EXPERIMENT NO 1

Date of performance:

Date of submission:

Aim: Study various -

i)Social Media platforms (Facebook, twitter, youtube etc)

ii)Social Media analytics tools (facebook insights, google analytics netlytic etc)

iii) Social Media Analytics techniques and engagement metrics (page level, post level, member level)

iv) Applications of Social media analytics for business.

e.g. Google Analytics

Exercises/Case Study

● Come up with five companies or brands you interact with regularly. For example, the companies could be a beverage bottler, restaurant, clothing brand, or technology company. For each of the five, find all the social media accounts you can. These will usually include a Facebook page, often a Twitter or YouTube account, and they may be present in many other types of social media.

1. List each company and their social media accounts.

○Find as many counts for each social media account as described in the section on measuring success.

○ How often does the company interact on their social network site? Is it many times a day, a few times a week, or never?

○ What kind of interaction is the company doing? Broadcast, request for input, direct interaction, or a combination? Provide an example of each.

Theory:

Exercises/Case Study

● Come up with five companies or brands you interact with regularly. For example, the companies could be a beverage bottler, restaurant, clothing brand, or technology company. For each of the five, find all the social media accounts you can. These will usually include a Facebook page, often a Twitter or YouTube account, and they may be present in many other types of social media.

1. List each company and their social media accounts.

2. Find as many counts for each social media account as described in the section on measuring success.

3. How often does the company interact on their social network site? Is it many times a day, a few times a week, or never?

4. What kind of interaction is the company doing? Broadcast, request for input, direct interaction, or a combination? Provide an example of each.

5. Assess the company's social media strategy. What are they doing well and why? What could they do better, why would that be better, and how should they do it?

Case Study

Following is the list of five Indian brands and their social media accounts:

Sr. No.	Brands	Social Media Accounts
1	Haldiram's	https://www.facebook.com/haldiramsofficial
		https://www.instagram.com/haldirams_nagpur/
		https://twitter.com/NagpurHaldirams
		https://www.youtube.com/user/haldiramsofficial
		https://in.pinterest.com/haldiramsnagpur/haldiram-winter-special/
2	Bikaner	https://www.facebook.com/Bikanerexpressofficial/
		https://www.instagram.com/bikanerexpressofficial/?hl=en\
3	Jumbo King	https://www.facebook.com/Jumbokingindia/
		https://www.instagram.com/jumboking_india/
		https://www.youtube.com/channel/UCV7uaBHz2cITiqps4JdHCKw
4	Monginis	https://www.facebook.com/monginis/
		https://twitter.com/MonginisIndia
		https://www.youtube.com/channel/UCL2K9zi9D94YBbmU1GJicAA
		https://www.instagram.com/monginiscelebrations/
5	Cafe Coffee day	https://www.instagram.com/cafecoffeeday/ https://www.facebook.com/cafecoffeeday
		https://www.youtube.com/user/cafecoffeeday
		https://twitter.com/CafeCoffeeDay
		https://www.cafecoffeeday.com/

Q2.

Sr no.	Followers	Social Media Handle Name	Likes	Rating
1	Fb : 122,960 Instagram : 75.5k Twitter : 10.1k Youtube : 21.3k Pinterest : 144	Fb: @haldiramsofficial Instagram: **haldirams_nagpur** Twitter: @NagpurHaldirams Youtube: @Haldiramsofficial Pinterest: @haldiraminternational	Fb: 115,432	Fb: 3.9
2	Fb : 14,962 Instagram : 7543	Fb: @Bikanerexpressofficial Instagram: **bikanerexpressofficial**	Fb: 14,857	
3	Fb : 434,603	Fb: @Jumbokingindia	Fb: 435,525	Fb: 4

	Instagram : 68.4k Youtube : 1.36k	Instagram: **jumboking_india** Youtube: @jumbokingindia5297		
4	Fb : 1,155,702 Instagram : 107k Twitter : 6,990 Youtube : 9.51k	Fb: @monginis Instagram: **monginiscelebrations** Twitter: @MonginisIndia Youtube: @monginis	Fb: 1,156,797	Fb: 4
5	Fb : 4,566,596 Instagram : 188k Twitter : 92.5k Youtube : 7.26k Pinterest : 1.3k	Fb: @cafecoffeeday Instagram: **cafecoffeeday** Twitter: @CafeCoffeeDay Youtube: @cafecoffeeday Pinterest: @cafecoffeeday	Fb: 4,630,088	

Q3.
Monthly posts after 23rd Dec:

Haldiram :

Fb : 14 posts
Instagram : 30 posts
Twitter: 1 post
Youtube: yearly videos (35 videos ⇒ 3yrs ago, 24 videos ⇒ 4yrs ago, 3 videos ⇒ 8yrs, 3 videos ⇒ 6yrs, 2 videos ⇒ 5yrs ago, 10 videos ⇒ 2yrs ago, 1 videos ⇒ 1yr ago)
Pinterest:

Bikaner:
Instagram: 10 posts
Facebook: 2 posts

JumboKing:

Facebook : 5 posts
Instagram : 4 posts
Youtube : (9 VIdeos past 11 months, 4 videos past 1 yrs, 22 past 2 yrs, 9 past 3yrs, 6 past 4yrs, 2 past 6yrs, 5 past 7yrs)

Monginis:

Facebook: 66 posts
Twitter: 2 posts
Instagram: 51 posts
Youtube: 35 videos in the past 11 months, 27 videos past 1 yrs, 4 past 2 yrs, 7 videos past 4 yrs, 4 videos past 6yrs, 9 videos past 7yrs, 1 past 9yrs, 8 past 10 yrs, 25 videos past 11 yrs.

Cafe Coffee Day:

Facebook: 26 posts
Instagram: 24 posts
Pinterest:
Twitter: 36 posts
Youtube: 3yrs ago 3 videos, 4yrs ago 3 videos, 5yrs ago 12 videos, 6yrs ago 10 videos, 9yrs ago 16 videos, 7yrs ago 1 video, 10yrs ago 21 videos, 11yrs ago 2 videos, 13yrs ago 1 video

Q4. Following is the list of Interaction Company is doing

Haldiram:

Broadcast:

Haldiram advertises its various food items through its official youtube channel https://www.youtube.com/@Haldiramsofficial, Television advertisements, or through its poster or ads on websites.

Request to input:
Haldiram interacts with its customers through taking input from customers about how they can improve their products and services to give a better experience towards customers. Haldiram takes feedback its from customers through its websites contact us page: https://www.haldirams.com/contact-us

Direct interaction with customers:
Haldiram is a food manufacturing and retail company that primarily sells Indian snacks and sweets. They have a number of physical retail stores where customers can purchase their products directly. Additionally, they also sell their products online through their website and other e-commerce platforms, which allows customers to place orders for delivery. They also have a phone number for customer service and a customer can also interact with the company via email or social media for customer support.

Jumbo King:

Broadcast:
Jumbo King mostly broadcasts its food items through instagram, facebook or youtube. Mostly the broadcasting and advertisement done for its food items is done by instagram https://www.instagram.com/jumboking_india/.

Request to input:
Jumbo King takes input from customers through its website. Customers can provide feedback to Jumbo King through this website: https://www.jumboking.co.in/feedback.

Direct interaction with customers:
Jumbo King has stores in various cities in India through which customers can directly purchase and eat burgers or other food items. Additionally, they also sell their products online through their website and other e-commerce platforms, which allows customers to place orders for delivery. Through which they can directly contact the customers without them coming to their store.

Bikaner:

Broadcast:
Bikaner broadcasts its food items through facebook and instagram social media channels. Bikaner is mostly active through social media on facebook and instagram.

Request to input:
Bikaner takes input from customers through various sites like Zomato or mainly through its official website where the customers can ask some queries or give feedback for Bikaner.

Direct interaction with customers:
Bikaner also has various shops and some online sites through which customers can order what they want. In bikaner shops people can directly purchase and eat the food. Through shopping online also they can get their order at their doorstep.

Monginis:

Broadcast:
In Monginis they mostly broadcast their food items cakes through advertisements on TV and on some websites. Also they broadcast about their products on social media accounts like instagram, facebook, twitter.

Request to input:
Monginis takes input from customers about their service through various platforms like Zomato, Swiggy and their website through which people can share their experience. Monginis has become a top cake brand in India by noticing the various requirements given by customers to them.

Direct Interaction with customers:
Monginis has shops and stores across India through which customers can directly purchase cakes through their shops. They can also give orders through websites or on call.

Cafe Coffee day:

Broadcast:
Cafe Coffee day mostly broadcasts its products through advertisements on various social media sites like instagram, facebook, twitter, youtube, its website or through various sites through its ads.

Request to input:
Cafe Coffee day takes input from customers regarding its service, food and many things so that they can improve their services, food etc. Customers can provide their inputs through their official site or their store.

Direct interaction with customers:
Cafe Coffee day store has many shops in India through which customers can purchase food and eat. Also customers can order through their websites and get it at their doorstep.

Q5. Assess the company's social media strategy. What are they doing?
well and why? What could they do better, why would that be better, and how should they do it?

Cafe Coffee Day:
Their entire social media strategy is focused towards youngsters. Be it their Facebook updates or tweets, they are very focused on whom to communicate to.
Coffee and conversations are the most resonant topics with the 'CCD Culture' and they capitalized on it in a very smart way. They are going all out to promote themselves as a place where people should 'Sit Down' and discuss topics/issues.
A huge majority of their social media communication is revolving around their 'Sit Down' culture.
CCD's strategy is very much in sync with its offline positioning. A place to have conversations and coffee, they are projecting themselves in a similar way on social media as well.And they know that they are the most favorite place to hangout for youngsters and couples. Which they have smartly integrated in their content strategy as well.All their social media strategy has resulted in generating quite a buzz, especially on Facebook. And with an overwhelming majority of people liking in, their perception is positive as well.
To conclude, CCD is a smart brand when it comes to Facebook. Now if only they could use Twitter in a better way and revamp their strategy for YouTube. One area where they could improve is in their use of customer generated content. Encouraging customers to share photos and reviews of their experiences at Cafe Coffee Day could help to build trust and credibility for the brand.

Jumbo King:
They use Social media to communicate the following to their target group: introducing the brand, introducing product launches, introducing store launches and introducing city specific consumer offers. 80% of their marketing spend is allocated towards social media.
Another way for Jumbo King to improve its social media strategy is by creating a more interactive and personalized experience for their customers. This could include things like polls, quizzes, or live Q&A sessions. Additionally, they could also consider using social media as a customer service channel by providing quick and efficient responses to customer inquiries and complaints.

Monginis:
Monginis has a presence on social media platforms such as Facebook, Instagram, Twitter and YouTube. They regularly post updates about their products, promotions, and events. They also have a good engagement rate with their audience by replying to comments and messages, which helps to build a loyal customer base.
Monginis could start by conducting market research to better understand their target audience and their preferences. They could also invest in a good camera, editing software, and other tools to help them create high-quality visual content. Additionally, they could also hire a social media specialist to help manage their accounts and create engaging content.

Haldiram's

Haldiram's has a strong presence on social media platforms such as Facebook, Instagram, Twitter and YouTube. They regularly post updates about their products, promotions, and events. They also have a good engagement rate with the audience by replying to comments and messages, this helps to build a loyal customer base. They also use their social media accounts to share information about their company, history, and values.

One area where they could improve is in creating more visually appealing and interactive content. They could consider using more images and videos to showcase their products and the process of making them. This could help to make their brand more relatable and attract a younger audience.

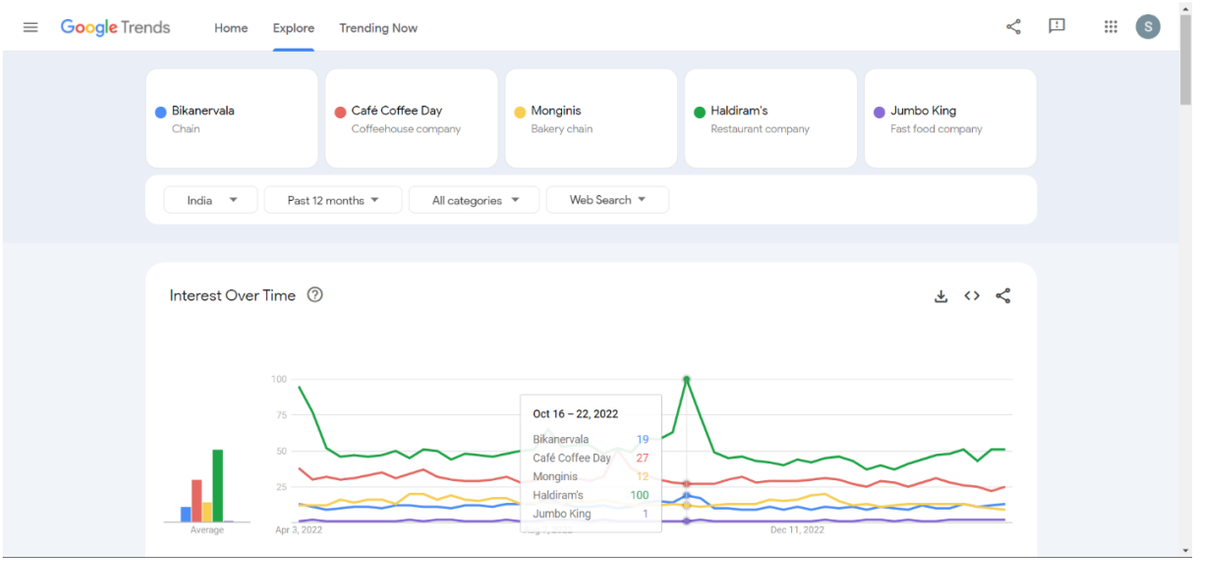
Bikanervala

Bikanervala has a strong presence on social media platforms such as Facebook, Instagram, Twitter and YouTube. They regularly post updates about their products, promotions, and events. They also have a good engagement rate with the audience by replying to comments and messages, this helps to build a loyal customer base. They also use their social media accounts to share information about their company, history, and values.

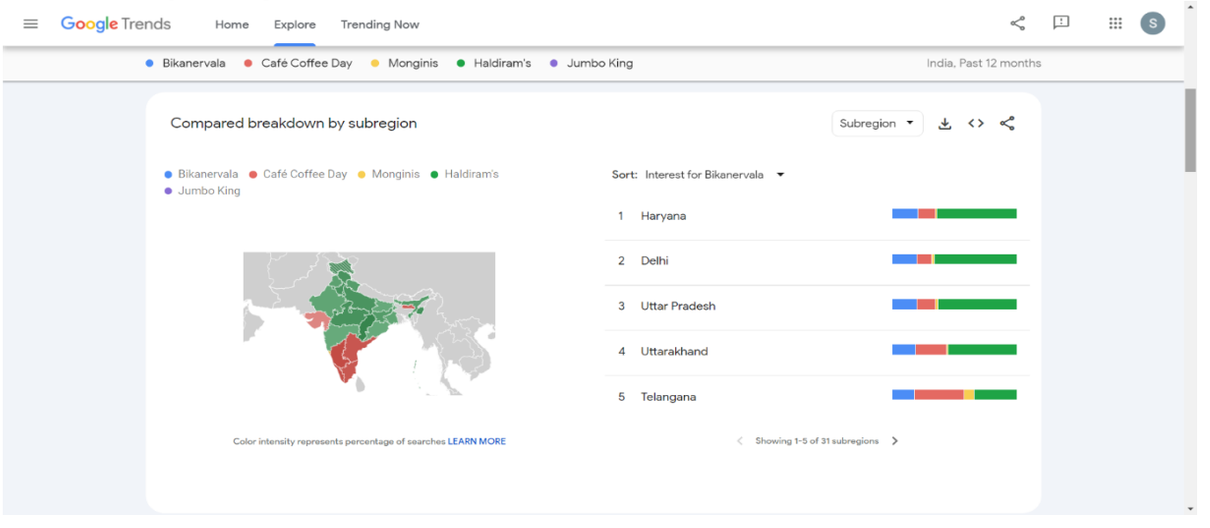
Another way for Bikanervala to improve its social media strategy is by creating a more interactive and personalized experience for their customers. This could include things like polls, quizzes, or live Q&A sessions. Additionally, they could also consider using social media as a customer service channel by providing quick and efficient responses to customer inquiries and complaints.

Output:

Comparison of 5 brands (Interest over time)

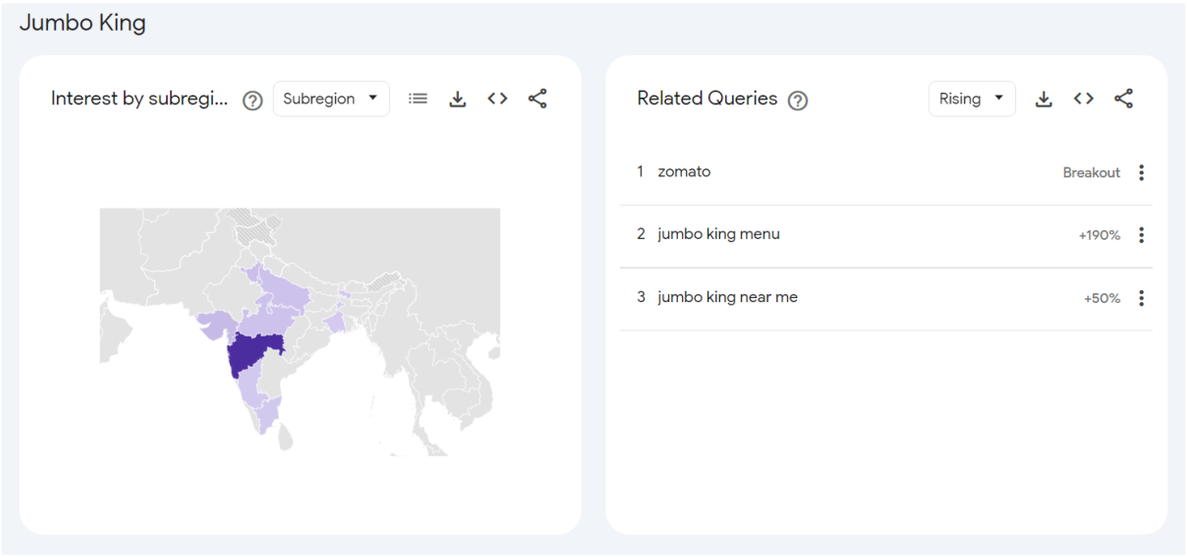


Breakdown by subregion:



Jumbo King (Interest by subregion and Related queries)

Conclusion:



Various social media platforms like Facebook, Youtube, Instagram, Twitter were studied. Social media analytics tools like Google analytics, Google trends were used to gather information to gather engagement metrics. Metrics like rise in certain queries, interest by subregion, comparative breakdown over subregion, interest over time pertaining to selected brands were understood and studied thoroughly. As a result, application of social media analytics in business was studied and implemented successfully.

R1 (3 Marks)	R2 (3 Marks)	R3 (3 Marks)	R4 (3 Marks)	R5 (3 Marks)	Total (15 Marks)	Signature

EXPERIMENT NO 2

Date of performance:

Date of submission:

Aim: Data collection - Select the social media platforms of your choice (Twitter, Facebook, LinkedIn, Youtube,Web blogs etc) ,connect to and capture social media data for business (scraping, crawling,parsing).

Theory:

Case Study

● Choose a popular current issue of public debate (a bill under consideration, an election, or a political issue). Search Twitter for posts about that issue.

○ What opinions are you able to find? Summarize them.

○ Is one opinion dominating the others?

○ Do you find a lot of content repeated? Perhaps one or two tweets that are repeated by many accounts? Does this appear suspicious, or is there a reason for it?

https://twitter.com/search?q=%23UddhavThackeray&src=trend_click&vertical=trends

- Uddhav Thackarey called Maharashtra CM's faction thieves. Eknath Shinde replied and suggests 'self-introspection' as to why such situation has come upon him.
- No. Majority seems to be on the side of Eknath Shinde.
- One might say a lot of tweets/content is being repeated and it comes from the fact that most replies or tweets are trolling the ex-CM of Maharashtra and his statements from previous interviews. Exact words or quotes are rarely repeated, but the sentiment driving such posts is present and is same. This concludes the fact that the tweets/content is not spam or bots with an agenda, but common people responding to the political situation in the state.

Social Media data collection:

Social media data is any information collected from social media platforms that provides insights into the activities of users on the platform.

Social media data collection is not as simple as it sounds, for many reasons. The type of data collected depends on the platform and the relevance of the data to an organization.

For instance, the data obtained on Facebook will include the number of likes, number of shares, and follower increases. Twitter's data would include the number of impressions, retweets, and likes. For Instagram, data like hashtag usage, engagement rates, and active followers are important.

Having access to raw data provides you with the opportunity to understand your customers better, and develop an effective content strategy. When you obtain data, you can know the interests of your audience and tailor the content you create for real people.

Here are five marketing goals that social media data collection can be used for:

1.Content optimization

2. Social media strategy development

3. Better SEO strategies

4. Brand image management

5. Easily spot effective social media influencers

Web Scraping:
1. Web Scraping :

Web Scraping is a technique used to extract a large amount of data from websites and then saving it to the local machine in the form of XML, excel or SQL. The tools used for web scraping are known as web scrapers. On the basis of the requirements given, they can extract the data from any website in a fraction of time. This automation of tasks is very helpful for developing data for machine learning and other purpose. They work in four steps:

1.Sending the request to the target page.

2.Getting response from the target page.

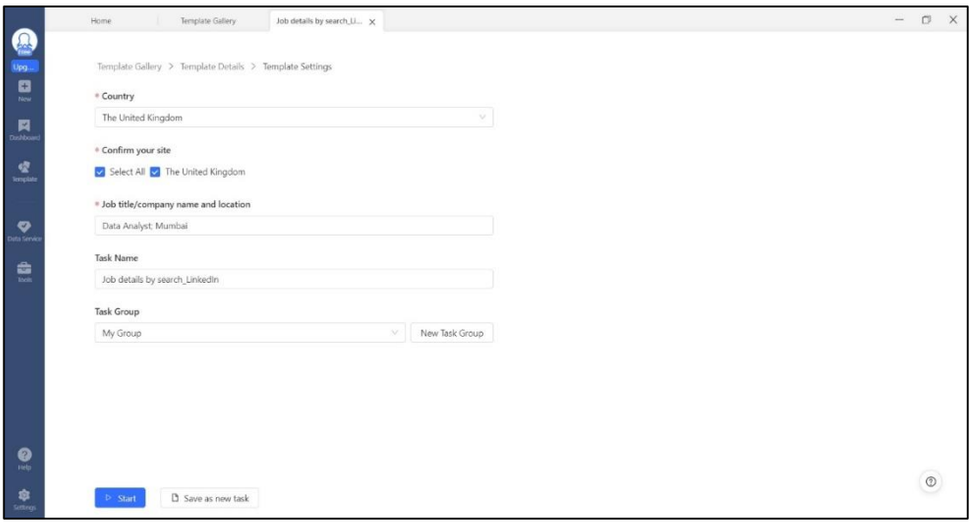
3.Parsing and extracting the response.

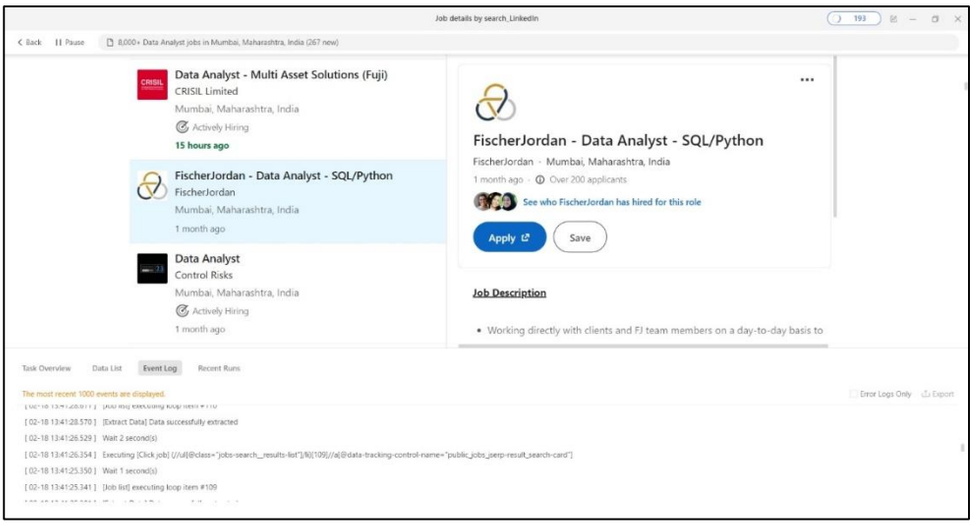
4. Download the data.

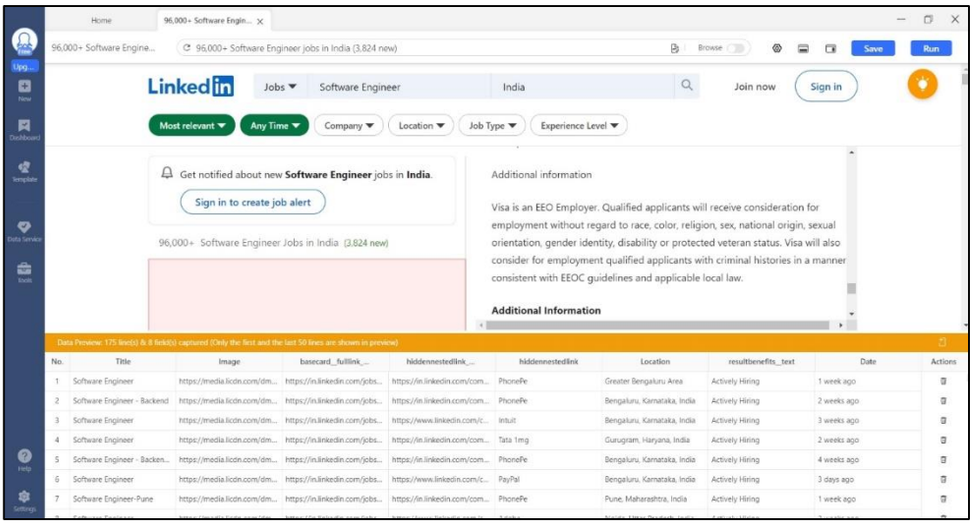
Some of the popular web scraping tools are ProWebScraper, Webscraper.io, Octoparse, etc.

Output:

Screenshots: (Octoparse Web Scraping)







Output: (Data Scraped using Octoparse)

#	Title	Image	basecard_fulllink_URL	hiddennestedlink_URL	hiddennestedlink	Location	resultbenefits_text	Date
1	Core Python Developer/Lea...		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Wipro	Pune, Maharashtra, India	Actively Hiring	3 days ago
2	Jr. Java Developer	https://media.licdn.com/dm...	https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Citi	Pune, Maharashtra, India	Actively Hiring	6 days ago
3	Jr. Java Developer	https://media.licdn.com/dm...	https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Citi	Pune, Maharashtra, India	Actively Hiring	6 days ago
4	Junior Software Engineer	https://media.licdn.com/dm...	https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Parallel Wireless	Pune, Maharashtra, India	Actively Hiring	1 day ago
5	Software Developer - Fronte...	https://media.licdn.com/dm...	https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Mintifi	Mumbai, Maharashtra, India	Be an early applicant	1 month ago
6	HTML Coder	https://media.licdn.com/dm...	https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	SVKM's Narsee Monjee Insti...	Mumbai Metropolitan Region		3 weeks ago
7	Jr. Java Developer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Citi	Pune, Maharashtra, India	Actively Hiring	6 days ago
8	Software Engineer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Emerson	Pune, Maharashtra, India	Actively Hiring	1 week ago
9	Python Developer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Deloitte	Pune, Maharashtra, India	Actively Hiring	2 weeks ago
10	Software Development Engi...		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Truemeds India	Pune, Maharashtra, India	Actively Hiring	2 months ago
11	Software Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	PharmEasy	Mumbai, Maharashtra, India	Be an early applicant	1 month ago
12	Software Engineer React		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Woliba	Pune, Maharashtra, India		1 month ago
13	Junior Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	CRISIL Limited	Mumbai, Maharashtra, India	Actively Hiring	4 weeks ago

#	Title	Image	basecard_fulllink_URL	hiddennestedlink_URL	hiddennestedlink	Location	resultbenefits_text	Date
1	Software Engineer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Medscape	Navi Mumbai, Maharashtra, ...	Be an early applicant	1 month ago
2	React.Js Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Persistent Systems	Pune, Maharashtra, India	Actively Hiring	1 month ago
3	Java/Springboot Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Infosys	Pune, Maharashtra, India	Actively Hiring	1 week ago
4	Software Engineer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Wipro	Mumbai, Maharashtra, India	Actively Hiring	2 weeks ago
5	Junior Java Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	LenDenClub	Mumbai, Maharashtra, India	Actively Hiring	1 month ago
6	Software Development Engi...		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Truemeds India	Mumbai, Maharashtra, India	Actively Hiring	3 months ago
7	Frontend Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Reliance Jio	Mumbai, Maharashtra, India	Actively Hiring	1 week ago
8	Software Engineer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Ivalua	Pune, Maharashtra, India	Actively Hiring	3 weeks ago
9	Software Engineer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Teqsar	Pune, Maharashtra, India	Be an early applicant	1 month ago
10	Python Developer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Deloitte	Mumbai, Maharashtra, India	Actively Hiring	2 weeks ago
11	Java Developer	https://media.licdn.com/dm...	https://in.linkedin.com/jobs...	https://fr.linkedin.com/com...	Atos	Pune, Maharashtra, India	Actively Hiring	3 weeks ago
12	India - Junior Java Backend ...		https://in.linkedin.com/jobs...	https://kn.linkedin.com/com...	Chain	Mumbai, Maharashtra, India		4 weeks ago
13	Java Jr and Sr Developers B...		https://in.linkedin.com/jobs...	https://uk.linkedin.com/com...	PwC	Mumbai, Maharashtra, India	Actively Hiring	1 week ago

#	Title	Image	basecard__fulllink_URL	hiddennestedlink_URL	hiddennestedlink	Location	resultbenefits__text	Date
1	Software Engineer I - (Javas...		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	TraceLink	Pune, Maharashtra, India	Actively Hiring	2 days ago
2	URGENT HIRING WITH TEC...		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Tech Mahindra Business Ser...	Pune, Maharashtra, India	Be an early applicant	4 weeks ago
3	Software Engineer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Mastercard	Pune, Maharashtra, India	Actively Hiring	2 days ago
4	Python Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Tudip Technologies	Mulshi, Maharashtra, India	Be an early applicant	4 weeks ago
5	Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Wipro	Pune, Maharashtra, India	Actively Hiring	3 days ago
6	Software Engineer Java SQL...		https://in.linkedin.com/jobs...	https://ch.linkedin.com/com...	Avaloq	Pune, Maharashtra, India		1 week ago
7	Developer		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	Tech Mahindra	Pune, Maharashtra, India	Actively Hiring	1 week ago
8	Frontend Developer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	Deloitte	Pune, Maharashtra, India	Actively Hiring	1 week ago
9	Software Engineer / Senior ...		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	NEC Software Solutions (Ind...	Mumbai, Maharashtra, India	Be an early applicant	1 month ago
10	System Software Engineer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	NVIDIA	Pune, Maharashtra, India	Actively Hiring	3 days ago
11	Software Engineer / Senior ...		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	NEC Software Solutions (Ind...	Mumbai, Maharashtra, India	Be an early applicant	2 weeks ago
12	Software Developer Fresher		https://in.linkedin.com/jobs...	https://in.linkedin.com/com...	G. P. KAPADIA & CO	Mumbai, Maharashtra, India		2 months ago
13	System Software Engineer		https://in.linkedin.com/jobs...	https://www.linkedin.com/c...	NVIDIA	Pune, Maharashtra, India	Actively Hiring	2 days ago

Conclusion: Octoparse has a user-friendly interface that allows users to create web scraping tasks without requiring any coding knowledge. Users can simply drag and drop elements on a webpage to create a template, and Octoparse will extract the relevant data.

R1 (3 Marks)	**R2 (3 Marks)**	**R3 (3 Marks)**	**R4 (3 Marks)**	**R5 (3 Marks)**	**Total (15 Marks)**	**Signature**

EXPERIMENT NO 3

Date of performance:

Date of submission:

Aim: Data Cleaning and Storage- Preprocess, filter and store social media data for business (Using Python, MongoDB, R, etc).

Theory:

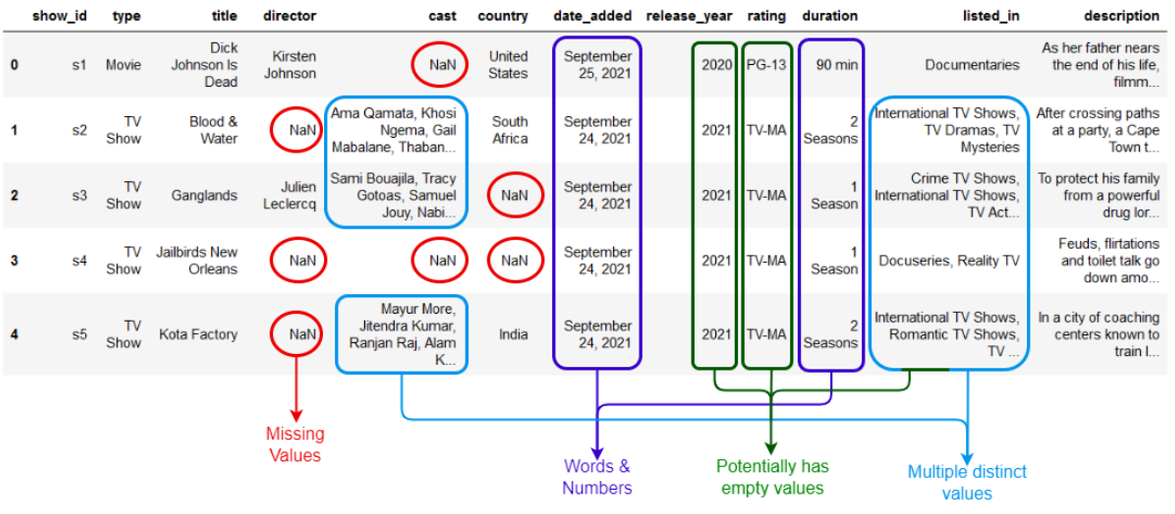
For this experiment, we will be working with the Netflix TV Shows and Movies Dataset which features many inconsistencies and missing data.

Table of Contents:

1. Look into your data
2. Look at the proportion of **missing data**
3. Check the **data type** of each column
4. If you have columns of strings, check for **trailing whitespaces**
5. **Dealing with Missing Values** (NaN Values)
6. **Extracting more information from your dataset to get more variables**
7. Check the **unique values** of columns

Step 1: Look into your data

Before even performing any cleaning or manipulation of your dataset, you should take a glimpse at your data to understand what variables you're working with, how the values are structured based on the column they're in, and maybe you could have a rough idea of the inconsistencies that you'll need to address or they'll be cumbersome in the analysis phase. Here, you might also be able to eliminate certain columns that you won't need depending on the analysis you want to do.
To stay organized, note the issues you see in your dataset (by taking a glimpse of your dataset).



	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

Step 2: Look at the proportion of missing data

```
# examining missing values
print("Missing values distribution: ")
print(netflix_titles.isnull().mean())
print("")
```

```
Output:
Missing values distribution:
show_id         0.000000
type            0.000000
title           0.000000
director        0.299080
cast            0.093675
country         0.094357
date_added      0.001135
release_year    0.000000
rating          0.000454
duration        0.000341
listed_in       0.000000
description     0.000000
dtype: float64
```

From the output, these are insights you can gather:

- director column has the highest percentage of missing data ~ 30%
- cast and country column also has a considerable percentage of missing data ~ 9%
- date_added, rating and duration don't have that much missing data ~ 0% - 0.1%

There are a few ways to deal with it:

1. Drop the column completely. If the column isn't that important to your analysis, just drop it.
2. Keep the column. In this case, because **the director, cast and country columns are quite important to my analysis**, I will keep them.
3. **Imputation — the process of replacing missing data with substituted values.** Here, it is not possible to do so because most of the data are string values and not numerical values. However, I will be writing an article that talks more about imputation in detail, why and when it should be used, and how you can use it in R and Python with the help of some packages.

Step 3: Check the data type of each column

```
# check datatype in each column
print("Column datatypes: ")
print(netflix_titles.dtypes)
```

```
Output:
Column datatypes:
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

Here, you can see that all the columns have object as their datatype aside from release_year. In pandas, object means either string or mixed type (numerical and non-numerical type mixed).

Step 4: If you have columns of strings, check for trailing whitespaces

After we know which data types we are dealing with, let's remove any trailing characters and whitespace using strip.

Step 5: Dealing with Missing Values (NaN Values)

Referring back to the columns of missing values, let's take a look at the columns: director, cast, country, date_added, rating, duration. We can segment these columns by whether they are a string or mixed type.

String: director, cast, country, rating (here, it's a string and not mixed because the numerical values won't have any meaning if separated)

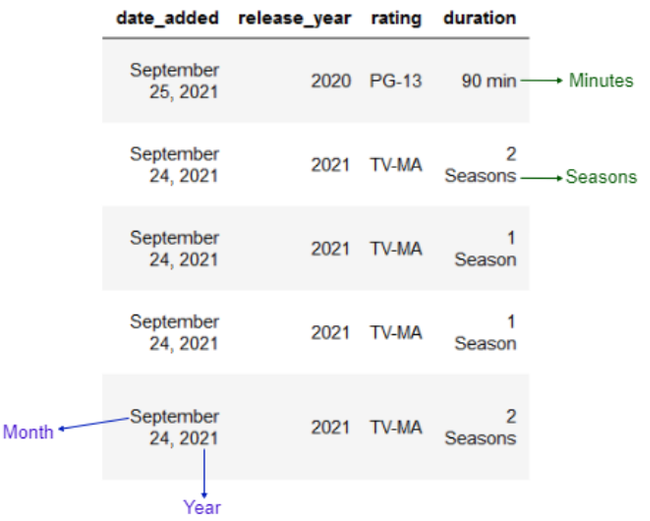
Mixed: date_added, duration

NaN means Not a Number in pandas. It is a special floating-point value that is different from NoneType in Python. NaN values can be annoying to work with, especially when you want to filter them out for plots or analysis. Let's **replace these NaN values with something else.**

For string type values, we can replace NaN values with "" or "None" or any string that can indicate to you that there isn't any value in that entry. Here, we chose to replace it with "" using the fillna function. Because it's not an in-place function, we reassigned the changed values to the column in the dataset.

Step 6: See if there are any other variables that you can obtain by extracting them from other variables

For mixed-type values, before we tackle the missing value issue, let's see if we can extract any data to make our analysis richer or process easier.



Looking at date_added, we can see that it contains the month, date, and year that the film/show was added. Instead of having all this information in one column, why not try to separate them? That way, we can choose to isolate how month or year interacts with the other variables instead of looking at date_added where its granularity will make it difficult for any trend to be discovered. Now, the new dataset contains the month_added and year_added columns.

Looking at duration, on top of it being a mixed type, there are also 2 different time units in this column. This is a problem because we are dealing with 2 different types of content that are measured differently for time. Thus, making graphs for durationwill be quite difficult to interpret if we keep them as it is.

By separating the type of content into 2 different datasets and naturally, the duration column will just be numerical and just have 1 type of time unit. So, we can easily plot using the values.

Because the duration column has both strings and numbers, I'll also have to create a function to extract the number from that column so that it can be inserted into the columns of the 2 new datasets.

Step 7: Check the unique values of columns

Beyond potentially missing values, there could be corrupted values that you can run into once you perform analysis. You can easily obtain the unique values of a column like rating using Python's built-in function, unique.

Looking at dataset, why are there UR (Unrated) and NR (Not Rated)? Aren't they supposed to mean the same thing? We should keep the consistency where NR is used and change UR values to NR.

After using the split function, we can easily obtain the unique values for the country and listed_in columns.

```
# getting unique country names
unique_countries = getUnique(netflix_titles['country'])
unique_countries
```

```
Output:
['',
 'Czech Republic',
 'Armenia',
 'Belgium',
 'Mozambique',
'East Germany'*,
 'West Germany'*,
 'Soviet Union'*,
 'Burkina Faso', etc.] (shortened for article)
```

We can see there are some issues with this list:

- There's both the Soviet Union and Russia
- There's both the West/East Germany and Germany

We can easily fix this with a few modifications to the dataset.

As for the list of genres, we can see that there are some genres we might not want or need to include. Thus, we can easily remove it from the dataset to make our analysis less confounding.

In both the TV shows and films dataset, there is a "TV Shows" and "Movies" genre. Technically, this isn't a genre but could be a label of the type of content. To confirm this, we should print out the counts of these "genres" appearing in the respective datasets.

The hypothesis is that if these "genres" appear in all the rows of the datasets, it means that they're simply labels. Otherwise, we'll have to investigate further as to what those "genres" represent.

Taking a look at the rows, it is now obvious that the "TV Shows" and "Movies" genre was used to signify that these contents didn't have a genre in the first place. Now that we understand what this meant, we can either choose to exclude or include it in our analysis. Here, we have chosen to include it because it doesn't affect my analysis.

Benefits of data cleaning include:

- Removal of errors when multiple sources of data are at play.
- Fewer errors make for happier clients and less-frustrated employees.
- Ability to map the different functions and what your data is intended to do.
- Monitoring errors and better reporting to see where errors are coming from, making it easier to fix incorrect or corrupt data for future applications.
- Using tools for data cleaning will make for more efficient business practices and quicker decision-making.

Dataset Used: netflix_titles.csv

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
s1	Movie	Dick Johns	Kirsten Johnson		United Sta	September	2020	PG-13	90 min	Document	As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive and comi
s2	TV Show	Blood & Water		Ama Qam	South Afri	September	2021	TV-MA	2 Seasons	Internatio	After crossing paths at a party, a Cape Town teen sets out to prove whether a private-school swimming
s3	TV Show	Ganglands	Julien Lecl	Sami Bouajila, Tracy		September	2021	TV-MA	1 Season	Crime TV S	To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of robbers are p
s4	TV Show	Jailbirds New Orleans				September	2021	TV-MA	1 Season	Docuserie	Feuds, flirtations and toilet talk go down among the incarcerated women at the Orleans Justice Center i
s5	TV Show	Kota Factory		Mayur Mo	India	September	2021	TV-MA	2 Seasons	Internatio	In a city of coaching centers known to train Indiaâ€™s finest collegiate minds, an earnest but unexceptic
s6	TV Show	Midnight	Mike Flana	Kate Siegel, Zach Gilf		September	2021	TV-MA	1 Season	TV Dramas	The arrival of a charismatic young priest brings glorious miracles, ominous mysteries and renewed religi
s7	Movie	My Little P	Robert Cul	Vanessa Hudgens, Ki		September	2021	PG	91 min	Children &	Equestria's divided. But a bright-eyed hero believes Earth Ponies, Pegasi and Unicorns should be pals â€
s8	Movie	Sankofa	Haile Geri	Kofi Ghan	United Sta	September	1993	TV-MA	125 min	Dramas, In	On a photo shoot in Ghana, an American model slips back in time, becomes enslaved on a plantation ar
s9	TV Show	The Great	Andy Devo	Mel Giedr	United Kin	September	2021	TV-14	9 Seasons	British TV	A talented batch of amateur bakers face off in a 10-week competition, whipping up their best dishes in
s10	Movie	The Starlin	Theodore	Melissa M	United Sta	September	2021	PG-13	104 min	Comedies,	A woman adjusting to life after a loss contends with a feisty bird that's taken over her garden â€" and a
s11	TV Show	Vendetta: Truth, Lies and		The Mafia		September	2021	TV-MA	1 Season	Crime TV S	Sicily boasts a bold "Anti-Mafia" coalition. But what happens when those trying to bring down organize
s12	TV Show	Bangkok B	Kongkiat K	Sukollawat Kanarot, S		September	2021	TV-MA	1 Season	Crime TV S	Struggling to earn a living in Bangkok, a man joins an emergency rescue service and realizes he must un
s13	Movie	Je Suis Karl	Christian S	Luna Wedl	Germany,	September	2021	TV-MA	127 min	Dramas, In	After most of her family is murdered in a terrorist bombing, a young woman is unknowingly lured into j
s14	Movie	Confessio	Bruno Gar	Klara Castanho, Lucc		September	2021	TV-PG	91 min	Children &	When the clever but socially-awkward TetÃª joins a new school, she'll do anything to fit in. But the quee
s15	TV Show	Crime Stories: India Detectives				September	2021	TV-MA	1 Season	British TV	Cameras following Bengaluru police on the job offer a rare glimpse into the complex and challenging in
s16	TV Show	Dear White People	Logan Bro		United Sta	September	2021	TV-MA	4 Seasons	TV Comedi	Students of color navigate the daily slights and slippery politics of life at an Ivy League college that's not
s17	Movie	Europe's	Pedro de Echave GarcÃa, Pablo			September	2020	TV-MA	67 min	Document	Declassified documents reveal the post-WWII life of Otto Skorzeny, a close Hitler ally who escaped to Sp
s18	TV Show	Falsa identidad		Luis Ernest	Mexico	September	2020	TV-MA	2 Seasons	Crime TV S	Strangers Diego and Isabel flee their home in Mexico and pretend to be a married couple to escape his
s19	Movie	Intrusion	Adam Salk	Freida Pinto, Logan M		September	2021	TV-14	94 min	Thrillers	After a deadly home invasion at a coupleâ€™s new dream house, the traumatized wife searches for ans
s20	TV Show	Jaguar		Blanca SuÃ¡rez, IvÃ¡n		September	2021	TV-MA	1 Season	Internatio	In the 1960s, a Holocaust survivor joins a group of self-trained spies who seek justice against Nazis fleei
s21	TV Show	Monsters I	Olivier Mega	ton		September	2021	TV-14	1 Season	Crime TV S	In the late 1970s, an accused serial rapist claims multiple personalities control his behavior, setting off a
s22	TV Show	Resurrection: Ertugru	Engin Alta		Turkey	September	2018	TV-14	5 Seasons	Internatio	When a good deed unwittingly endangers his clan, a 13th-century Turkish warrior agrees to fight a sulta
s23	Movie	Avvai Shan	K.S. Ravikur	Kamal Hassan, Meen		September	1996	TV-PG	161 min	Comedies,	Newly divorced and denied visitation rights with his daughter, a doting father disguises himself as a gray
s24	Movie	Go! Go! C	Alex Woo,	Maisie Benson, Paul K		September	2021	TV-Y	61 min	Children &	From arcade games to sled days and hiccup cures, Cory Carsonâ€™s curious little sister Chrissy speeds c
s25	Movie	Jeans	S. Shankar	Prashanth, Jaya	India	September	1998	TV-14	166 min	Comedies,	When the father of the man she loves insists that his twin sons marry twin sisters, a woman creates an
s26	TV Show	Love on the Spectrum	Brooke Sa		Australia	September	2021	TV-14	2 Seasons	Docuserie	Finding love can be hard for anyone. For young adults on the autism spectrum, exploring the unpredicta
s27	Movie	Minsara K	Rajiv Men	Arvind Swamy, Kajol,		September	1997	TV-PG	147 min	Comedies,	A tangled love triangle ensues when a man falls for a woman studying to become a nun â€" and she fall
s28	Movie	Grown Up	Dennis Du	Adam San	United Sta	September	2010	PG-13	103 min	Comedies	Mourning the loss of their beloved junior high basketball coach, five middle-aged pals reunite at a lake
s29	Movie	Dark Skies	Scott Stew	Keri Russel	United Sta	September	2013	PG-13	97 min	Horror Mo	A familyâ€™s idyllic suburban life shatters when an alien force invades their home, and as they struggle
s30	Movie	Paranoia	Robert Luk	Liam Hem	United Sta	September	2013	PG-13	106 min	Thrillers	Blackmailed by his company's CEO, a low-level employee finds himself forced to spy on the boss's rival a
s31	Movie	Ankahi Ka	Ashwiny Iy	Abhishek Banerjee, Ri		September	2021	TV-14	111 min	Dramas, In	As big city life buzzes around them, lonely souls discover surprising sources of connection and companio
s32	TV Show	Chicago Party Aunt		Lauren Ash, Rory O'M		September	2021	TV-MA	1 Season	TV Comedi	Chicago Party Aunt Diane is an idolized troublemaker with a talent for avoiding adulthood â€" and a sof
s33	TV Show	Sex Education		Asa Butter	United Kin	September	2020	TV-MA	3 Seasons	British TV	Insecure Otis has all the answers when it comes to sex advice, thanks to his therapist mom. So rebel Ma
s34	TV Show	Squid Game		Lee Jung-jae, Park Ha		September	2021	TV-MA	1 Season	Internatio	Hundreds of cash-strapped players accept a strange invitation to compete in children's games. Inside, a
s35	TV Show	Tayo and Little Wizar	Dami Lee, Jason Lee,			September	2020	TV-Y7	1 Season	Kids' TV	Tayo speeds into an adventure when his friends get kidnapped by evil magicians invading their city in se
s36	Movie	The Father	Daniel San	Adrian Titieni, Elena P		September	2021	TV-MA	110 min	Dramas, In	When his son goes missing during a snowy hike in the mountains, a retired intelligence officer will stop a
s37	Movie	The StrongÇÃ©dric Ji		Gilles Lellouche, Kari		September	2021	TV-MA	105 min	Action & A	Tired of the small-time grind, three Marseille cops get a chance to bust a major drug network. But lines
s38	TV Show	Angry Birds		Antti PÃ¤Ã	Finland	September	2018	TV-Y7	1 Season	Kids' TV, T	Birds Red, Chuck and their feathered friends have lots of adventures while guarding eggs in their nest th
s39	Movie	Birth of th	George No	Billy Magn	China, Can	September	2017	PG-13	96 min	Action & A	A young Bruce Lee angers kung fu traditionalists by teaching outsiders, leading to a showdown with a Sh
s40	TV Show	Chhota Bheem		Vatsal Dub	India	September	2021	TV-Y7	3 Seasons	Kids' TV	A brave, energetic little boy with superhuman powers leads his friends on exciting adventures to guard t
s41	TV Show	He-Man and the Mast	Yuri Lowe		United Sta	September	2021	TV-Y7	1 Season	Kids' TV, T	Mighty teen Adam and his heroic squad of misfits discover the legendary power of Grayskull â€" and the
s42	Movie	Jaws	Steven Spi	Roy Scheid	United Sta	September	1975	PG	124 min	Action & A	When an insatiable great white shark terrorizes Amity Island, a police chief, an oceanographer and a griz
s43	Movie	Jaws 2	Jeannot Sz	Roy Scheid	United Sta	September	1978	PG	116 min	Dramas, H	Four years after the last deadly shark attacks, police chief Martin Brody fights to protect Amity Island fro
s44	Movie	Jaws 3	Joe Alves	Dennis Qu	United Sta	September	1983	PG	98 min	Action & A	After the staff of a marine theme park try to capture a young great white shark, they discover its mother
s45	Movie	Jaws: The	Joseph Sar	Lorraine G	United Sta	September	1987	PG-13	91 min	Action & A	After another deadly shark attack, Ellen Brody has had enough of Amity Island and moves to the Caribbe
s46	Movie	My Heroes	Tyler Greco			September	2021	PG	23 min	Document	Robin Wiltshire's painful childhood was rescued by Westerns. Now he lives on the frontier of his dreams
s47	Movie	Safe Hous	Daniel Espi	Denzel Wa	South Afri	September	2012	R	115 min	Action & A	Young CIA operative Matt Weston must get a dangerous criminal out of an agency safe house that's com
s48	TV Show	The Smart Bunmi Aja	Osas Ighodaro, Ini Di			September	2020	TV-MA	1 Season	Internatio	Five glamorous millennials strive for success as they juggle careers, finances, love and friendships. Basec
s49	Movie	Training D	Antoine Fu	Denzel Wa	United Sta	September	2001	R	122 min	Dramas, T	A rookie cop with one day to prove himself to a veteran LAPD narcotics officer receives a crash course in
s50	TV Show	Castle and Castle		Richard M	Nigeria	September	2021	TV-MA	2 Seasons	Internatio	A pair of high-powered, successful lawyers find themselves defending opposite interests of the justice sy
s51	TV Show	Dharmakshetra		Kashmira I	India	September	2014	TV-PG	1 Season	Internatio	After the ancient Great War, the god Chitragupta oversees a trial to determine who were the battle's tru
s52	Movie	InuYasha t	Toshiya Sh	Kappei Ya	Japan	September	2002	TV-14	99 min	Action & A	With their biggest foe seemingly defeated, InuYasha and his friends return to everyday life. But the peac
s53	Movie	InuYasha t	Toshiya Sh	Kappei Ya	Japan	September	2003	TV-14	99 min	Action & A	The Great Dog Demon beaqueathed one of the Three Swords of the Fang to each of his two sons. Now t
s54	Movie	InuYasha t	Toshiya Sh	Kappei Ya	Japan	September	2004	TV-PG	88 min	Action & A	Ai, a young half-demon who has escaped from Horai Island to try to help her people, returns with poten

Code:

```
import pandas as pd

# importing dataset
netflix_titles = pd.read_csv("/content/netflix_titles.csv")
# printing the first 5 rows of dataset
netflix_titles.head()

# getting the columns of the dataset
columns = list(netflix_titles.columns)
columns

# examining missing values
print("Missing values distribution: ")
print(netflix_titles.isnull().mean())
print("")
# check datatype in each column
print("Column datatypes: ")
print(netflix_titles.dtypes)

# getting all the columns with string/mixed type values
str_cols = list(netflix_titles.columns)
str_cols.remove('release_year')

# removing leading and trailing characters from columns with str type
for i in str_cols:
    netflix_titles[i] = netflix_titles[i].str.strip()
```

```
# names of the columns
columns = ['director', 'cast', 'country', 'rating', 'date_added']

# looping through the columns to fill the entries with NaN values with ""
for column in columns:
    netflix_titles[column] = netflix_titles[column].fillna("")
from datetime import datetime

# examining rows with null values for date_added column
rows = []
for i in range(len(netflix_titles)):
    if netflix_titles['date_added'].iloc[i] == "":
        rows.append(i)

# examine those rows to confirm null state
netflix_titles.loc[rows, :]
# extracting months added and years added
month_added = []
year_added = []
for i in range(len(netflix_titles)):
    # replacing NaN values with 0
    if i in rows:
        month_added.append(0)
        year_added.append(0)
    else:
        date = netflix_titles['date_added'].iloc[i].split(" ")
        month_added.append(date[0])
        year_added.append(int(date[2]))

# turning month names into month numbers
for i, month in enumerate(month_added):
    if month != 0:
        datetime_obj = datetime.strptime(month, "%B")
        month_number = datetime_obj.month
        month_added[i] = month_number

# checking all months
print(set(month_added))
print(set(year_added))

# inserting the month and year columns into the dataset
netflix_titles.insert(7, "month_added", month_added, allow_duplicates = True)
netflix_titles.insert(8, "year_added", year_added, allow_duplicates = True)
netflix_titles.head()

# separating original dataset to tv show and movie dataset respectively
shows = []
films = []

# looping through the dataset to identify rows that are TV shows and films
for i in range(len(netflix_titles)):
    if netflix_titles['type'].iloc[i] == "TV Show":
        shows.append(i)
    else:
        films.append(i)

# grouping rows that are TV shows
netflix_shows = netflix_titles.loc[shows, :]

#grouping rows that are films
netflix_films = netflix_titles.loc[films, :]

# reseting the index of the new datasets
```

```
netflix_shows = netflix_shows.set_index([pd.Index(range(0, len(netflix_shows)))])
netflix_films = netflix_films.set_index([pd.Index(range(0, len(netflix_films)))])

# get length of movie or number of seasons of show
def getDuration(data):
    count = 0
    durations = []
    for value in data:
                # filling in missing values
        if type(value) is float:
            durations.append(0)
        else:
            values = value.split(" ")
            durations.append(int(values[0]))
    return durations

# inserting new duration type column for shows (renamed column)
netflix_shows.insert(11, 'seasons', getDuration(netflix_shows['duration']))
netflix_shows = netflix_shows.drop(['duration'], axis = 1)
netflix_shows.head()

# inserting new duration type column for films (renamed column)
netflix_films.insert(11, 'length', getDuration(netflix_films['duration']))
netflix_films = netflix_films.drop(['duration'], axis = 1)
netflix_films.head()

# getting the unique ratings for films
netflix_films['rating'].unique()

# getting the unique ratings for shows
netflix_shows['rating'].unique()

# printing more details of the rows that have incorrect ratings
incorrect_ratings = ['74 min', '84 min', '66 min']
for i in range(len(netflix_films)):
    if netflix_films['rating'].iloc[i] in incorrect_ratings:
        print(netflix_films.iloc[i])
        print("")

# getting the row indices
index = [3562, 3738, 3747]

# fixing the entries
for i in index:
    split_value = netflix_films['rating'].iloc[i].split(" ")
    length = split_value[0]
    netflix_films['length'].iloc[i] = length
    netflix_films['rating'].iloc[i] = "NR"

# double checking the entries again
for i in index:
    print(netflix_films.iloc[i])

# fixing the entries
for i in range(len(netflix_films)):
    if netflix_films['rating'].iloc[i] == "UR":
        netflix_films['rating'].iloc[i] = "NR"

# double checking
netflix_films['rating'].unique()

# function to get unique values of a column
def getUnique(data):
```

```
    unique_values = set()
    for value in data:
        if type(value) is float:
            unique_values.add(None)
        else:
            values = value.split(", ")
            for i in values:
                unique_values.add(i)
    return list(unique_values)
# getting unique country names
unique_countries = getUnique(netflix_titles['country'])
unique_countries
```

#Examining the list of unique countries to see if there are any inconsistencies or mistakes. We can see there are some issues with this list:

There's both the Soviet Union and Russia
There's both the West/East Germany and Germany

```
# converting soviet union to russia and east/west germany to germany
for i in range(len(netflix_titles)):
    if type(netflix_titles['country'].iloc[i]) is not float:
        countries = netflix_titles['country'].iloc[i].split(", ")
        for j in range(len(countries)):
            if "Germany" in countries[j]:
                countries[j] = "Germany"
            elif "Soviet Union" in countries[j]:
                countries[j] = "Russia"
        netflix_titles['country'].iloc[i] = ", ".join(countries)

# getting unique film genres
unique_genres_films = getUnique(netflix_films['listed_in'])
unique_genres_films

# getting unique show genres
unique_genres_shows = getUnique(netflix_shows['listed_in'])
unique_genres_shows
```

In both the TV shows and films dataset, there is a "TV Shows" and "Movies" genre. Technically, this isn't a genre but could be a label of the type of content. To confirm this, we should print out the counts of these "genres" appearing in the respective datasets.

```
# checking for TV shows
# replace netflix_shows with netflix_films to check for movies
count = 0
index = []
for i, value in enumerate(netflix_shows['listed_in']):
    genres = value.split(", ")
    if "TV Shows" in genres:
        count += 1
        index.append(i)
print("count %s" %count)
print("index %s" %index)

# checking for Movies
count = 0
index = []
for i, value in enumerate(netflix_films['listed_in']):
    genres = value.split(", ")
    if "Movies" in genres:
        count += 1
        index.append(i)
print("count %s" %count)
```

```
print("index %s" %index)

# printing the first 5 rows of all rows that have TV Shows as its genre
netflix_shows.iloc[index[0:5]]

# printing the first 5 rows of all rows that have Movies as its genre
netflix_films.iloc[index[0:5]]
```

Output:

getting the columns of the dataset

```
['show_id',
 'type',
 'title',
 'director',
 'cast',
 'country',
 'date_added',
 'release_year',
 'rating',
 'duration',
 'listed_in',
 'description']
```

examining missing values

```
Missing values distribution:
show_id         0.000000
type            0.000000
title           0.000000
director        0.299080
cast            0.093675
country         0.094357
date_added      0.001135
release_year    0.000000
rating          0.000454
duration        0.000341
listed_in       0.000000
description     0.000000
dtype: float64
```

inserting the month and year columns into the dataset

	show_id	type	title	director	cast	country	date_added	month_added	year_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	9	2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	9	2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...		September 24, 2021	9	2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans				September 24, 2021	9	2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory		Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	9	2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

printing more details of the rows that have incorrect ratings

```
show_id                                                    s5542
type                                                       Movie
title                                             Louis C.K. 2017
director                                               Louis C.K.
cast                                                   Louis C.K.
country                                           United States
date_added                                        April 4, 2017
month_added                                                   4
year_added                                                 2017
release_year                                               2017
rating                                                   74 min
length                                                        0
listed_in                                                Movies
description     Louis C.K. muses on religion, eternal love, gi...
Name: 3562, dtype: object

show_id                                                    s5795
type                                                       Movie
title                                       Louis C.K.: Hilarious
director                                               Louis C.K.
cast                                                   Louis C.K.
country                                           United States
date_added                                  September 16, 2016
month_added                                                   9
year_added                                                 2016
release_year                                               2010
rating                                                   84 min
length                                                        0
listed_in                                                Movies
description     Emmy-winning comedy writer Louis C.K. brings h...
Name: 3738, dtype: object
show_id                                                    s5814
type                                                       Movie
title                          Louis C.K.: Live at the Comedy Store
director                                               Louis C.K.
cast                                                   Louis C.K.
country                                           United States
date_added                                      August 15, 2016
month_added                                                   8
year_added                                                 2016
release_year                                               2015
rating                                                       NR
length                                                       66
listed_in                                                Movies
description     The comic puts his trademark hilarious/thought...
```

getting unique show genres

```
['TV Mysteries',
 'Reality TV',
 'Science & Nature TV',
 'Korean TV Shows',
 'Anime Series',
 'TV Shows',
 'TV Dramas',
 'Spanish-Language TV Shows',
 'British TV Shows',
 'TV Comedies',
 'TV Action & Adventure',
 'TV Sci-Fi & Fantasy',
 "Kids' TV",
 'International TV Shows',
 'TV Horror',
 'Romantic TV Shows',
 'Classic & Cult TV',
 'Crime TV Shows',
 'Teen TV Shows',
 'TV Thrillers',
 'Docuseries',
 'Stand-Up Comedy & Talk Shows']
```

printing the first 5 rows of all rows that have TV Shows as its genre

	show_id	type	title	director	cast	country	date_added	month_added	year_added	release_year	rating	seasons	listed_in
59	s149	TV Show	HQ Barbers	Gerhard Mostert	Hakeem Kae-Kazim, Chioma Omeruah, Orukotan Ade...		September 1, 2021	9	2021	2020	TV-14	1	TV Shows
110	s298	TV Show	Navarasa	Bejoy Nambiar, Priyadarshan, Karthik Narain, V...	Suriya, Vijay Sethupathi, Revathy, Prakash Raj...	India	August 6, 2021	8	2021	2021	TV-MA	1	TV Shows
272	s727	TV Show	Metallica: Some Kind of Monster	Joe Berlinger, Bruce Sinofsky	James Hetfield, Lars Ulrich, Kirk Hammett, Rob...	United States	June 13, 2021	6	2021	2014	TV-MA	1	TV Shows
286	s772	TV Show	Pretty Guardian Sailor Moon Eternal The Movie	Chiaki Kon	Kotono Mitsuishi, Hisako Kanemoto, Rina Satou,...		June 3, 2021	6	2021	2021	TV-14	1	TV Shows
452	s1332	TV Show	Five Came Back: The Reference Films			United States	February 9, 2021	2	2021	1945	TV-MA	1	TV Shows

printing the first 5 rows of all rows that have Movies as its genre

	show_id	type	title	director	cast	country	date_added	month_added	year_added	release_year	rating	length	listed_in
59	s108	Movie	A Champion Heart	David de Vos	Mandy Grace, David de Vos, Donna Rusch, Devan ...	United States	September 4, 2021	9	2021	2018	G	90	Children & Family Movies, Dramas
110	s175	Movie	Tears of the Sun	Antoine Fuqua	Bruce Willis, Monica Bellucci, Cole Hauser, Ea...	United States	September 1, 2021	9	2021	2003	R	121	Action & Adventure, Dramas
272	s420	Movie	Chhota Bheem: Bheem vs Aliens	Rajiv Chilaka	Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jig...		July 22, 2021	7	2021	2010	TV-Y7	69	Children & Family Movies, Sports Movies
286	s439	Movie	2 Weeks in Lagos	Kathryn Fasegha	Beverly Naya, Mawuli Gavor, Ajoke Silva, Jide ...		July 16, 2021	7	2021	2020	TV-PG	107	Dramas, International Movies, Romantic Movies
452	s723	Movie	Sir! No Sir!	David Zeiger	Troy Garity	United States	June 15, 2021	6	2021	2005	TV-MA	84	Documentaries

Conclusion:

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for your data cleaning process so you know you are doing it the right way every time. All such steps were studied and carefully implemented.

R1 (3 Marks)	R2 (3 Marks)	R3 (3 Marks)	R4 (3 Marks)	R5 (3 Marks)	Total (15 Marks)	Signature

EXPERIMENT NO 4

Date of performance:

Date of submission:

Aim: Data Cleaning and Storage- Preprocess, filter and store social media data for business (Using Python, MongoDB, R, etc).

Theory:

Exploratory data analysis is a method used to analyze and summarize data sets.

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions. EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them.

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

EDA helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modelling.

Some of the most common data science tools used to create an EDA include:

- **Python:** Python and EDA can be used together to identify missing values in a data set, which is important so you can decide how to handle missing values for instance.

Code:
Dataset Used: tweepy python module for Twitter API extraction.

	username	location	verified	tweet_date	text	hashtags
0	Rachel Roh	La Crescenta-Montrose, CA	False	20-12-2020 06:06	Same folks said daikon paste could treat a cyt...	['PfizerBioNTech']
1	Albert Fong	San Francisco, CA	False	13-12-2020 16:27	While the world has been on the wrong side of ...	NaN
2	eli???ñ???????¦???§????	Your Bed	False	12-12-2020 20:33	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf...
3	Charles Adler	Vancouver, BC - Canada	True	12-12-2020 20:23	Facts are immutable, Senator, even when you're...	NaN
4	Citizen News Channel	NaN	False	12-12-2020 20:17	Explain to me again why we need a vaccine @Bor...	['whereareallthesickpeople', 'PfizerBioNTech']

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import re
import os
import time
from datetime import datetime, date, timedelta
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from tqdm import tqdm
from gensim.models.doc2vec import LabeledSentence
import gensim
```

```
from sklearn.linear_model import LogisticRegression
from scipy import stats
from sklearn import metrics
from sklearn.metrics import mean_squared_error,mean_absolute_error,
make_scorer,classification_report,confusion_matrix,accuracy_score,roc_auc_score,roc_curve
from sklearn.model_selection import train_test_split,cross_val_score,KFold
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score
from sklearn.naive_bayes import BernoulliNB
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
import warnings
warnings.filterwarnings("ignore")

# For Twitter API extraction
import tweepy

# Tweet pre-processor
import preprocessor as p

# NLTK
import nltk
nltk.download('n'vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# TextBlob
from textblob import TextBlob
import copy

# Stanza
import stanza
stanza.download('en')
import csv

import stylecloud
from PIL import Image

# Twitter API Credentials
api_key = 'BipQLD1z5El5nlhxVadzAhaIc'
api_key_secret = 'akvTqwy**********rnFfK'
access_token = '11535****695366-3LEWVp**********lVJfc'
access_token_secret = 'wFR6yU7**********ZRRkdx8ov'

auth = tweepy.OAuthHandler(api_key, api_key_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

extract_datetime = datetime.today().strftime('%Y%m%d_%H%M%S')
filename = 'covid_vaccine_tweets_extracted_' + extract_datetime
tweets_df.drop_duplicates(inplace = True)
tweets_df.to_csv('/content/drive/MyDrive/TE Project/data.xls')

working_dir_path = '/content/drive/MyDrive/vaccination.csv'
tweets_df = pd.read_csv(working_dir_path,encoding = 'latin')
tweets_df = tweets_df.drop_duplicates('text')

tweets_df.info()

tweets_df.tweet_date.value_counts()
```

```
def get_hashtags(text):
    list_of_hashtags = []
    temp = text.split()
    for word in temp:
        if word[0] == '#':
            list_of_hashtags.append(word)
    return list_of_hashtags

texts = list(tweets_df['text'])
tweets_df['hashtags'] = [get_hashtags(text) for text in texts]

#Overall hashtags for all months
tweets_df['hashtags_extracted'] = tweets_df['hashtags'].apply(lambda x: ' '.join(x))
long_string = ','.join(list(tweets_df['hashtags_extracted'].values))
stylecloud.gen_stylecloud(text=long_string, icon_name='fas fa-hashtag', max_words=100,
palette='cartocolors.diverging.Earth_2', output_name='hashtags.png', size = 1000, collocations=False)
Image.open('hashtags.png')
#tweets_df['hashtags_texts'].values
#long_string

# we do not care about the exact time of each tweet, we just want the date
tweets_df['tweet_date'] = pd.to_datetime(tweets_df['tweet_date']).dt.date

# create a copy of the dataframe
#df_time = tweets_df.copy()
df_time = copy.deepcopy(tweets_df)

# set the timestamp column as the index and delete the column
df_time.index = df_time['tweet_date']
del df_time['tweet_date']

sns.catplot("tweet_date", data=tweets_df, kind="count", height=8)

tweets_df['location'].value_counts()

tweets_df['verified'].value_counts().head(n=10).plot.bar()

# Heat Map for missing values
plt.figure(figsize=(17, 5))
sns.heatmap(tweets_df.isnull(), cbar=True, yticklabels=False)
plt.xlabel("Column_Name", size=14, weight="bold")
plt.title("Places of missing values in column",fontweight="bold",size=17)
plt.show()

# Top 10 locations of tweet
Top_Location_Of_tweet= tweets_df['location'].value_counts().head(10)

sns.set(rc={'figure.figsize':(12,8)})
sns.set_style('white')
Top_Location_Of_tweet.head(10)

Top_Location_Of_tweet_df=pd.DataFrame(Top_Location_Of_tweet)
Top_Location_Of_tweet_df.reset_index(inplace=True)
Top_Location_Of_tweet_df.rename(columns={'index':'Location', 'Location':'Location_Count'},
inplace=True)
Top_Location_Of_tweet_df

viz_1=sns.barplot(x="Location", y="location", data=Top_Location_Of_tweet_df, palette='Blues_d')
viz_1.set_title('Locations with most of the tweets')
```

```
viz_1.set_ylabel('Count of listings')
viz_1.set_xlabel('Location Names')
viz_1.set_xticklabels(viz_1.get_xticklabels(), rotation=90)

# Text pre-processing
tweets_df = tweets_df_final.copy()

#Clean tweet text with tweet-preprocessor
tweets_df['text_cleaned'] = tweets_df['text'].apply(lambda x: p.clean(x))

#Remove duplicate tweets
tweets_df.drop_duplicates(subset='text_cleaned', keep="first", inplace = True)
len(tweets_df)

#Remove unnecessary characters
#Note: Need to remove % as Stanford CoreNLP annotation encounters error if text contains some of these characters
punct =['%','/',':','\\','&amp;','&',';', '?']

def remove_punctuations(text):
    for punctuation in punct:
        text = text.replace(punctuation, '')
    return text

tweets_df['text_cleaned'] = tweets_df['text_cleaned'].apply(lambda x: remove_punctuations(x))

#Drop tweets which have empty text field
tweets_df['text_cleaned'].replace('', np.nan, inplace=True)
tweets_df['text_cleaned'].replace(' ', np.nan, inplace=True)
tweets_df.dropna(subset=['text_cleaned'], inplace=True)
len(tweets_df)
# 37279

#Drop tweets which have empty location field
tweets_df['location'].replace('', np.nan, inplace=True)
tweets_df['location'].replace(' ', np.nan, inplace=True)
tweets_df.dropna(subset=['location'], inplace=True)
len(tweets_df)
# 29126

tweets_df = tweets_df.reset_index(drop=True)

sns.set_style('whitegrid')
%matplotlib inline
def plot_10_most_common_words(count_data, count_vectorizer):
    import matplotlib.pyplot as plt
    words = count_vectorizer.get_feature_names()
    total_counts = np.zeros(len(words))
    for t in count_data:
        total_counts+=t.toarray()[0]

    count_dict = (zip(words, total_counts))
    count_dict = sorted(count_dict, key=lambda x:x[1], reverse=True)[0:10]
    words = [w[0] for w in count_dict]
    counts = [w[1] for w in count_dict]
    x_pos = np.arange(len(words))

    plt.figure(2, figsize=(20, 20))
    plt.subplot(title='10 most common words')
    sns.set_context("notebook", font_scale=4, rc={"lines.linewidth": 2.5})
```

```
    sns.barplot(x_pos, counts, palette='husl')
    plt.xticks(x_pos, words, rotation=90)
    plt.xlabel('words')
    plt.ylabel('counts')
    plt.show()# Initialise the count vectorizer with the English stop words
count_vectorizer = CountVectorizer(stop_words='english')# Fit and transform the processed titles
count_data = count_vectorizer.fit_transform(tweets_df['text_cleaned'])# Visualise the 10 most
common words
plot_10_most_common_words(count_data, count_vectorizer)
plt.savefig('saved_figure.png')

# Bi-grams
import cufflinks as cf
cf.go_offline()
cf.set_config_file(offline=False, world_readable=True)

def get_top_n_bigram(corpus, n=None):
    vec = CountVectorizer(ngram_range=(2, 4), stop_words='english').fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:n]
common_words = get_top_n_bigram(tweets_df['text_cleaned'], 20)
mydict={}
for word, freq in common_words:
    bigram_df  = pd.DataFrame(common_words, columns = ['ngram' , 'count'])
bigram_df.groupby('ngram').sum()['count'].sort_values(ascending=False).sort_values().plot.barh(title
='Top 20 bigrams in review after removing stop words', color='orange', width=.9, figsize=(12, 8),
stacked = True)

#iplot(kind='bar', yTitle='Count', linecolor='black', title='Top 20 bigrams in review after removing
stop words')

# Location Analysis
#composite_df_copy = composite_df.copy()
nltk_df['location'] = nltk_df['location'].apply(lambda x: (x.split()[-1]).strip() if x.split() else None )
nltk_df

# AFTER MERGING LOCATIONS
top_location_of_tweet = nltk_df['location'].value_counts()
top_location_of_tweet.head(20)
loc_analysis = pd.DataFrame(nltk_df['location'].value_counts().sort_values(ascending=False))
loc_analysis = loc_analysis.rename(columns={'location':'count'})
loc_analysis

import matplotlib.pyplot as plt

group_names=loc_analysis.index[:5]
group_size=loc_analysis['count'][:5]
subgroup_names=['Positive', 'Neutral','Negative','Positive', 'Neutral','Negative','Positive',
'Neutral','Negative','Positive', 'Neutral','Negative',
            'Positive', 'Neutral','Negative']

subgroup_size=[]

composite_df = nltk_df
for city in group_names:
  # positive df
```

```
  pos_len = len(composite_df[(composite_df.nltk_sentiment == 'Positive') & (composite_df.location
== city)].axes[0])
  subgroup_size.append(pos_len)

  # neutral df
  neu_len = len(composite_df[(composite_df.nltk_sentiment == 'Neutral') & (composite_df.location
== city)].axes[0])
  subgroup_size.append(neu_len)

# negative df
  neg_len = len(composite_df[(composite_df.nltk_sentiment == 'Negative') & (composite_df.location
== city)].axes[0])
  subgroup_size.append(neg_len)


subgroup_new_names =
['Pos','Neu','Neg','Pos','Neu','Neg','Pos','Neu','Neg','Pos','Neu','Neg','Pos','Neu','Neg']
subgroup_colors = ['#28AE7B', '#F0F5F3', '#ff0000']


# Create colors
fig, ax = plt.subplots(figsize=(8, 5))
ax.axis('equal')
theme = plt.get_cmap('bwr')
ax.set_prop_cycle("color", [theme(1. * i / len(group_size)) for i in range(len(group_size))])
# First Ring (outside)
fig.subplots_adjust(0.02,0,1,1)
mypie, _ = ax.pie(group_size, radius=1.3, labels=group_names, colors=
[theme(1. * i / len(group_size)) for i in range(len(group_size))] )
plt.setp( mypie, width=0.3, edgecolor='white')

# Second Ring (Inside)
mypie2, _ = ax.pie(subgroup_size, radius=1.3-0.3,
labels=subgroup_new_names, labeldistance=0.7, colors=['#28AE7B', '#F0F5F3', '#ff0000'])
plt.setp( mypie2, width=0.4, edgecolor='white')
plt.margins(0,0)

handles, labels = ax.get_legend_handles_labels()
plt.legend(loc=(1.7, 0.1)) #-> GIVES EVERY COUNTRY KA percent.

def autopct_generator(limit):
    """Remove percent on small slices."""
    def inner_autopct(pct):
        return ('%.2f%%' % pct) if pct > limit else ''
    return inner_autopct


box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 1.3, box.height])

total = sum(group_size)

#   CODE TO DISPLAY ONLY COUNTRIES WITH %
plt.legend(
    loc='upper right',
    labels=['%s, %1.1f%%' % ( l, (float(s) / total) * 100) for l, s in zip(group_names, group_size)],
    prop={'size': 12},
    bbox_to_anchor=(0.0, 1),
    bbox_transform=fig.transFigure
)
```

```
# plt.legend(
#     loc='upper right',
#     labels=['%s, %1.1f%%' % ( l, (float(s) / total) * 100) for l, s in zip(subgroup_names,
subgroup_size)],
#     prop={'size': 12},
#     bbox_to_anchor=(0.0, 1),
#     bbox_transform=fig.transFigure
# )

#   CODE TO DISPLAY ONLY positive -ve neutral
plt.legend(loc=(1.2, 0.1))
plt.show()
```

 # Sentiment Analysis

```
# Define function to get value counts
def get_value_counts(col_name, analyzer_name):
    count = pd.DataFrame(tweets_df[col_name].value_counts())
    percentage = pd.DataFrame(tweets_df[col_name].value_counts(normalize=True).mul(100))
    value_counts_df = pd.concat([count, percentage], axis = 1)
    value_counts_df = value_counts_df.reset_index()
    value_counts_df.columns = ['sentiment', 'counts', 'percentage']
    value_counts_df.sort_values('sentiment', inplace = True)
    value_counts_df['percentage'] = value_counts_df['percentage'].apply(lambda x: round(x,2))
    value_counts_df = value_counts_df.reset_index(drop = True)
    value_counts_df['analyzer'] = analyzer_name
    return value_counts_df
sia = SentimentIntensityAnalyzer()

# Obtaining NLTK scores
tweets_df['nltk_scores']= tweets_df['text_cleaned'].apply(lambda x: sia.polarity_scores(x))

# Obtaining NLTK compound score
tweets_df['nltk_cmp_score']= tweets_df['nltk_scores'].apply(lambda score_dict:
score_dict['compound'])

neutral_thresh = 0.05

# Categorize scores into the sentiments of positive, neutral or negative
tweets_df['nltk_sentiment'] = tweets_df['nltk_cmp_score'].apply(lambda c: 'Positive' if c >=
neutral_thresh else ('Negative' if c <= -(neutral_thresh) else 'Neutral'))

tweets_df['nltk_cmp_score'].describe()
nltk_df = tweets_df[['text_cleaned', 'nltk_sentiment', 'location']]
nltk_sentiment_df = get_value_counts('nltk_sentiment','NLTK Vader')
sns.set_theme(style="dark")
ax = sns.barplot(x="sentiment", y="percentage", data=nltk_sentiment_df)
ax.set_title('NLTK Vader')

for index, row in nltk_sentiment_df.iterrows():
    ax.text(row.name,row.percentage, round(row.percentage,1), color='black', ha="center")
```

 # Visualization after Analysis

```
stop_words = stopwords.words('english')
not_stopwords = {}
common_words  =['com','twitter','please',  'give',  'say',  'still',  'via',  'COVID-19',  'vaccine',  'Vaccine',
'Covid-19', 'new','the','would','could','can','may','must', 'The', 'one', 'take', 'getting', 'doses', 'coronavirus',
'many']
```

```
stop_words.extend(word for word in common_words if word not in stop_words)
final_stop_words = set([word for word in stop_words if word not in not_stopwords])
new_df.text_cleaned = new_df.text_cleaned.apply(lambda x: ' '.join([word for word in x.split() if word not in final_stop_words]))

from wordcloud import WordCloud,STOPWORDS
STOPWORDS.update(['india', 'friends', '2wks', '8c', 'ph3', 'vaccines','citizens', 'states', 'false', 'day', 'data', 'vck', 'absence', 'opposition', 'wants', 'safet', 'distributed', 'bharat', 'ignore', 'biotech', 'bharatbiotech', 'government', 'taken', 'covid', 'amp', 'take', 'vaccine', 'hammering', 'yet', 'european','ji', 'decade', 'covaxin', 'sh', 'eg', 'prev', 'paying', 'voice', 'govt', 'lot', 'taking', 'given', 'less', 'long', 'one', 'days', 'new', 'today',’tweet', 'us', 'takeaway', 'must', 'many', 'add', 'will', 'without', 'known', 'speaking', 'even', 'little', 'none', 'people', 'made', 'common', 'got', 'back','india', 'friends', 'taken', 'covid', 'based', 'take', 'S', 'example', 'lik', 'modi', 'phase', 'covidvaccine', 'covid19india', 'show', 'well', 'now', 'already','t', 'decade', 'covishield', 'sh', 'rs', 'astra', 'zeneca', 'oxford', 'countries', 'pe', 'rt', 'covid19vaccine', 'took', 'two', 'look', 'ye', 've', 'make', 'giving', 're','indian', 'sii', 'mean', 'know', 'example link', 'hand', 'chest', 'set', 'bharat biotech', 'COVID-19', 'vaccine', 'Vaccine', 'vaccination', 'covid19', 'Covid-19'])

normal_words = ' '.join([text for text in new_df['text_cleaned'][new_df['nltk_sentiment'] == 'Neutral']])
wordCloud = WordCloud(max_words = 100, width=1000, height=600, random_state=42,background_color='black',colormap='Set2', max_font_size=110, stopwords=STOPWORDS).generate(normal_words)
plt.figure(figsize=(20,10))
plt.imshow(wordCloud, interpolation='bilinear')
plt.axis('off')
```

 # Sentiment Analysis of Covishield and Covaxin

```
all_vax = ['covaxin', 'covishield']

# Function to filter the data to a single vaccine
def filter_by_vaccy(df, vax):
    df_filt = pd.DataFrame()
    for v in vax:
        df_filt = df_filt.append(tweets_df[tweets_df['text'].str.lower().str.contains(v)])
    other_vax = list(set(all_vax)-set(vax))
    for o in other_vax:
        df_filt = df_filt[~df_filt['text'].str.lower().str.contains(o)]
#     df_filt = df_filt.drop_duplicates()
    timeline = df_filt.groupby(['tweet_date']).agg(np.nanmean).reset_index()

    timeline = timeline[['tweet_date']]
    return df_filt, timeline
covaxin_df, covaxin_timeline = filter_by_vaccy(tweets_df, ['covaxin'])
covishield_df, covishield_timeline = filter_by_vaccy(tweets_df, ['covishield'])

from langdetect import detect

#Extracting only the tweet column from the dfs

covishield_df = covishield_df['text']
covishield_df = covishield_df.to_frame().reset_index()
del covishield_df['index']

covaxin_df = covaxin_df['text']
covaxin_df = covaxin_df.to_frame().reset_index()
```

```
del covaxin_df['index']

def cleanTweets(text):

    if text == '' or text=='...' or text== None:
        return 'None'
    try:
        lang = detect(text)
        if lang !='en':
            return 'None'
    except Exception as e:
        print(e)
        print(text)
        return 'None'

    # Convert to lowercase
    text = text.lower()
    # Remove mentions
    text = re.sub(r'@[A-Za-z0-9_]+', '', text)
    # Remove hashtags
    text = re.sub(r'#', '', text)
    # Remove retweets:
    text = re.sub(r'RT : ', '', text)
    # Remove urls
    text = re.sub(r'https?:\/\/[A-Za-z0-9\.\/]+', '', text)
    # Removing extra spaces from start and end
    text = re.sub(r"^\s+", "", text)

    return text

covishield_df['text'] = covishield_df['text'].apply(cleanTweets)
covaxin_df['text'] = covaxin_df['text'].apply(cleanTweets)

# Define function to get value counts
def get_value_counts_covi(col_name, analyzer_name):
    count = pd.DataFrame(covishield_df[col_name].value_counts())
    percentage = pd.DataFrame(covishield_df[col_name].value_counts(normalize=True).mul(100))
    value_counts_df = pd.concat([count, percentage], axis = 1)
    value_counts_df = value_counts_df.reset_index()
    value_counts_df.columns = ['sentiment', 'counts', 'percentage']
    value_counts_df.sort_values('sentiment', inplace = True)
    value_counts_df['percentage'] = value_counts_df['percentage'].apply(lambda x: round(x,2))
    value_counts_df = value_counts_df.reset_index(drop = True)
    value_counts_df['analyzer'] = analyzer_name
    return value_counts_df

# Define function to get value counts
def get_value_counts_cov(col_name, analyzer_name):
    count = pd.DataFrame(covaxin_df[col_name].value_counts())
    percentage = pd.DataFrame(covaxin_df[col_name].value_counts(normalize=True).mul(100))
    value_counts_df = pd.concat([count, percentage], axis = 1)
    value_counts_df = value_counts_df.reset_index()
    value_counts_df.columns = ['sentiment', 'counts', 'percentage']
    value_counts_df.sort_values('sentiment', inplace = True)
    value_counts_df['percentage'] = value_counts_df['percentage'].apply(lambda x: round(x,2))
    value_counts_df = value_counts_df.reset_index(drop = True)
```

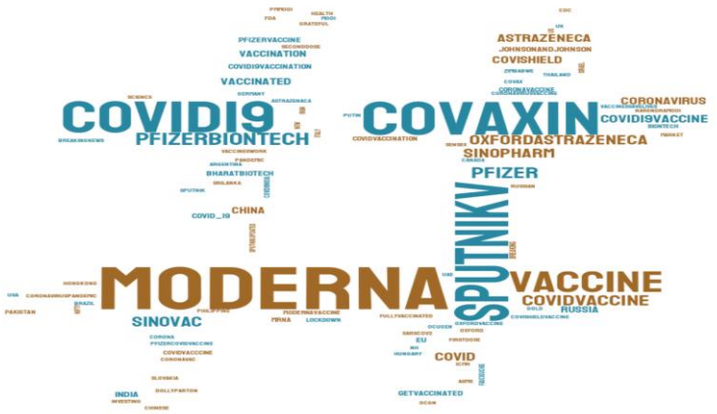
```
    value_counts_df['analyzer'] = analyzer_name
    return value_counts_df
sia = SentimentIntensityAnalyzer()

# Obtaining NLTK scores
covishield_df['nltk_scores'] = covishield_df['text'].apply(lambda x: sia.polarity_scores(x))
covaxin_df['nltk_scores'] = covaxin_df['text'].apply(lambda x: sia.polarity_scores(x))

# Obtaining NLTK compound score
covishield_df['nltk_cmp_score'] = covishield_df['nltk_scores'].apply(lambda score_dict:
score_dict['compound'])
covaxin_df['nltk_cmp_score'] = covaxin_df['nltk_scores'].apply(lambda score_dict:
score_dict['compound'])

neutral_thresh = 0.05

# Sentiment distribution for Covaxin
sns.set_theme(style="dark")
ax = sns.barplot(x="sentiment", y="percentage", data=covaxin_df)
ax.set_title('Covaxin sentiment distribution')

for index, row in covaxin_df.iterrows():
    ax.text(row.name,row.percentage, round(row.percentage,1), color='black', ha="center")

# Sentiment distribution for Covishield
sns.set_theme(style="dark")
ax = sns.barplot(x="sentiment", y="percentage", data=covishield_df)
ax.set_title('covishield sentiment distribution')

for index, row in covishield_df.iterrows():
    ax.text(row.name,row.percentage, round(row.percentage,1), color='black', ha="center")
```

Output:

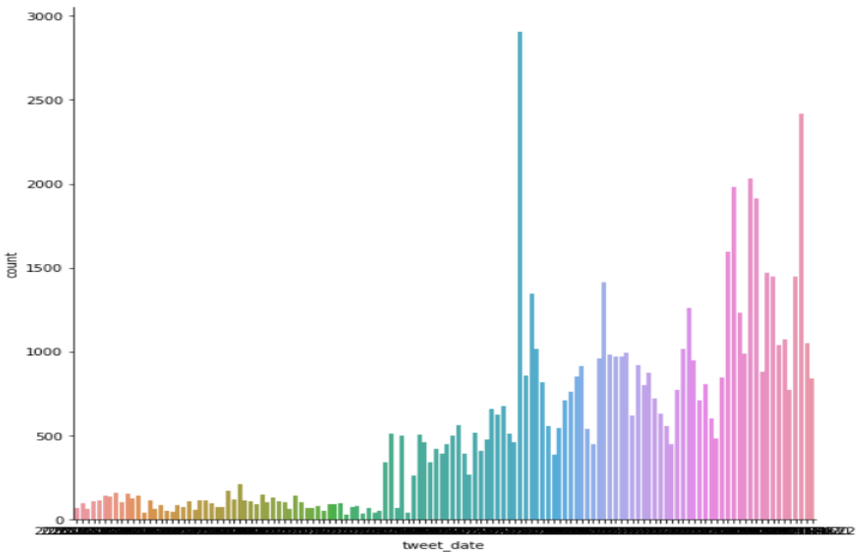
tweets_df.head()

	username	location	verified	tweet_date	text	hashtags
0	Rachel Roh	La Crescenta-Montrose, CA	False	20-12-2020 06:06	Same folks said daikon paste could treat a cyt...	['PfizerBioNTech']
1	Albert Fong	San Francisco, CA	False	13-12-2020 16:27	While the world has been on the wrong side of ...	NaN
2	eli???ñ???????¡???§????	Your Bed	False	12-12-2020 20:33	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf...
3	Charles Adler	Vancouver, BC - Canada	True	12-12-2020 20:23	Facts are immutable, Senator, even when you're...	NaN
4	Citizen News Channel	NaN	False	12-12-2020 20:17	Explain to me again why we need a vaccine @Bor...	['whereareallthesickpeople', 'PfizerBioNTech']

hashtags.png



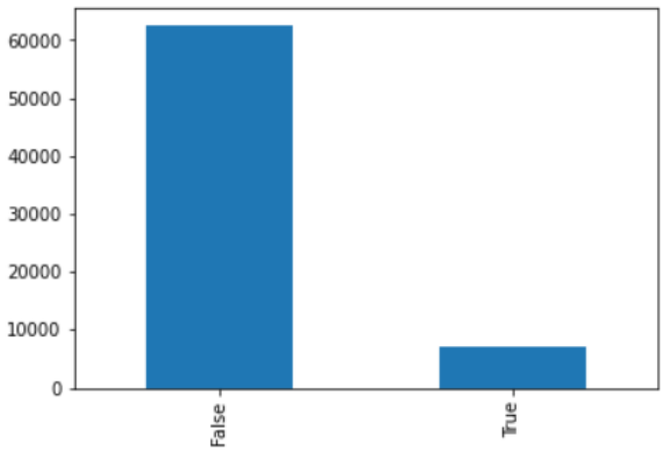
sns.catplot("tweet_date", data=tweets_df, kind="count", height=8)



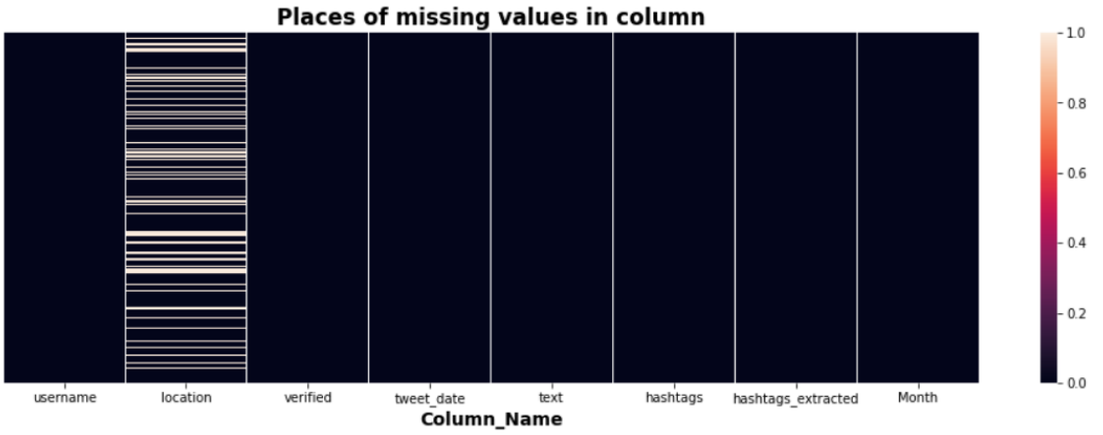
tweets_df['location'].value_counts()

```
India                                2289
Toronto, Canada  and Worldwide      2080
New Delhi, India                     914
United States                        721
Mumbai, India                        586
                                     ...
Tampa,FL to Delft, NL                  1
bikaner rajasthan                      1
Behind the "Offensive Replies"         1
New Delhi, India (Bharat)              1
Lawrenceville, GA                      1
Name: location, Length: 12748, dtype: int64
```

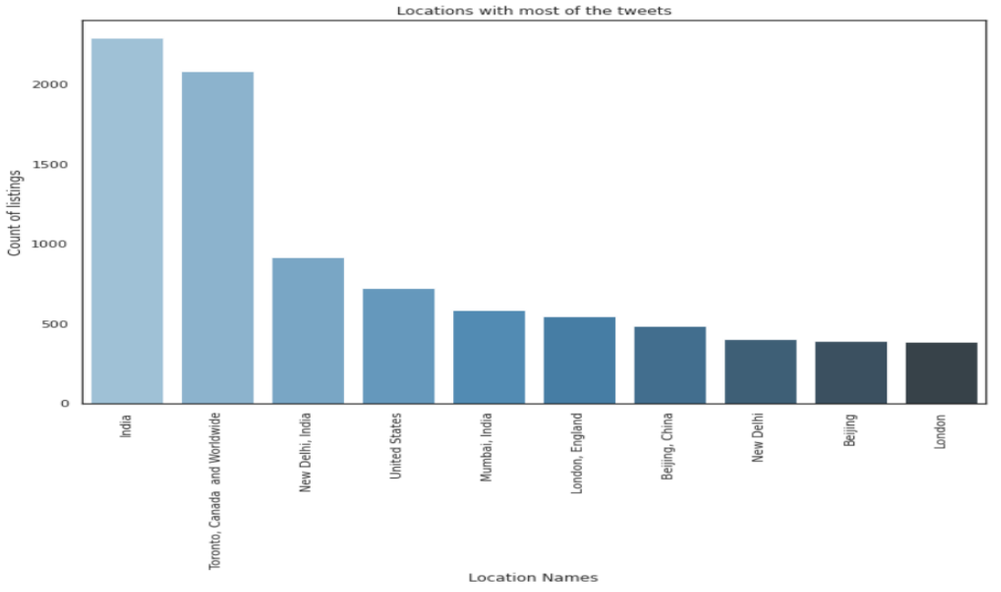
Verified and Non-verified Users
tweets_df['verified'].value_counts().head(n=10).plot.bar()



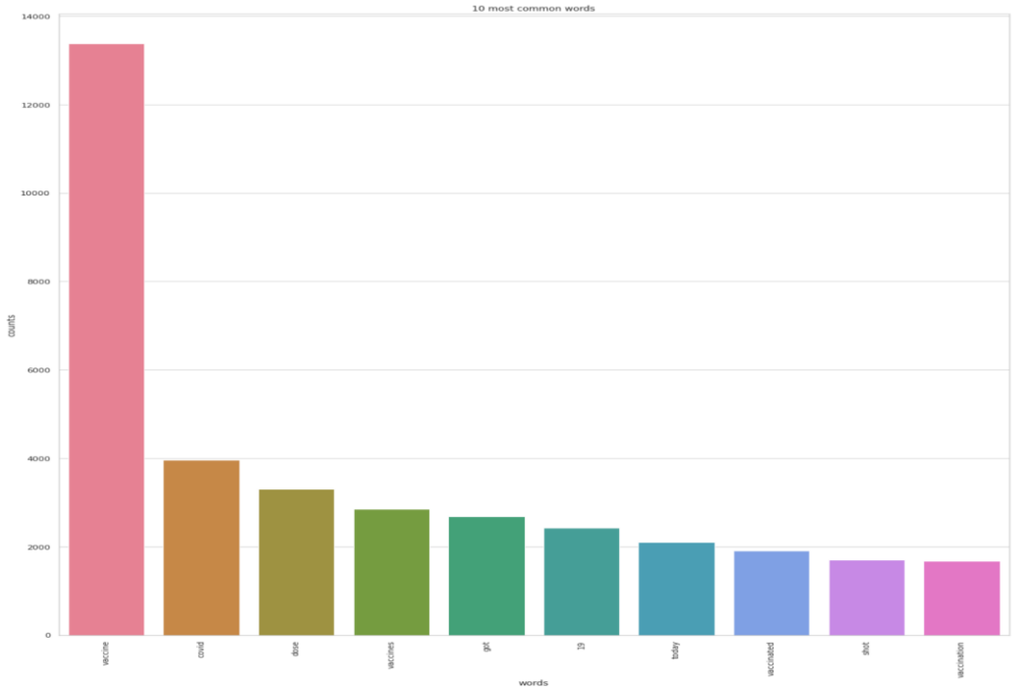
Heatmap for missing values



Top Locations of Tweet



Most recurring words in tweets



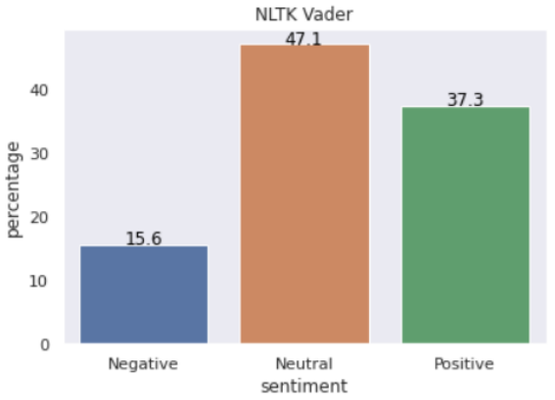
Bi-grams

Sentiment Analysis

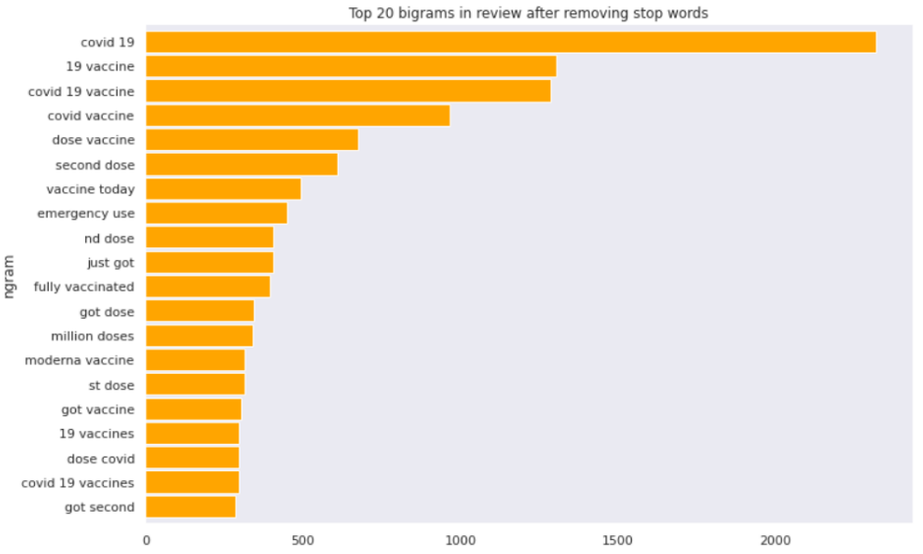
nltk_sentiment_df = get_value_counts('nltk_sentiment','NLTK Vader')

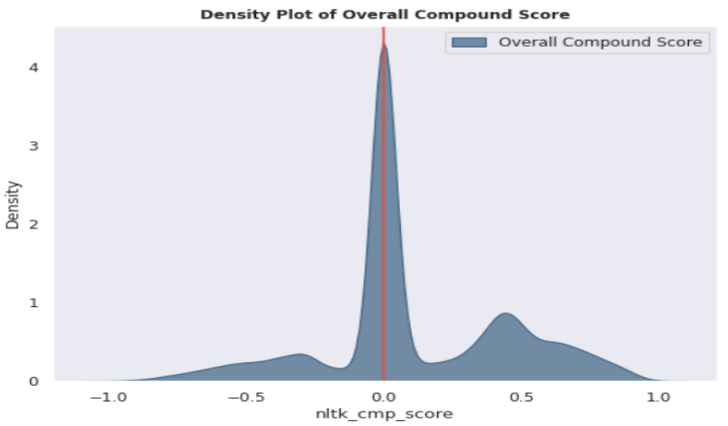
	sentiment	counts	percentage	analyzer
0	Negative	4552	15.63	NLTK Vader
1	Neutral	13706	47.06	NLTK Vader
2	Positive	10868	37.31	NLTK Vader

ax = sns.barplot(x="sentiment", y="percentage", data=nltk_sentiment_df)



Density Plot of Overall Compound Score



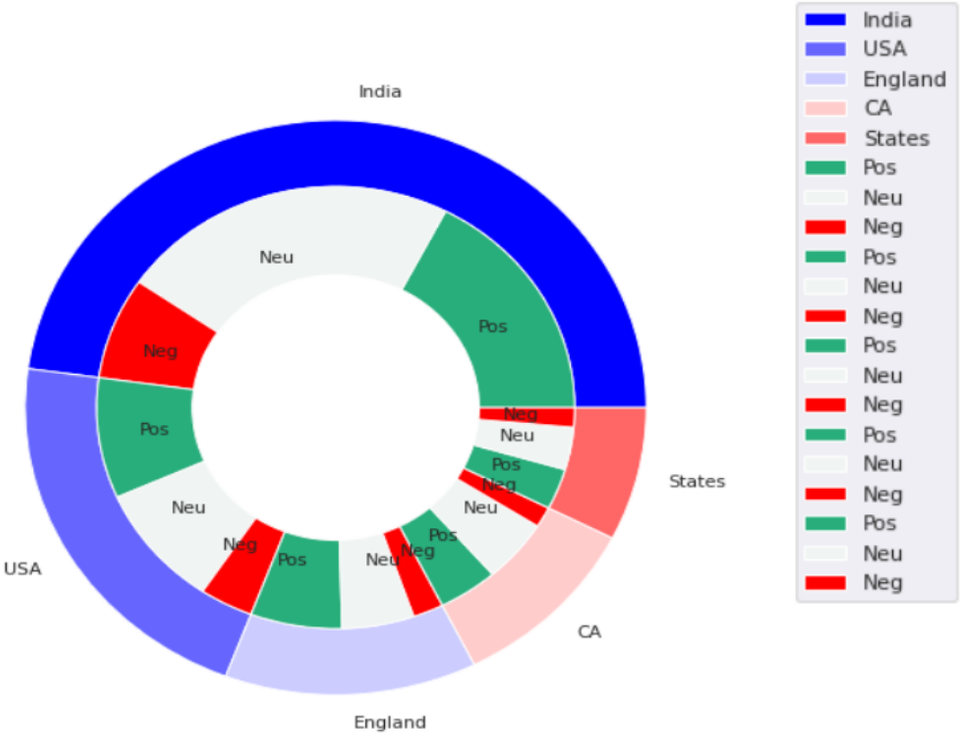


Location Analysis

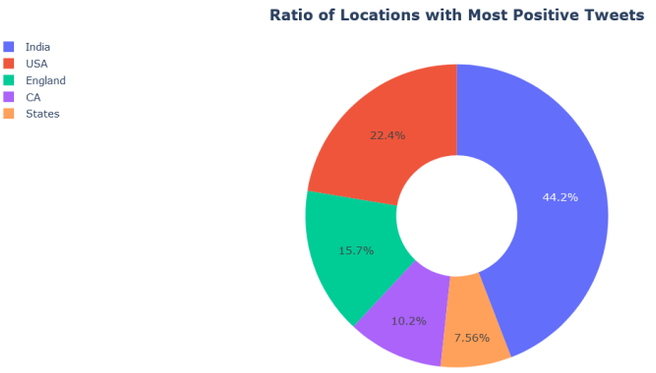
top_location_of_tweet = nltk_df['location'].value_counts()

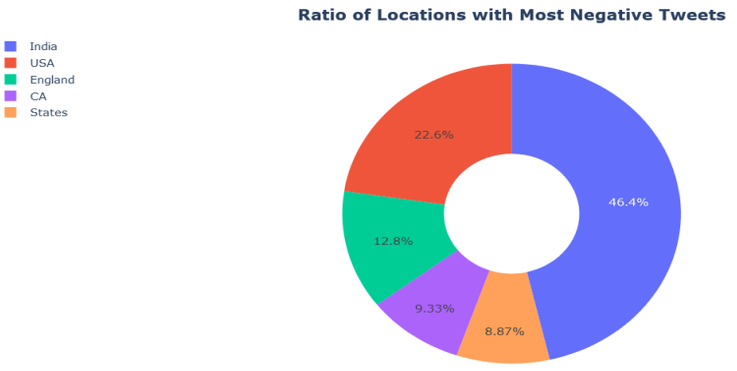
```
India           3258
USA             1458
England          894
CA               690
States           511
China            509
UK               454
NY               418
Canada           400
TX               398
Delhi            397
Kingdom          361
Beijing          322
London           314
Pakistan         306
FL               255
Russia           219
Philippines      218
Ireland          198
Worldwide        195
Name: location, dtype: int64
```

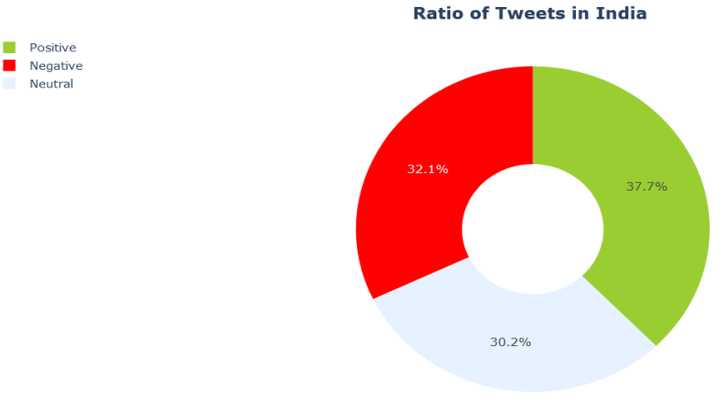
#PIE chart of tweets Worldwide



Visualization after Analysis

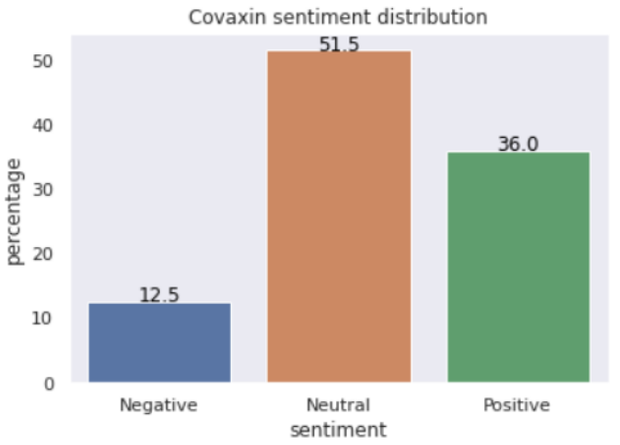




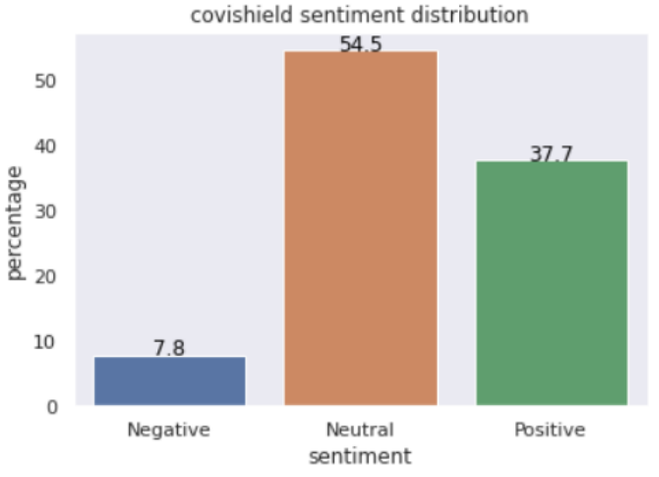


Wordcloud of normal words from Tweets

ax = sns.barplot(x="sentiment", y="percentage", data= covaxin_df)



ax = sns.barplot(x="sentiment", y="percentage", data= covishield_df)



Conclusion:

It was studied and understood that exploratory analysis ensures the results produced are valid and applicable to any desired business outcomes and goals. EDA along with various visualization techniques was performed to get a deeper understanding of that data. Various techniques were used like heatmaps, density plots, word cloud, pie-chart, bar chart, bi-grams to make sense out of the data, find missing values etc, which ultimately helps in making robust social media analytics model.

R1 (3 Marks)	R2 (3 Marks)	R3 (3 Marks)	R4 (3 Marks)	R5 (3 Marks)	Total (15 Marks)	Signature

EXPERIMENT NO 5

Date of performance:

Date of submission:

Aim: Develop Content(text, emoticons, image, audio, video) based social media analytics model for business. (e.g.Content Based Analysis :Topic , Issue ,Trend, sentiment/opinion analysis, audio, video, image analytics)

Theory:

Sentiment analysis is a technique that uses natural language processing and machine learning algorithms to identify and extract subjective information from text. In the context of social media analytics, sentiment analysis tools can be used to analyze the sentiment of social media posts, comments, and messages.

Social media analytics is the process of collecting, analyzing, and interpreting data from social media platforms. It provides insights into how people perceive and interact with brands, products, and services. Sentiment analysis tools can enhance social media analytics by providing information on the overall sentiment of social media conversations, as well as the sentiment of individual posts or messages.

Here are some ways in which sentiment analysis tools can be used in social media analytics:

- Monitoring brand reputation: Sentiment analysis tools can be used to monitor social media conversations about a brand or product. By analyzing the sentiment of these conversations, companies can identify areas where they need to improve their products or services, as well as areas where they are doing well.

- Understanding customer feedback: Social media is a platform where customers often share their opinions and feedback about a product or service. Sentiment analysis tools can be used to analyze this feedback and identify trends in customer sentiment. Companies can then use this information to improve their products or services and address any customer concerns.

- Market research: Sentiment analysis tools can be used to analyze social media conversations related to a particular topic, product or service. This can help companies understand consumer preferences, opinions, and behaviors, and make informed decisions based on this data.

- Identifying influencers: Sentiment analysis tools can be used to identify social media influencers who have a positive or negative impact on a brand's reputation. Companies can then work with these influencers to promote their products or services or address any negative sentiment.

- Competitive analysis: Sentiment analysis tools can be used to compare a brand's sentiment with that of its competitors. This information can be used to identify areas where a brand is falling behind its competitors and take steps to improve its products or services.

- Political analysis: Sentiment analysis can also be used in political analysis to understand public opinion and sentiment towards a particular candidate or issue.

- Crisis management: Sentiment analysis tools can be used to monitor social media conversations during a crisis. By analyzing the sentiment of these conversations, companies can identify the extent of the crisis and take steps to address any negative sentiment.

In conclusion, sentiment analysis tools can be a valuable addition to social media analytics. They provide insights into customer sentiment and can help companies improve their products, services, and reputation.

Steps to build sentiment analysis models:

1. Data collection: Collect a large dataset of text documents with associated sentiment labels. This dataset should be representative of the domain or application for which the model will be used.

2. Text pre-processing: Pre-process the text data by removing stop words, stemming, and lemmatizing the words. This step also involves cleaning the text data by removing special characters, URLs, and any other unwanted elements.

3. Feature extraction: Transform the pre-processed text data into numerical features that can be used as inputs to machine learning models. Common techniques for feature extraction in sentiment analysis include bag-of-words, term frequency-inverse document frequency (TF-IDF), and word embeddings.

4. Model training: Select an appropriate machine learning algorithm, such as logistic regression, decision trees, or deep learning, and train the model on the labeled dataset. The model should be optimized for the specific application, such as binary or multi-class classification, and the appropriate performance metrics should be selected.

5. Model evaluation: Evaluate the performance of the trained model on a held-out test set. Common evaluation metrics for sentiment analysis include accuracy, precision, recall, F1-score, and AUC-ROC.

6. Model deployment: Deploy the trained model in a production environment, such as a web application or API, where it can be used to classify the sentiment of new text data.

7. Model maintenance: Monitor the performance of the deployed model and retrain it periodically with new data to ensure it stays up to date and accurate.

It is important to note that building an effective sentiment analysis model is an iterative process, and requires experimentation with different techniques and parameters to achieve the desired performance. Additionally, the quality and size of the labelled dataset is critical to the success of the model, and obtaining high-quality labelled data can be a time-consuming and expensive process.

Code and Output:

Sentiment Analysis on Movie Review Data

Using dedicated GPU

In [1]:
```
import tensorflow as tf
from tensorflow import keras
```

In [2]:
```
print("Num GPUs Available: ", len(tf.config.experimental.list_physical_devices('GPU')))
```

```
Num GPUs Available:  1
```

In [3]:
```
tf.test.is_built_with_cuda()
print(tf.version.VERSION)
```

```
2.6.0
```

In [4]:
```
import sys
sys.version
```

Out[4]:
```
'3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]'
```

In [5]:
```
import os
import pandas as pd
import numpy as np
import gc
import matplotlib.pyplot as plt
import operator
import seaborn as sns
from wordcloud import WordCloud,STOPWORDS

import re


import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow import keras

from keras.callbacks import EarlyStopping,ModelCheckpoint
# from keras.utils import to_categorical
from tensorflow.keras.utils import to_categorical
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Embedding,Conv1D,LSTM,GRU,BatchNormalization,Flatten,Dense


for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Data Preprocessing

In [7]:
```
df= pd.read_csv("dataSentimental/IMDB Dataset/IMDB Dataset.csv")
df.head()
```

Out[7]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   review     50000 non-null  object
 1   sentiment  50000 non-null  object
dtypes: object(2)
memory usage: 781.4+ KB
```

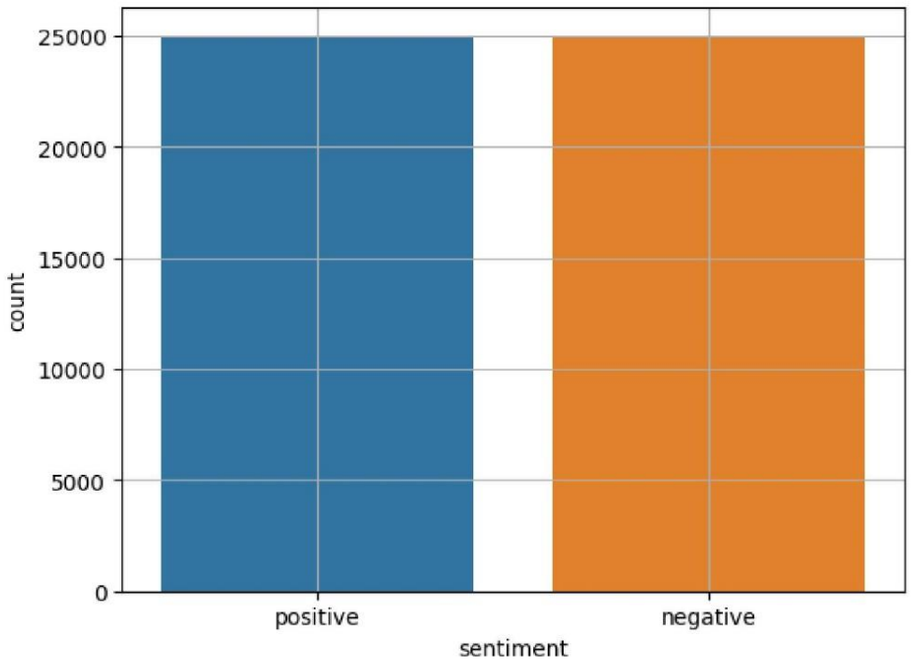
Let's find if the data contains any missing value

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
review       0
sentiment    0
dtype: int64
```

We will find the count of each type of sentiment in the dataset using seaborn library

In [10]:
```
sns.countplot(x=df['sentiment'])
plt.grid()
```



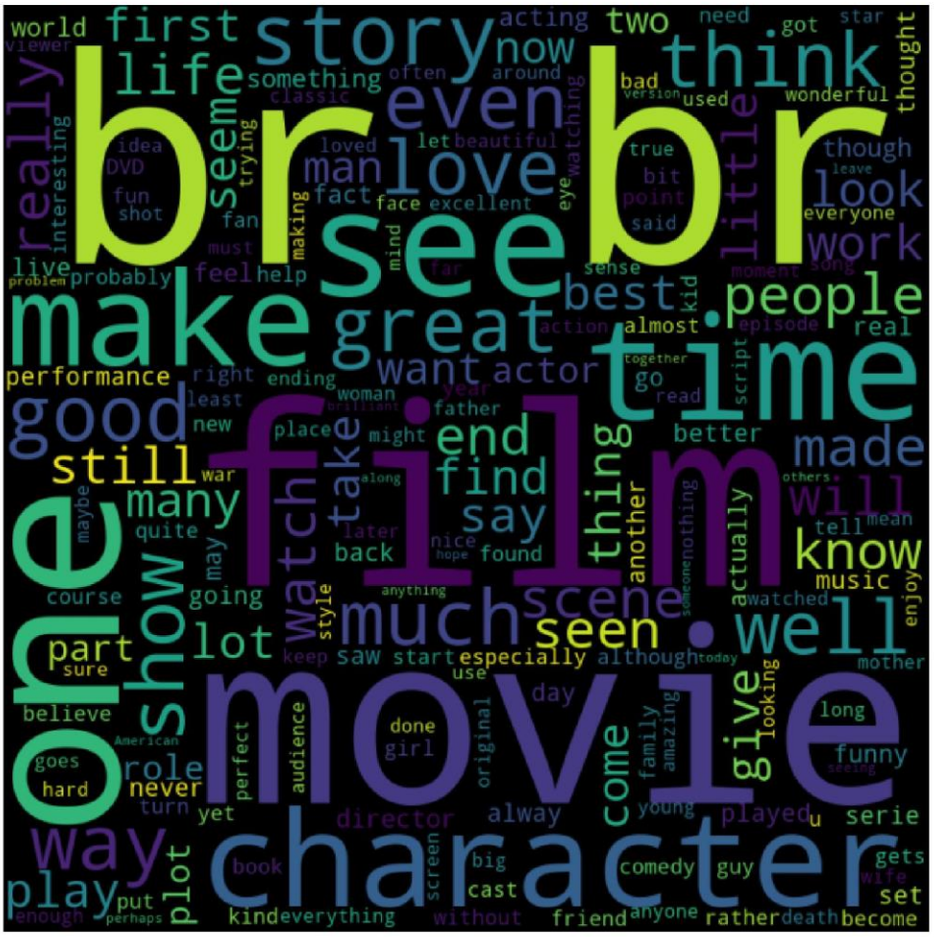
In [11]:
```
sentences=df['review']
le=LabelEncoder()
df['sentiment']= le.fit_transform(df['sentiment'])
```

Data visualization using word cloud for finding the most used words for each type of sentiment

In [12]:
```
stopwords = set(STOPWORDS)

pos=' '.join(map(str,sentences[df['sentiment']==1]))
neg=' '.join(map(str,sentences[df['sentiment']==0]))

wordcloud1 = WordCloud(width = 800, height = 800,
                background_color ='black',
                stopwords = stopwords,
                min_font_size = 10).generate(pos)

plt.figure(figsize=(8,8))
plt.imshow(wordcloud1)
plt.title('Positive Sentiment')
plt.axis('off')
```

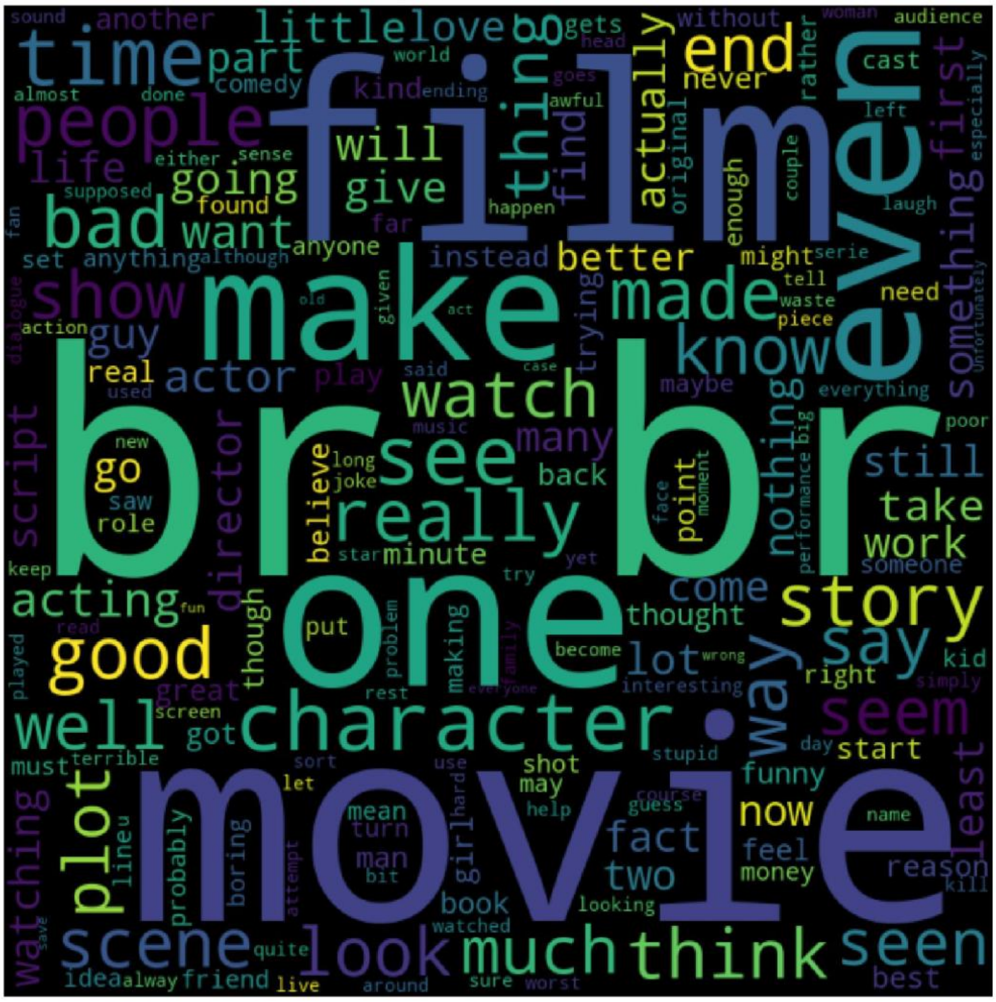
Out[12]:
```
(-0.5, 799.5, 799.5, -0.5)
```

Positive Sentiment



In [13]:
```
plt.figure(figsize=(8,8))
wordcloud2 = WordCloud(width = 800, height = 800,
                background_color ='black',
                stopwords = stopwords,
                min_font_size = 10).generate(neg)

plt.imshow(wordcloud2)
plt.title('Negative Sentiment')
plt.axis('off')

plt.show()
```

Negative Sentiment




```
In [14]: labels=to_categorical(df['sentiment'],num_classes=2)
X_train,X_test,Y_train,Y_test = train_test_split(df['review'],labels,test_size=0.1,random_state=10)
```

Data Cleaning

Note: In this model we will be using glove embeddings.
It has a large vocabulary and we can find the words from our data which are not present in the glove
(these words are contractions, misspelled words, concated words or emojis which can decrease our model's performance).
We will then use re library to remove these words from the dataset.

```
In [16]: glove_embeddings= np.load('dataSentimental/glove/glove.840B.300d.pkl',
                               allow_pickle=True)
```

We will build vocabulary and count of each vocabulary using the below function

```
In [17]: def vocab_build(review):

    comments = review.apply(lambda s: s.split()).values
    vocab={}

    for comment in comments:
        for word in comment:
            try:
                vocab[word]+=1

            except KeyError:
                vocab[word]=1
    return vocab
```

Embedding Coverage tells how much percentage of the words in our data are covered by the vocabulary.
sorted_oov is the list of words which we need to do text cleaning on.

In [18]:
```
def embedding_coverage(review,embeddings):

    vocab=vocab_build(review)

    covered={}
    word_count={}
    oov={}
    covered_num=0
    oov_num=0

    for word in vocab:
        try:
            covered[word]=embeddings[word]
            covered_num+=vocab[word]
            word_count[word]=vocab[word]
        except:
            oov[word]=vocab[word]
            oov_num+=oov[word]

    vocab_coverage=len(covered)/len(vocab)*100
    text_coverage = covered_num/(covered_num+oov_num)*100

    sorted_oov=sorted(oov.items(), key=operator.itemgetter(1))[::-1]
    sorted_word_count=sorted(word_count.items(), key=operator.itemgetter(1))[::-1]

    return sorted_word_count,sorted_oov,vocab_coverage,text_coverage
```

In [19]:
```
train_covered,train_oov,train_vocab_coverage,train_text_coverage=embedding_coverage(X_train,glove_embeddings)
test_covered,test_oov, test_vocab_coverage, test_text_coverage = embedding_coverage(X_test,glove_embeddings)

print(f"Glove embeddings cover {round(train_vocab_coverage,2)}% of vocabulary and {round(train_text_coverage,2)}%
print(f"Glove embeddings cover {round(test_vocab_coverage,2)}% of vocabulary and {round(test_text_coverage,2)}% t
```

```
Glove embeddings cover 26.74% of vocabulary and 87.62% text in training set
Glove embeddings cover 42.72% of vocabulary and 87.73% text in testing set
```

train_oov shows the words which we need to preprocess

In [20]:
```
train_oov[:10]
```

Out[20]:
```
[('/><br', 90971),
 ('/>The', 12949),
 ('film,', 7318),
 ('movie,', 7167),
 ('/>I', 6565),
 ("isn't", 5241),
 ('"The', 4379),
 ("he's", 3981),
 ('/>This', 3969),
 ("wasn't", 3788)]
```

In [21]:
```
def clean_sentences(line):

    line=re.sub('<.*?>','',line) # removing html tags

    #removing contractions
    line=re.sub("isn't",'is not',line)
    line=re.sub("he's",'he is',line)
    line=re.sub("wasn't",'was not',line)
    line=re.sub("there's",'there is',line)
    line=re.sub("couldn't",'could not',line)
    line=re.sub("won't",'will not',line)
    line=re.sub("they're",'they are',line)
    line=re.sub("she's",'she is',line)
    line=re.sub("There's",'there is',line)
    line=re.sub("wouldn't",'would not',line)
    line=re.sub("haven't",'have not',line)
    line=re.sub("That's",'That is',line)
    line=re.sub("you've",'you have',line)
    line=re.sub("He's",'He is',line)
    line=re.sub("what's",'what is',line)
    line=re.sub("weren't",'were not',line)
    line=re.sub("we're",'we are',line)
    line=re.sub("hasn't",'has not',line)
    line=re.sub("you'd",'you would',line)
    line=re.sub("shouldn't",'should not',line)
    line=re.sub("let's",'let us',line)
    line=re.sub("they've",'they have',line)
    line=re.sub("You'll",'You will',line)
```

```
    line=re.sub("i'm",'i am',line)
    line=re.sub("we've",'we have',line)
    line=re.sub("it's",'it is',line)
    line=re.sub("don't",'do not',line)
    line=re.sub("that´s",'that is',line)
    line=re.sub("I´m",'I am',line)
    line=re.sub("it’s",'it is',line)
    line=re.sub("she´s",'she is',line)
    line=re.sub("he’s'",'he is',line)
    line=re.sub('I’m','I am',line)
    line=re.sub('I’d','I did',line)
    line=re.sub("he’s'",'he is',line)
    line=re.sub('there’s','there is',line)

    #special characters and emojis
    line=re.sub('\x91The','The',line)
    line=re.sub('\x97','',line)
    line=re.sub('\x84The','The',line)
    line=re.sub('\uf0b7','',line)
    line=re.sub('¡¨','',line)
    line=re.sub('\x95','',line)
    line=re.sub('\x8ei\x9eek','',line)
    line=re.sub('\xad','',line)
    line=re.sub('\x84bubble','bubble',line)

    # remove concated words
    line=re.sub('trivialBoring','trivial Boring',line)
    line=re.sub('Justforkix','Just for kix',line)
    line=re.sub('Nightbeast','Night beast',line)
    line=re.sub('DEATHTRAP','Death Trap',line)
    line=re.sub('CitizenX','Citizen X',line)
    line=re.sub('10Rated','10 Rated',line)
    line=re.sub('_The','_ The',line)
    line=re.sub('1Sound','1 Sound',line)
    line=re.sub('blahblahblahblahblahblahblahblahblahblahblahblahblahblahblahblah','blah blah',line)
    line=re.sub('ResidentHazard','Resident Hazard',line)
    line=re.sub('iameracing','i am racing',line)
    line=re.sub('BLACKSNAKE','Black Snake',line)
    line=re.sub('DEATHSTALKER','Death Stalker',line)
    line=re.sub('_is_','is',line)
    line=re.sub('10Fans','10 Fans',line)
    line=re.sub('Yellowcoat','Yellow coat',line)
    line=re.sub('Spiderbabe','Spider babe',line)
    line=re.sub('Frightworld','Fright world',line)

    #removing punctuations

    punctuations = '@#!~?+&*[]-%._-:/£();$=><|{}^' + '''"“”’`'''
    for p in punctuations:
        line = line.replace(p, f' {p} ')

    line=re.sub(',',' , ',line)

    # ... and ..
    line = line.replace('...', ' ... ')

    if '...' not in line:
        line = line.replace('..', ' ... ')

    return line
```

After cleaning the dataset we can see that now our vocabulary covers almost 87% on training set and 95.5% on testing set which initially was far less.

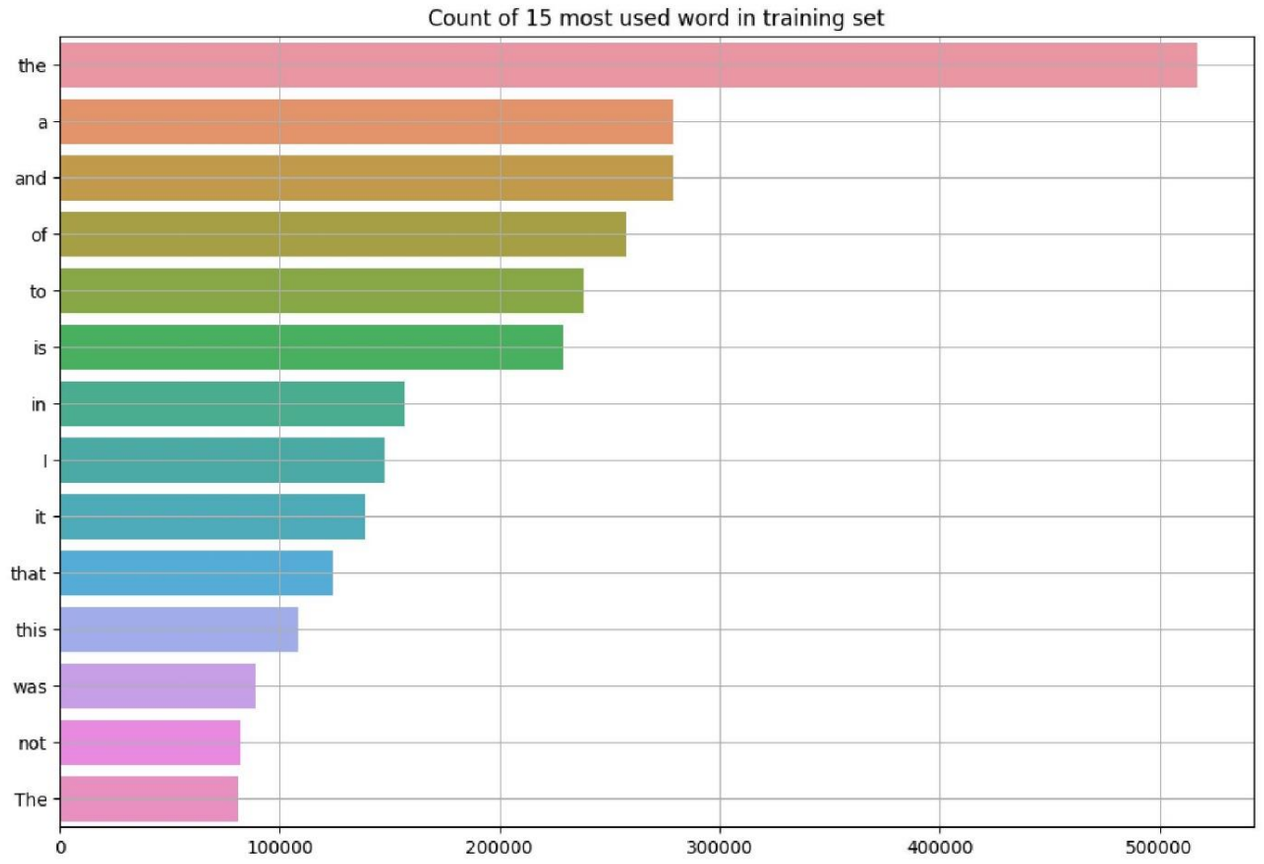
In [22]:
```
X_train=X_train.apply(lambda s: clean_sentences(s))
X_test=X_test.apply(lambda s: clean_sentences(s))

train_covered,train_oov,train_vocab_coverage,train_text_coverage=embedding_coverage(X_train,glove_embeddings)
print(f"Glove embeddings cover {round(train_vocab_coverage,2)}% of vocabulary and {round(train_text_coverage,2)}%

test_covered,test_oov,test_vocab_coverage,test_text_coverage=embedding_coverage(X_test,glove_embeddings)
print(f"Glove embeddings cover {round(test_vocab_coverage,2)}% of vocabulary and {round(test_text_coverage,2)}% t
```

```
Glove embeddings cover 86.92% of vocabulary and 99.82% text in training set
Glove embeddings cover 95.5% of vocabulary and 99.82% text in training set
```

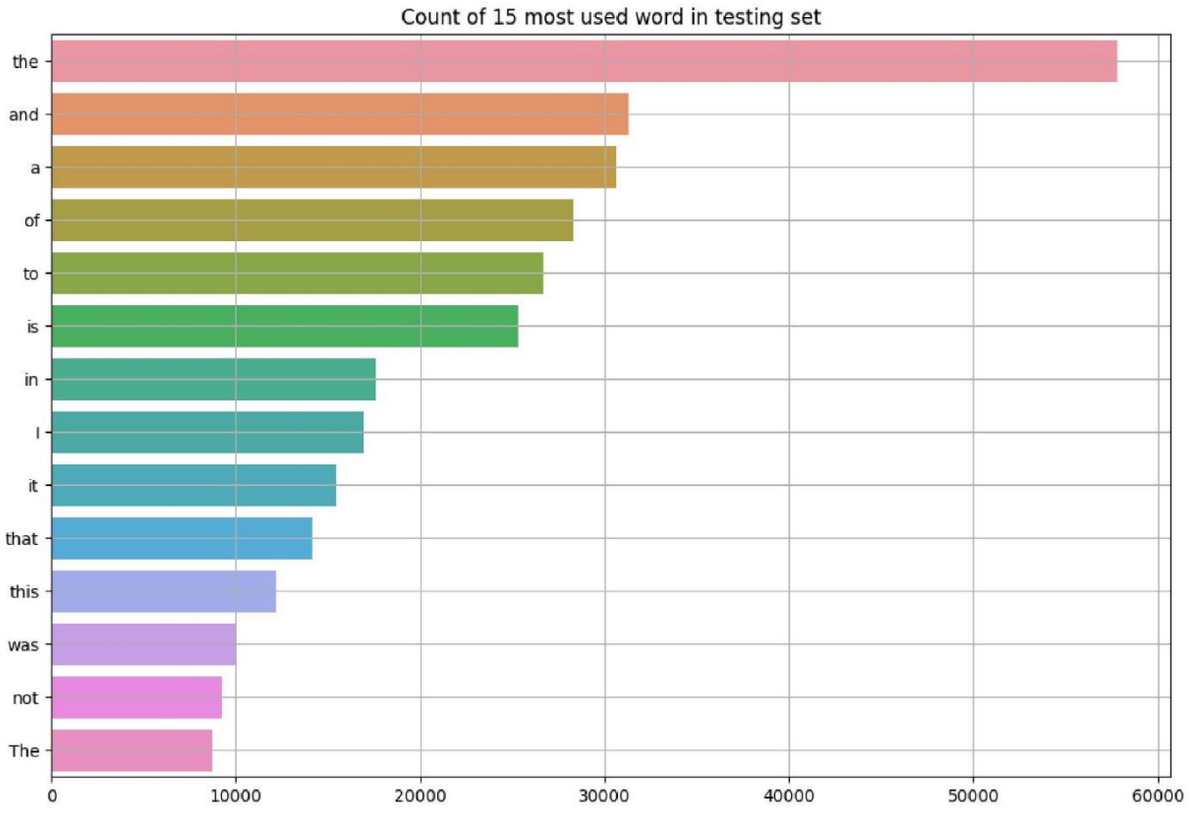
using seaborn's barplot let's find out the count of 10 most used words in training and testing set

In [23]:
```
punctuations = '@#!~?+&*[]-%._-:/£();$=><|{},^' + '''"“”‘`'''
train_word=[]
train_count=[]

i=1
for word,count in train_covered:
    if word not in punctuations:
        train_word.append(word)
        train_count.append(count)
        i+=1
    if(i==15):
        break

test_word=[]
test_count=[]

i=1
for word,count in test_covered:
    if word not in punctuations:
        test_word.append(word)
        test_count.append(count)
        i+=1
    if(i==15):
        break
```

In [24]:
```
plt.figure(figsize=(12,8))
sns.barplot(x=train_count,y=train_word).set_title('Count of 15 most used word in training set')
plt.grid()
```



In [25]:
```
plt.figure(figsize=(12,8))
sns.barplot(x=test_count,y=test_word).set_title('Count of 15 most used word in testing set')
plt.grid()
```



We will delete the embeddings as it takes too much memory

In [26]:
```
del glove_embeddings,train_oov,test_oov
gc.collect()
```

Out[26]:
```
12090
```

Model Building

In [27]:
```
num_words=80000
embeddings=256
```

In [28]:
```
tokenizer=Tokenizer(num_words=num_words,oov_token='<OOV>')
tokenizer.fit_on_texts(X_train)
word_index=tokenizer.word_index
total_vocab=len(word_index)
```

In [29]:
```
print("Vocabulary of the dataset is : ",total_vocab)
```

```
Vocabulary of the dataset is :  100954
```

In [30]:
```
sequences_train=tokenizer.texts_to_sequences(X_train)
sequences_test=tokenizer.texts_to_sequences(X_test)

max_len=max(max([len(x) for x in sequences_train]),max([len(x) for x in sequences_test]))

train_padded=pad_sequences(sequences_train,maxlen=max_len)
test_padded=pad_sequences(sequences_test,maxlen=max_len)
```

In [31]:
```
X_train,X_val,Y_train,Y_val=train_test_split(train_padded,Y_train,
                                             test_size=0.05,random_state=10)
```

We will 2 LSTM layers and Conv1D layer for training the model.

Using Dropout reduces the overfitting by decreasing the bias and is a must since there is lot of variance seen.

In [32]:
```
model= keras.Sequential()
model.add(Embedding(num_words,embeddings,input_length=max_len))
model.add(Conv1D(256,10,activation='relu'))
model.add(keras.layers.Bidirectional(LSTM(128,return_sequences=True)))
model.add(LSTM(64))
```

```
model.add(keras.layers.Dropout(0.4))
model.add(Dense(2,activation='softmax'))
```

In [33]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 2527, 256)         20480000
_________________________________________________________________
conv1d (Conv1D)              (None, 2518, 256)         655616
_________________________________________________________________
bidirectional (Bidirectional (None, 2518, 256)         394240
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                82176
_________________________________________________________________
dropout (Dropout)            (None, 64)                0
_________________________________________________________________
dense (Dense)                (None, 2)                 130
=================================================================
Total params: 21,612,162
Trainable params: 21,612,162
Non-trainable params: 0
_________________________________________________________________
```

In [34]:
```
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy']
             )
```

In [35]:
```
es= EarlyStopping(monitor='val_accuracy',
                  patience=2
                 )

checkpoints=ModelCheckpoint(filepath='./',
                            monitor="val_accuracy",
                            verbose=0,
                            save_best_only=True
                           )

callbacks=[es,checkpoints]
```

In [36]:
```
history=model.fit(X_train,
                  Y_train,
                  validation_data=(X_val,Y_val),
                  epochs=5,
                  callbacks=callbacks)
```

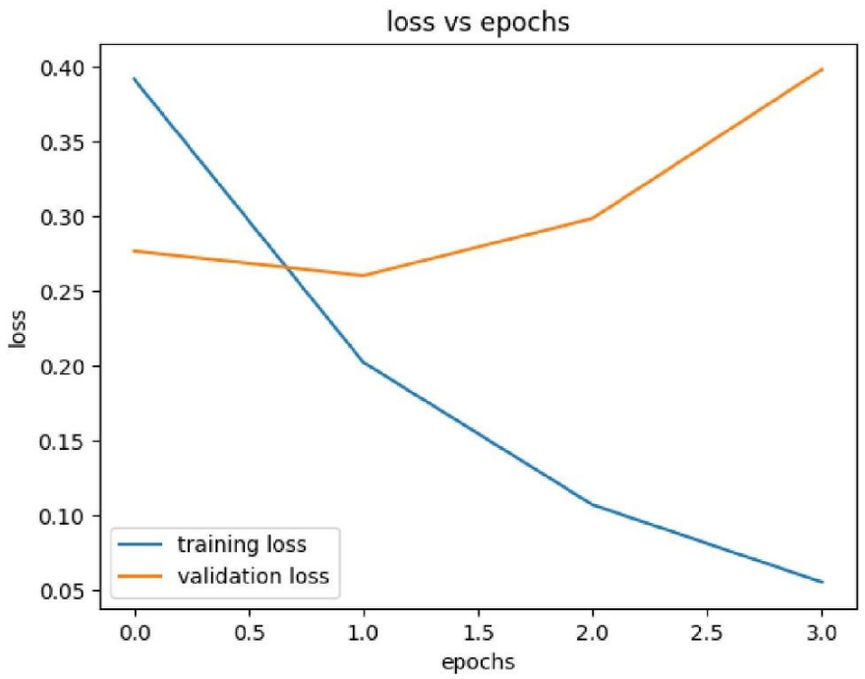
```
Epoch 1/5
1336/1336 [==============================] - 865s 636ms/step - loss: 0.3920 - accuracy: 0.8237 - val_loss: 0.277
0 - val_accuracy: 0.8902
WARNING:absl:Found untraced functions such as lstm_cell_3_layer_call_fn, lstm_cell_3_layer_call_and_return_condi
tional_losses, lstm_cell_1_layer_call_fn, lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_2_laye
r_call_fn while saving (showing 5 of 15). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: .\assets
INFO:tensorflow:Assets written to: .\assets
Epoch 2/5
1336/1336 [==============================] - 856s 641ms/step - loss: 0.2026 - accuracy: 0.9250 - val_loss: 0.260
5 - val_accuracy: 0.8924
WARNING:absl:Found untraced functions such as lstm_cell_3_layer_call_fn, lstm_cell_3_layer_call_and_return_condi
tional_losses, lstm_cell_1_layer_call_fn, lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_2_laye
r_call_fn while saving (showing 5 of 15). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: .\assets
INFO:tensorflow:Assets written to: .\assets
Epoch 3/5
1336/1336 [==============================] - 838s 627ms/step - loss: 0.1073 - accuracy: 0.9631 - val_loss: 0.298
8 - val_accuracy: 0.8907
Epoch 4/5
1336/1336 [==============================] - 829s 620ms/step - loss: 0.0556 - accuracy: 0.9818 - val_loss: 0.398
4 - val_accuracy: 0.8782
```

In [38]:
```
model.save('trained_model/Sentiment_Analysis/imdb_model.h5')
```

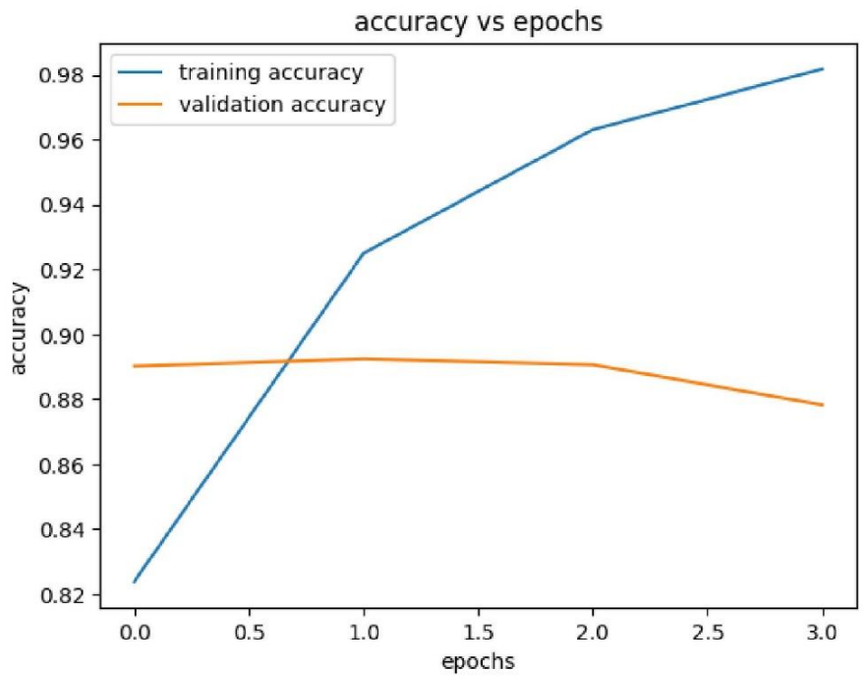
Model Evaluation

In [39]:
```
def plot_graph(history,string):

    plt.plot(history.history[string],label='training '+string)
    plt.plot(history.history['val_'+string],label='validation '+string)
    plt.legend()
    plt.xlabel('epochs')
    plt.ylabel(string)
    plt.title(string+' vs epochs')
    plt.show()
```

In [40]:
```
plot_graph(history,'loss')
```



In [41]:
```
plot_graph(history,'accuracy')
```



In [42]:
```
print("Model Performance on test set")
result = model.evaluate(test_padded,Y_test)
print(dict(zip(model.metrics_names, result)))
```

```
Model Performance on test set
157/157 [==============================] - 30s 190ms/step - loss: 0.3693 - accuracy: 0.8862
{'loss': 0.3692818582057953, 'accuracy': 0.8862000107765198}
```

Conclusion:

Social media sentiment analysis is the process of retrieving information about a consumer's perception of a product, service, or brand. Here content (text)based social media model – sentiment analysis was done on movie review data from IMDB. Here, the model developed helped in understanding the audience. Content based model also helps in gathering actionable data and get meaningful insights about brand messaging. Methods like Sentiment analysis helps brands in stepping up their customer service, helps in conducting comprehensive competitive analysis and monitor long-term brand health.

R1 (3 Marks)	R2 (3 Marks)	R3 (3 Marks)	R4 (3 Marks)	R5 (3 Marks)	Total (15 Marks)	Signature

EXPERIMENT NO 6

Date of performance:

Date of submission:

Aim: Develop Structure based social media analytics model for any business.(e.g. Structure Based Models - community detection, influence analysis)

Theory:

Network X is python package for the creation, manipulation and study of the structures, dynamics and functions of complex networks.

It is used to study large complex networks represented in the form of graph with nodes and edges. Using network X we can load and share complex networks. We can generate many types of random and classic network, analyze network structure, build network models, design new network algorithm and draw networks.

We can add notes using

1) G add-node (1) - Addressing one known at a time
2) G add node from (2,3) Adding list of nodes

We can also add edges to the graph using: G add-edge ( ) OR G add – (edge- frame)

We can then visualize the graph using network (the package) X draw (G) (The graph created)

There are also various graph generators which are inbuilt in network X. Some of these are:

1) Complex graph
2) Multipartite graph
3) Barbell graphs
4) Complete multi tight graph etc.

Node analysis:

After visualizing our network clearly it may be of interest so as to characterize the notes. There are multiple matrix that describes the characteristics of the nodes. Some of their includes

1. Degree: Most basic matric for node (No. of incoming or outgoing (or both) relationships from a nodes)
   Network X. degree (G)
   Network X. degree_ centrality (G)
2. Eigenvector Centrality: Influence / importance of a node in the network.
   Network X. eigenvector _ centrality (G)
3. Betweenness Centrality: No. of times a node appears in the shortest path between other nodes.
   Network X. bewteenness _ centrality (G)

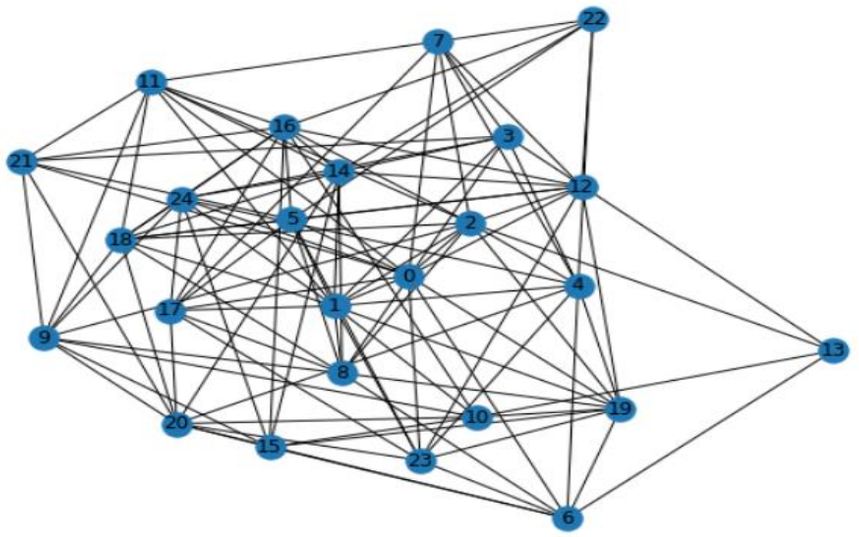
➔ **Community Detection:**
Network also provides community detection (group of nodes that are highly connected to each other but minimally connected with nodes outsides their community algorithms. One of the most popular ones is label propagation (In this, each node starts with a unique label, in a community of one. The labels of the nodes are iteratively updated according to the majority of the labels of the neighboring nodes).

Label_propagation _ communities (G)
Network library has many other use cases like finding shortest path between nodes, clustering etc.

Code and Output:

```
[ ]  1 import networkx as nx
     2 import matplotlib.pyplot as plt
     3 import pandas as pd
     4 pd.options.display.max_rows = 10
```

```
1 G = nx.gnp_random_graph(25, 0.4)
2 plt.figure(figsize=(8, 6))
3 nx.draw(G, with_labels=1)
```



Degree

```
[ ]  1 nx.degree(G)
```

```
DegreeView({0: 12, 1: 14, 2: 12, 3: 8, 4: 10, 5: 10, 6: 8, 7: 8, 8: 10, 9: 9, 10: 7, 11: 9, 12: 10, 13: 4, 14: 11, 15: 8, 16: 12, 17: 9, 18: 11, 19: 10, 20: 10, 21: 7, 22: 6, 23: 9, 24: 10})
```

Degree Centrality

```
[ ]  1 dc_df = pd.DataFrame.from_dict(nx.degree_centrality(G), orient="index", columns=["Degree Centrality"])
     2 dc_df.sort_values(by="Degree Centrality", ascending=False)
```

	Degree Centrality
1	0.583333
0	0.500000
2	0.500000
16	0.500000
18	0.458333
...	...
3	0.333333
10	0.291667
21	0.291667
22	0.250000
13	0.166667

```
[ ]  1 e_df.iloc[e_df.idxmax()]
```

	Eigen Vector Centrality
1	0.296762

Communities

```
[ ]  1 from networkx.algorithms import community
     2
     3 communities = community.k_clique_communities(G, 4)
     4 print("Communities:")
     5 for _ in communities:
     6   print(list(_))
```

```
Communities:
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24]
[9, 10, 20, 15]
```

Conclusion:

In the above experiment, we have looked at the working of NetworkX library and how it can be used for Social Network Analysis. Once we represent data as a graph using the NetworkX library in Python, a few short lines of code can be illuminating. We can visualize our dataset, measure and compare node characteristics, and cluster nodes sensibly via community detection algorithms. In the way, we have successfully performed the required experiment.

R1 (3 Marks)	R2 (3 Marks)	R3 (3 Marks)	R4 (3 Marks)	R5 (3 Marks)	Total (15 Marks)	Signature

EXPERIMENT NO 7

Date of performance:

Date of submission:

Aim: Develop a dashboard and reporting tool based on real time social media data.

Theory:

Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. The data might be an Excel spreadsheet, or a collection of cloud-based and on-premises hybrid data warehouses. Power BI lets one easily connect data sources, visualize and discover what's important, and share that with anyone.

Social Media Monitoring & Analytics with Power BI for Digital Marketing

Power BI can also be used to create dashboards for social media monitoring and analytics. For example, the social media monitoring dashboard from Cheetos has helped the brand to understand sales, product's quality, crisis control and understand the audience opinion about the product.

Power BI for Facebook data:

- Power Bi has a connector for Facebook which is limited to public facing data and this was used on the overview page, simply login with your details and point to any website.
- To get the internal data you need to do a data dump from Facebook business > page insights > export data. Save all Facebook exports in the same folder and load as 'folder' in Power BI. Each week do an export and save it in the same folder and Power BI will include all files in the folder in its next refresh.

Power BI for Twitter data:

- Unfortunately, the Twitter API is not built into Power BI, the free version is very limited but like Facebook we can do a data dump by going to Twitter Analytics > Tweet > Export Data

Power BI for Google Analytics:

- Power BI has a connector for Google Analytics, once you have logged in, the trick is in understanding which fields you need to use.

Power BI DAX:

DAX (Data Analysis Expressions) is a formula expression language and can be used in different BI and visualization tools. DAX is also known as function language, where the full code is kept inside a function. DAX programming formula contains two data types: Numeric and Other. Numeric includes - integers, currency and decimals, while Other includes: string and binary object.

6 key metric DAX formula examples:

Following are the examples of six DAX expressions to create a comprehensive set of metrics to allow you to develop dashboard using LinkedIn data.

Totals

```
Total LinkedIn New Followers =
SUM(LinkedIn_Data[LinkedIn New Followers])
```

You can use SUMX in lieu of SUM if you wish here, noting you will need to provide a table and expression in lieu of a column.

Year to Date (YTD)

```
LinkedIn New Followers YTD =
TOTALYTD(
   [Total LinkedIn New Followers],
   LinkedIn_Data[Date]
)
```

Latest Month (MTD)

```
LinkedIn New Followers This Month =
Calculate(
   [Total LinkedIn New Followers],
   LASTDATE(LinkedIn_Data[Date])
)
```

Previous Month

```
LinkedIn Followers Previous Month =
Calculate(
   SUM(LinkedIn_Data[LinkedIn New Followers])
   ,PREVIOUSMONTH(LinkedIn_Data[Date]
   )
)
```

Month Over Month Difference

```

LinkedIn Followers Diff MoM =
VAR CurrentS = Sum(LinkedIn_Data[LinkedIn New Followers])
VAR PreviousS = [LinkedIn Followers Previous Month]
VAR Result = CurrentS - PreviousS
Return
   Result
```

Month Over Month % Growth

```
LinkedIn Followers MoM Growth % =
Divide(
   [LinkedIn Followers Diff MoM],
   [LinkedIn Followers Previous Month]
)
```

Social Media Dashboard:

Audience of the dashboard:

The social media dashboard is for social media managers and is used to help them understand the high-level impact each of their social networks has on their audience.

Purpose of the dashboard:

The purpose of the report is to provide an overview of how the social media teams efforts are impacting the number of impressions over time. Additionally, the dashboard is designed to help identify the best performing social networks. In return, this helps the team better understand where to focus their marketing efforts.

Dashboards key insights are:

1. Ribbon chart - Impressions by month, split by platform:

   To help prioritize which platforms to focus on throughout the year.

2. Table - Impressions by content type:

   To determine which content types perform the best.

3. Stacked Column Chart - Impressions by quarter, split by platform.:

   To provide a high level view of impressions across the year to help resource allocation.

4. Bar Chart - Impressions by platform:

   To determine the highest performing platform and prioritise accordingly (supports the below).

5. Donut Chart - Impressions % by platform:

   To determine the split of impressions and prioritise accordingly (supports the above).

Output:

Conclusion:

A social media analytics dashboard is the place where you can track, measure, and analyze the performance of your social media platforms. Some of the benefits of social media dashboard for brands are gaining insights with a single glance, sharing performance with media team, increasing conversion rate and generating revenue. It also aids in performance tracking, content creation, sharing effective reports with stakeholders, competitive analysis, improved collaboration and formulating better social media marketing strategies. Hence, a social media dashboard was successfully created using Power BI.

R1 (3 Marks)	R2 (3 Marks)	R3 (3 Marks)	R4 (3 Marks)	R5 (3 Marks)	Total (15 Marks)	Signature

EXPERIMENT NO 8

Date of performance:

Date of submission:

Aim: Design the creative content for promotion of your business on social media platform

Theory:

Content is being created, published, and refreshed every second of every day. In 2019, more than four million blog posts were published on a daily basis — not to mention the abundance of ebooks, templates, podcast episodes, infographics, and videos that go live every minute, as well.

Social Media Content Promotion Ideas

Social media is an ideal place to promote your content, as those following your accounts already know your brand and likely look forward to the content that you post. Here are nine ideas to help you promote content on social media.

1. Update your cover photo to promote the content. If using Facebook, link to the offer in the photo's description.

2. Pin a post to the top of your page linking to the landing page of your content.

3. Add a link to the offer in your business page's bio.

4. Post several times a week to the piece of content, varying the images and copy you use in each post, for several weeks (or months).

5. Create a hashtag for the campaign or the offer and utilize it in your posts.

Join a group on Facebook or LinkedIn and start a discussion.

6. If you have a group of your own, send out the content to all group members.

Promote the offer in your social media stories.

7. Make the content offer appear on your Facebook Messenger when visitors come to your page.

Blogging Promotion Ideas

Blogging can drive traffic to your landing pages and website better than any other medium. Each time you blog, you give Google and other search engines one more opportunity to find you. Each blog post gives you the opportunity to rank for more and more keywords and grow your reach.

Here are seven tips for promoting your blog posts.

1. Blog about the piece of content multiple times, targeting a new angle or keyword each time to avoid redundancy. Spread out posts over a few weeks or months to continue a steady stream of traffic to the landing page.

2. Create a CTA at the bottom of each relevant blog post that links to the offer.

3. Create anchor text CTA towards the top of the page, using hyperlinked words to point to the offer. HubSpot's blogging team has found this to be one of the most successful CTA placements.

4. Create a trigger-based CTA to appear or slide in at a certain point. For example, the CTA could appear when a reader has been on the page for 15 seconds, has scrolled through 20% of the page, or is about to exit the page.

5. Make existing CTAs smart so that users who have already downloaded content from your site see your new offer instead.

6. Encourage social sharing of blog posts with built-in social share buttons and tweetable quotes.

7. Guest blog on related sites to increase awareness of the content.

Video Marketing Promotion Ideas

Video promotion typically requires more planning and effort than other promotion types, since you'll have to script, shoot, and edit with the help of a dedicated videographer.

However, if you have video experts at your disposal, it's one of the most effective methods for promoting content.

Here are three tips to ensure you're correctly promoting your videos once you've made them.

1. Make a video for YouTube, capitalizing on a term with high search volume and promoting the content as a featured resource. Don't forget to link to the content landing page in the description of your video.

2. Make a video for your social media channels with appropriate dimensions, messaging, and length.

3. Make a video for your website and content landing page describing the value of the offer.

Paid Promotion Ideas

Ideally, your content should be optimized for SEO and promoted within your existing channels. However, for content offers that aren't supported by ample search volume, or for smaller companies looking to build awareness, paid promotion can be a useful way to get your content in front of a larger audience.

Luckily, there are plenty of sites with billions of users that could be perfect for your content promotion.

Here are a few examples of the ways you can advertise on them:

1. Set up Google Ads campaign for search, with multiple ad groups targeting different keywords and angles.

2. Use display ads to show off the visuals of your offer with Google Display Network.

3. Set up promoted tweets for your Twitter account and collab with designers for visual assets.

4. Use LinkedIn's Sponsored Updates feature to promote the Company Page update related to the content and collab with designers for visual assets.

5. Create YouTube pre-roll ads with the help of your video team or a video agency.

6. Boost the engagement of an organic Facebook post by putting some money behind it.

7. Create a Facebook ad or a dedicated group of Facebook ads.

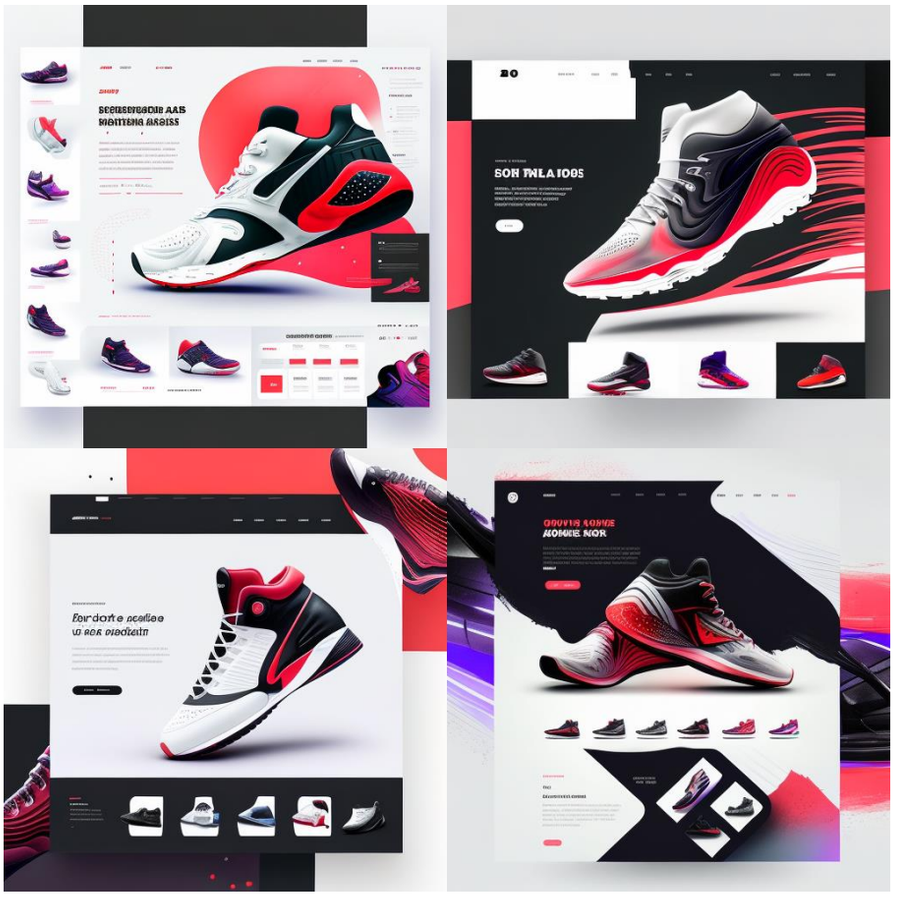
8. Get even more visual with Instagram ads. You can also try GIF and video ads here rather than just images.

Output:

Creative content for shoe brand:

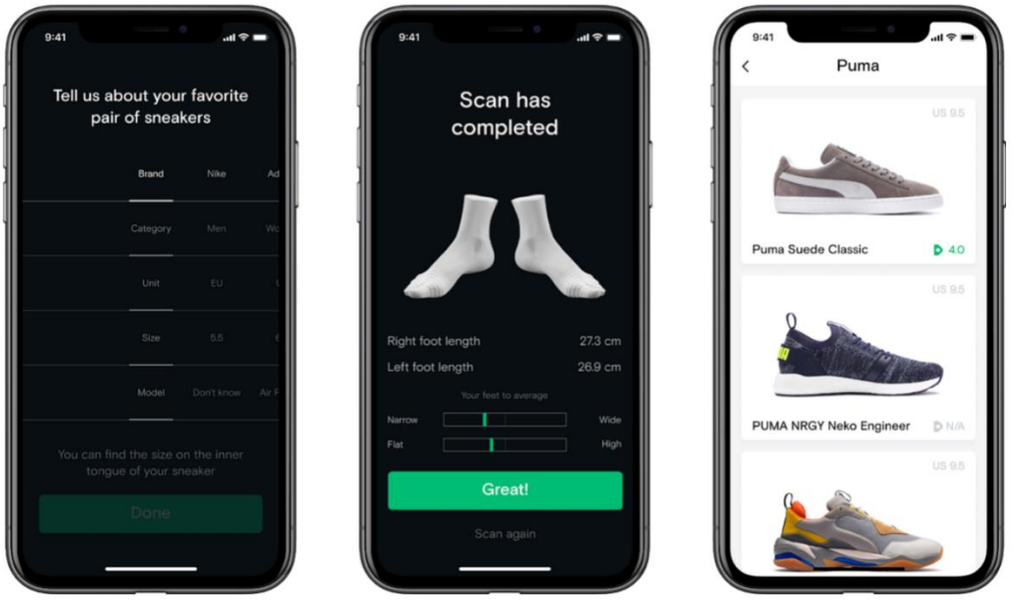
1.Minimalsitic advertisement where focus is on the product with only necessary information.

2. Image with elegant color scheme to potray various colour and model of shoes available.



FINALLY! Real History To Be Taught In Indian Schools By 2025

3. Prototype making product more accessible and user friendly while prioritizing user's need be it with respect to brand, size, category, model etc



Conclusion:

Content plays a crucial role in achieving business and social media objectives such as brand recognition, thought leadership, audience engagement, and lead generation. It allows a business to cement their position as a go-to destination for its customers, whether it be for entertainment, education, or inspiration. Effective content will help to build long term relationships with the audience, ultimately leading to an increase in revenue for the organization. Hence, creative content for promotion of business (shoe brand) on social media platform was made successfully.

R1 (3 Marks)	R2 (3 Marks)	R3 (3 Marks)	R4 (3 Marks)	R5 (3 Marks)	Total (15 Marks)	Signature

EXPERIMENT NO 9

Date of performance:

Date of submission:

Aim: Analyze competitor activities using social media data.

Theory:

Competitive analysis is more than just comparing yourself to others. And it's definitely not about copying the competition. By analyzing the competition with a social media competitive analysis, brands can uncover valuable insights that empower brands to improve your social strategy, and your entire business.

Here, we will be using Similarweb's powerful **competitive analysis** tools, metrics, and insights to find your top digital competitors and analyze traffic performance and strategies of your competitor set.

Within Similarweb's **Competitive Analysis** module, brands can:

- find who they are competing with for valuable digital traffic
- see who's leading in traffic across marketing channels, in organic and paid search, in different geographical regions, and more
- discover what's working to drive traffic and find growth opportunities

.. and then optimize strategy accordingly and win more traffic!

How to perform a social media competitive analysis:

Step 1: Identify your competitors

Before you can win in the digital world you need to know who you are competing with. And, your digital traffic competitors may look a little different than your well-known business competitors -- because in the digital world, a competitor is *any* site that pulls clicks away from your site.

So, let's start by identifying competitors:

1. Go to **Competitive Analysis** and enter your website in the search bar. Click Enter.

The **Website Performance** page for your site is presented. Here you'll get a birds-eye-view of many traffic and engagement metrics and insights for your site.

2. Next, use the sidebar to navigate to the **Organic Competitors** page.

   The **Organic Competitors page** reveals a list of sites competing for the same website traffic as the analyzed site. Take note of the sites that top the list.

Once you've taken note of the top sites competing for *organic traffic*, navigate to the **Paid Competitors page** to see who tops that list.

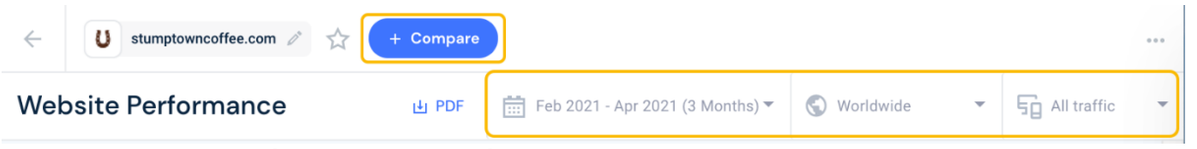
With the list of competitors in hand, let's compare the sites to see who is earning the most volume and quality traffic, and how the traffic breaks down across various marketing channels.

Step 2: Benchmark your performance

Similarweb's expansive and reliable data helps reveal how your website traffic stacks up against the competition. Compare the traffic volume and (more importantly) traffic quality, and see how that traffic is distributed across marketing channels.

Website Performance

1. Go to **Competitive Analysis** and enter your website in the search bar.
2. Click **+Compare** and add up to 4 competitor websites.



3. Use the filters at the top of the page to set desired date range, geographical region, and traffic type (Desktop, Mobile, or both).

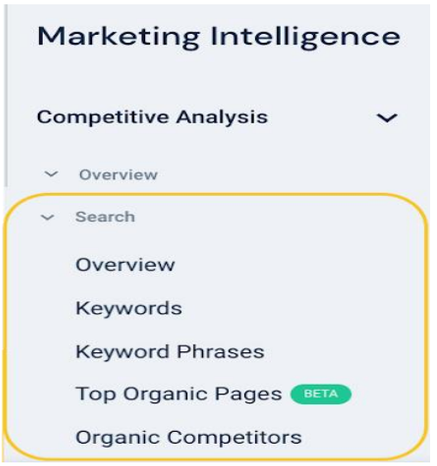
Step 3: Discover and Optimize

Now you know who is getting traffic and which channels that traffic is coming from. It's time to take a deep dive into each channel to see what's driving traffic and how you can use these insights to build your winning marketing strategy.

Let's look at the traffic insights for each of the Marketing Channels driving traffic to the competitive set of websites.

Organic Search and Keywords

Which keywords are driving the most organic search traffic (not just the search volume) to your competitors? What keyword questions are hot, new, or trending? How can I use these insights to increase my traffic?



Winning in organic search is the foundation of a good SEO campaign. When done right, leveraging organic traffic insights will help rank your pages for more keywords within search engines and consequently get you more traffic.

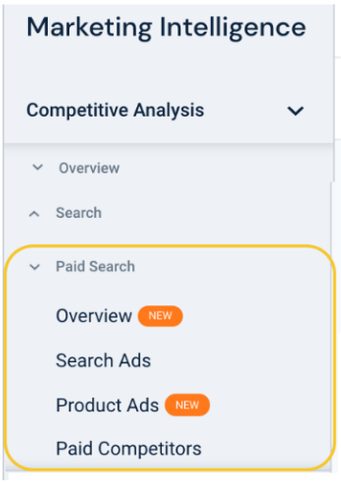
Select to the **Search -> Overview**.

Key insights and using the data to discover opportunities and optimize your search strategies::

- **Search Overview**: Discover the search traffic metrics of your competitors.
- **Performance**: Evaluate the distribution of the different types of search traffic among your competitors. Look at Paid v Unpaid, Branded v. Non-Branded, and different Search Types (regular, video, image, shopper, etc.)
- **Discover** New Keywords and Keyword Phrase opportunities.
- **Search Engines**: See what search engines are driving traffic to your competitors

Paid Search

The **Paid Search Overview page** provides key metrics and insights needed to analyze your competitor's paid search strategy.



See the distribution of traffic sent from paid search ads to each site in the competitor set. Competitors who receive large amounts of traffic from Paid Search spend on ad budgets to increase brand awareness and target relevant audiences.

Paid Search campaigns are targeted to high intent users and can result in higher conversion rates.

Key insights and using the data to discover opportunities and optimize your PPC strategies:

- Evaluate the list of top-paid keywords, search ads, and product ads
- Gain visibility into a competitor's product focus and competitive messaging. See what is working for your competitors.
- Uncover opportunities to optimize your campaign spending and KPI's (ad impressions, clicks, conversions, etc.) to target high intent users

Social

Discover insights on traffic sent from social media sites such as Facebook or Reddit (organic and paid). Including direct media buying from Facebook.

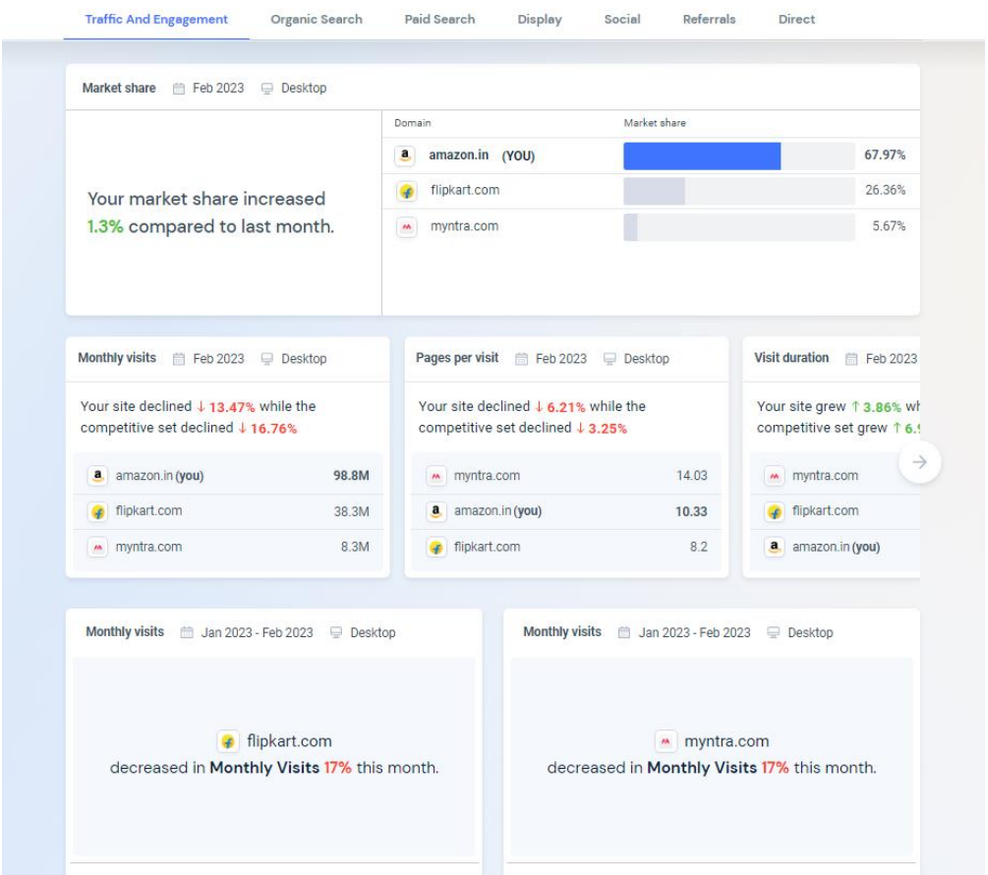
Key insights and using the data to discover opportunities and optimize traffic from Social media sites.

- Visits from Social are considered to be easily influenced (as a result of a viral article, meme, image, etc.). Thus, a website that generates high and consistent traffic from Social is likely to have a loyal community of users. View the list of social domains to understand the websites' core communities.
- Additionally, social networks tend to attract specific audience types. For example, Pinterest is known to attract an audience that skews more females with an interest in crafts. Social traffic referrals are yet another factor in understanding a particular competitor's audience profile.

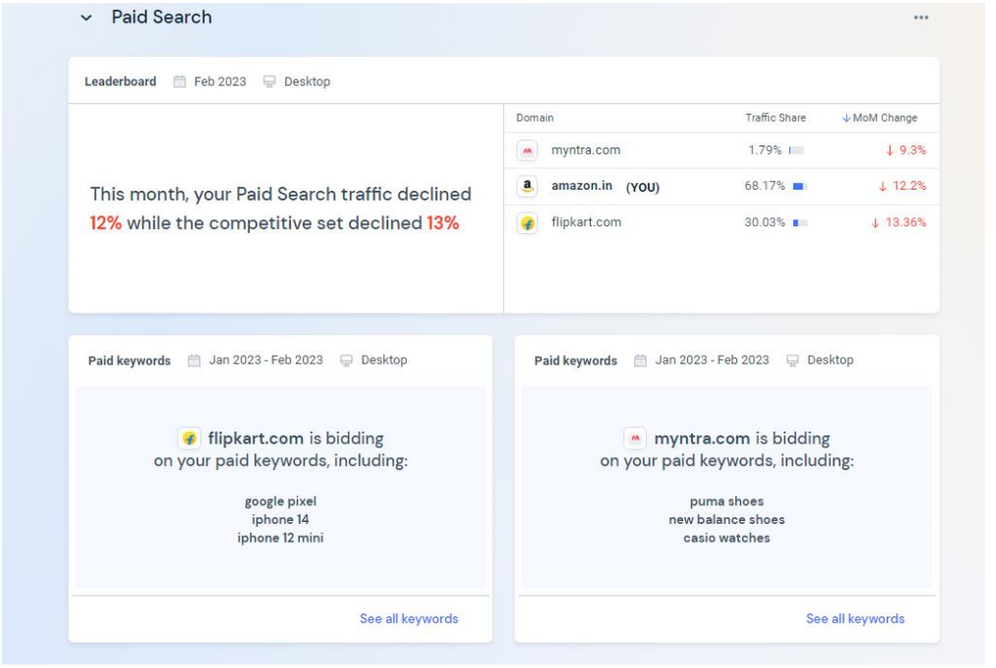
Output:

For competitive analysis, Amazon, Flipkart and Myntra are considered

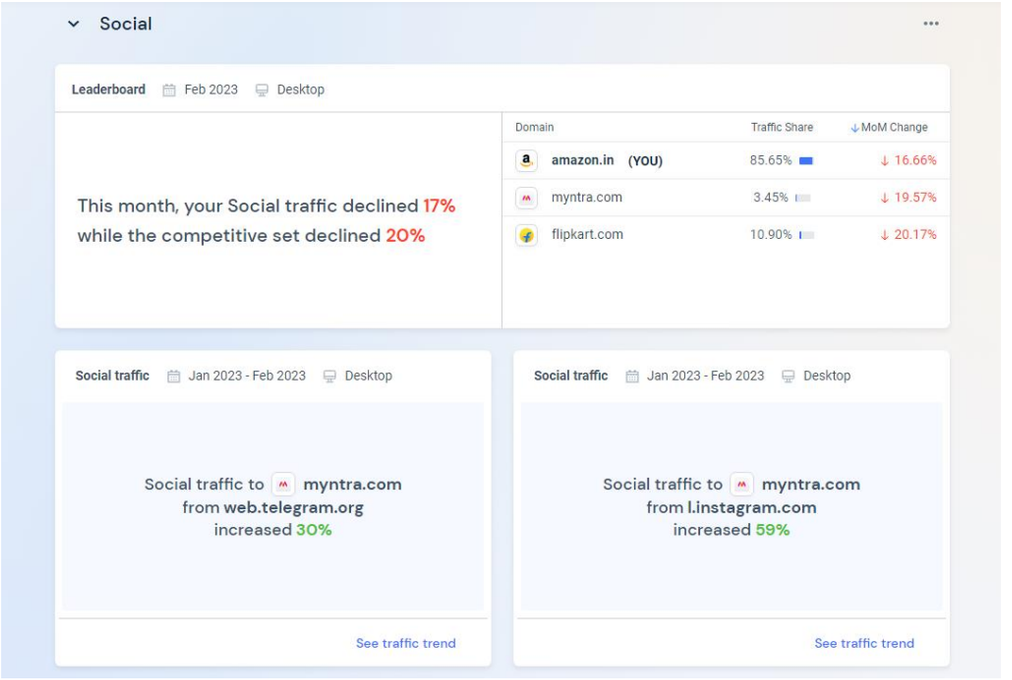
Traffic and Engagement comparison



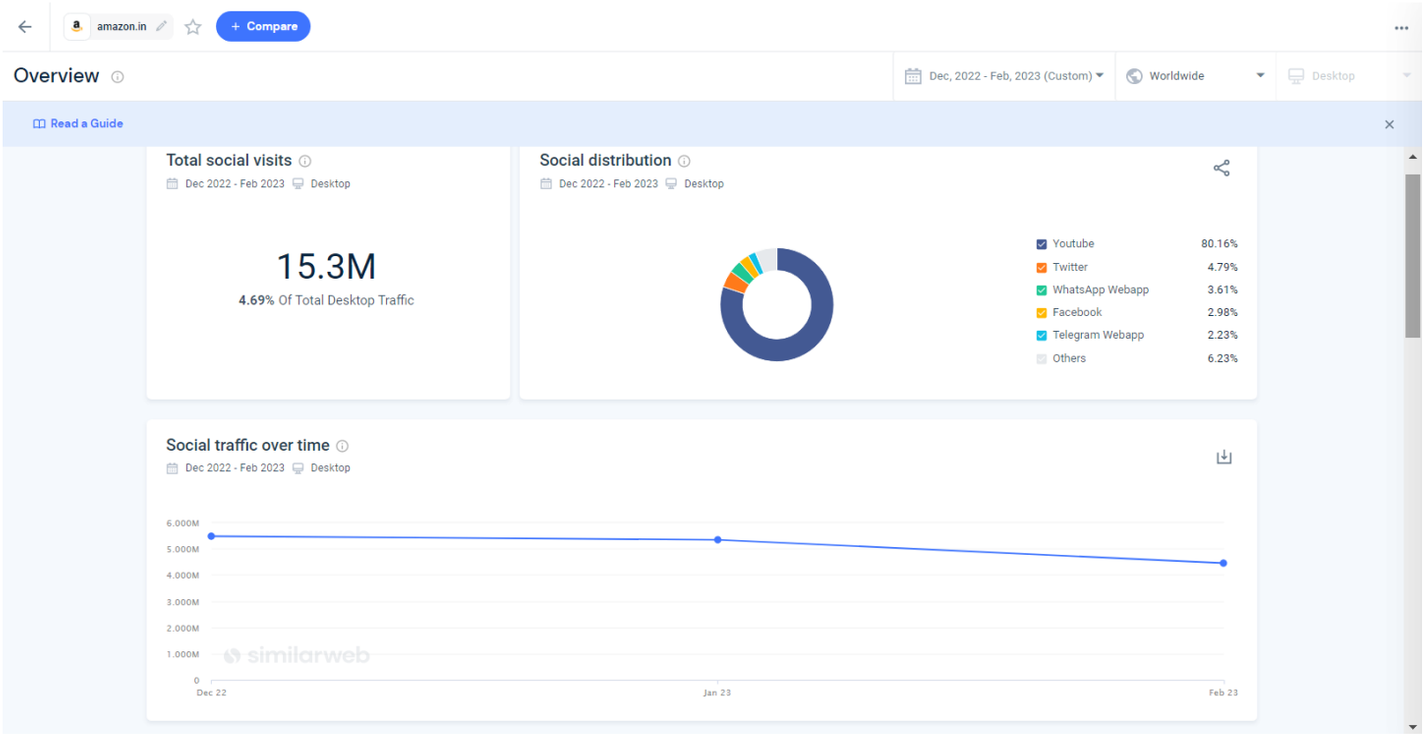
Paid Search



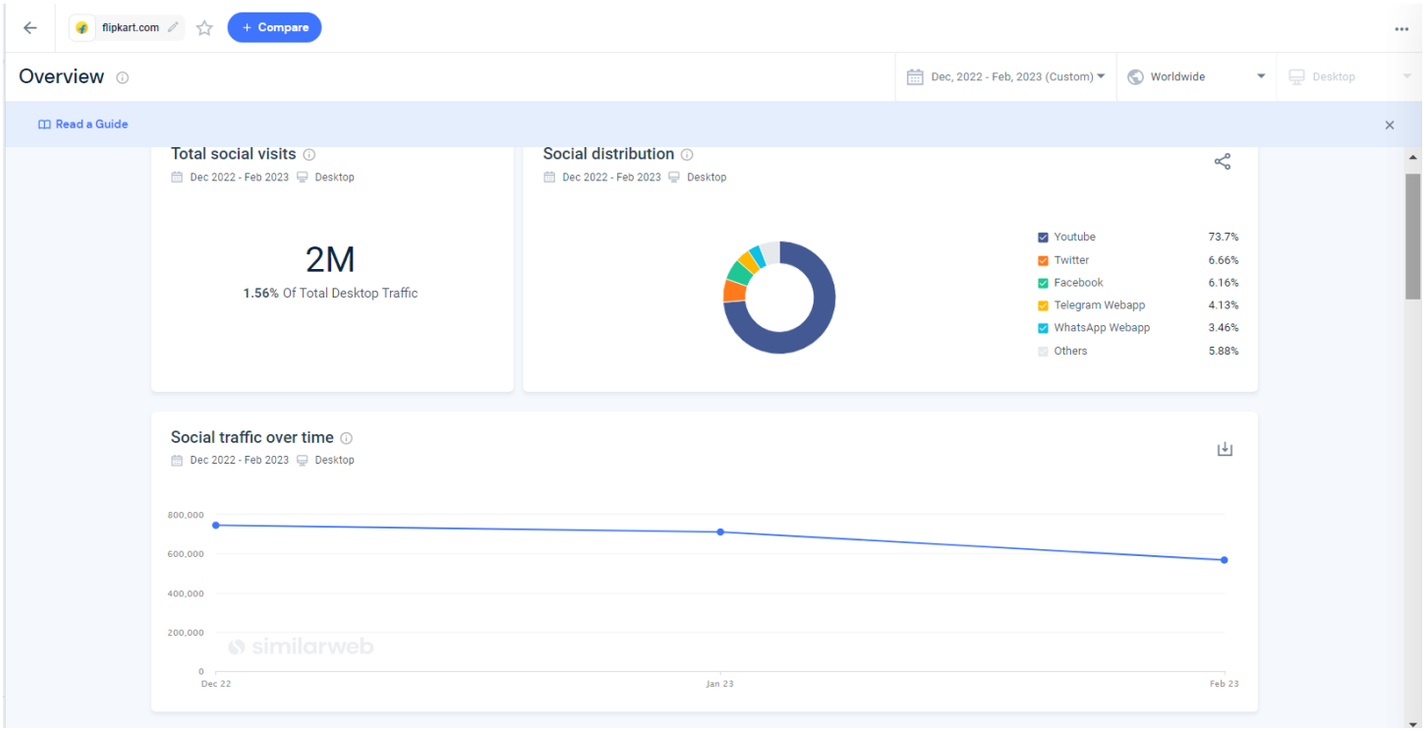
Social



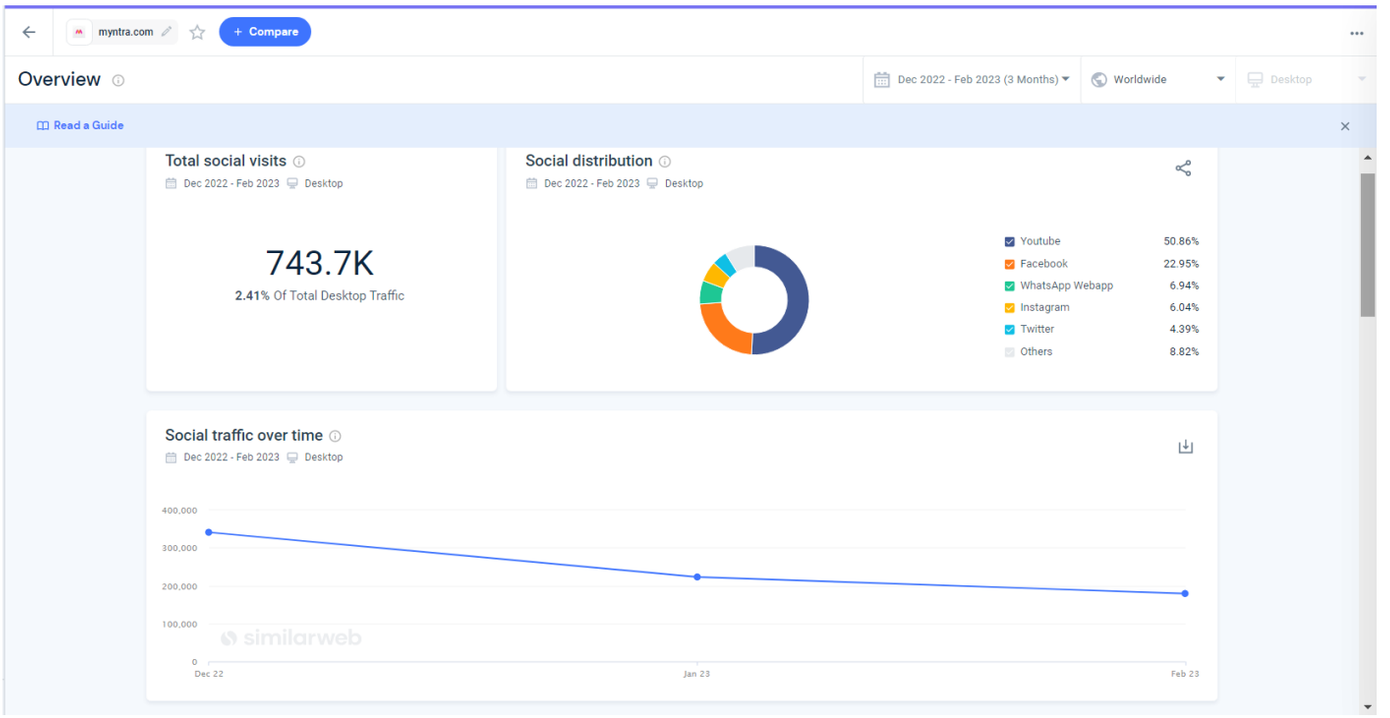
Amazon's social metrics:



Flipkart's social metrics:



Mytnra's social metrics:



Conclusion:

A social media competitor analysis is an analysis of your competition on *social* media to find out what their strengths and weaknesses are, and how those strengths and weaknesses compare to your own brand. Social media competitive analysis can give brands insights like performance benchmarks, ideas for best times to post on social media, understanding potential customer pain points, new and better ideas for content, ideas for ways to differentiate your brand. Hence, competitive analysis using social media data was performed using Similar web platform.

R1 **(3 Marks)**	**R2** **(3 Marks)**	**R3** **(3 Marks)**	**R4** **(3 Marks)**	**R5** **(3 Marks)**	**Total** **(15 Marks)**	**Signature**

EXPERIMENT NO 10

Date of performance:

Date of submission:

Aim: Develop social media text analytics models for improving existing product/ service by analyzing customers reviews/comments.

Theory:

In a globally networked world, where the internet is a boon to the immediate feedback of sentiments that are based on emotions. People review the services and products being offered to them on various websites. These websites not only help customers to share their reviews but also allows other.
Customers and business owners to improve the range of their services. Users with different backgrounds furnish their reviews in different languages and scripts which are gold for the opinion miners.

Customer satisfaction is the key in assessing how a product or service of a company meets customer expectations and is an important tool that can give organisations major insights into every part of their business, thus helping them to increase earnings or minimise marketing expenses. Customer feedback might help in reviewing the factors that were not previously considered, such as shipping, safe packing, politeness and available customer service consultants and a user-friendly website. Nothing can make customers feel that they are important than asking for their views and valuing their comments. When a customer is asked for any opinion on a product or experience, they feel valued and connected to the organisation.

In the food industry, customers often look into restaurant reviews before placing their orders. Nowadays, restaurants or food delivery services (FDSs) have a review or feedback system that is integrated in their portal or social media platforms; however, only a few act on customer opinions due to the presence of a large amount of review data across various platforms and the lack of customer service consultants that will go through each of these comments and act on them.

Here, restaurant review data is taken from Kaggle, specific restaurants in Bangalore, and will be analyzed using the Naïve Bayes in Python Scikit-Learn and analyze it with accuracy and precision-recall. There are several steps:

Data Collection:
Data that we got from Kaggle is a platform for predictive modeling and analytics competitions in which companies and researchers post data and statisticians and data miners compete to produce the best models for predicting and describing the data. Data that we collect specific is about reviews on Zomato Bangalore and collect 1000 reviews to be analyzed

Workflow Process:
The study was conducted and processed in Python 3.6 and with the Scikit-Learn library using the Naïve Bayes method to implement the model. The following figure is a block diagram of the stages of research. Figure 1. Workflow Process From data review we are pre-processing text then if the data clean, we split it to 80% training data and 20% test data. Then the 80% data training data we train using Naïve Bayes. After the machine model finish trained, we are testing it to data testing and evaluate the accuracy and precision-recall to see how the metrics of our machine model.

Text Preprocessing:
Text preprocessing is the first stage of text mining. The purpose of text preprocessing is to prepare unstructured text documents into structured data that is ready to be used for processes then by eliminating noise, homogenize word forms and reduce word volume (Putraranti & Jonathan, Sihotang & Martin, Sentiment Analysis of… 1837 Winarko, 2014). Stages of preprocessing text used in this study are lowercase, tokenization, remove punctuation, stopwords removal, pos tags and stemming.

Bag of Words Model:
The bag-of-words model is a way of representing text data when modeling text with machine learning algorithms.
The bag-of-words model is simple to understand and implement and has seen great success in problems such as language modeling and document classification problem with modeling text is that it is messy, and techniques like machine learning algorithms prefer well defined fixed-length inputs and outputs.
Machine learning algorithms cannot work with raw text directly; the text must be converted into numbers. Specifically, vectors of numbers.
A popular and simple mehod of feature extraction with text data is called the bag-of-words model of text.
A bag-of-words model, or BoW for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms.
The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents.
A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:
1. A vocabulary of known words.
2. A measure of the presence of known words.
It is called a “bag” of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

Limitations of Bag-of-Words:
The bag-of-words model is very simple to understand and implement and offers a lot of flexibility for customization on your specific text data.
It has been used with great success on prediction problems like language modeling and documentation classification.
Nevertheless, it suffers from some shortcomings, such as:
Vocabulary: The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations.
Sparsity: Sparse representations are harder to model both for computational reasons (space and time complexity) and also for information reasons, where the challenge is for the models to harness so little information in such a large representational space.
Meaning: Discarding word order ignores the context, and in turn meaning of words in the document (semantics). Context and meaning can offer a lot to the model, that if modeled could tell the difference between the same words differently arranged (“this is interesting” vs “is this interesting”), synonyms (“old bike” vs “used bike”), and much more.

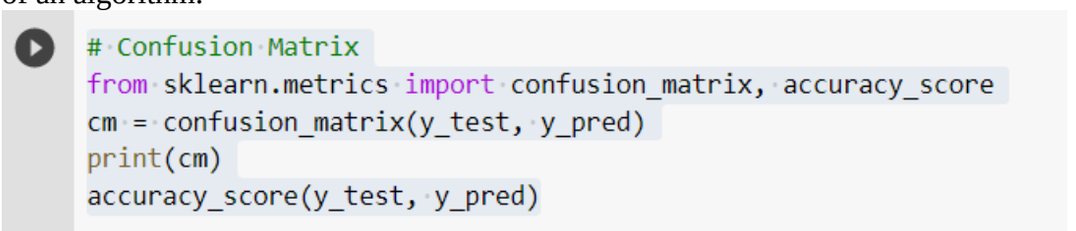
Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as ply
dataset = pd.read_csv('Zomato Review Analysis.csv', encoding =
'unicode_escape')
dataset
import re
```

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
corpus = []
for i in range(1000):
zomato_ratings = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
zomato_ratings = zomato_ratings.lower()
zomato_ratings = zomato_ratings.split()
all_stopwords = stopwords.words('english')
all_stopwords.remove('not')
ps = PorterStemmer()
zomato_ratings = [ps.stem(word) for word in zomato_ratings if not word
in set(all_stopwords)]
zomato_ratings = ' '.join(zomato_ratings)
corpus.append(zomato_ratings)
print(corpus)
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1600)
x = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, -1].values
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
0.20, random_state = 51)
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)
print(np.concatenate((y_pred.reshape(len(y_pred),                    1),
y_test.reshape(len(y_test), 1)), 1))
# Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

Output:

Confusion Matrix is a metric that shows the true and prediction errors of data from the results of an algorithm.




```
[[57 44]
 [14 85]]
0.71
```

```
[21] y_pred = classifier.predict(x_test)
     print(np.concatenate((y_pred.reshape(len(y_pred), 1), y_test.reshape(len(y_test), 1)), 1))
```

```
[0 1]
[0 0]
[1 1]
[0 0]
[1 1]
[1 1]
[1 1]
[1 1]
[1 1]
[0 0]
[1 0]
[0 0]
[0 0]
[1 1]
[1 1]
[1 1]
[0 0]
[1 1]
[1 1]
[1 0]
```

Conclusion:

Customer review analysis helps brands understand customer and product experience feedback from multiple channels to uncover intelligent business insights. Customer review analysis helps in product innovation**,** effective marketing, track customer motivations, improved customer service**,** wholesome brand experience, product-fit & market gap analysis**,** competitor analysis**,** discover market trends, measure business stability and build customer loyalty. Hence, we used Naïve Bayes algorithm to classify user's sentiment of restaurants on Zomato (Bangalore region). We also found words that affects the classifier model using Bag of words model.

R1 **(3 Marks)**	**R2** **(3 Marks)**	**R3** **(3 Marks)**	**R4** **(3 Marks)**	**R5** **(3 Marks)**	**Total** **(15 Marks)**	**Signature**