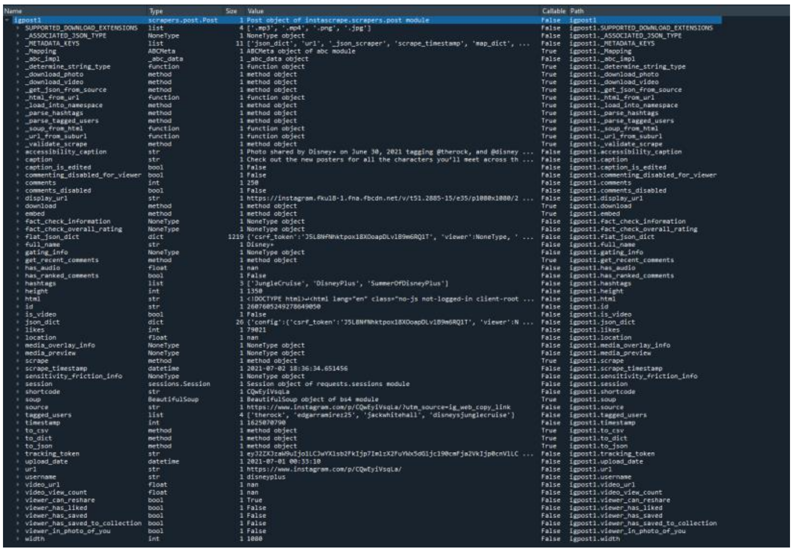


Figure 5: Object returned after scraping

```python
flat_json1 = igpost1.flat_json_dict
basename = "_node_text"
insta_comment1 = []
for i in range(96):
    current = str(i) + basename
    if current in flat_json1:
        insta_comment1.append(flat_json1[current])
    else:
        break
```

Figure 6: Code snippet to extract comments through the object returned

For the scraping of data from Instagram, we did the data scraping by using a tool called “instascrape” that requires no API too. Unlike “facebook_scraper”, “instascrape” is able to scrape the data of a certain post directly with an Instagram post link parsed into the function as shown in figure 4. By parsing the links as attached above, the function returns an object consists of all the data related to the link as shown in figure 5. Among all the variables returned within the object, the comments are located inside a dictionary named ”flat_json_dict”. Then, within the dictionary also includes not only the text of the comment but also other data of the comments like their link, user, id etc. However, only the text of the comment itself is our interest thus a for loop as shown in figure 6 was used to iterate into the dictionary to obtain the text and append into our own list. To obtain the count statistics, we can just obtain the values by calling the variable inside the object shown in figure 5 directly and store in our own list. The process was repeated from the start by using different links in the scraping tool’s function to obtain all data from all 3 posts in Instagram. Lastly, all the created lists were also exported as CSV file for easier analytics later.

```python
import snscrape.modules.twitter as sntwitter
import pandas as pd

''' 1. obtain replies in the form of tweets to disney '''

tweets_list1 = []
for i,tweet in enumerate(sntwitter.TwitterSearchScraper('to:disneyplus').get_items()):
    if i>10000:
        break
    tweets_list1.append([tweet.url, tweet.date,  tweet.renderedContent, tweet.id, tweet.user, tweet.replyCount, tweet.retweetCount, tweet.likeCount, tweet.quoteCount, tweet.conversationId, tweet.hashtags])

# Creating a dataframe from the tweets list above
tweets_df1 = pd.DataFrame(tweets_list1, columns=['Link', 'Datetime', 'Content', 'Tweet Id', 'User', 'replyCount', 'retweetCount', 'LikeCount', 'quoteCount', 'conversationId', 'hashtags'])


''' 2. obtain the tweets posted by disney to get the conversation id '''

tweets_list2 = []
for i,tweet in enumerate(sntwitter.TwitterSearchScraper('from:disneyplus').get_items()):
    if i>40:
        break
    tweets_list2.append([tweet.url, tweet.date,  tweet.renderedContent, tweet.id, tweet.user, tweet.replyCount, tweet.retweetCount, tweet.likeCount, tweet.quoteCount, tweet.conversationId, tweet.hashtags])

# Creating a dataframe from the tweets list above
tweets_df2 = pd.DataFrame(tweets_list2, columns=['Link', 'Datetime', 'Content', 'Tweet Id', 'User', 'replyCount', 'retweetCount', 'LikeCount', 'quoteCount', 'conversationId', 'hashtags'])
```

Figure 7: Code snippet to scrape twitter data

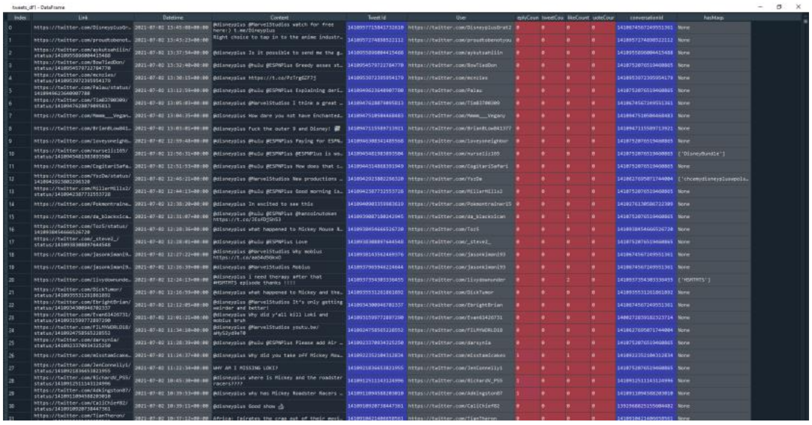


Figure 8: Dataframe of tweets tweeted by Disney

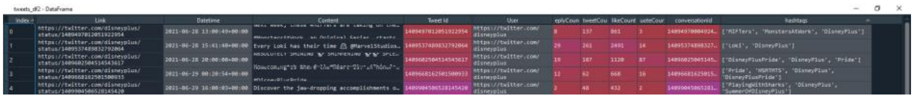


Figure 9: Dataframe of tweets tweeted to Disney

To scrape Twitter data, we used a tool called "snscrape" that requires no API similarly shown in figure 7. The tough part of twitter scraping tools available online is that they cannot obtain the replies of a certain post directly. This was the case with this tool too. So, to scrape the replies of the posts, we first need to scrape the tweets tweeted by Disney+ themselves. Then, we continue the scraping process by scraping all tweets tweeted to Disney+. And now we had 2 lists, with the former being a list of tweets by Disney+ shown at figure 8, with the latter being a list of tweets tweeted to Disney+ shown at figure 9. Then, we look into the 1st list we got and extract the "conversationId" of the tweet we wanted. After that, we search for tweets from the 2nd tweet with the "conversationId" obtained earlier and append them to our own list once a match of them was found by using a loop. With that, we are able to obtain the replies/tweets on that post with a limitation that some older replies might be left out because we only scraped the first 10000 tweets to Disney+. Moving forward, scraping of the statistics counts for our targeted tweets wasn't tough. The 1st list of tweets tweeted by Disney+ already included the figures inside, so what we need to do is just subsetting the list into a new list of our own. Finally, all created lists were exported similarly for analytics.

3.2.2 Analytics

Right now, we already have 2 types of data scraped and available. The first one was the text of all the comments and the second was the statistic counts on the posts. The statistical counts were cleaned, ready and self-explanatory which is why the focus of analytics is on the text comments. The analytics was mainly focusing on the sentiment distribution. But before we can proceed with the analytics, some data pre-processing was performed to lemmatize sentences, tokenize words, and remove noise.

```
def lemmatize_sentence(tokens):
    lemmatizer = WordNetLemmatizer()
    lemmatized_sentence = []
    for word, tag in pos_tag(tokens):
        if tag.startswith('NN'):
            pos = 'n'
        elif tag.startswith('VB'):
            pos = 'v'
        else:
            pos ='a'
        lemmatized_sentence.append(lemmatizer.lemmatize(word,pos))
    return lemmatized_sentence


def remove_noise(tweet_tokens, stop_words):
    cleaned_tokens = []
    for token in tweet_tokens:
        token = re.sub('http[s]', '', token) #remove https
        token = re.sub('//t.co/[A-Za-z0-9]+', '', token) # remove remaining link
        token = re.sub('(@[A-Za-z0-9_]+)', '', token) # remove mentions
        token = re.sub('[0-9]', '', token) #remove numbers
        if (len(token) > 3 ) and (token not in string.punctuation) and (token.lower() not in stop_words):
            cleaned_tokens.append(token.lower())
    return cleaned_tokens

stop_words = stopwords.words('english')

fb_token = fb['Value'].apply(word_tokenize).tolist()


cleaned_tokens = []
for tokens in fb_token:
    rm_noise = remove_noise(tokens, stop_words)
    lemma_tokens = lemmatize_sentence(rm_noise)
    cleaned_tokens.append(lemma_tokens)

def get_all_words(cleaned_tokens_list):
    for tokens in cleaned_tokens_list:
        for token in tokens:
            yield token

tokens_flat = get_all_words(cleaned_tokens)
```

Figure 10: Code snippet for data pre-processing

During data pre-processing shown at figure 10, we first define a function to lemmatize the sentences to their stemming while respecting the original content and meaning. Then, we defined another function to remove noise like hyperlinks and symbols. After the definition of functions. The real process starts by getting a list of stop words and then tokenizing all the comments. Once they are all tokenized, a loop was used to remove noise, and then lemmatized accordingly. After this, a cleaned list of tokens will be returned which ends the process of the text pre-processing. To begin obtaining results, we import one of the CSV comment files as a dataframe from the CSV file that we exported out earlier. Then, the dataframe runs through the process of data preprocessing before it proceeds for the next stage.

To obtain the top 20 words in the list of comments, the texts must go through data pre-processing and make sure a clean list is returned. After that, we define a function to yield the single token that is in a list of words. Then, we just run the "FreqDist" function in the nltk library and subset the result with 20 words in the "most_common" function. However, this process is not under the area of interest for the research topic and is not further analyzed.

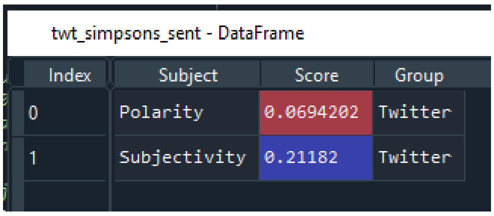
```
text_blob = []
for cmt in fb['Value'].tolist():
    analysis = TextBlob(cmt)
    if analysis.sentiment.polarity == 0:
        sentiment = "Neutral"
    elif analysis.sentiment.polarity > 0:
        sentiment = "Positive"
    elif analysis.sentiment.polarity < 0:
        sentiment = "Negative"
    text_blob.append(sentiment)

fb['Sentiment'] = text_blob

polarity = []
subjectivity = []
for i, value in fb.iterrows():
    text = TextBlob(value['Value'])
    polarity.append(text.sentiment.polarity)
    subjectivity.append(text.sentiment.subjectivity)

c = list(zip(polarity, subjectivity))
c = pd.DataFrame(c, columns=['Polarity', 'Subjectivity'])
labelled_fb_cmt = pd.concat([fb, c], axis=1)

m = []
m.append(mean(c['Polarity']))
m.append(mean(c['Subjectivity']))
m = pd.DataFrame(m)
m = m.rename(index={0: 'Polarity', 1: 'Subjectivity'})
```

Figure 11: Code snippet for sentiment analytics

After obtaining the top 20-word counts to have an overview of the words used in the comments, we then proceed to work on obtaining the sentiment values of comments on each post on different social media sites using the code shown at figure 11. We take the comments and convert it into a list and use a for loop to loop through each of the comments only to convert it into a text blob. Once it is converted to text blob, we store it in a variable and check the sentiment polarity score of it using an if-else condition statement to label the text as "positive", "negative", "neutral" according to their sentiment polarity. Once each of the comments is labelled with their sentiment polarity, we append them into the original data so that we can see clearly which comment is positive, negative, or neutral.

After the labelling process, we started a similar process of counting the polarity and subjectivity of each comment by looping through each row of the comment data so that the exact score is recorded for comparison and analytics later. Once all the comments have their respective polarity and subjectivity score, the mean of the score is calculated and stored in another list so that we get to know the score of the post on that platform for comparison and discussion. For now we have the comment data updated with new column values of their sentiment type, polarity, and subjectivity score. At the same time, we had another dataframe of their mean scores for polarity and subjectivity. To end the process, we once again export these data frames as csv for the next stage and this process is repeated another 8 times for the other posts.

3.2.3 Visualizations


twt_simpsons_sent - DataFrame

Index	Subject	Score	Group
0	Polarity	0.0694202	Twitter
1	Subjectivity	0.21182	Twitter

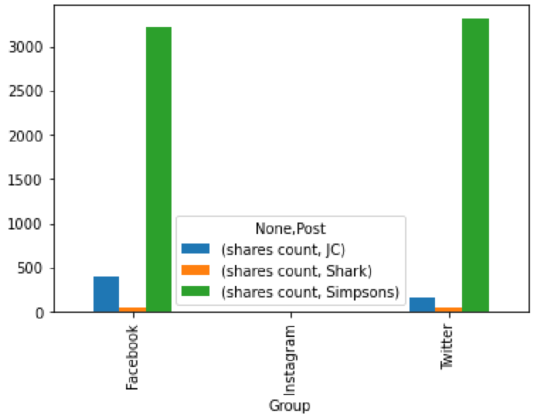
Figure 12: Assigning the name of the platform to the datasets

After data scraping and processing, we finally came to the last stage of analytics with the final analytics and visualizations. We had 2 types of visualizations to produce in mind. The first one was a bar plot of the statistic counts of different posts grouped by the platform. Thus for the first type of visual, we will be having 3 different bar plots for comment count, like count, and share count. Then for the 2nd type of visualization, we wanted to perform a bar plot comparing the sentiment score of different posts grouped by platform. Thus, we are having 3 more plots for 3 different posts stated at the start of analysis.

We first import all the datasets (mean sentiment score and statistic counts) and assign names to them respectively. Then, we renamed the name for the columns so that all of the datasets are the same for easier data querying later. Once they have a new unified column name, a new variable was created in respective dataframes with the name of their platform. For example, we imported the comments of Simpsons that were scraped from Instagram. Then, the column names for their mean score were renamed for better readability. After the rename process, the name of the platform was added to the dataframe so that we do not get lost once we combine all the datasets.

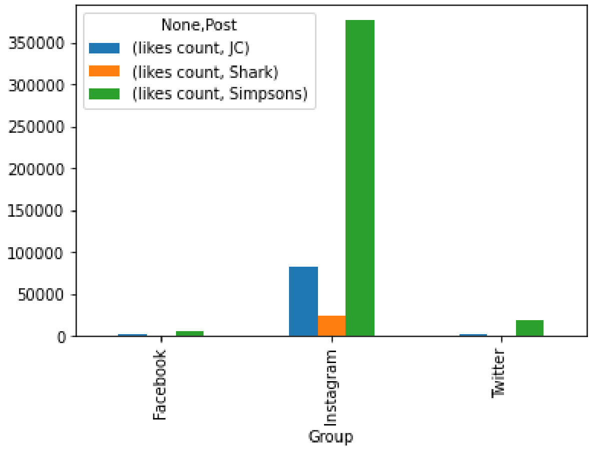
To obtain the bar plot for statistical counts, we combine the dataframes into a single big dataset and only perform the plotting using the python's default plotting library and feature. A similar approach was used to obtain the bar plot for sentiment score whereby the dataframes are concatenated based on the posts into a single big dataframe.

3.3 Results



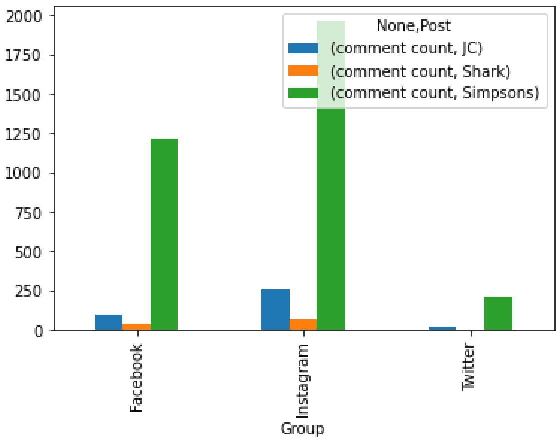
Share Count			
Jungle Cruise	403	0	159
Playing with Sharks	53	0	48
Simpsons	3211	0	3313
Social Media Platform	Facebook	Instagram	Twitter

Figure 13: Share count of 3 different posts grouped by platform



Like Count			
Jungle Cruise	2101	81248	1504
Playing with Sharks	458	23585	435
Simpsons	5824	377155	18038
Social Media Platform	Facebook	Instagram	Twitter

Figure 14: Like count of 3 different posts grouped by platform



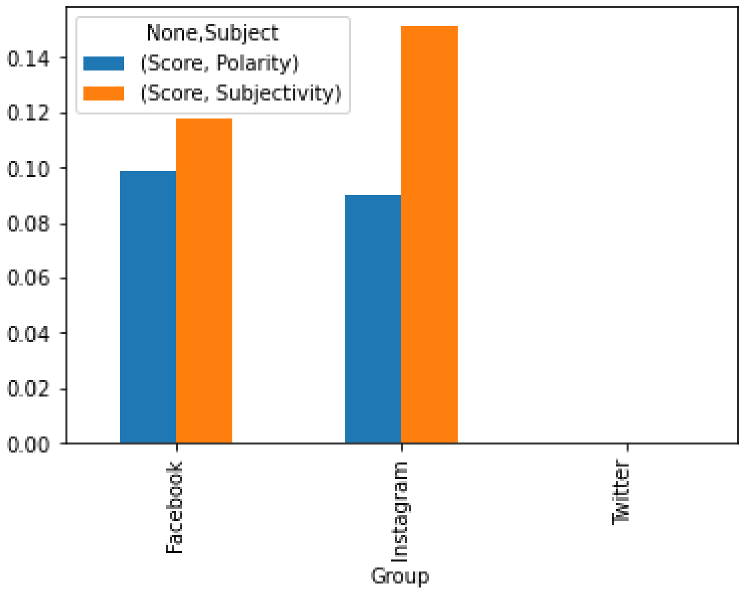
Comment Count			
Jungle Cruise	99	253	15
Playing with Sharks	38	65	2
Simpsons	1212	1965	209
Social Media Platform	Facebook	Instagram	Twitter

Figure 15: Comment count of 3 different posts grouped by platform

After a series of processes in the analytics process, we finally are able to come up with our end results. As proposed earlier, we will be looking at the statistical count of posts of each platform first. Looking at Figure 15 13, it is seen that Twitter has a higher share count on the "Simpsons" post at 3313 shares as compared to Facebook at 3211. However, the situation was reversed where Facebook has a higher share count at 403 shares than Twitter at 159 for the "Jungle Cruise" post. It is obvious to see that we do not have any results for Instagram too. This was due to the nature of Instagram not having a sharing feature like Facebook's share or Twitter's retweet. Thus, no figures can be obtained for Instagram for this comparison.

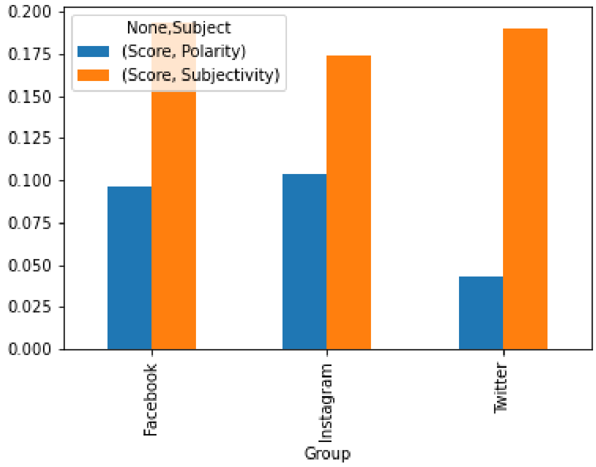
Looking at figure 15, we had another bar plot to compare the like counts of the posts grouped by their platform. This plot was very obvious that Instagram has much higher like counts when compared to Facebook and Twitter. For the "Simpsons" post, Instagram recorded 377155 likes, Facebook recorded 5824 likes, while Twitter had 18038 likes. For the "Jungle Cruise" post, Instagram had 81248 likes, Facebook had 2101 likes, and Twitter had 1504 likes. For the "Playing with sharks" post, Instagram recorded 23585 likes, Facebook recorded 458 likes, while Twitter recorded a like count of 435 which is very close to Facebook.

Moving forward to figure 15, is a bar plot that visualizes the number of comments of different posts grouped by the social media platform. Once again we are seeing a similar pattern at figure 14, where Instagram once again crowned as champion of comment counts when compared to Facebook and Twitter. Going deeper, for the "Simpsons" post, Instagram had 1965 comments, Facebook had 1212 comments, while Twitter had 209 comments (replies). For another post "Jungle Cruise", Instagram had 253 comments, Facebook had 99 comments, Twitter had 15 comments. Lastly for the "Playing with sharks" post, Instagram recorded 65 comments, Facebook recorded 38 comments, while Twitter had only a mere comment count of 2 only.



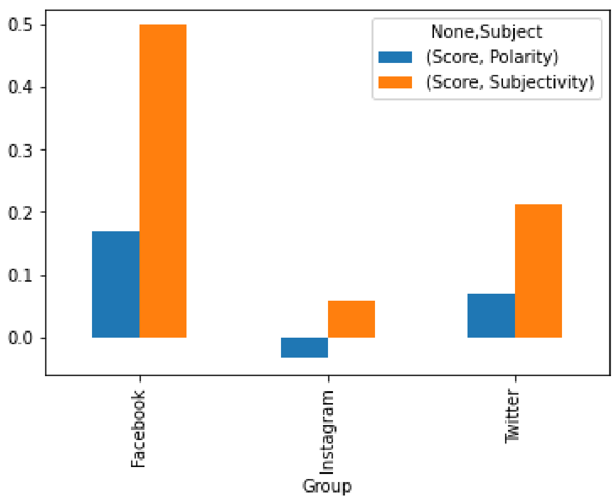
Playing with Sharks			
Polarity	0.0987	0.0897	0
Subjectivity	0.1177	0.1512	0
Social Media Platform	Facebook	Instagram	Twitter

Figure 16: Mean sentiment score of the "Playing with sharks" post grouped by platform



Jungle Cruise			
Polarity	0.0959	0.1038	0.0433
Subjectivity	0.1935	0.1739	0.1896
Social Media Platform	Facebook	Instagram	Twitter

Figure 17 : Mean sentiment score of the "Jungle Cruise" post grouped by platform



Simpsons			
Polarity	0.1683	-0.0333	0.0694
Subjectivity	0.4973	0.0583	0.2118
Social Media Platform	Facebook	Instagram	Twitter

Figure 18 : Mean sentiment score for the post "Simpsons" grouped by platform

Based on the results on figure 13, figure 14, figure 15, we can see an obvious pattern that Instagram is the platform that has the highest popularity and engagement with their users and audiences. However, our research topic does not stop there, and we will proceed to look at the sentiments to understand how the audiences are feeling. Therefore, we had figure 16, figure 17, and figure 18 plotted to continue the research. These 3 plots were plotted to investigate the polarity and subjectivity score each post got on average which is grouped by platform similarly.

At **figure** 16, was a bar plot of the average sentiment score of the "Playing with sharks" post grouped by different platforms. At first glance we can immediately notice that we do not have any scores for Twitter, and this is because of the comments available on the post. As shown in figure 15 and explained above, Twitter had only 2 comments on the post and both the comments were non text, thus making it scoring a 0 in terms of polarity and subjectivity because there is nothing to be analyzed. Then when we look at Facebook and Instagram, it is seen that Instagram has a lower polarity while having a higher subjectivity when compared to Facebook.

Then at figure 17, is a bar plot of the average sentiment score for the "Jungle Cruise" post grouped by platform. Looking at the polarity, Twitter has the lowest polarity at 0.0433, which is followed by Facebook at 0.0959 and only Instagram at 0.1038. If we are comparing the subjectivity of those comments on the post, Facebook's comments are highly subjective if compared to Instagram and Twitter because it has the highest subjectivity score of 0.1935. That is followed by Twitter with a subjectivity score of 0.1896 and Instagram at 0.1739. Thus for the "Jungle Cruise" post, Instagram has the highest polarity while also having the lowest subjectivity.

Moving on to the last result which was shown at figure 18, is a bar plot of the average sentiment score for the "Simpsons" post grouped by platform. This time the result was a little different at first glance when compared to the previous posts because we finally had a negative polarity result. Facebook topped and scored 0.1683 for polarity which is followed by Twitter at 0.0694 and Instagram at -0.0333. In terms of subjectivity, Instagram has the lowest subjectivity at 0.0583, followed by Twitter at 0.2118, and only Facebook at 0.4973. It is interesting to see a negative polarity result because we might discover interesting insights when we go deeper in the discussion. In the end, Facebook has the highest polarity while Instagram has the lowest subjectivity.

3.4 Discussions

Looking back on the whole process of analytics, manipulation, output result formation, all of it starts from one single point, the research topic. Just to recap our process from the research topic till now, the whole idea of this research topic is to understand how Disney+ social media is doing internally. To investigate that, we went with the approach of comparing their accounts' performance on the 3 main social media platforms that Disney+ had heavily invested in advertising and marketing that was mentioned in the literature review. Those platforms were Facebook, Instagram, and Twitter. We decided to measure their success-fullness by using statistics such as like count, share count, and comment count as metrics. However, we decided to go deeper to justify the results of those statistics by looking at the sentiment of the comment section and perform a comparison. By now, we already had a successful identification of research topics and methodology, we then moved forward to analytics.

During our analytics, we split the process into 3 parts, with the 1st one being data scraping from 3 platforms, 2nd one being the statistics and sentiment analytics, and the last one being result and visualizations. The toughest part during the whole analytics process would be scraping. We are dealing with 3 different platforms, which requires us to familiarize which 3 different tools, and 3 different data formats returned. For example, Facebook scraping is different because you can only get the posts through "number of pages" instead of an exact amount. For Instagram scraping, there is so much data returned with a single command and that requires expertise and strong knowledge to deal with the object returned upon each scrape. Then for Twitter scraping, the process is rather indirect if we wanted to obtain the replies(comment) of each tweet. Thus, we need to do 2 different scrapes and compare them by the "conversation id" to know which comment belongs to which tweet. At the end of the process, we are able to obtain the required data which were the statistical counts and comments, which all of them were cleaned and exported to CSV to get ready for the statistics and sentiment analytics process. Despite having high challenges during data scraping, those obstacles were overcome, and the statistics and sentiment analytics process was much easier. The code is highly usable because we cleaned and removed most of the data during the scraping process that removes a lot of unnecessary data for this process. With a clear objective in mind on how we should analyze our cleaned data, the process was straight-forwarded, quick and easy. We used several user-defined functions and libraries like nltk, textblob, and matplotlib packages to ease our process of analyzing. In the end of this process, we updated our original dataset of comments with new columns of their sentiment, and their polarity and subjectivity score. Not only that, but we also obtained the average polarity and subjectivity of the comment section of each post in each social media platform. After analyzing the scraped data, we compiled them together and produced plots for easier visualization of the results which we ended up with 6 insightful plots about how Disney+ is performing across the social media platforms.

The statistical counts of like, share and comment plots showed a unified result of Instagram being the platform that interacted the most with their audience. Instagram topped in all posts when compared to Twitter and even Facebook that they heavily invested on. Despite Instagram not having a "sharing" feature, they still performed much better than other platforms. Looking at the statistics of the popular post which is the "Simpsons" post, the like counts on Facebook is only a mere 1% of Instagram's while Twitter is only 4% of the total likes Instagram gets. This is very strong evidence that Disney+'s younger audience on Instagram is interacting much more than the audience of Facebook and Twitter. This trend of Instagram being the best performing platform for Disney+ is continued for the other post we investigated. Facebook's and Twitter's performance can be considered as a draw because both of them performed similarly. Facebook had more comment counts and slightly more share counts. Twitter was performing a little bit better in terms of like count when compared to Facebook but the figures are very close with a difference of not more than 10% which makes them a draw in terms of performance.

Now we know that Instagram is the top performer among all platforms, it is time to look into the sentiment of their performance to see how their audiences behave on different platforms. Overall, most of the comments on all 3 platforms were positive but very close to being neutral. Only Instagram had a negative polarity for the post about The Simpsons. Then, most of the comments on all 3 platforms also had a relatively low subjectivity score which means that those comments are very close to being an objective comment instead of being very subjective that differs from person to person. Only Facebook had a high subjectivity score for the post about Simpsons also, at 0.4973. Facebook generally has a higher polarity and subjectivity which reflects that the audience and user on Facebook tend to comment positive comments and show their optimistic part of them in an objective manner. The comments of the audiences on Instagram tend to be slightly less positive than Facebook and being in a more subjective manner. Twitter on the other hand shows that their users behave similar to Instagram with having lower sentiment polarity than Facebook. However, we 2 highlights were found on our sentiment analytics, which is Instagram being the only platform having a negative sentiment polarity on a post that we research on, and the other highlight was Facebook having high subjectivity comments on a post.

Coincidentally, both highlights were from the same post which is the "Simpsons". Instagram having a negative sentiment polarity and overall lower score might be due to their user base too. The younger generations tend to express opinions much more freely and straight forward than older generations and at the same time, Instagram is a much-preferred platform by younger generations. This makes Instagram have lower sentiment polarity because Instagram's users, which mostly are younger generations, express their opinions freely at the comments whether it being positive or negative. Facebook on the other hand has older audiences, have more matured minds and they tend to think before they speak which makes them leave "nicer" or more positive-sentimental comments that are highly subjective.

3.5 Implications

At the end of the whole analysis and discussion, we can see something from our work and provide comments to Disney+. Firstly, like our research in literature review, Disney+ spent 52% of their digital advertising budget on Facebook and only 7% on Instagram and 3% on Twitter. Their strategy proved to be less effective because their audience are still more actively engaging with them on Instagram despite them investing heavily on Facebook. Although Disney's target market is almost all age groups with their quote being "You're dead if you aim only for kids. Adults are only kids grown up, anyway", our data and result shows that the younger generations are reacting with them more as compared to the older generations. However, we should also notice that Facebook has a higher sentiment polarity on average which proves that their Facebook advertising is successful in sentiment terms. Thus, the take home advice to Disney+ for Facebook usage is they should keep up the good work of their Facebook account management, but also figure on new ways to increase user engagement. They could probably post Facebook-limited posts that are solely interactions with Facebook users. By the time of writing this report, Disney+ Instagram had 4.2 million followers while Disney+ Facebook had 4.5 million followers. This once again proved that despite Facebook being the platform that has the largest followers among other platforms, the user does not interact and engage much with them. Higher user engagement will always help in the growth of user loyalty and adherence.

As mentioned in our previous comment, Instagram is the most engaged platform between Disney+ and its users. This is seen from the statistics count we obtained in results where Disney+ on Instagram literally had 65 times more likes than Facebook and 21 times on Twitter. This is like an undiscovered market that has huge potential for them to further into and explore. However, our results also show that the sentiment polarities on Instagram are lower than Facebook and Twitter and this is where our advice comes into play. Instagram is definitely a platform that has huge potential and Disney+ must improve their management especially user engagement. Our result showed that Instagram users are willing to interact with them and show their presence. Thus, Disney+ can interact more with their users in the comment section of Instagram to create more topics or conversations to spark loyalty and increase their feelings or sentiment towards Disney+. Unlike our advice on Facebook is to increase user engagement, Disney+ can spark topics and attract Instagram users to join their conversation to increase popularity. Instagram is the platform that they invested less on digital marketing but outperforms the other platform which they invested 7 times more. This is very strong evidence showing that they should increase more of their exposure on Instagram, while reacting with their users at the comment sections to increase their sentiments.

Lastly, Disney+'s presence on Twitter is a little bit awkward since they do not have a clear goal or advantage over other platforms. They either outperform Instagram in sentiment polarities or underperform Facebook in statistical counts. Disney+ should really look into focusing on 1 element on Twitter, either capturing more followers or gaining a loyal user base. However, we would actually advise Disney+ to work on gaining a loyal user base on Twitter. This is because of the nature of Twitter being the only social media platform that limits the number of words on a single post. This feature that sometimes seemed as their weakness was actually their strength. All tweets(posts) can be made in a very short and sweet manner so that users don't need to spend so much time reading, understanding, and digesting messages. With the limitation of 140 characters, it actually feels more like typing a message for a conversation with your friend than making a comment that can be viewed by everyone else publicly. This is why we advise Disney+ to connect more with their users that are active twitter users by replying, reacting(like), or even retweet their tweets. This makes their Twitter users feel like they gained Disney's attention and are connected to them. Disney+'s replies make them feel like they are part of the conversation and the retweets made them feel like their opinion is well respected.

In the end, we can see that Disney+ is doing well on Facebook, Instagram, and Twitter, but not as well as their competitors like Netflix. Thus, we advise them to increase user engagement on Facebook, increase popularity on Instagram by sparking conversations in comments, and gain a loyal user base on Twitter through tweets. We would not deny the fact that Disney+ is doing better and better among the competition, by offering better packages and prices, but in the end everything is about the users. Their digital advertising was heavily invested on Facebook, and they should maintain it. Instagram is outperforming, which is why they should explore it. Twitter is the combination of Facebook and Instagram, so they should play safe and steady by securing a loyal user base. We believed that the implications of our analytics and the advice, will take Disney+'s social media marketing and usage to a whole other level and close the gap between them with the competitors.

4. Research Topic 2 : Date Time Analysis with Disney+ Instagram

4.1 Methodology

In this research topic, we are interested in discovering the actions of social media from followers of Disney+ by quantifying the support that they are able to show through social media such as the amount of likes, comments, and views of the posts from Disney+ social media account. In this case, the social media platform that contains all these 3 actions would be the Instagram social media platform. The reason for identifying the amount of likes, comments, and views of each post is to compare it with one another and look for reasons for why a particular post has higher counts compared to the other. Of course, comparing those quantities would not provide any valuable outputs. That is why another important factor of this analysis topic is the date and time of the posts. This is because we believe that the date and time to post content is crucial for the amount of support that they receive. Many followers from Disney+ may not be able to view posts posted on the exact time due to multiple reasons such as time difference of countries, followers may have overflowing posts from different following accounts on their feed, and many more. Due to Disney+ expanding to the worldwide market, the date and time to post contents is extremely important for them to capture and keep their audience updated.

We decided to approach this analysis by performing the scraping and other processes with Instagram as Instagram is currently one of the leading social media platforms and scraping data from it can certainly reward a high value of insight or predictions. We would first scrape all the contents from the Disney+ social media account, capturing all necessary data points that we could but the most important data points needed are the amount of likes, comments, and views as well as the upload date of each post. We do not filter out any posts from Disney+ Instagram account as we believe that the more data we can obtain, the higher the value of the output would be.

Now, we are interested in discovering the patterns and trends of the support from Disney+ followers through social media. We would resume our approach by using the four main data points to assist us in the analysis. Mainly, amount of likes, comments, and views acting as the respondent variables against the dependent variable, upload date. The reason the data points are labelled as dependent and respondent variables is because of the nature of our analysis as we are planning to perform multiple plot and graph visualisations to spot out the patterns and trends and come up with solutions for Disney+. Naturally, plots would then be produced and we would investigate that if as time progresses, the amount of likes, comments, views would be in an increasing manner or decreasing manner from the data scraped. We would not stop here if a strong output pattern or trend is identified.

The approach would continue to find out the average likes, comments, and views of each post grouping by the month, day name, and the hour of the upload date of each post. By doing so, we would be able to obtain a stronger result such that it shows the highest and lowest average of the support gained on which month, day, and hour from previous uploaded contents.The average data in this case is deemed to be superior to figure out the most popular date and time for Disney+ to post important contents or updates so that it can reach the most audience it possibly can. A total of multiple visualisations are essential in this research topic to help us visualise and support our deduction or predictions as well as suggestion and advice to provide to Disney+ Instagram social media account.

4.2 Analysis

4.2.1 Scraping

Since we are interested in scraping data from Disney+'s Instagram social media account, we will be utilizing the "instascrape" scraping tool to scrape the data from Instagram. It does not require any API to use it but certain module imports will be needed.

```
from selenium.webdriver import Chrome
from instascrape import Profile, scrape_posts
import pandas as pd
```

Figure 19: Imports of insta data scraping

For scraping the data, we would approach this by importing the instascrape module and the selenium webdriver that uses Google Chrome to automate the scraping process. Other than that, the pandas library is also imported in order to create dataframes from the data scraped with the "instascrape" tool as shown in Figure 19.

```
# Creating our Google Chrome webdriver
webdriver = Chrome("C:/Users/Asus/OneDrive/BSDA/Chrome Driver/chromedriver.exe")

# Scraping DisneyPlus's profile
SESSIONID = '1390298927%3ATKpvSMWZNX348b%3A2'
headers = {"user-agent": "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Mobile Safari/537.36 Edg/87.0.664.57",
           "cookie": f"sessionid={SESSIONID};"}
disney = Profile("disneyplus")
disney.scrape(headers=headers)
```

Figure 20: Initialize Google Chrome webdriver and scrape Disney+ Profile

In order to utilize the selenium Google Chrome webdriver, we must first download the webdriver through this link. After the completion of the download, unzip the zip file and it is ready to use anytime by specifying the location of the driver. The webdriver line of code from Figure 20 will prompt out a window platform that will be used in scraping data from instagram. Now, an important procedure to take note is to obtain the Instagram session id from your browser. Simply head over to the google chrome instagram page follow the steps at this link to obtain the session id and paste it inside the headers dictionary as shown in Figure 20. Identify and insert Disney+'s username into the Profile function in order to scrape the correct Instagram account.

```
# Scraping the posts
posts = disney.get_posts(webdriver=webdriver, login_first=True)
```

Figure 21: Code for scraping all individual posts

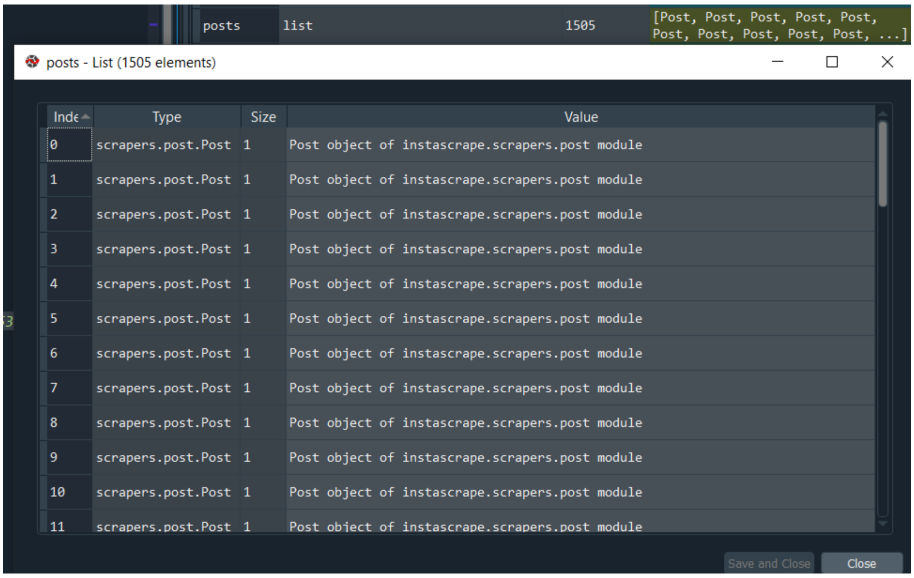


Figure 22: List of individual posts

Moving on would be the most time-consuming process that will get us the data needed for the analysis. According to Figure 21, The "login_first=True" will require us to manually login to our Instagram account before we can scrape the posts. This process should run for a few minutes as it will scrape all the individual posts from the Disney+ account into a list named "posts" as shown in Figure 22 and the total posts that it returned was 1505. The process can also be seen in real-time if we navigate to the Chrome webdriver window due to the nature of Selenium.

```
scraped, unscraped = scrape_posts(posts, silent=False, headers=headers, pause=10)
```

Figure 23: Scrape obtained list of posts

scraped	list	1505	[Post, Post, Post, Post, Post, Post, Post, Post, Post, Post, ...]
unscraped	list	0	[]

Figure 24: List of scraped and unscraped posts

After obtaining the list of individual posts, the final step to get our much needed data is to use the scrape_posts function that we imported earlier to turn the list of individual posts into another list that contains all the data points such as the main attributes likes, comments, views, and many more. As shown in Figure 23, we performed a precaution step to create two list which are namely "scraped" and "unscraped". This is because if an error occurs and some posts are not scraped, those particular posts will be in the list of “unscraped”. A pause of 10 seconds is integrated into the process so that the process will have a 10 second pause between each scrape. This is because we do not want Instagram to temporarily IP block us and interfere with our scrape. Thus, having an interval pause of 10 seconds between each scrape will result in prolonging the scraping process to hours in order to get the “scraped” output list. According to Figure 24, we can see that all the posts are successfully scraped due to “scraped” list containing all 1505 posts and “unscraped” being an empty list.

```
# Turns list into a dataframe
posts_df = pd.DataFrame([post.to_dict() for post in scraped])
```

Figure 25: Turn list into dataframe using pandas library


posts_df - DataFrame

Index	id	shortcode	height	width	ɟating_infı	ɛck_overal	eck_inforı	vity_frictic	a_overlay	media_preview
0	2608308651926914684	CQykuZGroZ8	952	612	None	None	None	None	None	ABsq6N5FjGWIA6ZNRSXKoQPvEjIwR09eSOK54XEkj
1	2607860475814061894	CQw-0kkradG	952	612	None	None	None	None	None	ABsqxqMUVcjHmgZUbVOOTgE+hPoK0NEr1PB69jSVq
2	2607770041293173831	CQwqQk3L3RH	607	1080	None	None	None	None	None	ACoX6GmsCQQpwex64p1FAGcTc56AkehA/AjI/19Ku
3	2607619249999576404	CQwH-Rhsf1U	1080	1080	None	None	None	None	None	ACoqltgPNkB9VP5ipoSB5h9Hb+Qqsh23DD+8oP5Go
4	2607605249278649050	CQwEyiVsqLa	1350	1080	None	None	None	None	None	None
5	2607568877925656181	CQv8hQ4H2Z1	1919	1080	None	None	None	None	None	ABcq59Isn5wQPXFSbYx2ZvwxWjDD5uXbkDgD1P8Ag
6	2607561471422140249	CQv61fCHI9Z	1922	1080	None	None	None	None	None	ABcqwI4GJ+YMBjrg/hipfKQZGGP4EY//AF/yrSimG
7	2607528164849699575	CQvzQz3sUr3	1350	1080	None	None	None	None	None	ACEqy4Yl8suRkg4/TI/rT4VEjbSOvH/1/wAOtLbOC
8	2606987591986334728	CQt4WcKMkwI	1350	1080	None	None	None	None	None	ACEqTUbcrtkQdeGOB+GePwzUEVtvDbQCQOjADb14P
9	2606924577776809797	CQtqBdmnzdF	1080	1080	None	None	None	None	None	ACoqxcVJHHvz7An8qaoqzBIse4tnlSBj1qShhtnAy
10	2606865110171255849	CQtcgGFHgQp	1920	1080	None	None	None	None	None	ABcq5mitJbKMj5nIOPQ8VG1nt5zlSeD2PpTsyboo0
11	2606863788513735960	CQtcM3MMLUY	1350	1080	None	None	None	None	None	ACEqrrbVYW09RWgI8c1Nub/P+fetLmVjLNqfSojBj
12	2606417664271287749	CQr2w5nMYXF	1350	1080	None	None	None	None	None	None
13	2606260027937553942	CQrS6_VhH4W	952	612	None	None	None	None	None	ABsqqUlS71DYCkjGSSf8BVkxr/d/U16qqJtrsZWa1
14	2606135395320935538	CQq21WKrTRy	952	612	None	None	None	None	None	ABsq5mirKQrkAnr68YpzW4BODuA7ildDsVKKspHGR
15	2606048604458853902	CQqi2X4LCYO	1340	1080	None	None	None	None	None	None
16	2605700978396274411	CQpTzv4Mp7r	1080	1080	None	None	None	None	None	None

Figure 26: Instagram posts dataframe

Finally, we turn the “scraped” list into a dataframe so that we can proceed to use the dataframe in our analysis as shown in Figure 25 and 26.

4.2.2 Analytics and Visualizations

Now that we have our dataframe to work with, we can proceed to perform our analysis.

```
import matplotlib.pyplot as plt
from datetime import datetime
import datetime
```

Figure 27: Import matplotlib and datetime library

```python
# Upload date vs Likes Plot
x = posts_df["upload_date"]
y = posts_df["likes"]

fig, ax = plt.subplots()
ax.plot_date(x, y)
fig.autofmt_xdate() #fix date visualization
plt.title('Likes vs Upload Date')
plt.xlabel('Upload Date')
plt.ylabel('Likes')
plt.grid()
plt.show()

# (Year 2020-2021)
fig, ax = plt.subplots()
ax.plot_date(x, y)
fig.autofmt_xdate() #fix date visualization
ax.set_xlim([datetime.date(2020, 1, 1), datetime.date(2021, 1, 1)])
plt.title('Likes vs Upload Date (Year 2020-2021)')
plt.xlabel('Upload Date')
plt.ylabel('Likes')
plt.grid()
plt.show()
```

Figure 28: Code for Upload Date vs Likes plot visualisation

Firstly, the first approach would be to identify the upload dates and likes of each post and plot the upload date as the x-axis and the likes of each post as the y-axis into a scatter plot graph. Library matplotlib and datetime would assist us in producing those visualisations necessary as shown in Figure 27. According to Figure 28, we must first instantiate the x and y for the plot, which are upload date and likes respectively in order to generate the plot. In this case, the first plot to produce is by plotting each post into a scatter plot with the upload date and likes of each post. The second plot to generate at the same time is to view how Disney+ performs in terms of the likes they receive in the year 2020. "fig.auto_fmt_xdate()" is used to organize the x-axis visualisation in order to avoid overlapping of the dates. Other than that, "ax.set_xlim()" refers to the date to be plotted and in this case the start date and end date is specified which are 1st of January 2020 and 1st of January 2021 respectively. Matplotlib module provides multiple customizations to plot and graphs such as the naming of labels, title, and showing grid lines for the plot. We plan to utilize all of them in our plot to have a better and clearer understanding of the output.

```
# Upload date vs Comments Plot
x = posts_df["upload_date"]
y = posts_df["comments"]

fig, ax = plt.subplots()
ax.plot_date(x, y)
fig.autofmt_xdate() #fix date visualization
plt.title('Comments vs Upload Date')
plt.xlabel('Upload Date')
plt.ylabel('Comments')
plt.grid()
plt.show()

# (Year 2020-2021)
fig, ax = plt.subplots()
ax.plot_date(x, y)
fig.autofmt_xdate() #fix date visualization
ax.set_xlim([datetime.date(2020, 1, 1), datetime.date(2021, 1, 1)])
plt.title('Comments vs Upload Date (Year 2020-2021)')
plt.xlabel('Upload Date')
plt.ylabel('Comments')
plt.grid()
plt.show()
```

Figure 29: Code for Upload Date vs Comments plot visualisation

According to Figure 29, now we would want to investigate how Disney+ performs in terms of comments received from its audiences. Likewise, we visualise the same plots for comments against the upload date of each post. An important matter to look for is to reinitialize the x and y axis to the correct columns from the dataset unless a different data point is used. As shown in Figure 29, we used the same x and y series object to store our data point. Thus, we would need to instantiate the x and y axis again each time we want to use a different data point against the upload date. For the comments section, we repeat the steps for likes and produce two plots of comments against the upload date. One as a whole and another for the year 2020. The customization steps are repeated according to the requirements of the plot.

```
# Upload date vs Video view count Plot
x = posts_df["upload_date"]
y = posts_df["video_view_count"]

fig, ax = plt.subplots()
ax.plot_date(x, y)
fig.autofmt_xdate() #fix date visualization
plt.title('Video view count vs Upload Date')
plt.xlabel('Upload Date')
plt.ylabel('Video view count (million)')
plt.grid()
plt.show()

# (Year 2020-2021)
fig, ax = plt.subplots()
ax.plot_date(x, y)
fig.autofmt_xdate() #fix date visualization
ax.set_xlim([datetime.date(2020, 1, 1), datetime.date(2021, 1, 1)])
plt.title('Video view count vs Upload Date (Year 2020-2021)')
plt.xlabel('Upload Date')
plt.ylabel('Video view count (million)')
plt.grid()
plt.show()
```

Figure 30: Code for Upload Date vs Views Count

According to Figure 30, the same steps are repeated in this case for view count on each video post posted by Disney+. Instantiate the x and y axis again to plot the graph with views data point as the y-axis. Two scatter plots will also be produced in this section and that totals up to six visualisations that may provide some significant value of result.

```
posts_df['year'] = posts_df['upload_date'].dt.year
posts_df['month'] = posts_df['upload_date'].dt.month
posts_df['day'] = posts_df['upload_date'].dt.day
posts_df['hour'] = posts_df['upload_date'].dt.hour
posts_df['minute'] = posts_df['upload_date'].dt.minute

posts_df['weekday'] = posts_df['upload_date'].dt.weekday
posts_df['day_name'] = posts_df['upload_date'].dt.day_name()
posts_df['dayofyear'] = posts_df['upload_date'].dt.dayofyear

posts_df = posts_df.set_index('upload_date')
```

Figure 31: Code for creating other date time data points

Our approach is to analyze the time series against data points such as likes, comments, and views. Upload date itself does not give us that much specific value for us. Thus, we decided to break the upload date into all parts of data points that we could get and use those data to perform more visualisations.

According to Figure 31, we use the upload date data point as our main column and perform some processing to create more data points into the "posts_df" dataframe. Firstly, we want the exact year, month, day, hour, and minute from the upload date. Next, our approach is to also obtain the weekday, day name, and the day of the year from the upload date. These columns will then be generated as simple as a line of code for each specific data point as shown in Figure 31. Lastly, we would relocate the upload date to be our index of the dataframe.

```
import calplot
```

Figure 32: Import calplot library for heatmap

```
# Heatmap
likes = pd.Series(posts_df.likes)
comments = pd.Series(posts_df.comments)
video_view_count = pd.Series(posts_df.video_view_count)

calplot.calplot(data=likes, cmap='Greens', suptitle='Calendar Heatmap for Likes')
calplot.calplot(data=comments, cmap='Reds', suptitle='Calendar Heatmap for Comments')
calplot.calplot(data=video_view_count, cmap='PiYG', suptitle='Calendar Heatmap For Video View Count')
```

Figure 33: Code for Heatmap Visualisation

Moving on, we would want to approach the research topic by providing different kinds of visualisation possible to illustrate the importance of support gains from audiences. After thoroughly researching different kinds of visualisation, we decided to perform a heatmap plot to see how the result differs.

First, we import the calplot module for us to produce the heatmap as shown in Figure 32. Secondly, we create three new variables namely "likes", "comments", and "video view count". These three variables contain upload date as their index and like, comment, and view count respectively in their own variables as shown in Figure 33.
We furthered our analysis by plotting the heatmap using the calplot library. Firstly, insert the data according to which variable that was created earlier and specify three different colours for the heatmaps with "cmap" function. In addition to that, we insert the name for each heatmap with the "suptitle" function. A total of 3 plots will be generated from the code as shown in Figure 33.

```python
# Average Likes,Comments,Views (bar/scatter plot)
'''
Month
'''
posts_month = posts_df.groupby('month').mean()

plt.bar(posts_month.index, posts_month['likes'])
plt.xlabel('Month')
plt.ylabel('Likes')
plt.title('Likes vs Months', y=1.1)
plt.grid()
plt.show()

plt.bar(posts_month.index, posts_month['comments'])
plt.xlabel('Month')
plt.ylabel('Comments')
plt.title('Comments vs Months', y=1.1)
plt.grid()
plt.show()

plt.bar(posts_month.index, posts_month['video_view_count'])
plt.xlabel('Month')
plt.ylabel('Video View Count')
plt.title('Video View Count vs Months', y=1.1)
plt.grid()
plt.show()
```

Figure 34: Code for Month vs Average Likes, Comments, Video View Count Bar Plot

posts_month - DataFrame

month	height	width	is_video	tion_is_edi	ınked_comr	comments	nents_disa	g_disabled	timestamp	likes	wer_has_li	ve
1	973.951	917.235	0.490196	0.235294	0	613.794	0	0	1.59246e+09	70049.1	0	0
2	1027.45	950.929	0.482143	0.125	0	724.429	0	0	1.60059e+09	80456.7	0	0
3	970.102	947.492	0.516949	0.144068	0	660.864	0	0	1.60121e+09	84544.8	0	0
4	1051.95	958.743	0.514286	0.307143	0	512.457	0	0	1.60218e+09	74303.2	0	0
5	994.762	935.205	0.503311	0.211921	0.456954	530.51	0	0	1.60541e+09	79820.3	0	0
6	1025.45	942.033	0.528455	0.227642	0.788618	518.707	0	0	1.61725e+09	67508.5	0	0
7	992.523	928.123	0.430769	0.246154	0.0923077	684.477	0	0	1.59764e+09	65887.3	0	0
8	1071.56	972.351	0.467532	0.188312	0	657.994	0	0	1.5866e+09	60854.7	0	0
9	1053.72	963.901	0.425743	0.207921	0	571.337	0	0	1.59253e+09	61279.7	0	0

Figure 35: "posts_month" dataframe

According to Figure 34, we will now create a new dataframe "posts_month" that loads the previous dataframe and groups it by the month data point. Moreover, we will add a "mean()" function to calculate all the means for each data point. The calculated means would be crucial in our journey to create the visualisations. The newly created dataframe is as shown in Figure 35.

Using matplotlib library, create the first bar plot by specifying the index month as the x-axis and the average likes calculated as the y axis. Rename the x-and y-axis according to the data points used. Provide a title to avoid messing up the plots. Use the grid function and show function at the end to print the plot. Repeat the steps and create the second and third plot for comments and video view count on the y-axis as shown in Figure 34.

```
'''
Day Name
'''
posts_dayname = posts_df.groupby('day_name').mean()

plt.scatter(posts_dayname.index, posts_dayname['likes'])
plt.xlabel('Day Name')
plt.ylabel('Likes')
plt.title('Likes vs Day Name', y=1.1)
plt.grid()
plt.show()

plt.scatter(posts_dayname.index, posts_dayname['comments'])
plt.xlabel('Day Name')
plt.ylabel('Comments')
plt.title('Comments vs Day Name', y=1.1)
plt.grid()
plt.show()

plt.scatter(posts_dayname.index, posts_dayname['video_view_count'])
plt.xlabel('Day Name')
plt.ylabel('Video View Count')
plt.title('Video View Count vs Day Name', y=1.1)
plt.grid()
plt.show()
```

Figure 36: Code for Day Name vs Average Likes, Comments, Video View Count Scatter Plot

posts_dayname - DataFrame

day_name	height	width	is_video	tion_is_edi	anked_comr	comments	nents_disa	g_disablec	timestamp	likes	wer_has_li
Friday	1008.66	925.369	0.55	0.192308	0.111538	619.735	0	0	1.59894e+09	70417.6	0
Monday	986.703	941.397	0.473684	0.167464	0.0909091	590.928	0	0	1.59759e+09	71941.7	0
Saturday	1004.47	914.62	0.604651	0.217054	0.0813953	557.922	0	0	1.59435e+09	59880.6	0
Sunday	976.124	992.516	0.33871	0.209677	0.107527	559.769	0	0	1.59912e+09	79447.9	0
Thursday	1041.76	940.476	0.452381	0.214286	0.17619	568.857	0	0	1.5996e+09	66100.2	0
Tuesday	1025.52	947.653	0.491329	0.202312	0.115607	618.341	0	0	1.59906e+09	70669.3	0
Wednesday	1082.04	933.895	0.555024	0.263158	0.124402	620.005	0	0	1.59724e+09	63807.6	0

Figure 37: "posts_dayname" dataframe

Similarly in this case, we are interested in looking at the plots for day name against average likes, comments, and video view count. Likewise, create a new dataframe from the initial "posts_df" dataframe and group it according to the day name while also specifying the "mean()" function to get the average of all the data points. The new dataframe would then have the day name as the index. The newly created dataframe is as shown in Figure 37.

To have a change in plotting the visualisations, we decided to perform scatter plots to see how the plots will differ compared to bar plots. According to Figure 36, The steps are similar for the three plots of average likes, comments, and video view count. Firstly, insert the x--axis and y-axis into "plt_scatter(x,y)" function, namely the index day name and average likes respectively. Correctly label the x and y axis as well as the title for the plots. Repeat the steps to create the other two plots for y-axis average comments and video view count while keeping the same x-axis, day name.

```
'''
Hours
'''
posts_hours = posts_df.groupby('hour').mean()

plt.bar(posts_hours.index, posts_hours['likes'])
plt.xlabel('Hour')
plt.ylabel('Likes')
plt.title('Likes vs Hour', y=1.1)
plt.grid()
plt.show()

plt.bar(posts_hours.index, posts_hours['comments'])
plt.xlabel('Hour')
plt.ylabel('Comments')
plt.title('Comments vs Hour', y=1.1)
plt.grid()
plt.show()

plt.bar(posts_hours.index, posts_hours['video_view_count'])
plt.xlabel('Hour')
plt.ylabel('Video View Count')
plt.title('Video View Count vs Hour', y=1.1)
plt.grid()
plt.show()
```

Figure 38: Code for Hour vs Average Likes, Comments, Video View Count Bar Plot

posts_hours - DataFrame

hour	height	width	is_video	tion_is_edi	anked_comr	comments	nents_disa	g_disablec	timestamp	likes	wer_has_li	ve
0	983.26	912.947	0.550296	0.213018	0.118343	587.775	0	0	1.59683e+09	64414.8	0	0
1	1015.44	943.796	0.532847	0.284672	0.131387	458.511	0	0	1.59936e+09	62456.3	0	0
2	1040.61	913.682	0.522727	0.159091	0.113636	456.864	0	0	1.6001e+09	61375.7	0	0
3	1010.81	927.404	0.596154	0.201923	0.0769231	427.298	0	0	1.59815e+09	55442.8	0	0
4	987.931	952.851	0.465347	0.158416	0.148515	417.881	0	0	1.60076e+09	60891.4	0	0
5	1021.44	938.273	0.522727	0.238636	0.0909091	485.898	0	0	1.59742e+09	64215.1	0	0
6	1006.56	959.5	0.486111	0.111111	0.125	603.5	0	0	1.59767e+09	65244.2	0	0
7	990.956	946.615	0.472527	0.186813	0.10989	638.275	0	0	1.59718e+09	76552.9	0	0
8	949.204	912.584	0.460177	0.221239	0.106195	643.301	0	0	1.59846e+09	83582.1	0	0
9	1050.48	955.624	0.4	0.2	0.117647	859.318	0	0	1.59816e+09	79475.3	0	0

Figure 39: "posts_hours" dataframe

Moving on, we decided that hours for the contents to be posted on social media is crucial so the hour is now taken into account to perform an analysis of visualisations with the average likes, comments, and video view count. Likewise, a new dataframe is created by grouping the initial dataframe, "posts_df" by the hour column and averaging all the other data points. The newly created dataframe is as shown in Figure 39.

According to Figure 38, similar processes are implemented but the bar plot has made its return in this case. We insert the x and y axis into the "plt.bar(x,y)", hour index column and average likes respectively. Correctly label the x-axis, y-axis, and the title of the plot. Perform similar changes for average comments and video view count. Finally, produce 3 plots to complete the visualisations.

4.3 Results

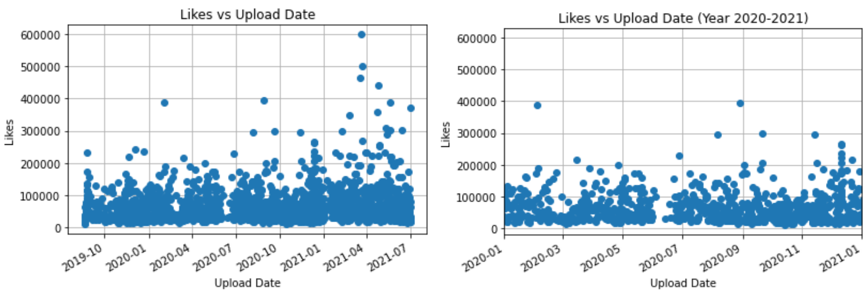


Figure 40: Plots for Likes vs Upload Date, Upload Date (Year 2020)

To recap, we are interested in investigating the upload date of each post and plot it with the likes, comments, and video view count. According to Figure 40, it is seen that the plots are all over the place with a few posts reaching over the 300 thousand threshold while most of the posts receive likes of under 200 thousand. There is only one post that reached the highest like account that they had in the year 2021 with 600 thousand likes and more. Similarly in the year 2020, most of the posts received an amount of likes under 200 thousand while no other posts received above an amount of 400 thousand.

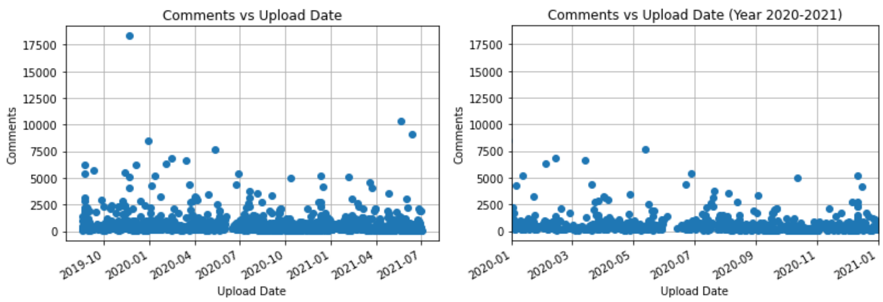


Figure 41: Plots for Comments vs Upload Date, Upload Date (Year 2020)

Looking at Figure 41, there is only one post with a number of higher than 17500 comments. Other posts all fall below the 12500 comment threshold. Most of the posts receive under 2500 comments. Likewise in the year 2020, the comments received does not even exceed 10000 comments with only one post that got over 7500 comments.

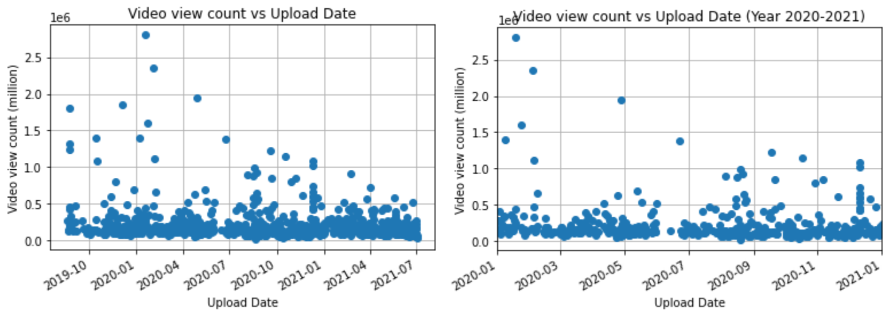


Figure 42: Plots for Video View Count vs Upload Date, Upload Date (Year 2020)

According to Figure 42, the video view count has the highest count as compared to likes and comments. Most of the posts contain views that fall under the 1 million benchmark. However, quite a few posts rack up some high numbers of views with the highest achieving more than 2.5 millions views in the year 2020. We can also see that in the year 2020, the first quarter of the year contains some of the highest view count that Disney+ has.

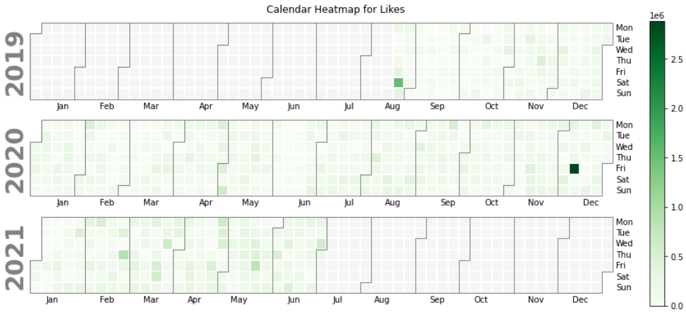


Figure 43: Calendar Heatmap for Likes

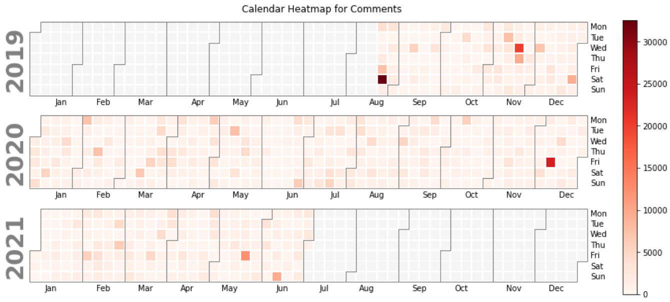


Figure 44: Calendar Heatmap for Comments

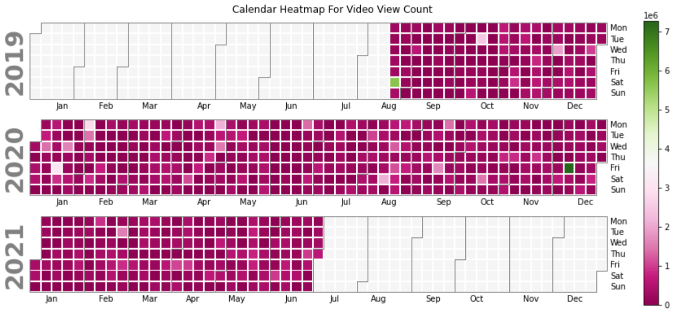


Figure 45: Calendar Heatmap for Video View Count

We decided to have a different approach in visualisation and obtained 3 heatmap calendars for us to investigate on. The 3 visualisations consist of the three main data points, likes, comments, and video view count. Looking at Figures 43, 44, and 45, there are no specific patterns as most of the plots are in the similar contrast of colors except for some blocks that have a different or darker colour indicating a high count for the responding variables. For the likes calendar heatmap, the highest count can be seen on a Friday of December, Year 2020. The comments section have their highest count on a Saturday of

August, Year 2019. Lastly, the highest video view count falls similarly on a Friday in December, Year 2020.

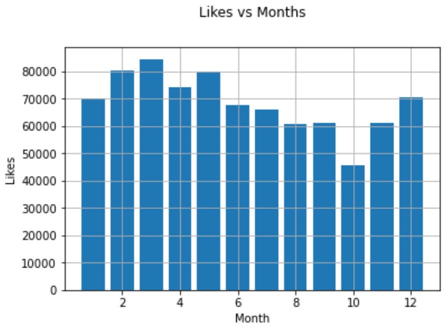


Figure 46: Bar Plot for Average Likes vs Month

Moving on to the average data points that we had plotted to investigate further. Looking at Figure 46, a bar plot for average likes against month is generated. From the plot, we can see that the highest average like count is in the third month, March followed by February, May and so on to the lowest average month October. The top three highest average likes count all contain average likes of over 80 thousand while the lowest average likes month had just achieved under 50 thousand average likes.

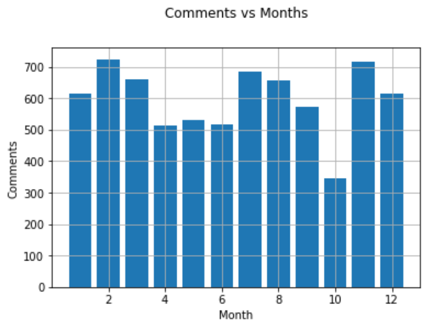


Figure 47: Bar Plot for Average Comments vs Month

According to Figure 47, we can see that the highest average comments is on the month of February with November slightly losing out on a few comments. Both of the months mentioned received over 700 comments while the lowest month, October receives just below 400 comments. Other months received total comments ranging from 500 to 700 on average.

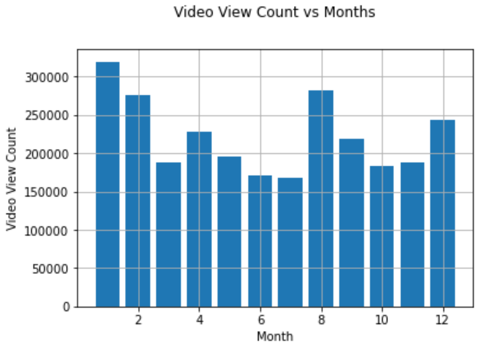


Figure 48: Bar Plot for Average Video View Count vs Month

As seen in Figure 48, the month of January obtained the highest average video view count followed by August and February all amounting over 250 thousand views. The lowest video view count falls under the month of July with a little over 150 thousand views. Other months of average video view count all ranging around 150 thousand to 250 thousand views.

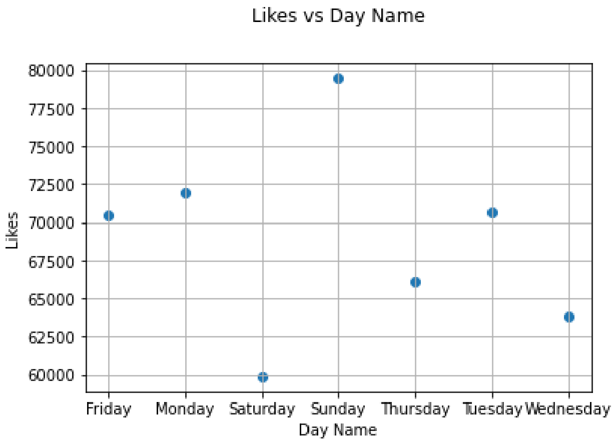


Figure 49: Scatter Plot for Average Likes vs Day Name

Next, we are interested in looking at which day performs the best in terms of average likes, comments and video view count. Looking at Figure 49, we can clearly see the winner here which is Sunday totalling up to an amount of a little below 80 thousand average likes. The lowest average likes received falls under Saturday with 60 thousand average likes. Other days receive average likes ranging from 62.5 thousand to 72.5 thousand.

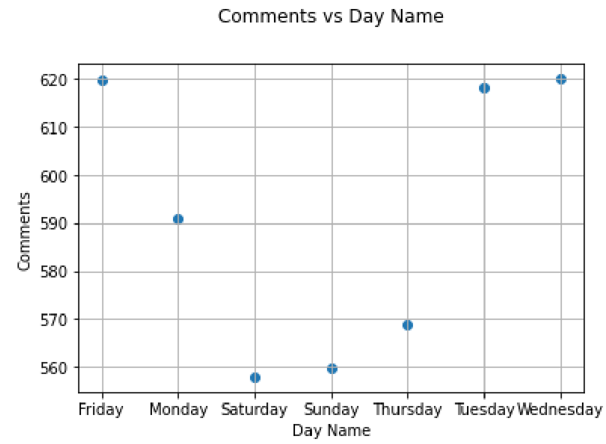


Figure 50: Scatter Plot for Average Comments vs Day Name

As for the comments section, a close competition is seen on Figure 50 with Friday and Wednesday having the same amount of average comments, 620. Similarly, Saturday remains at the bottom receiving below 560 average comments in this case.

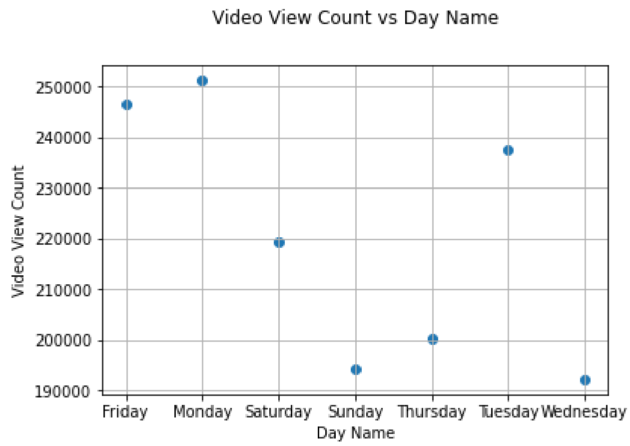


Figure 51: Scatter Plot for Average Video View Count vs Day Name

According to Figure 51, Monday has the highest average video view count with a number of over 250 thousand followed by Friday, Tuesday, Saturday, Thursday, Sunday, and finally Wednesday. The lowest average video view count consists of only slightly above 190 thousand views on Wednesday.

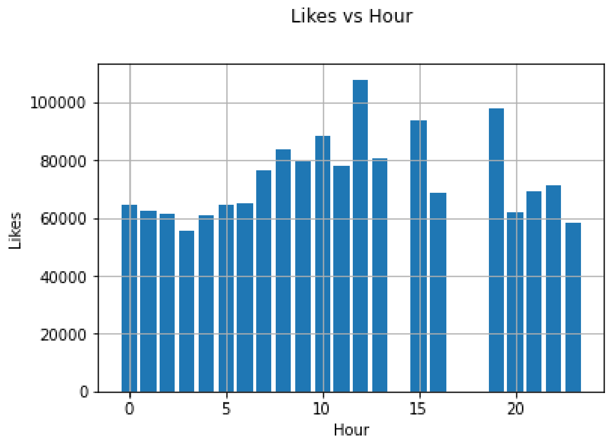


Figure 52: Bar Plot for Average Likes vs Hour

Other than that, we furthered our analysis to look at the hours against average likes, comments, and video view count. The visualisations utilizes the 24 hour system. According to Figure 52, we can clearly see that on the 12th hour which is noon, it has the highest average likes received with a number of over 100 thousand being the only one to go over 6 digits. The lowest average likes amounting to below 60 thousand falls under the 3rd and 23th hour which is 3am and 11pm respectively. Other hours range from 60 thousand to 100 average likes. Looking at the visualisation, we can see that the hour 13th, 17th, and 18th does not have any contents posted on their social media.

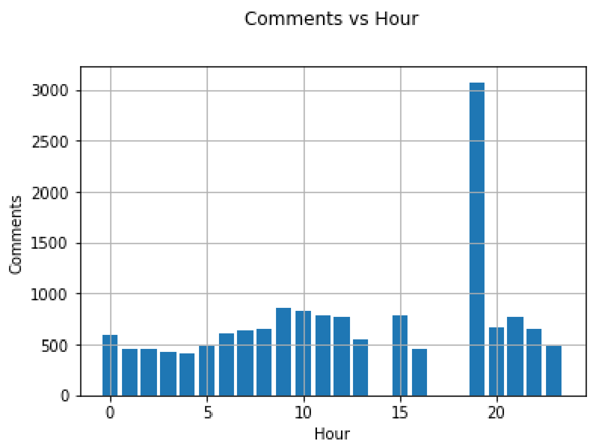


Figure 53: Bar Plot for Average Comments vs Hour

Based on Figure 53, we can see a clear lead on the 19th hour with an average comment of over 3 thousand. Other hours are clearly not close to reaching that 3 thousand benchmark as they range from 0 to only 1 thousand comments. The lowest average comments can be seen on the 4th hour which is 4am based on the Malaysia time. There are no results for the hours, 13th, 17th, and 18th as no contents are posted on those particular hours.

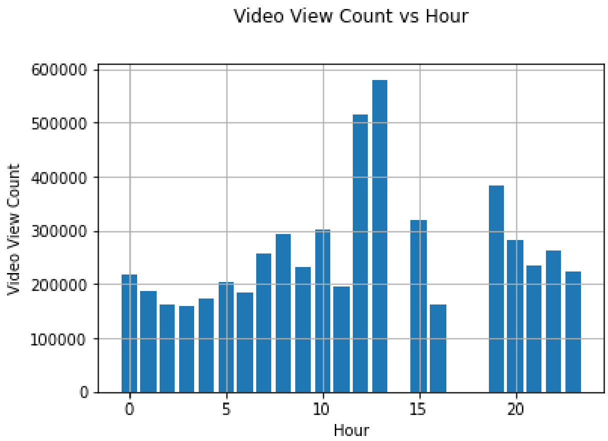


Figure 54: Bar Plot for Average Video View Count vs Hour

Lastly, we have our final visualisation produced that plots the average video view count to hours. Likewise as shown in Figure 54, there is quite a clear lead with slightly below 600 thousand average views that falls under the 13th hour followed closely by the previous hour, 12th of slightly over 500 thousand average views. The lowest average video view count can be seen to have a pretty tight competition among the 3rd and 16th hour totalling below 200 thousand average views. Similarly in the average likes and comments visualisations, there are some hours that did not see any contents posted namely the 14th, 17th and 18th hour.

4.4 Discussions

Before discussing the outputs and results that we obtained, we must look back at our objective as to why we are approaching the matter in this way. Firstly, we wanted to look out the relationship between the upload date and three other main data points which are likes, comments, and video view count from the Instagram Disney social media account. We successfully scraped all 1505 posts from their Instagram social media profile with instascrape tool. The scraped data ranges from the first post to the latest post that we are able to scrape in terms of its upload date, 2019-08-19 and 2021-07-01 respectively. The results are separated into 3 sets or sections to be analysed.

Firstly, from the first six plot visualisations ranging from Figure 40 to 42, we could not really find any significant pattern or trend by looking at the plots. The results of the plots were plotted neither in an increasing or decreasing manner. All of the plottings were congested on the lower half of the visualisation with a few notable ones seen on higher counts. The results were not as expected and we could not identify a strong enough conclusion for the visualisations. We continued to look at only the year 2020 as a whole to look at their performance for that specific year. As a result, we still could not find any sort of patterns even though we shortened the x-axis value to look at an even detailed plot.

Moving on, we decided to perform some calendar heatmaps based on the likes, comments, and video view count. The heatmaps can help us to see on which day and month specifically that Disney+ got good values of support from their followers. The first heatmap containing the likes eventually did not provide any valuable insights as the colour contrasts or brightness of the blocks are almost the same across the heatmap except for a few darker ones as shown in Figure 43. We utilized cmap in calplot to use some other colours for comments and video view count to get a better visualisation on the heatmap. We swapped out the Greens colour scheme and proceeded with Reds and PiYG but the results were still not strong enough to provide a significant conclusion as shown in Figure 44 and 45. The brightness of the blocks are similarly almost the same again.

Finally, our approach is to get the average likes, comments, and video view count and group them into 3 sections, namely months, day name, and hours that will result in 3 new dataframes. Firstly, we decided to perform these analyses from the largest value in terms of the date. Meaning to say we start off with the analysis of months, day name, and narrowing it down to hours. The results produced on the visualisations were enlightening and insightful in this case.

Based on Figures 46 to 48, we found out that on the contents posted on the month of March obtained the most average likes possible followed by February and May. Moreover, February was the top in terms of average comments received followed closely by the month of November. Lastly, January became the leading month of average video view count followed by August and February. On the other hand, we found out that the month of October was receiving poor results from 2 of the 3 plots that we generated, average likes and comments respectively.

To change up things a little, we decided to try out the scatter plot when producing the average likes, comments, and video view count against day names. Based on Figures 49 to 51, we discovered that Sunday received a commanding lead compared to other days for average likes. Wednesday and Monday were tied on the highest average comments and lastly Monday was in the lead for average video view count. On the flip side, Saturday was the only day to be ranked the lowest in terms of average likes and comments and that totals up to 2 of the 3 plots produced in this case. Not to mention that the day Saturday was ranked only in the middle for the final video view count plot.

Moving onto the last dataframe, hours. We switched it back to bar plot for visualisations. According to the final 3 Figures 52 to 54, we found that the 12th hour equivalent to 12pm got the highest average likes followed by 7pm and 3pm. Furthermore, a significant lead can be seen for average comments on the 19th hour equivalent to 7pm. No other timing can be compared to it according to the average comments plot. Finally, we have our last plot that shows the 13th hour equivalent to 1pm having the most average video view count followed closely by 12pm. Nonetheless, we identified that the lowest average counts for likes, comments, and video view count lies under the time of 10pm to 5am. These timing can be seen to always be at the lower end of the plots as shown in Figures 52 to 54. As for the hours that do not consist of any data in the plot are due to no contents being posted during those particular hours, 2pm, 5pm and 6pm.

4.5 Implication

At the final section after the analyses and discussion together with explanations of output, we could establish some solid and strong suggestions. Before moving onto the implications, we can see that Disney+ performs exceptionally well in terms of their views followed by likes received and finally the comments of interaction from their followers. We could also see that the amount of support they receive is not small in terms of numbers with the highest that they achieved based on our results are 599020 likes, 18339 comments, and 2.8 million.

Moving onto the suggestions that we would propose is for Disney+ to post contents carefully with significant considerations and thinking before choosing the right timing to bring content to their audience. In order to do so, we suggest three segments to consider before posting their next blockbuster news or update.

Based on our analyses, we would advise Disney+'s Instagram social media account to post important contents on the month of March as that month produced the highest average likes and comments. Another advice similarly on the month's context is to post contents on the first quarter of the year. This is because the highest average for likes, comments, and video view count all comes under the first quarter of the year. So the planning of posting or releasing exclusive contents should be focused on this timing. On the other side, we would also suggest Disney+ to avoid posting valuable content in the month of October. The reason is because October has the lowest average likes and comments.

Moving on to days that Disney+ should post after identifying the months, it is seen in the output that Disney+ performs exceptionally well on days such as Monday, Tuesday, and Friday. This is because these two days are always performing on average or above average as shown in Figure 49 to 51. The days that Disney+ should refrain from posting important contents are on Thursday and Saturday. These two days have the worst average counts as compared to other days.

Finally, we will move on to suggest what hour Disney+ should post on Instagram. We have identified a few key hours and we suggest that Disney+ should be posting their contents at 12pm, 1pm, 3pm, and 7pm. Our output shows that these timings are essential for Disney+ to hit a high count for average likes, comments, and video view counts. According to our result from Figure 52 to 54, we investigated and came up with an advice for Disney+ to refrain from posting their blockbuster contents for time, 10pm to 5am. Do note that all the timings stated here are in Malaysia time.

To summarise the suggestions given, we recommend Disney+ to post huge updates, important contents, blockbuster clips or teasers, and others in the first quarter of the year. In addition to that, are to post them on days like Monday, Tuesday, or Friday as well as best timings, 12pm, 1pm, 3pm, and 7pm. We believe that if Disney+ made these changes to their posting habits or activities, they can gain an advantage against competitors while boosting their follower count as well as improving their use of social media.

5. Research Topic 3 : How is Disney+ performing on Facebook

5.1 Methodology

Based on the literature review, we can see that among all of the social media platforms, Disney+ invests half of their budget in Facebook advertisement alone. Visual material is an important criterion to increase interaction on social media platforms. Images or videos can easily boost the views, likes, and comments when it is compared to standard text posts, this is because people tend to stop and pay attention to the strong image or interesting video when they are scrolling through their social media. Hence, it is very important for Disney+ to know which type of posts will get more or better response to be strategically successful in their Facebook page. Aside from that, Facebook is one of the social media platforms with the largest user bases. According to Statista, as of April 2021, Facebook has more than 2.7 billion users worldwide, which is also the most used social media platform among all of the famous social media platforms across the globe. Based on these facts, we can say that by fully utilizing ways of promoting on Facebook, Disney+ can easily gain a lot more profit. Because of this, we wanted to know in Disney+ Facebook platform whether posts with images or posts with videos get better response from the customers. So, Disney+ have to be more precise with their way of promoting on Facebook to utilize the advantages. To find out which type of posts in Disney+ Facebook page is more engaging, we wanted to look into the average likes and comments of both types of posts, then compare them. What we will do for this research is pretty simple, which is just to scrape their latest 50 Facebook posts, then calculate and compare the mean like and mean comment for both image and video type of posts.

5.2 Analysis

As mentioned, the steps for this research are not much, it is very minimal and straightforward. In order to scrape data from Facebook, a tool named "facebook_scraper" is used to perform the scraping process. The benefit of using this tool is that we can scrape data from Facebook without needing an API. The feature of this tool is that instead of scraping Facebook posts by the website link, it scrapes Facebook posts of a specific page or user pages by pages. What we can do is to specify a desired number to decide how many pages of data we want to scrape from that specific user or page. Later on, the tool will return the data scraped in the form of a list. Inside the list, we extract the number of comments as well as the number of likes of each post. Other than that, we can differentiate whether the Facebook post is an image post or a video post by looking into the image_id or video_id of every post. If the image_id of a specific post is not blank and actually has a value, it means that the post is actually an image post, and on the other hand, it is the same case of video. Since Facebook's feature allows posts to be text only, or allow posts to have both image and video altogether, conflicts like what if there is actually a post that has both image and video exists together, or vice versa might happen, how do we handle that kind of posts or how do we take into account. However, after investigating thoroughly, this kind of case does not happen in Disney+ Facebook page, which means that there are only two types of posts in their page, either a post with image only, or a post with video only. After scraping the comments and likes amount of the latest 50 posts of Disney+ Facebook page, we categorize them into 2 types of posts, which are image post and video post. Then, we will need to calculate the mean comment and mean like for each category and compare them to see which type of post is doing better.

```
from facebook_scraper import get_posts
import pandas as pd

listposts = []
for post in get_posts("disneyplus", pages=14, options={"comments": True}):
    print(post['text'][:50])
    listposts.append(post)

post_df = pd.DataFrame(listposts)
```

Figure 55: Code snippet for Facebook scraping

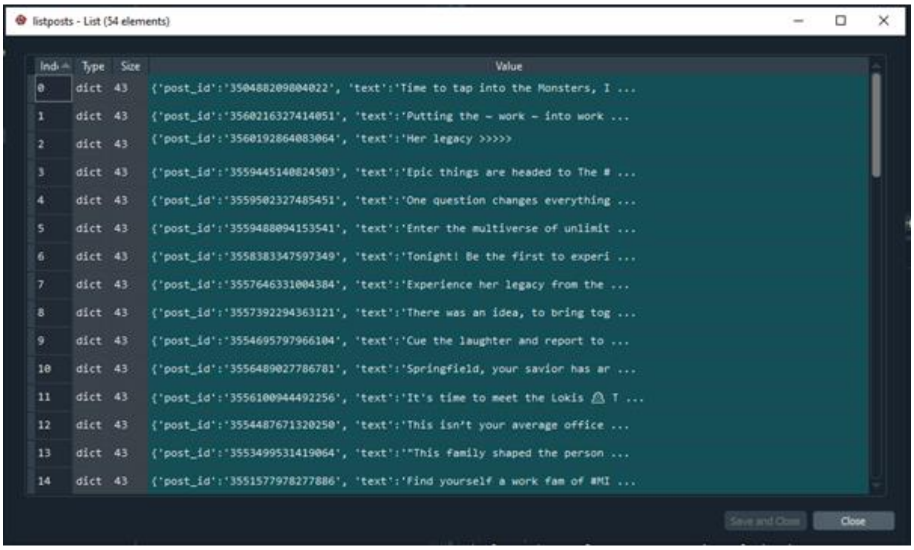


Figure 56: list of returned posts from scraper

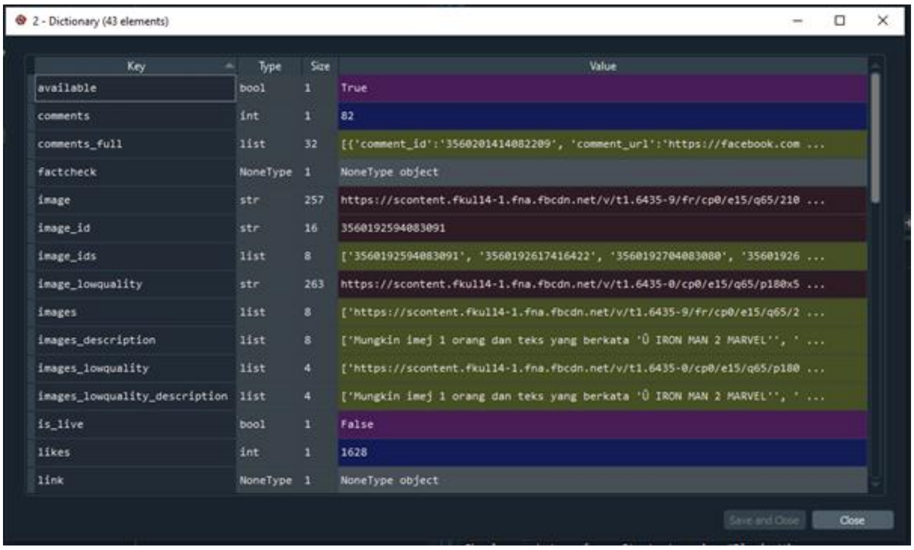
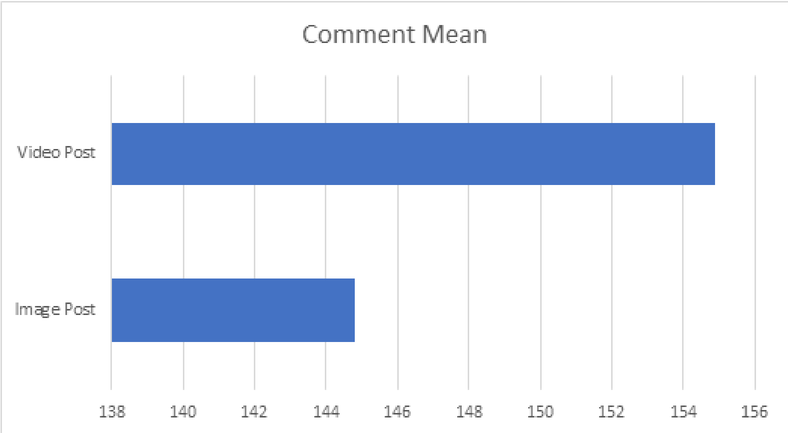


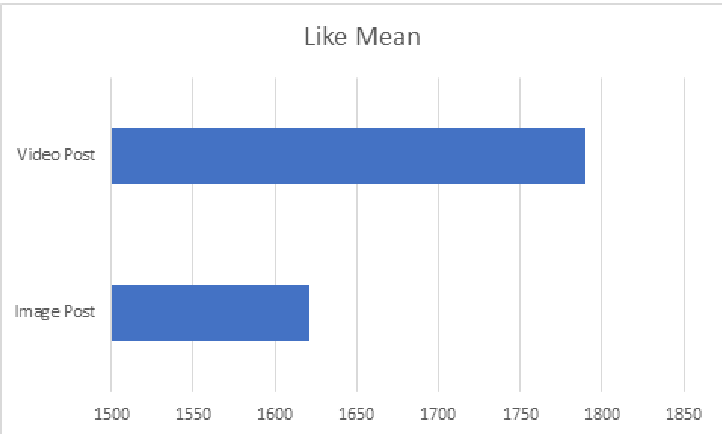
Figure 57: Data of each posts in lists recorded in dictionary format

5.3 Results



Comment Mean	
Image Post	144.81
Video Post	154.91

Figure 58: Mean of comments grouped by type of posts



Like Mean	
Image Post	1621.31

Video Post	1790.26

Figure 59: Mean of likes grouped by type of posts

Based on the first graph and table above, we can see that for both comment and like categories, video type of post stands out more than image type of post by a slight margin. For the comment section, post with image has a mean of 144.81 while post with video has a mean of 154.91. This means that on average, for every post that contains an image, there are approximately 145 people commented on it, meanwhile for every post that contains video, there are approximately 155 people who commented on it. From these data, if we were to compare how the two types of videos are doing for comments, the video type of post has a better performance as compared to the image type of post

Moving on to the second graph and table, which shows the mean like count for both of the two types of posts. For this case, it is similar to the previous case, in which posts with videos are doing slightly better than posts with images. From the table, we can see that on average, there are about 1621 people who comment on every image type of post. On the other hand, for video type of posts, there are about 1790 people to comment on every video type of post. In general, we can say that video posts are overall doing better than image posts. Hence, to conclude, we can say that video type of posts always get more attention and better feedback than image type of posts.

5.4 Discussions

When it comes to determining whether which type of post is the better way of promoting on social media, the answer will always be a myth, and it might be vary for different type of business. Hence, it is important for us to perform social media analytics in order to find out whether how is Disney+ doing on their social media platform and which type of promotion is getting better response for them, specifically on Facebook. The reason why Facebook is chosen among a variety of social media platform is because we found out that 62% of the marketers thinks that Facebook is the most important social media channel for their business. By using Facebook as their main social media platform for their promotion, they are able to extend out to a larger market, not only being able to reach out to their loyal customers, followers or fans.

After generating and comparing the results from the visualization above, we can come to a conclusion, which is for Disney+ on their Facebook page, video type of post will always be the most effective way to attract the audience and to boost their customer engagement. No matter whether it is for the likes amount or comments amount, the outcome is post with video will triumph over post with images all the time. Hence, by looking at the comparison outcome, we can think of a number of possible reasons why video tends to perform better.

If we were to compare both images and videos, they actually have their own unique characteristics that can provide benefits. The first possible reason why video is performing slightly better than image is that they deliver a string of content causing the audience to stop and read or watch it. In other words, videos are able to catch user attention easily because of their simple and straightforward title, attractive thumbnail, or interesting storyline. Other than that, it is human nature to be easily attracted to movement, hence the movement made by videos will naturally draw people's attention.

Moreover, by using video, marketers are able to pass on more messages in a short amount of time. By the nature of video, a bunch of information can be bundled up, then to be conveyed to the customers, making them more memorable. The rationale behind this is that because of human nature, people are more likely to remember a story instead of a list of facts. Thus, we can say that videos have the ability to make it easier to convey a story. The absolute fact is that a video can have sound and movement, which is impossible for an image to do so. Not only can people easily reminisce the story, content, or message of the videos they have watched in the past, they can also enhance their imagination and creativity.

Furthermore, a video tends to get more exposure as compared to an image. This is because video type of content has the ability to induce people to share them. Based on statistics by WordStream, a social video can generate 12 times more shares than text and images combined. When a video utilises its innate advantage, viewers can understand the message of the video easily, leading to more people willing to share the video. As mentioned, internet users are usually more inclined to share videos, thus it will be simple for a video to gain more exposure than an image. Exposure is an essential element for a brand to become more successful as the more brand exposure you have, the more revenue you will be able to make from both existing and new customers.

5.5 Implication

After all the analysis, we can see the pattern of how customers are engaging with Disney+'s Facebook page, As mentioned in the beginning, Disney+ spent almost half of their digital advertising budget on Facebook alone, so it is crucial for them to make sure their marketing strategy on Facebook can be successful, effective and efficient otherwise the huge investment in Facebook alone will be wasted. Hence, there are some recommendations that can be done by Disney+ in order to keep up their good work and improve their marketing strategy on Facebook.

In this topic we researched about the comparison between image post and video post, specifically the average like count and comment count of the two types of posts. For the outcome, we found out that in normal circumstances, video post will always do better than image post, which is video post will have more comments and likes from the community as compared to image post. Thus, Disney+ must make sure that their quality of video content marketing is superior enough to keep their loyal consumers, as well as attract new consumers. The reason behind this is because video content can be used in a variety of ways depending on how the marketers want, and also videos actually have a good return on investment. Without being said, we know that people tend to love video over other marketing material as they are informative, educational, entertaining, and inspiring. Also, videos are extremely good storytellers.

Moving on to the suggestion for Disney+ to improve their current usage of social media, we would suggest they invest more in video marketing. Based on Cisco's annual internet report, online videos make up more than 82% of the consumer internet traffic by 2022. Thus, we can say that video marketing is the current digital marketing trend and it is an essential element for a marketing strategy to be successful. In a nutshell, we believe that Disney+ can enhance their marketing strategy by making use of video marketing, which is to post more engaging videos. Ultimately, they will be able to build their brand reputation and boost their brand awareness.

6. Research Topic 4 : Why should Disney plus retweet more?

6.1 Methodology

The amount of social media platforms coming out every year, social media getting more popular than ever. This has grown to the extent that most businesses are using more of Facebook, Twitter, and Instagram to advertise their business which includes products and services to potential customers. In this research, we found out Disney plus' Twitter activity doesn't include much retweeting. Therefore, we would be looking into Disney plus' Twitter account and how not utilizing an important feature on Twitter could affect them. After that, we would show an example of their competitor,Netflix who took advantage of retweeting.

The research we conducted used a mix of quantitative methodology based on the data collected through scraping data from Twitter using Selenium on Python. The research data was primary data that we collected ourselves. The data that we required before performing any analysis was a large number of tweets found from Netflix and Disney plus Twitter accounts which include whether it was a retweet, captions, reply count, retweet count, and share count. We used this as a standard to measure their engagement rate on Twitter.

In this research, we used the Selenium webdriver on python to scrape 565 ± 10 tweets for each Netflix and Disney plus official Twitter accounts. The process is fully automated like a Twitter bot which starts scraping data from the first tweet and scrolls until the very bottom until the last tweet. The method used was to scrape the tweets using xpath and locate the username, handle, date of post, caption, replies, retweets, and shares. This allows us to get all the data necessary to run through a round of analysis to see the patterns and make our analysis.

The method that we carry out our analysis for this research problem is to first look at how well Netflix and Disney plus perform on Twitter based on statistics such as reply count, retweet count, and like count. We acknowledged that Disney plus is a rather new service from Disney compared to Netflix that is why it is perfect for us to gauge how effectively they are performing. This opened up more possibilities for us to analyze different patterns in how both of them manage their Twitter account. As a result, we can look at how retweeting is important in today's day and age.

This methodology allows us to look at the bigger picture rather than just analyzing data based on a mere few posts. The amount of tweets scraped proves that. It shows us the high and lows or the uncertainties that they face on their social media. We first assess the huge data collected and proceed to find a pattern which fits perfectly into our research question.

6.2 Analysis

Before we analyze the data, we will be showing how we scrape the data from both Netflix and Disney plus Twitter accounts. We used the Selenium webdriver on python for this process. First, a brief introduction on Selenium. Selenium is an open-source portable framework to test web applications. Jason Huggins was the first to invent Selenium in 2004. He was working as an engineer at ThoughtWorks on a web application that required frequent testing. He designed a JavaScript programme that would automatically control the browser's behaviour after seeing that the repetitive Manual Testing of their application was getting increasingly inefficient. He wrote a JavaScript programme to control the browser's operations automatically. This application was given the name JavaScriptTestRunner" by him. He released JavaScriptRunner open-source, later renamed Selenium Core, after seeing promise in the notion to automate other online applications. (What Is Selenium?, 2021) https://www.guru99.com/introduction-to-selenium.html We will be including the Selenium WebDriver as one of the tools since it aligns with the goals we wish to achieve. It implements a modern and solid approach to automated browser actions. It controls the browser by directly communicating with it. The languages that it supports are Java, Python, C# and more. We will be using it together with Python due to the fact that Python is the main programming language we are learning this semester. We would be working with Selenium to scrape data off both Netflix and Disney plus Twitter accounts.

These are the links to both of the accounts:
Disney plus - https://twitter.com/disneyplus
Netflix - https://twitter.com/netflix

```python
import csv
from selenium import webdriver
from getpass import getpass
from time import sleep
from selenium.webdriver.common.keys import Keys
from selenium.common.exceptions import NoSuchElementException
```

Figure 60: Code to import modules and packages

We need to import the required libraries because this allows Python to access code from another module by importing the function using import. The csv module allows us to output our extracted data in a csv format. The webdriver is used to access Selenium. The other functions will be further explained as we demonstrate the code.

```python
PATH = "C:\Program Files (x86)\chromedriver.exe"
driver = webdriver.Chrome(PATH)
```

Figure 61: Code to set up Selenium webdriver

This step is to create an instance of the web driver. We are using the Google Chrome browser so we installed the web driver for Chrome.

```python
driver.get("https://twitter.com/disneyplus")
driver.maximize_window()
```

Figure 62: Code to navigate to Disney plus Twitter account

This code is used to navigate to the Disney plus Twitter account. Then, it proceeds to maximize the window to prevent alignment issues.

```python
sleep(2)

cards = driver.find_elements_by_xpath('//div[@data-testid="tweet"]')

card = cards[0]
```

Figure 63: Code to create variables for tweet

The ‘sleep’ function adds a delay to the automation in seconds depending on the number inputted in the bracket. The ‘cards’ and ‘card’ variables are meant to address each tweet. The ‘xpath’ syntax will be mainly used in this twitter scraper.

```python
def get_tweet_data(card):
    """Extract data from tweet data"""
    username = card.find_element_by_xpath('.//span').text
    handle = card.find_element_by_xpath('.//span[contains(text(), "@")]').text
    try:
        postdate = card.find_element_by_xpath('.//time').get_attribute('datetime')
    except NoSuchElementException:
        return
    comment = card.find_element_by_xpath('.//div[2]/div[2]/div[1]').text
    responding = card.find_element_by_xpath('.//div[2]/div[2]/div[2]').text
    text = comment + responding
    reply_cnt = card.find_element_by_xpath('.//div[@data-testid="reply"]').text
    retweet_cnt = card.find_element_by_xpath('.//div[@data-testid="retweet"]').text
    like_cnt = card.find_element_by_xpath('.//div[@data-testid="like"]').text

    tweet = (username, handle, postdate, text, reply_cnt, retweet_cnt, like_cnt)
    return tweet

    get_tweet_data(card)

    tweet_data = []

    for card in cards:
        data = get_tweet_data(card)
        if data:
            tweet_data.append(data)

    tweet_data[0]

    driver.execute_script('window.scrollTo(0, document.body.scrollHeight);')
```

Figure 64: Code to get the tweet data

Next, we create a function called 'get_tweet_data' to extract data from each tweet. The 2 variables that we are sorting it by are the users' Twitter username and handle. The 'try' block tests the block of code for errors which is the time and date of the tweet. We then handle the error in the 'except' block by using the function 'NoSuchElementException' which returns the tweet's content, reply count, retweet count, and like count. All these elements add up to make a complete tweet. We then test the function on our 'card' which is the tweet. The for loop is to apply the model on all of 'cards' and append the data on our 'tweet_data' list. We have our first set of data collected so now we have to make it continuous scrolling. We executed it using javascript to instruct the browser to scroll down the page until it reaches the end.

```
data = []
tweet_ids = set()
last_position = driver.execute_script("return window.pageYOffset;")
scrolling = True

while scrolling:
    page_cards = driver.find_elements_by_xpath('//div[@data-testid="tweet"]')
    for card in page_cards[-15:]:
        tweet = get_tweet_data(card)
        if tweet:
            tweet_id = ''.join(tweet)
            if tweet_id not in tweet_ids:
                tweet_ids.add(tweet_id)
            data.append(tweet)
```

Figure 65: Code to manage scrolling

We want to extract all available tweets according to the search term while scrolling down. The process will be repeated until we extract all the tweets on the page. The '-15' is to ensure that no repeated tweets are getting extracted because new tweets are constantly loading into the page because the background HTML is not changing. Therefore, we can just look at the last 15 items instead of every single tweet which will save us time by not having to recheck all of it. Despite the fact of not having a tweet ID, we can create one ourselves by concatenating elements of the tweet into one long string which generates a unique identifier. Then an 'if' statement to check and see the tweet ID has already been collected by checking it in a set of tweet ID's and if has not we can append the tweet into the tweet data list and also add the tweet ID into the tweet ID set. The 'last_position' variable is for us to check if we have already reached the end of the scroll region and if the scroll position has not changed we will know we reached the end and can break the loop. This will be explained further.

```
scroll_attempt = 0
while True:
#check scroll position
    driver.execute_script('window.scrollTo(0, document.body.scrollHeight);')
    sleep(1)
    curr_position = driver.execute_script("return window.pageYOffset;")
    if last_position == curr_position:
        scroll_attempt += 1

        #end of scroll region
        if scroll_attempt >= 3:
            scrolling = False
            break
        else:
            sleep(2) #attempt to scroll again
    else:
        last_position = curr_position
        break
```

Figure 66: Code to stop scrolling

We set the variable ‘scroll_attempt’ at 0 to make sure it is not scrolling anymore. If true, a script will run to check scroll length. The ‘curr_position’ is to compare the current position and the last position to check if it is the same which will prompt it to break the loop assuming we reach the end. In order to ensure it works, we need to allow a certain amount of scroll attempts before it finally breaks the loop. If the ‘scroll_attempt’ is greater or equal to 3 we stop the scrolling by setting it ‘False’ which breaks the outer loop. The ‘else’ statement is to check if the scrolls attempted are less than 3 it will ‘sleep’ for 2 seconds and continue cycling the loop. The last ‘else’ statement is to check if the last position that we scrolled is the same as the current position we are at. In the case that it matches, the loop will break which means we reached the end. Now we let the bot run until it has finished scraping all the data.

```
len(data)

##Saving tweet data
with open('disneyplus_tweets.csv', 'w', newline='', encoding='utf-8') as f:
    header = ['Username', 'Handle', 'Timestamp', 'Comments', 'Replies', 'Retweets', 'Likes']
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerows(data)
```

Figure 67: Code to store data

The ‘len’ function returns the data we collected in an object and when the object is a string the function returns the numbers of characters in a string. We also save the data using the csv library by categorizing our data into ‘Username’, ‘Handle’, ‘Timestamp’, ‘Comments’, ‘Likes’, ‘Retweets’, and ‘Text’.

For the Netflix account, the process is similar but with just a few changes to the search term. We can now move on to analyze the results and we can see a pattern.

6.3 Result

	A	B	C	D	E	F	G
1	Username	Handle	Timestamp	Comments	Replies	Retweets	Likes
2	Disney+	@disneyplus	2021-07-01T15:53:45.000Z	Weâ€™re not lying	69	261	1600
3	Disney+	@disneyplus	2021-07-05T14:00:00.000Z	Their mission is	5	20	194
4	Disney+	@disneyplus	2021-07-04T21:00:00.000Z	You ready for this?	42	253	2300
5	Disney+	@disneyplus	2021-07-04T18:00:00.000Z	His spark is back	56	372	2100
6	Disney+	@disneyplus	2021-07-04T17:34:44.000Z	Fireworks, food, and	7	64	633
7	Disney+	@disneyplus	2021-07-02T01:02:13.000Z	Will your memory	11	48	473
8	Disney D23	@DisneyD23	2021-07-01T16:42:16.000Z	NEW: #HSMTMTS	13	86	757
9	Disney and Pixar's Luca	@PixarLuca	2021-07-01T19:00:04.000Z	Ready to take a trip?	205	97	968
10	Disney+	@disneyplus	2021-07-01T19:00:00.000Z	It's time to take	35	142	1800
11	Disney+	@disneyplus	2021-07-01T01:00:00.000Z	This performance	9	51	450
12	Disney+	@disneyplus	2021-06-30T23:36:08.000Z	The future is now.	12	79	719
13	Disney+	@disneyplus	2021-06-30T22:00:00.000Z	It's *almost* time to	16	309	1700
14	Black Widow	@theblackwidow	2021-06-30T20:14:39.000Z	Check out photos fro	32	559	3900
15	Disney+	@disneyplus	2021-06-30T17:00:02.000Z	Can you believe	64	357	1600
16	Disney+	@disneyplus	2021-06-30T16:36:49.000Z		7	48	549
17	Disney+	@disneyplus	2021-06-30T16:36:49.000Z		6	50	505
18	Disney+	@disneyplus	2021-06-30T00:20:10.000Z	Hear from	6	56	503
19	Disney+	@disneyplus	2021-06-29T22:00:01.000Z	Start the shot clock	18	65	598
20	Disney+	@disneyplus	2021-06-29T20:04:37.000Z	In 10 days, the wait	55	687	5800
21	Disney+	@disneyplus	2021-06-29T18:00:00.000Z	Apparently, the God	26	253	1600
22	Disney+	@disneyplus	2021-06-29T16:05:00.000Z	She loves what we	15	138	1000
23	Disney+	@disneyplus	2021-06-29T16:00:03.000Z	Discover the jaw-	2	50	438
24	Marvel Studios	@MarvelStudios	2021-06-29T13:00:05.000Z	this tweet to be the	860	7600	86400
25	Disney+	@disneyplus	2021-06-29T00:20:54.000Z	Homecoming is	12	78	672
26	Disney+	@disneyplus	2021-06-28T20:00:00.000Z	ABSOLUTELY	20	277	1100
27	Disney+	@disneyplus	2021-06-28T15:41:40.000Z	Every Loki has their	29	275	2400
28	Disney+	@disneyplus	2021-06-28T13:00:49.000Z	Next week, these	8	140	873
29	Disney+	@disneyplus	2021-06-28T00:24:49.000Z	.	2	22	170
30	Disney+	@disneyplus	2021-06-28T00:24:09.000Z	A PERFECT	3	16	100

Figure 68: Disney plus scraped data in csv

This is a part of the output on Disney plus that we got from the scrape just now in csv file format. It would be impossible to show all as there are over 500 rows of data collected just for Disney plus.

	A	B	C	D	E	F	G	H
1	Username	Handle	Timestamp	Comments	Replies	Retweets	Likes	
2	Netflix	@netflix	2021-07-02T13:07:45.000Z	The summer of fear is	47	254	1500	
3	ì´ë°™ ë§ˆí‹´ à¸-à¸µà¸§à¸²à¸™à¸¡à¸²à¸£à¹Œà¸•à¸´à¸™	@ivanovskie24	2021-06-30T13:19:08.000Z	Episode 3, 20:00 of #Sex	28	98	654	
4	DK	@nkeleakaDk	2021-06-30T19:10:57.000Z	#sexlife to us who paus	52	113	1100	
5	Netflix	@netflix	2021-07-01T21:19:19.000Z	Maryland School for	53	96	737	
6	Maitreyi Ramakrishnan	@ramakrishnannn	2021-07-01T20:52:11.000Z	I just met a young fan	32	94	2000	
7	Netflix	@netflix	2021-07-01T20:00:02.000Z	Fairly certain you can gu	109	403	2400	
8	Con Todo	@contodonetflix	2021-06-30T15:00:01.000Z	based on ProPublica's	8	33	377	
9	Netflix	@netflix	2021-07-01T15:53:17.000Z	This prince faces a life-	89	343	1700	
10	NetflixFilm	@NetflixFilm	2021-07-01T15:00:04.000Z	John David Washington	20	542	1500	
11	Atypical	@Atypical	2021-06-30T16:45:02.000Z	Cazzie relationship stat	327	5500	15700	
12	Netflix	@netflix	2021-06-30T20:28:49.000Z	don't think you need a r	110	203	3300	
13	The Drew Barrymore Show	@DrewBarrymoreTV	2021-06-30T18:24:34.000Z	Hmmm....this feels	19	80	898	
14	Netflix	@netflix	2021-06-30T19:02:43.000Z	The founding of	96	214	1200	
15	Most	@Most	2021-06-30T18:00:00.000Z	Finally found those I	3	13	222	
16	Netflix	@netflix	2021-06-30T17:29:08.000Z	And this isn't even the	137	204	2500	
17		@filwe__	2021-06-30T17:14:49.000Z	Me watching Too Hot Tc	12	58	655	
18		@protozoamicrobe	2021-06-25T18:07:12.000Z	every person walking	9	197	2000	
19	Jhanoi	@Missoconnor_x	2021-06-28T17:59:14.000Z	Episode 3 shower scene	9	70	529	
20	Keisha	@KeishaMonae5921	2021-06-27T14:59:55.000Z	Episode 3 Gym shower s	11	94	699	
21	Ashley	@ASHD159	2021-06-28T06:02:23.000Z	Sex/Life - Netflix - epis	32	193	972	
22	Netflix	@netflix	2021-06-29T19:04:01.000Z	Which of these best cap	108	117	1200	
23	looseleaf	@SideCG	2021-06-29T18:11:36.000Z	watched	4	15	350	
24	Netflix	@netflix	2021-06-29T17:23:22.000Z	The most important det	138	423	6100	
25	Netflix Canada	@Netflix_CA	2021-06-29T14:34:28.000Z	three more sleeps till fe	39	76	932	
26	Netflix	@netflix	2021-06-29T15:18:16.000Z	These hijackers planed	393	2200	5300	
27	bri (saka stan account)	@Xhakaed	2021-06-28T21:37:10.000Z	Netflix understood the	45	832	4700	
28	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	

Figure 69: Netflix scraped data in csv

This would be the one for Netflix data consisting of 558 rows of records. The empty values or symbols in the username column mean that their name is either an emoji or in a different language such as korean or japanese.

These table outputs show exactly what we defined when scraping. Now we can look at the results it could give.

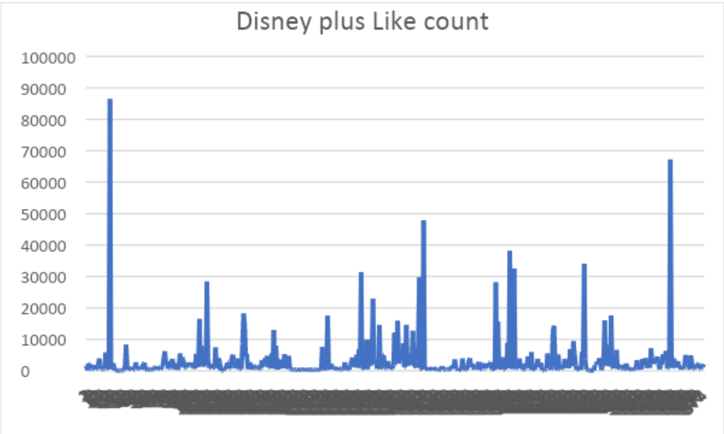


Figure 70: Disney plus like count of all scraped tweets

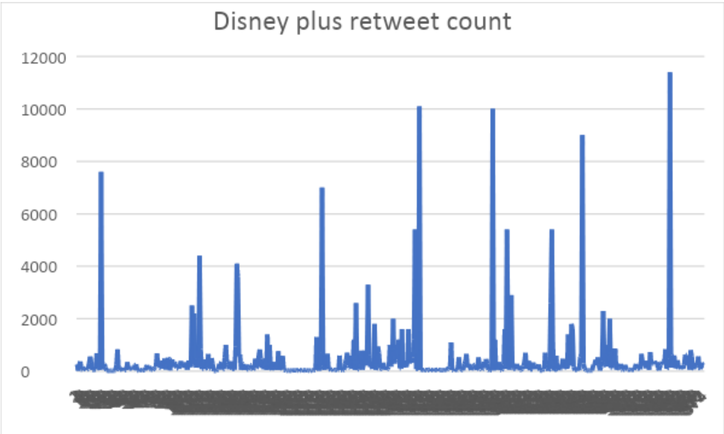


Figure 71: Disney plus retweet count of all scraped tweets

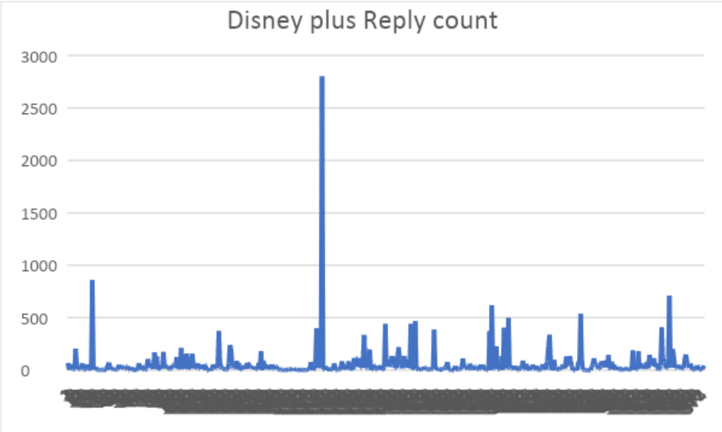


Figure 72: Disney plus reply count of all scraped tweets

	Replies	Retweets	Likes
Total	30158	237453	1641257
AVG	52.54006969	413.6811847	2859.333

Figure 73: Disney plus like, retweet, and reply stats

Disney has a total like count of 1641257, a retweet count of 237453, a reply count of 30158 from 574 tweets on Disney plus' Twitter account. The averages for likes are 2859.333, retweets are 413.681, replies are 52.54. These statistics include retweeted tweets from Disney plus. Disney plus tweets' highest likes have more than 80000 likes, highest retweeted tweet of more than 10000, and the tweet with the highest replies have more than 2500.

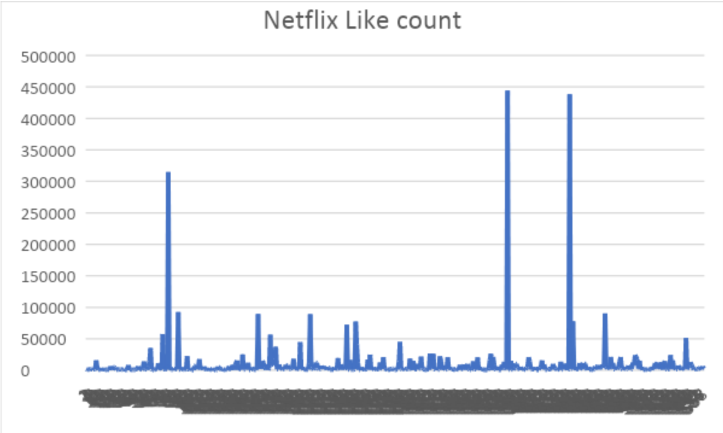


Figure 74: Netflix like count of all scraped tweets

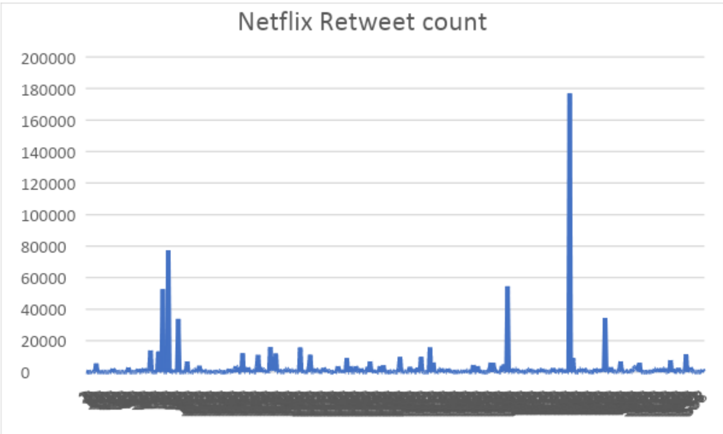


Figure 75: Netflix retweet count of all scraped tweets

Figure 76: Netflix reply count of all scraped tweets

	Replies	Retweets	Likes
Total	99394	997192	4416676
AVG	178.125	1787.082	7915.189

Figure 77: Netflix like, retweet, reply stats

Netflix has a total like count of 4416676, a retweet count of 997192, a reply count of 99394 from 558 tweets on Netflix's Twitter account. The averages for likes are 7915.189, retweets are 1787.082, replies are 178.125. These statistics include retweeted tweets from Netflix.

Netflix tweets' highest likes is close to 450000, highest retweeted tweet is close to180000, and the tweet with the highest replies has more than 10000.

From here, we can see that Netflix has more likes, retweets, and replies than Disney plus' Twitter account. This is due to the fact that Netflix has been in the streaming service industry longer than Disney plus.

Disney plus tweets	489
Disney plus retweets	85

Figure 78: Disney plus tweet stats

This table shows the tweets that appear in Disney plus' Twitter feed. Disney plus tweets their own tweets 489 times and 85 times that they retweet other tweets. This is separate from the retweet count show above as this is the tweets they are retweeting rather than other users tweeting their tweets.

Netflix tweets	287
Netflix retweets	271

Figure 79: Netflix tweet stats

This table shows the tweets that appear in Netflix's Twitter feed. Netflix tweets their own tweet 287 times and 271 times they retweet other tweets. This is separate from the retweet count shown above as this is the tweets they are retweeting rather than other users tweeting their tweets.

6.4 Discussions

We can see that Disney plus and Netflix both have high amounts of interaction from Twitter users based on their likes, retweets, and replies on each tweet with Netflix possessing a huge edge over them. We can also see that both of their Twitter account tweet statistics have highs and lows with different peak times. These highs indicate the new releases they are pushing out gaining more attention and the lows are during the low season when there is nothing new from them resulting in lower hype and interaction from their fans. Netflix is the benchmark that Disney plus should aim for and surpass if they want to be the leading streaming service in the world.

Since the dawn of time, word of mouth has influenced what people buy. A potential customer is more likely to try a service or product if it is loved and recommended by their peers. Brands like Disney must ensure that their tweets are engaging and convincing in order to have an impact and be shared by their followers. A customer will only spread the word if they enjoy or find it informative, and they believe their followers will as well. A retweet of a brand's tweet is more essential than the original tweet. This is because a retweet functions as a recommendation from a company's present followers, assisting the company in forming new relationships and expanding its customer base in the future. If Disney plus has a good grasp of its target demographic's Twitter activity, it may make posts go viral, resulting in more followers and a larger customer base. At the present, Disney plus is trailing behind in those 2 aspects. Disney plus can make sure their tweets are completely optimised if they want to see their messages become viral. To make a tweet more appealing to customers and persuade them to retweet, use photos, tweet at times when they are most active, use hashtags, and add links. The higher the number of retweets, the more likely a post will go viral, resulting in increased brand exposure, prospective new customers, and future purchases. In order to completely connect with potential customers and get the most value from their digital marketing strategy, Disney plus must consider all facets of social media platforms, including Twitter and its retweets.

The second part we are going to look at the statistics of Disney plus and Netflix retweeting other tweets which may be from fans or their subsidiaries. There is a stark difference in this aspect on how both of them run their Twitter accounts. Disney plus only retweeted 85 times out of 574 tweets whereas Netflix retweeted 271 times out of the 558. This shows that Netflix has a higher interaction rate with their fans compared to Disney plus on Twitter. This is very significant because keeping track of retweets, both those that have retweeted your initial tweet and popular ones that are now trending, is a smart idea. It can aid in the future positioning of your branding and social media strategy. Seeing which of your tweets receive the most retweets can help you understand what attracts and engages your target market so you can reproduce it in the future. Understanding the most popular retweets should have a comparable impact on your marketing plan.

In today's world, almost every blogger, business user, and companies on Twitter now tweet their own content, and there's nothing wrong with that as that is what Disney plus is currently doing. However on Twitter, it is a social media platform, not a broadcasting platform. When someone exclusively tweets about themselves, they appear to be someone who is only having a one-sided conversation who only wants to chat about themselves. Retweeting benefits the people you retweet. It gives others both social and practical benefits, such as increasing traffic to their work. It's a lovely gesture that's usually appreciated. This cannot be found in Disney plus' Twitter activity at the moment. It is also beneficial to your following. Tweeting and retweeting useful and fascinating content from social media influencers benefits your followers. When you open the eyes of your followers in discovering new knowledge and information, you can establish yourself as a trusted source of information. Besides that, It helps you get more retweets. Others are more inclined to retweet your stuff if they see you as someone who is trying to help others rather than merely advertising your own content. The likelihood of making others follow your account increases as well. People want their tweets to be retweeted (Pick, 2010). Retweets and other forms of interaction in your tweetstream make you a far more appealing person or entity to follow than someone whose tweets are one-way and clearly automated.

6.5 Implication

In this research, we want to suggest to Disney plus to retweet more on Twitter as it is a powerful tool to take advantage of and we want to prove why they should do so.

For businesses, social media has become a valuable ally. Facebook and LinkedIn are also important to follow, but Twitter, in particular, may help you build a better following by allowing you to engage with a larger audience. The retweet is one of the most important factors. The two distinct parts of a retweet are rather straightforward concepts. The first is when you retweet another person's tweet. The second is when a retweet of your own tweet occurs. Although the concept is simple, the power of a retweet is enormous. In this research, we will be covering both of them but mainly focusing on the first one where a company retweets another person's tweet.

When a person's tweet is retweeted by a company. The company shows recognition for that person thereby showing it is open to others. The company can utilise retweeting as a curation tool to pick and share relevant content by retweeting. Thus, the company shows it is a reliable source, worth following.

When a company retweets a client's compliment, retweeting can be utilised as a strategy for self-promotion. However, mentioning the client in a thank you tweet is preferable to simply retweeting the compliment. The retweet can start a conversation with the individual who was retweeted, allowing a wider range of individuals to participate, which can assist in building a relationship and cultivating a sort of friendliness attitude. You might be promoting someone else's material when you retweet them, but that's not all you're doing. You're also building a rapport with them. It's still a meaningful relationship, even if it's an online one. If you frequently retweet, they may notice you and visit your page. Following that, there's a strong chance they'll start following you and retweeting your tweets as well.

You're suddenly not only marketing your work to your followers, but also to their followers, perhaps resulting in a whole new client base for you. Particularly with people you might not have had the opportunity to interact with if it weren't for Twitter.

You're also marketing your industry when you retweet someone else's tweets. It's not like you're going to just retweet a low-quality tweet; you will definitely filter those out, so you're just sharing the best of your field's information. This will improve people's perceptions of your company simply for being in that industry, which will seem important after a few favourable retweets. The more customers that are attracted to the industry means there is a higher market segment. Furthermore, if you want to liven things up around your company, retweeting something amusing might be a terrific technique to use.

Many twitter users have a significant impact on their followers. When a well-known musician, for example, retweets a video of a fan's cover on YouTube, their fans will share it as well, sky-rocketing the video's views to the millions. So a huge company like Disney plus would have a huge impact on anything they retweeted. Therefore, improving their brand image and forging a closer relationship with their fan base.

It is clear that the retweet's power is so great that it can help you grow your business and bring more attention to your content. All you need to know is how to use it properly. Everything you retweet does not have to be serious. Netflix, for example, does this almost too frequently and they don’t seem to be getting a lot of backlash. Humor is always welcome, as long as it isn't disrespectful, profane, or overheard by everyone. You might come across some industry-related humour or anything relevant to the news (Levy, 2014). Sometimes it can be fine to throw in a few amusing ones now and then to watch how your audience reacts.

7. Research Topic 5 : Utilising Twitter to obtain key insights on suitable Social Media Strategies

7.1 Methodology

When it comes to the social media analysis, it is important that we scrutinise and make full use of the data that is available online. There has been a tremendous increase in social media traffic with users utilising the platform to provide reviews, feedback and to simply air out their comments. This has escalated to a level where there are reviews being provided by specific accounts on social media to complement the movies and items that are being floated around. For example, there are Social Media accounts that focus solely on providing reviews for the movies that have gone by. This will then be shared and reviewed with multiple users then sharing their feedback and comments on the post. This is definitely an interesting insight which we can work with and utilise for the greater good of the brand itself. In this case, we are looking at Disney Plus on Twitter particularly as we plan to utilize the feedback and opinion via the users' tweets to drive our analysis.

Our method when approaching this research topic is to utilise the Twitter platform and to obtain Tweets that had the phrase "Disney Plus". We then scraped the data and obtained sufficient tweets we ran a sentiment analysis so as to identify the general feelings that each users have towards Disney Plus. Hence, our Research Topic of "Utilising Twitter to obtain key insights on suitable Social Media Strategies". Through this analysis, we are hopeful that we are able to obtain sufficient data which will ultimately help in provided key feedback based on the Tweets. The Tweets that are analysed would include Tweets that are tweeted, retweeted and even comments. We would then use this to obtain the positive and negative words so as to go about our sentiment analysis.

As part of the research, we had utilised Tweepy and Selenium to further our analysis. We utilised Tweepy to analyse 7000 tweets. We believe that the number of Tweets obtained is more than sufficient and it would be best utilised to provide and credible analysis as to the actual sentiment amongst the users.

Prior to going about the analysis we had conducted extensive research as to the various pre-processing phases. This would include minor areas so as to ensure that links, names, duplicate rows, punctuation, stopwords can be removed. To extend our research, we will also look into popular hashtags that are being used alongside with a quick analysis of the Confusion Matrix. The Confusion Matrix would be a good strategy when it comes to obtaining the summary of the prediction results on a couple of classification problems. We were quick to include analysis on the TF-IDF (Term Frequency - Inverse Document Frequency) and the BOW (Bag of Words).

We believe that this analysis is a step in the right direction as it provides us with the raw feelings and sentiments of the users. Hence, allowing us some form of credible info that can be provided to the brand as a recommendation. With the huge pool of tweets that is to be scrapped, it would be crucial for us to assess and analyse the darta and proceed to find credible and key insights which can fit our research question.

7.2 Analysis

The main basis of our analysis revolves around utilising Tweepy and Selenium to go about the Twitter Sentiment Analysis. We have identified that text is in abundance in the form of opinions, news and many more. This abundance of text is mostly in an unstructured format which can then be used to benefit society and organisations. We utilised Natural Language Processing (NLP) which includes text processing and analysis. NLP is a popular library for text processing in Python. By utilising the Natural Language Toolkit imports we were able to go about its basic operations such as text pre-processing which includes the Bag of Words (BOW) and the TF-IDF feature. We utilised Tweepy and Selenium concurrently throughout the analysis to broaden our search as well as to provide better analysis. Tweepy and Selenium were used to scrape Twitter for data while we utilised Selenium to go about the usage of the Natural Language Toolkit's (NLTK) SentimentIntensityAnalyzer.

In general, the focus of this analysis would be to analyse any and all tweets that utilise the phrase "Disney Plus" on Twitter from all users.

```
[ ]  !pip install selenium
     !pip install tweepy
```

Figure 80: Install selenium and tweepy

We first started off by installing selenium and tweepy. This process would include importing pertinent libraries. We ran this command so as to kickstart our analysis. We managed to install Tweepy and the Selenium-3.141.0 version.

```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import re
    import time
    import string
    import warnings

    # for all NLP related operations on text
    import nltk
    from nltk.corpus import stopwords
    from nltk.tokenize import sent_tokenize, word_tokenize
    from nltk.sentiment.vader import SentimentIntensityAnalyzer
    from nltk.stem import WordNetLemmatizer
    from nltk.stem.porter import *
    from nltk.classify import NaiveBayesClassifier
    from wordcloud import WordCloud

    from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
    from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import f1_score, confusion_matrix, accuracy_score
    from sklearn.svm import SVC
    from sklearn.naive_bayes import GaussianNB

    # To mock web-browser and scrap tweets
    from selenium import webdriver
    from selenium.webdriver.common.keys import Keys
```

Figure 81: Import necessary libraries

We then proceeded to install the necessary Python libraries as this would allow for the the library to access the necessary part of the code and perform the necessary form of analysis. We have proceeded to look at the initial imports. We then proceeded to go about the necessary Natural Language Processing (NLP) related operations on text. This would allow for the text processing to proceed. We then followed it with the imports for Selenium which would help to scrap the web browser for tweets.

```python
# To consume Twitter's API
import tweepy
from tweepy import OAuthHandler

# To identify the sentiment of text
from textblob import TextBlob
from textblob.sentiments import NaiveBayesAnalyzer
from textblob.np_extractors import ConllExtractor

# ignoring all the warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

# downloading stopwords corpus
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('vader_lexicon')
nltk.download('averaged_perceptron_tagger')
nltk.download('movie_reviews')
nltk.download('punkt')
nltk.download('conll2000')
nltk.download('brown')
stopwords = set(stopwords.words("english"))

# for showing all the plots inline
%matplotlib inline
```

Figure 82: Import Twitter API to retrieve Twitter data

The next set of imports includes importing Tweepy in order to consume the Twitter's API. We also had to include the textblob import which will assist us when it comes to identifying the sentiment of the texts. The next set of imports will include utilising the NLTK library to import all the stopwords corpus. Stopwords refer to words which do not provide useful information when it comes to classifying a word type.

```python
[ ] # keys and tokens from the Twitter Dev Console
    consumer_key = 'Sec3MvclRIx2RVlgu9l0SJX6D'
    consumer_secret = 'ayoPNWtBm7fWpMBoK6EwRmegu3SW8Rw9mzJkottkv97quPe941'
    access_token = '736550752760406018-so5CPJrEbJKb3c3Pq8va3VFr0yk4S0E'
    access_token_secret = 'Cgr8tz0h6FTU7kxAjDzpHnjffNTHxWsBytXnu4Ihd1TFb'
```

Figure 83: Connecting to the Twitter Development Console

This process includes utilising the necessary data such as consumer key, consumer secret, access token and access token secret to establish a connection. These data are obtained from the Twitter Development Console.

```python
[ ] class TwitterClient(object):
        def __init__(self):
            #Initialization method.
            try:
                # create OAuthHandler object
                auth = OAuthHandler(consumer_key, consumer_secret)
                # set access token and secret
                auth.set_access_token(access_token, access_token_secret)
                # create tweepy API object to fetch tweets
                # add hyper parameter 'proxy' if executing from behind proxy "proxy='http://172.22.218.218:8085'"
                self.api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)

            except tweepy.TweepError as e:
                print(f"Error: Tweeter Authentication Failed - \n{str(e)}")

        def get_tweets(self, query, maxTweets = 1000):
            #Function to fetch tweets.
            # empty list to store parsed tweets
            tweets = []
            sinceId = None
            max_id = -1
            tweetCount = 0
            tweetsPerQry = 100

            while tweetCount < maxTweets:
                try:
                    if (max_id <= 0):
                        if (not sinceId):
                            new_tweets = self.api.search(q=query, count=tweetsPerQry)
                        else:
                            new_tweets = self.api.search(q=query, count=tweetsPerQry,
                                                    since_id=sinceId)
                    else:
                        if (not sinceId):
                            new_tweets = self.api.search(q=query, count=tweetsPerQry,
                                                    max_id=str(max_id - 1))
                        else:
                            new_tweets = self.api.search(q=query, count=tweetsPerQry,
                                                    max_id=str(max_id - 1),
                                                    since_id=sinceId)
                    if not new_tweets:
                        print("No more tweets found")
                        break

                    for tweet in new_tweets:
                        parsed_tweet = {}
                        parsed_tweet['tweets'] = tweet.text

                        # appending parsed tweet to tweets list
                        if tweet.retweet_count > 0:
                            # if tweet has retweets, ensure that it is appended only once
                            if parsed_tweet not in tweets:
                                tweets.append(parsed_tweet)
                        else:
                            tweets.append(parsed_tweet)

                    tweetCount += len(new_tweets)
                    print("Downloaded {0} tweets".format(tweetCount))
                    max_id = new_tweets[-1].id

                except tweepy.TweepError as e:
                    # Just exit if any error
                    print("Tweepy error : " + str(e))
                    break

            return pd.DataFrame(tweets)
```

Figure 84: Analysis on the Initialization process for the Twitter Development Console

This code goes about the Initialization method as we create the OAuthHandler object and utilise the access token and the access token secret. This allows for the Tweepy API to fetch the tweets. There is also the "get_tweets" function which functions to fetch the tweets and to utilise the info on the empty list to store the parsed tweets. We also added a function to go about the retweets. This function would ensure that if the tweets that are scraped are retweets then it would only be appended only once.

b. Using 'tweepy'

```
[ ] twitter_client = TwitterClient()

    # calling function to get tweets
    tweets_df = twitter_client.get_tweets('Disney Plus', maxTweets=7000)
    print(f'tweets_df Shape - {tweets_df.shape}')
    tweets_df.head(10)
```

Figure 85: Scrape 7000 tweets using Tweepy

Now that the connection has been established to the Twitter Development Console, we went about obtaining the 7000 tweets with the keyword "Disney Plus". We had set the maxTweets to be 7000. The figure above is the function that would call and obtain all the tweets. The command "tweets_df.head(10)" is used to print the first 10 tweets that are scraped.

```
[ ] # 1 way
    def fetch_sentiment_using_SIA(text):
        sid = SentimentIntensityAnalyzer()
        polarity_scores = sid.polarity_scores(text)
        return 'neg' if polarity_scores['neg'] > polarity_scores['pos'] else 'pos'

    # 2 way
    def fetch_sentiment_using_textblob(text):
        analysis = TextBlob(text)
        return 'pos' if analysis.sentiment.polarity >= 0 else 'neg'
```

Figure 86: Utilising NLTK's SentimentIntensityAnalyzer (SIA)

To go about our Sentiment Analysis, we utilized NLTK's SentimentAnalysisAnalyzer which helped us to identify the sentiment type of each tweet. This process would be where utilise the SIA library in order to analyse the polarity scores between that of negative and positive sentiment.

```
[ ] sentiments_using_textblob = tweets_df.tweets.apply(lambda tweet: fetch_sentiment_using_textblob(tweet))
    pd.DataFrame(sentiments_using_textblob.value_counts())
```

	tweets
pos	4206
neg	318

Figure 87: Utilising Textblob so as to understand the sentiment of the tweets

This process would include utilising the textblob library so as to fetch the sentiment of the tweets that are scraped. This would be a good approach when it comes to providing us with sufficient information on the sentiment. Based on the tweets that have been analysed, we have obtained 4206 tweets with positive sentiments and 318 tweets with negative sentiments.

```
tweets_df['sentiment'] = sentiments_using_textblob
tweets_df.head()
```

	tweets	sentiment
0	RT @MCU_Direct: #MarvelStudios has "great hope...	pos
1	@contasparavoce Disney Plus	pos
2	RT @contasparavoce: SORTEIO DE HBO MAX + DISNE...	pos
3	RT @DisneyParks: Today's @Disneyland Resort Di...	pos
4	Disney+ lanzará un episodio especial de Luke S...	pos

Figure 88: Sample tweets that have been analysed

Based on the tweets that have been scraped, we can obtain a brief idea as to the type of data that has been identified. This would include a brief idea of the tweets along with the category of sentiment. However, as noted here, it is seen that the tweets are still not processed yet as it still includes tags and short forms.

Pre-Processing Phase

This stage would involve removing unnecessary information which would not assist in the analysis. This includes removing names, links, tweets with empty text, punctuation and the stop words.

```
[ ] def remove_pattern(text, pattern_regex):
        r = re.findall(pattern_regex, text)
        for i in r:
            text = re.sub(i, '', text)

        return text
```

```
[ ] # We are keeping cleaned tweets in a new column called 'tidy_tweets'
    tweets_df['tidy_tweets'] = np.vectorize(remove_pattern)(tweets_df['tweets'], "@[\w]*: | *RT*")
    tweets_df.head(10)
```

Figure 89: Removing @names

This segment would include removing the unnecessary @names which is of no use to the analysis since it brings about no meaning. Hence, it would be useful for it to be removed. This would also include creating a new column called 'tidy_tweets' in order to store all the cleaned tweets.

```
[ ] cleaned_tweets = []

    for index, row in tweets_df.iterrows():
        # Here we are filtering out all the words that contains link
        words_without_links = [word for word in row.tidy_tweets.split() if 'http' not in word]
        cleaned_tweets.append(' '.join(words_without_links))

    tweets_df['tidy_tweets'] = cleaned_tweets
    tweets_df.head(10)
```

Figure 90: Removing Links

This process includes removing and filtering out the words that contain the links that would not bring about a positive impact to the analysis. This process would include utilising the cleaned_tweets column that was created.

```
[ ] tweets_df = tweets_df[tweets_df['tidy_tweets']!='']
    tweets_df.head()
```

Figure 91: Removing tweets with empty texts

This figure would assist in removing any empty text. This would help enhance the level of analysis and quality of results that we can produce.

```
[ ] tweets_df['absolute_tidy_tweets'] = tweets_df['tidy_tweets'].str.replace("[^a-zA-Z# ]", "")
```

Figure 92: Removing Punctuations

This process is crucial as it allows for proper analysis. This strategy is useful since we are going about sentiment analysis, it is crucial that we do analysis on the key phrases. We will create an additional column ‘absolute_tidy_tweets’ which will ensure that it consists of only the tidy words which will be used to go about the sentiment analysis.

```
stopwords_set = set(stopwords)
cleaned_tweets = []

stopwords.extend(['Disney Plus', 'HBO Max', 'Netflix', 'Amazon', 'iFlix'])


for index, row in tweets_df.iterrows():

    # filerting out all the stopwords
    words_without_stopwords = [word for word in row.absolute_tidy_tweets.split() if not word in stopwords_set and '#' not in word.lower()]

    # finally creating tweets list of tuples containing stopwords(list) and sentimentType
    cleaned_tweets.append(' '.join(words_without_stopwords))

tweets_df['absolute_tidy_tweets'] = cleaned_tweets
tweets_df.head(10)
```

Figure 93: Removing stop words

Now that we have cleaned the tweets into the column named “tidy_tweets”, we can now go about the proper analysis. Removing the stop words would include removing unnecessary words which bring no significance to the analysis. To ensure that we are able to rid off any errors in the analysis, we have also extended the database for the stopwords to include stop words such as “Disney Plus”, “HBO Max”, “Netlfix”, “Amazon”, and “iFlix”. This would play a huge role when it comes to the descriptive analysis.

7.3 Result

Word Cloud

We were able to analyse the tweets with the key phrase “Disney Plus” that has been pre-processed into a word cloud.

```
def generate_wordcloud(all_words):
    wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=100, relative_scaling=0.5, colormap='Dark2').generate(all_words)

    plt.figure(figsize=(14, 10))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis('off')
    plt.show()
```

```
all_words = ' '.join([text for text in tweets_df['absolute_tidy_tweets'][tweets_df.sentiment == 'pos']])
generate_wordcloud(all_words)
```

Figure 94: Code for the Word Cloud with the Positive Words based on the analysis

Figure 95: Word Cloud with the Positive Words based on the analysis

This figure includes the visual representation of the words that are utilised in abundance in the overall tweets that have been analysed. As we can see here, the positive words are put out in a word cloud. The word cloud is utilised effectively in order to provide proper representation. The words which are reflected in a huge text indicate that it has been mentioned more frequently compared to the other texts.

```
[ ] all_words = ' '.join([text for text in tweets_df['absolute_tidy_tweets'][tweets_df.sentiment == 'neg']])
    generate_wordcloud(all_words)
```

Figure 96: Code for the Word Cloud of the Negative Words based on the analysis

Figure 97: Word Cloud of the Negative Words based on the analysis

The word cloud provides a proper representation of the negative comments that are said about Disney Plus. This process is effectively displayed as it includes the various words that are mentioned and a quick display as to its frequency.

Most popular hashtag

```
[ ] # function to collect hashtags
    def hashtag_extract(text_list):
        hashtags = []
        # Loop over the words in the tweet
        for text in text_list:
            ht = re.findall(r"#(\w+)", text)
            hashtags.append(ht)

        return hashtags

    def generate_hashtag_freqdist(hashtags):
        a = nltk.FreqDist(hashtags)
        d = pd.DataFrame({'Hashtag': list(a.keys()),
                          'Count': list(a.values())})
        # selecting top 15 most frequent hashtags
        d = d.nlargest(columns="Count", n = 25)
        plt.figure(figsize=(16,7))
        ax = sns.barplot(data=d, x= "Hashtag", y = "Count")
        plt.xticks(rotation=80)
        ax.set(ylabel = 'Count')
        plt.show()
```

```
[ ] hashtags = hashtag_extract(tweets_df['tidy_tweets'])
    hashtags = sum(hashtags, [])
```

```
[ ] generate_hashtag_freqdist(hashtags)
```

Figure 98: Analysis to obtain the most popular hashtag graph

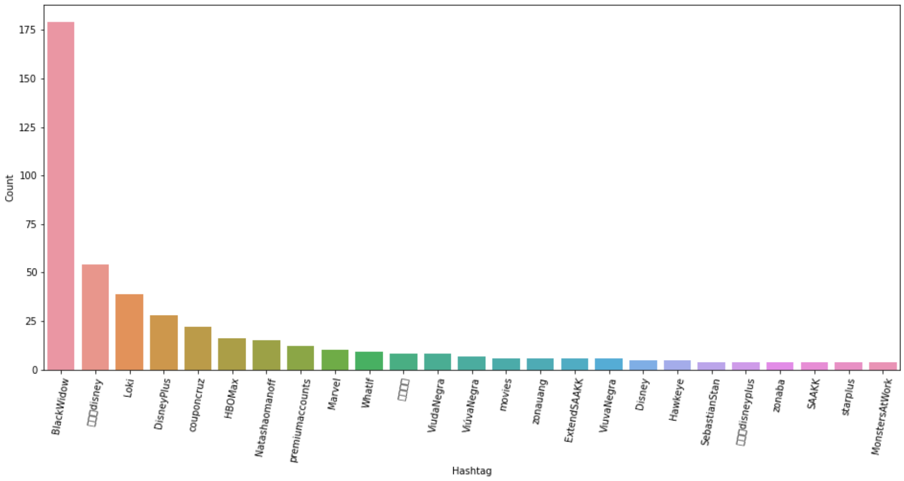


Figure 99: Most popular hashtag graph

As we can notice here, the graph here provides a good representation of the popular hashtags that are utilised when it comes to Tweets which utilises the phrase “Disney Plus”.

Confusion Matrix - BOW and TF-IDF

```
X_train, X_test, y_train, y_test = train_test_split(bow_word_feature, target_variable, test_size=0.3, random_state=272)
naive_model(X_train, X_test, y_train, y_test)
```

Figure 100: Command for BOW word features

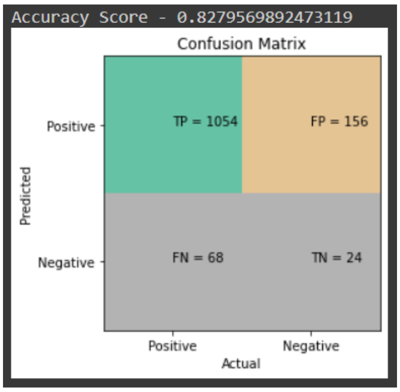


Figure 101: BOW word features

Bag of words is the simplest way to represent text in numbers. This indicates that there is an accuracy score of 82.8 %. This shows that this model is accurate in predicting the number of positive and negative words being used.

```
X_train, X_test, y_train, y_test = train_test_split(tfidf_word_feature, target_variable, test_size=0.3, random_state=272)
naive_model(X_train, X_test, y_train, y_test)
```

Figure 102: Command for the TF-IDF Word Features

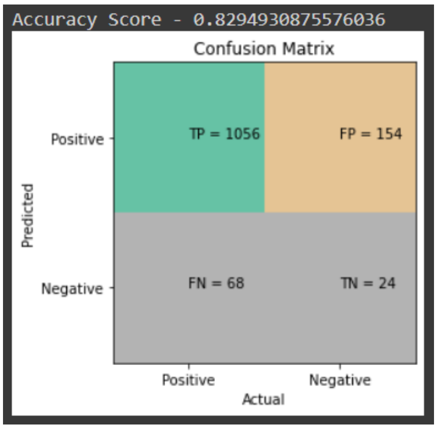


Figure 103: TF-IDF word features

The TF-IDF method is whereby a word is quantified so as to compute the weight of each word to correlate it to the importance of the word in the document. The TF-IDF Confusion Matrix has an accuracy score of 82.95%. This indicates that the model is significantly accurate in predicting the positive and negative words.

7.4 Discussions

As part of the analysis, we have identified multiple different areas where we can look into. We have 4 different outputs. The first being the analysis on the word clouds. We have presented 2 word clouds. One of which is to represent the Positive Words and the other to represent the Negative Words. We believe that it is the best representation of the feedback and opinion from the users.

Referring to the word cloud that was generated. The positive wordcloud highlights the key benefits and areas whereby the users are happy to work with and utilise the platform. This is done through the process of identifying the various different areas and scraping for the phrase “Disney Plus” on Twitter. There were feedback such as convenient, cheap, trusted and interesting. The negative word cloud indicates the various different words that were utilised to show how the users were not happy with the platform. For example, there were feedback such as cluttered, expensive, confusing and empty. Ultimately this shows how the feedback has been structured.

We extended our research by adding in the analysis for the graphical representation for the most popular hashtag. This was represented in a bar chart. Based on our analysis, the term “Black Widow” was utilised the most based on the 7000 tweets that we had scraped. This indicates that the popular hashtag is with regards to the new release. We can also see a new release as well, Loki, trending amongst the top hashtags for Disney Plus.

Furthermore, we also extended our research into identifying the Confusion Matrix for the Bag of Words and the Term Frequency - Inverse Document Frequency analysis as well. This included a quick predictive analysis on the accuracy score of any future prediction based on the current analysis. Both these analyses had a healthy accuracy score of above 80%.

7.5 Implication

Based on the analysis that we have conducted, the recommendation that we would like to make is to utilise the data, feedback and opinion that is provided by Twitter so that we can best come up with recommendations to improve Disney.

For example, the word clouds can be utilised as raw feedback and opinion from the users to best understand their sentiment. This would include the positive and negative words. In its essence, Disney can take a look at the positive word cloud to understand where they are on the right track and the negative word cloud to understand where they can improve. This would be a good strategy for them to understand and to take a good step forward into improving their brand in itself. The approach taken could be to improve or address the negative comments that has been posted by the users.

Next, the bar chart with the most popular hashtags indicates that there is a growing sentiment amongst the users when it comes to latest release. For example, in this case, the term “Black Widow” was utilised as it was very much popular when it was released. Hence, this can be a good strategy for Disney to actively market out their new release so that it can rapidly gain traction and improve.

Finally, there is also the element of the BOW and TF-IDF. These areas help to improve on the future analysis and any form of improvements. With the introduction of the accuracy scores, this indicates how important the analysis is and how we can improve from there.

8. Conclusion

All in all, we have done an in depth analysis into the various areas that we can cover when it comes to the social media analytics for Disney Plus. We have done extensive research into the various fields and believe that our approach and recommendation provided is sufficient and strong.

We now understand the vast role that Social Media Analytics play when it comes to online media. It allows us to obtain key insights through proper analysis and discussion. This is also achieved via data visualization. Furthermore, we are also able to help provide good recommendations that can best help the brands. Finally, social media analytics is also a powerful and credible method for us to analyse a platform and provide data driven analysis which is highly beneficial.

We are confident that our research is credible and would be able to effectively help Disney Plus to be the one stop option for online streaming.