SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Opportunities for Design Improvements for Nothing Phone 1 using Artificial Intelligence

AUTHORS	STUDENT ID
He Tianli	1006151
Lim Yu Jie	1005999
Aditya Kumar	1006300
Ethan Choo E-Rhen	1006324
Caitlin Daphne Tan Chiang	1006537
Chin Wei Ming	1006264

All team members contributed equally to the task.

Singapore University of Technology and Design

60.002: AI Applications in Design

Dr. Kwan Wei Lek, Dr Edwin Koh, Mr Michael Alexander Reeves

April 16, 2023

1.Introduction

In recent times, technology has been advancing at a rapid rate and in recent times,there has been a sudden boom in interest towards Artificial Intelligence and Machine Learning. This is not unexpected given how society is progressing towards digitization where more and more people are becoming glued to their smartphones, be it to browse through social media, doing online shopping etc. With this immense surge of people going online, there is also a huge amount of data being uploaded every single second, and these data are the honeypots of feedback and consumer taste and preferences where many companies would be interested in obtaining to develop their product.

However, the issue lies with the immense amount of data online, how are we able to discern what is relevant and irrelevant to what the companies are focusing on for design improvements. In this project we will be exploring and categorising large quantities of data extracted from online platforms through the use of Artificial Intelligence to find possible design improvements of a product.

1.1 Reason for choosing the Nothing Phone(1)

We have chosen to implement AI machine learning on the Nothing Phone (1), a phone that was announced in January in 2021 and released in July 2022. It was a highly anticipated phone that circulated a lot of discussions among the technology geek community, and the general view for this phone is that it is very controversial in the sense where it is nothing like we have seen before. People were excited for the final reveal and release of the phone back in July 2022, but just to be disappointed on how it actually came out.

The company that came out with this product is called ‘Nothing’. This company was founded on 29 October 2020, so it's still a relatively new company when compared to other phone companies. As of March 2023, the company has only released 4 products, 3 of which are wireless earbuds, and the last product being a smartphone. Being a product which does not have a subsequent model, there are bound to be issues that the smartphone faces when used by consumers in the mass. Thus, our group decided to ride on the opportunity and focus on *Nothing Phone (1)*.

One application we can implement AI and machine learning is what we will be exploring in this project where we would be utilising machine learning models to find rooms for improvements of the product Nothing Phone(1).

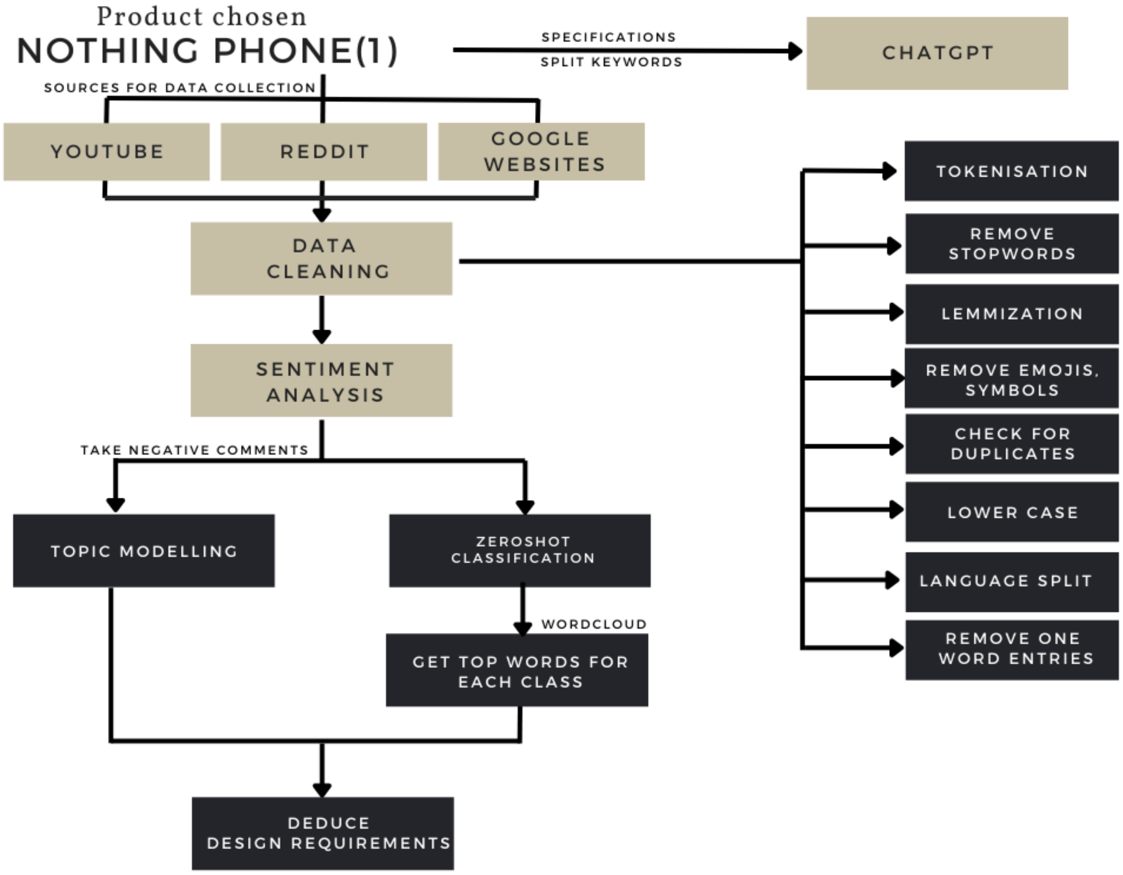
We have chosen to find design opportunities and improvements on “Nothing Phone 1” as it got a lot of traction during its pre release due to its unique design. However, after it got released there was a lot of bad feedback. Hence we think that this product has potential areas for design opportunities and improvements. To identify design opportunities and improvements, we used data-driven text analytics and classification models. Ultimately we wish to seek specific design requirements which can be used as a guideline when designing the next iteration of nothing phones.

1.2 Methodology

Firstly, we decided to get comments and reviews from various sources to get the general idea of how people view the Nothing Phone 1. We have narrowed down to 3 main data sources : Youtube, Reddit and Websites (web scraping). There are 3 main steps to go through in order to do text analysis and classification on a product.

Data extraction, data cleaning, data analysis.

1.3 Overview and Workflow of project



<u>Setting up:</u>

This project requires the use of multiple libraries, classification models as well as text models that need to be installed at the beginning of the project.

Library/Models	Description/Use
googlesearch	‘Search’:searches the term on google “build”:The googleapiclient.discovery module is used to build a service object that can interact with a Google API. The build() function is used to create this service object.
requests	Send http requests to interact with web APIs to scrape web pages
Beautifulsoup4	Converts HTML code into an object that can be used to find elements in a page
pickle	To serialize/deserialize python objects and save state of programs for sending data between processes.
os	This library provides a way to interact with the operating system in Python. It includes functions for working with files and directories, launching subprocesses, and managing environment variables.
pandas	Data structures for efficiently storing and manipulating large datasets ,and functions for cleaning, transforming and analysing data.

google_key	Use of Google API keys to authenticate and authorize access to google services, in this project, it will be used when web scraping google search results
re	Used for pattern matching and string manipulation, to extract information from text, validate user inputs etc.
Numpy	Library for working with arrays and numerical operations
praw	Reddit API to obtain comments from reddit
langdetect	Library for detecting language of a given text string,used for content filtering and language-specific text analysis
nltk	Natural Language Toolkit for text analysis
openai	To enable us to access to ChatGPT using python

Before gathering data to perform text analysis on, a variety of websites were considered for data collection, we mainly chose websites that are popular and tend to have a lot of reviewers. However during our selection phase, we realised that certain mainstream websites do not have anything to offer for the Nothing Phone(1), such as the case of Amazon, Shopee and Lazada, where searching the term does not give back the product.

Hence we decided to do data extraction on Youtube, Reddit and Websites that do have reviews and comments on the product.

YouTube Data Extraction:

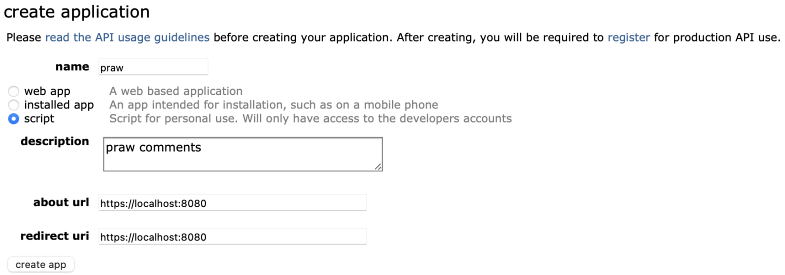
Before we are able to extract data from youtube, we have to set up the YouTube Data API. APIs are unique to every individual's google account. It is needed for us to be able to access all youtube's data for example videos, comments, titles and many more of youtube's data.

The first step of extracting youtube data is first to specify what is the search terms that you are searching for, which in our case will be the 'nothing phone (1)'. Next is to specify how many videos do you want the code to run through under the variable "max_results". Afterwhich the code will essentially iterate through each video's comments, and even furthermore each comment's replies. Which all these sentences will be appended one by one to an initially empty list.

We utilised the pandas Dataframe to store each of the comments with the list of comments formed when running the code. And saved it as a csv file in the end to avoid needing to run it again in the future as this code takes awhile to finish running.

Reddit Data Extraction:

Firstly, create a reddit account.Then go to https://www.reddit.com/prefs/apps on your web browser and click on apps>create apps
Choose the "script" option and set both urls as "https://localhost:8080", then click create app



create application

Please read the API usage guidelines before creating your application. After creating, you will be required to register for production API use.

name praw

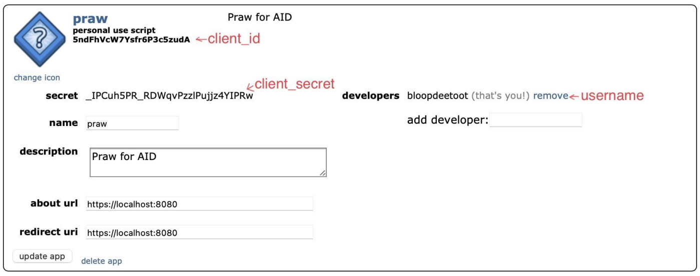
○ web app — A web based application
○ installed app — An app intended for installation, such as on a mobile phone
● script — Script for personal use. Will only have access to the developers accounts

description praw comments

about url https://localhost:8080

redirect uri https://localhost:8080

After creation we will be given the client_secrets and client_ID which we will be using later.



praw Praw for AID
personal use script
5ndFhVcW7Ysfr6P3c5zudA ←client_id

secret _IPCuh5PR_RDWqvPzzlPujjz4YIPRw ↙client_secret

developers bloopdeetoot (that's you!) remove←username

add developer:

name praw

description Praw for AID

about url https://localhost:8080

redirect uri https://localhost:8080

Next, pip install PRAW library, this is the library created specifically to web scrape reddit.

In attempts to keep data extraction to be as automated as possible, r/all which is our homepage
The function 'search_term' is the value we would need to change depending on the product you choose.

Inputs: 'reddit.subreddit('all')', searches through the whole of reddit, if we want to search through a specific subreddit we just replace 'all' with the subreddit you want to do the scrapping on.'search(search_term, sort='relevance', limit=100)', 'sort' allows you to choose which recommendation of subreddits to look into, relevance is just to keep the results closely related to the product we have search. Whereas 'limit' just sets the maximum number of subreddits we will be scrapping from.

The function for post in post takes in the title and the selftext and collates it as 1 comment
Then the comments.replace_more(limit=25) function loads that post and their comments
.append({'Comment':comment.body}) adds comments post and self text under a dataframe column called Comments
df['comment'].str.replace('\n', ' ') removes the spaces and line breaks from the comments

Website Extraction

We also decided that the websites obtained from google will be useful for our project. Therefore we created a code that is essentially able to get the comments and text from these websites to aid our design process.

Firstly, we specified our search_terms, which is 'nothing phone (1)' for this project. Then using the google_search library, we will get a list of URL of the top 10 websites of our search result on google. Afterwhich, we defined a get_sentences(link) function which will be able to get all the sentences in a website into a list when given a website link. And we iterate for each URL found using the google_search library, we run it through the get_sentences(link) function and eventually end up with a list of all the sentences of all 10 websites.

Finally using pandas dataframe, we store the list of sentences into a dataframe and export it as a csv file.

Combination of all 3 data sources

The final step of our data extraction process is combining all comments retrieved from youtube, reddit and web scraping into a single dataframe and eventually saving it as a csv file as 'All_CombinedRaw_Data.csv'.

How we did it is that we read all the csv files that were created when extracting data from youtube, reddit and web scraping as a pandas dataframe. And concatenating all 3 data frames into 1 dataframe and exporting it to csv.

2. Data Cleaning

Before running analysis and classification models on the datasets, it is essential to do data cleaning to remove anything that does not provide value to our project goals of finding areas of improvements for the Nothing Phone (1). Our data cleaning process involves these steps below:

1. Tokenisation
2. Removal of Stopwords
3. Lemmatization
4. Split neutral terms
5. Language split
6. Remove 1 word comments

The code for the data cleaning is all in the "Clean_data.py" code folder. Our inputs for this process will be the extracted comments CSV file from our data collection process. In the code, all these functions would run on the same file but we will be going in depth to what each of these cleaning functions does and the justification in doing so.

2.1 Tokenisation, Removal stopwords, Lemmatization

Tokenization is the first step to do when doing data cleaning, it splits the comments into individual words or terms called tokens. It is an important first step in data preprocessing as it is required for lemmatization. Lemmatization is used to remove inflection by determining the part of speech and utilising a detailed database of the language, reducing a given word to its root meaning to identify similarities e.g. from 'better' to 'good', it derives the meaning of a word from a dictionary. The function 'Lemmastise_sentence' tokenise the sentences and lemmatises it and then joins the words back together.

Stopwords are frequent words used in the english language that do not have specific semantics such as "the","is" etc, removing them prevents common words from showing up for common words used during the data analysis step such as when we wish to detect the most used word. The function "remove_stop_words" identifies non-semantic words and removes them from the sentences. Doing all of this will increase search performance as there are lesser words to process.

It is important to note that we also remove the words "nothing", "phone", "1", and "one" from each sentence in the data too. Reason being we predict that these words will affect our result when finding out the words that appear the most in the later stage. As our search term is "nothing phone (1)", most comments describing this product will have a high probability of having the name of the product in their sentences too. So in order to avoid the top words being the name of the product, we decided to remove the name of the product in our data.

2.3 Removal of emojis,links ,duplicated comments and lower casing words

Links, emojis and duplicated/repeated comments are then removed from our dataset through the "clean_comment()" function. These are removed as it will not be relevant to us; although emojis can give us an emotion towards something, it will not be able to provide us exact details in what they are reacting to, i.e. what specifications, hence we remove it as it will be difficult to obtain specifications from that.

As we are focusing on text emojis, we cannot really discern much from emojis so we will be removing it to reduce our data. Additionally, links in comments would not be relevant to our projects as most of them would be for advertisement. which is not useful for our project.

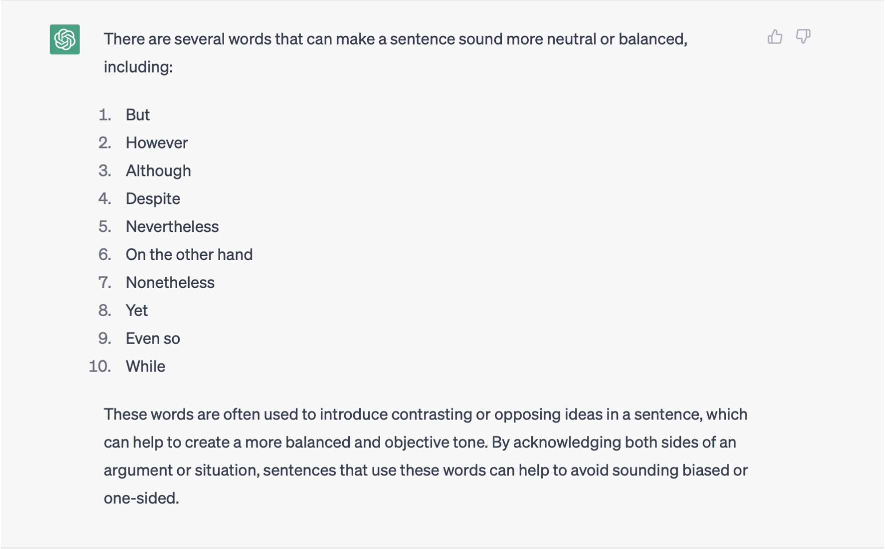
We also used the pandas dataframe's drop_duplicate() function to remove all the entries that were repeated. And later on resetting the index of the pandas dataframe to make it easier to visualise how many entries are left in the dataframe.

We also converted all characters in the data to be lowercase. This is so that there wont be any error of upper and lower case when we try to search for any specific words within a comment in the later stage. Therefore we standardise for every single character of every comment in the data to be lowercase.

2.4 Split neutral terms

We also have split long balanced comments with neutral words like 'but' and 'however'' to rule out the comments that are neutral, essentially splitting up one long comment into 2 comments; positive and negative.

We proceeded to ask Chatgpt what are the possible words or terms that can make a sentence sound neutral. We then check if any comments consist of any of these neutral words, and if there is, we will split the sentence with that word. Below is a picture of chatgpt's response of the neutral terms.



There are several words that can make a sentence sound more neutral or balanced, including:

1. But
2. However
3. Although
4. Despite
5. Nevertheless
6. On the other hand
7. Nonetheless
8. Yet
9. Even so
10. While

These words are often used to introduce contrasting or opposing ideas in a sentence, which can help to create a more balanced and objective tone. By acknowledging both sides of an argument or situation, sentences that use these words can help to avoid sounding biased or one-sided.

2.5 Removal of non-english comments

After this, in the dataframe of comments, we added another column named "language". In this column, essentially we used the function detect() from the langdetect library in python, so that it is able to go through each sentence in the dataframe to determine what language it is in, and label each sentence's language in the column "language".

After we have a column of language labels for each sentence, we proceed to only take the sentences which are only classified as English to move further into our project.

2.6 Remove 1 word comments

This step is the last step of our data cleaning. How we decided to remove the 1 worded entries in the data is because when we looked at the data, we realised that these one wording will not give us good information regarding how to improve the product.

Take for example in the data there is the entry "battery". Without any more words, we will definitely be able to deduce whether this is a positive or negative comment. Which will be analysed in the further part of this project. So considering the time taken to run analysis and the size of the data, we decided that it is a waste of time and computational effort to let the model try to deduce a one-worded entry whether it is a positive or negative comment. We also tried thinking of one-worded entries that may be beneficial data to our design requirements but we could not think of an example. Which led us to ultimately decide to remove 1 word comments from the data.

How we did it is we check if there is any white space in each comment. And if there is no white space in that comment, indicating that it is a one-worded entry in the data, and so we remove this entry.

3. Data Processing & Analysis

At this stage, we will have a huge data of all the cleaned and ready to be processed and analyse comments from various sources.

Our group is left with a total of 34425 comments from all 3 data sources. However, it is from here onwards we have decided to remove the data from scraping websites from our data set as it contains a lot of irrelevant text due to it including words from advertisements, headers, directory etc and it also requires huge amounts of time and computer resources to run the whole data set through Sentient Analysis.

Removing it leaves us with 29155 comments. But the issue is that because the model that we will be putting these comments into is a deep learning model. Meaning that it will take a considerably longer time to analyse one comment. Given that we have 29155 comments, we do not have the time and resources to run it for all of these comments as we estimated the time taken for all 29155 comments to be analysed by the model is around 22 hours.

Therefore, we decided to randomise our comments from Youtube and Reddit and randomly picked out 15% of the data. Leaving us with a sample of 3340 comments from the huge 29155 comments of the original dataset. We believe that this is a smart move as it will hugely reduce the hours required to do the analysis and classification of our data and we believe that it will not affect the final result of our project. But we noted that in an ideal situation, we should run all of the data into the model to get the maximal possible accurate final output of the project given that we have enough time and strong enough computers to run the computations.

For this part of the project, we did research and decided to try 4 models on our dataset. Namely, the Siebert Sentiment Analysis and Zero shot classification which we retrieved from a website called huggingface, Vader Analysis, as well as topic modelling.

3.1 Siebert Sentiment Analysis

To run the Siebert Sentiment Analysis on our dataset using the huggingface website, our python code uses a pipeline to connect the huggingface model to our code. There also is a function called analyze_sentiment(comment) so that we can iterate through all comments and pass it on to this function for analysis. The function will then analyse whether a comment is positive or negative and label it in a new dataframe column as either “POSITIVE” or “NEGATIVE”.

3.2 VADER model

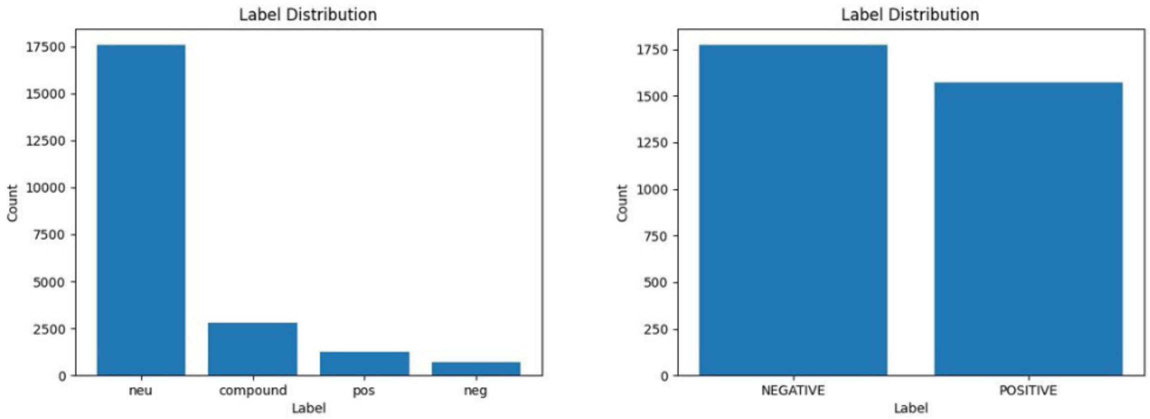
At the same time, we have decided to use the VADER model which is an alternative to Sentiment Analysis that is more suitable in discerning social media comments . It is during this process we realised that the VADER model took lesser time in classifying the sentiment values hence we decided to run our full Dataset through it (29155). The key difference in terms of output is that this model classified the neutral comments as well.

3.3 Extracting negative sentiment comments from both VADER and Siebert Sentiment Analysis

Comparing the 2 models which can do the same thing can be done very differently. We decided to use Siebert's Sentiment Analysis from huggingface over Vader. Here are the reasons why.

1. As seen in the figure below, Siebert’s Sentiment Analysis is able to identify more negative comments in the dataset and Vader actually classified the least negative comments on the other hand. Looking at this bar chart, we deduced that Siebert is able to analyse a comment better than Vader, which is why Siebert’s model will take a longer time to run.
2. Siebert Sentiment Analysis is a deep learning model, which means it is able to analyse comments more accurately.
3. Too many comments that went through the Vader model are classified as Neutral. Because we believe that we already successfully ruled out all the neutral comments in the cleaning data part of the process. Therefore the fact that so many comments are still classified as neutral tells us that the Vader model may not be suitable to use in our case.

Therefore moving on further into our project, we utilised the data that ran through the Siebert Sentiment Analysis model. The output after running this model is a column in our dataframe, indicating whether that comment is a positive comment “POSITIVE” or a negative comment “NEGATIVE”. We then took out all the comments that were labelled as “NEGATIVE” because these will be the comments that will be more valuable to us when deducing the final design requirements. Whereas the positive comments will not be able to tell us how we might improve the phone.



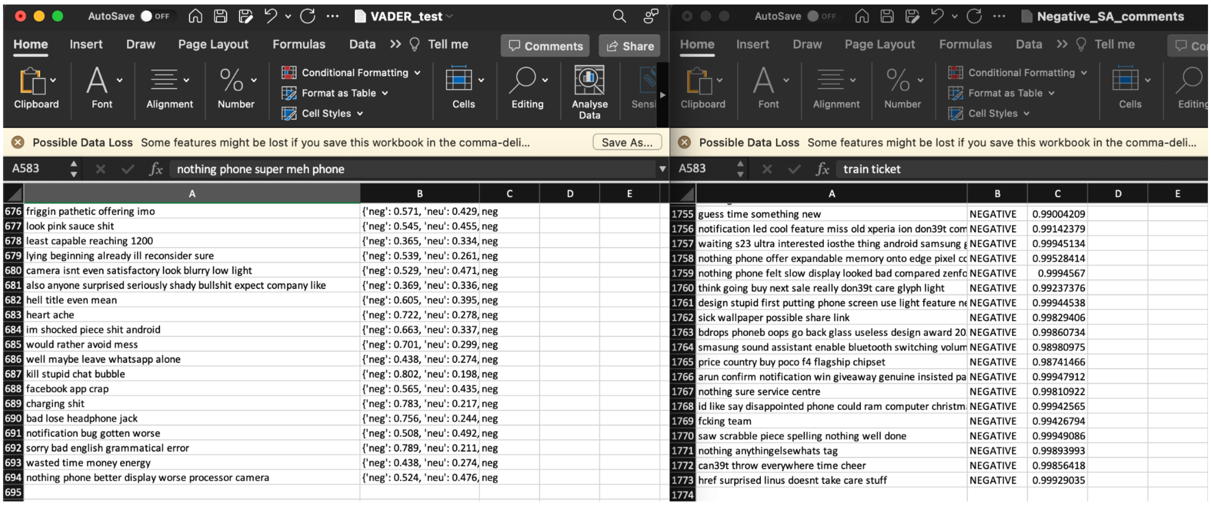
VADER Barchart

Sentiment Analysis Barchart

Figure: VADER vs Sentiment Analysis polarity results

3.4 Comparisons of results between SA and VADER for extracted negative comments

The VADER model returns us with only 494 negative comments out of the full 29155 comments within 5 seconds and sentiment analysis returns us with 1773 negative comments out of 3340 which took 68 minutes.



A583 nothing phone super meh phone

	A	B	C	D	E
676	friggin pathetic offering imo	{'neg': 0.571, 'neu': 0.429,	neg		
677	look pink sauce shit	{'neg': 0.545, 'neu': 0.455,	neg		
678	least capable reaching 1200	{'neg': 0.365, 'neu': 0.334,	neg		
679	lying beginning already ill reconsider sure	{'neg': 0.539, 'neu': 0.261,	neg		
680	camera isnt even satisfactory look blurry low light	{'neg': 0.529, 'neu': 0.471,	neg		
681	also anyone surprised seriously shady bullshit expect company like	{'neg': 0.369, 'neu': 0.336,	neg		
682	hell title even mean	{'neg': 0.605, 'neu': 0.395,	neg		
683	heart ache	{'neg': 0.722, 'neu': 0.278,	neg		
684	im shocked piece shit android	{'neg': 0.663, 'neu': 0.337,	neg		
685	would rather avoid mess	{'neg': 0.701, 'neu': 0.299,	neg		
686	well maybe leave whatsapp alone	{'neg': 0.438, 'neu': 0.274,	neg		
687	kill stupid chat bubble	{'neg': 0.802, 'neu': 0.198,	neg		
688	facebook app crap	{'neg': 0.565, 'neu': 0.435,	neg		
689	charging shit	{'neg': 0.783, 'neu': 0.217,	neg		
690	bad lose headphone jack	{'neg': 0.756, 'neu': 0.244,	neg		
691	notification bug gotten worse	{'neg': 0.508, 'neu': 0.492,	neg		
692	sorry bad english grammatical error	{'neg': 0.789, 'neu': 0.211,	neg		
693	wasted time money energy	{'neg': 0.438, 'neu': 0.274,	neg		
694	nothing phone better display worse processor camera	{'neg': 0.524, 'neu': 0.476,	neg		
695					

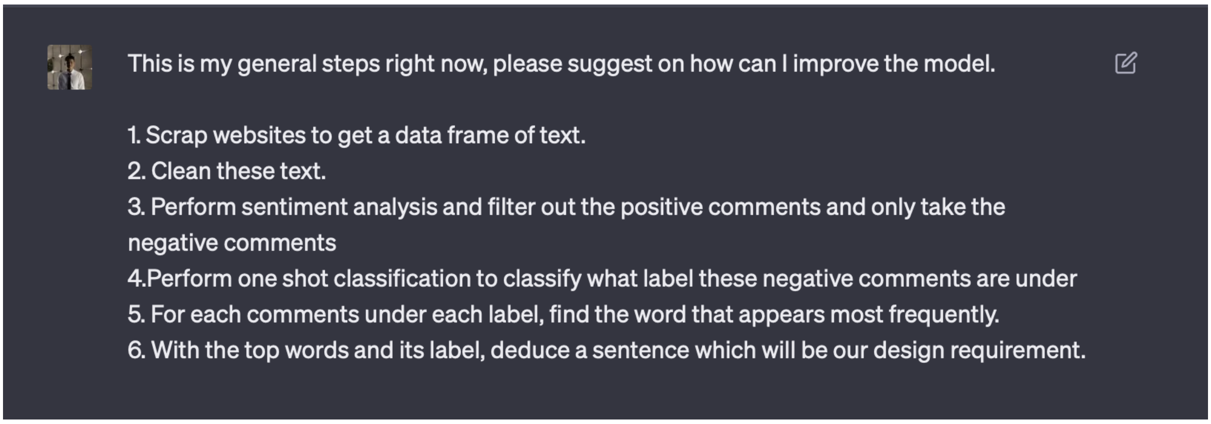
A583 train ticket

	A	B	C	D	E
1755	guess time something new	NEGATIVE	0.99004209		
1756	notification led cool feature miss old xperia ion don39t com	NEGATIVE	0.99142379		
1757	waiting s23 ultra interested iosthe thing android samsung	NEGATIVE	0.99945134		
1758	nothing phone offer expandable memory onto edge pixel cc	NEGATIVE	0.99528414		
1759	nothing phone felt slow display looked bad compared zenfo	NEGATIVE	0.9994567		
1760	think going buy next sale really don39t care glyph light	NEGATIVE	0.99237376		
1761	design stupid first putting phone screen use light feature ne	NEGATIVE	0.99944538		
1762	sick wallpaper possible share link	NEGATIVE	0.99829406		
1763	bdrops phoneb oops go back glass useless design award 20	NEGATIVE	0.99860734		
1764	smasung sound assistant enable bluetooth switching volum	NEGATIVE	0.98980975		
1765	price country buy poco f4 flagship chipset	NEGATIVE	0.98741466		
1766	arun confirm notification win giveaway genuine insisted pa	NEGATIVE	0.99947912		
1767	nothing sure service centre	NEGATIVE	0.99810922		
1768	id like say disappointed phone could ram computer christm	NEGATIVE	0.99942565		
1769	fcking team	NEGATIVE	0.99426794		
1770	saw scrabble piece spelling nothing well done	NEGATIVE	0.99949086		
1771	nothing anythingelsewhats tag	NEGATIVE	0.99893993		
1772	can39t throw everywhere time cheer	NEGATIVE	0.99856418		
1773	href surprised linus doesnt take care stuff	NEGATIVE	0.99929035		
1774					

Figure: Comparing VADER and SA models

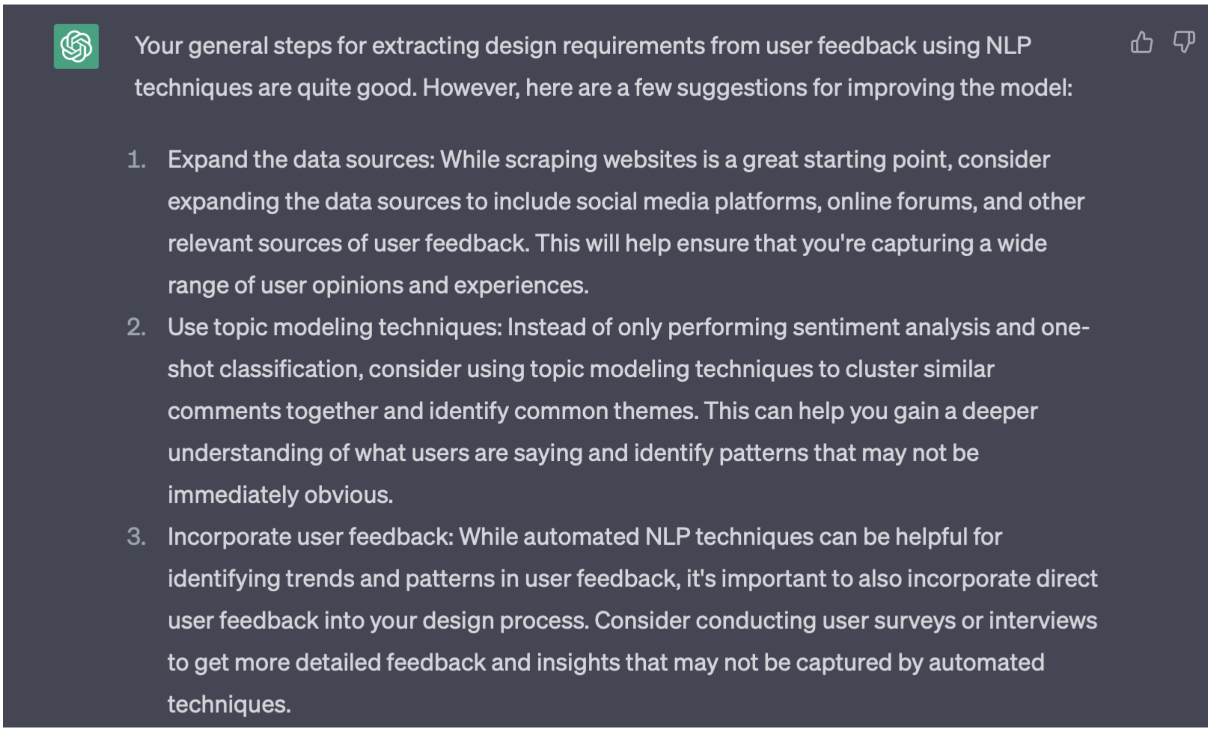
3.5 Topic Modelling

This model is actually suggested by Chatgpt. Upon finalising our work plan for this project, we wondered if there is any other way we could go about this project in a better way. So we turned to ChatGPT. Below is a screenshot of our conversation with ChatGPT.



This is my general steps right now, please suggest on how can I improve the model.

1. Scrap websites to get a data frame of text.
2. Clean these text.
3. Perform sentiment analysis and filter out the positive comments and only take the negative comments
4.Perform one shot classification to classify what label these negative comments are under
5. For each comments under each label, find the word that appears most frequently.
6. With the top words and its label, deduce a sentence which will be our design requirement.



Your general steps for extracting design requirements from user feedback using NLP techniques are quite good. However, here are a few suggestions for improving the model:

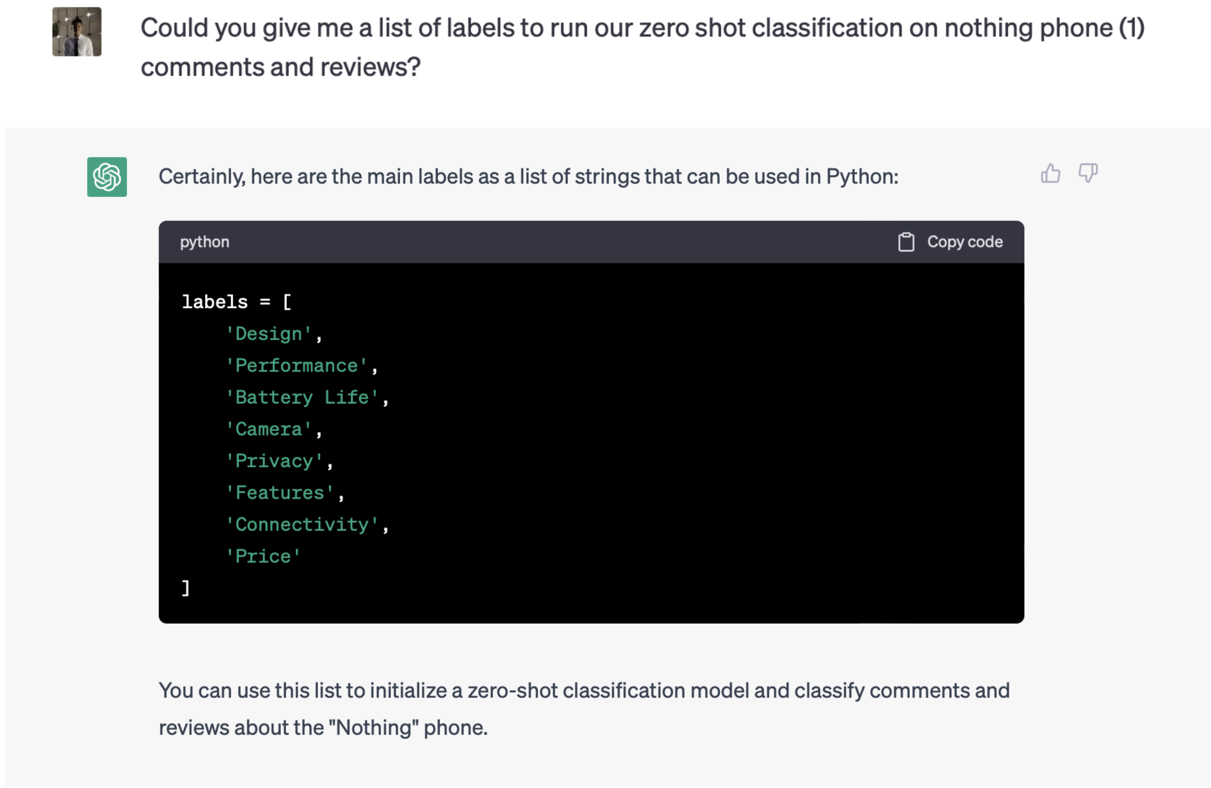
1. Expand the data sources: While scraping websites is a great starting point, consider expanding the data sources to include social media platforms, online forums, and other relevant sources of user feedback. This will help ensure that you're capturing a wide range of user opinions and experiences.
2. Use topic modeling techniques: Instead of only performing sentiment analysis and one-shot classification, consider using topic modeling techniques to cluster similar comments together and identify common themes. This can help you gain a deeper understanding of what users are saying and identify patterns that may not be immediately obvious.
3. Incorporate user feedback: While automated NLP techniques can be helpful for identifying trends and patterns in user feedback, it's important to also incorporate direct user feedback into your design process. Consider conducting user surveys or interviews to get more detailed feedback and insights that may not be captured by automated techniques.

Essentially what this model does is it goes through all of our comments, then it will try to find out what are the key words in each of these sentences and gives a list of keywords for each comment.

3.6 Zeroshot Classification

Forming categories to classify comments into before putting into Zeroshot:
We managed to write a python code that is able to ask ChatGPT to provide us with categories to put into when doing our Zeroshot Classification. Reason why we did this is so that even when the product of interest changes, it will still be able to give us categories regardless of whatever product.

To perform Zeroshot Classification on our model, we first have to have a list of categories so that the model knows what are the options it can classify each sentence into and later on proceed to classify them into the category which the model thinks the sentence is more related to. So for the output of this model after we ran it with our data, it is an additional column in the dataframe, labelling what is the comment talking about given that the categories were given by ChatGPT.



Could you give me a list of labels to run our zero shot classification on nothing phone (1) comments and reviews?

Certainly, here are the main labels as a list of strings that can be used in Python:

```python
labels = [
    'Design',
    'Performance',
    'Battery Life',
    'Camera',
    'Privacy',
    'Features',
    'Connectivity',
    'Price'
]
```

You can use this list to initialize a zero-shot classification model and classify comments and reviews about the "Nothing" phone.

3.7 Zeroshot Classification or Topic Modelling and the interpretation of the result

We eventually decided that Zeroshot Classification will be better than Topic Modelling in our project. And here is the reason why.

The keywords of Topic Modelling are too wide. Because for Topic Modelling the model is identifying the keywords for each sentence by itself. Therefore for every sentence, there will be a very diverse range of keywords, which we will not be able to really deduce what this sentence is talking about. Whereas for Zeroshot classification, the categories are given by us. So it will classify each sentence to what we need to know about. Therefore each sentence will be fine tuned to our categories which will be used to identify design opportunities.

From the result of the Zeroshot classification, we did a barchart to find out which is the category that has the most comments. In other words, which are the categories of the phone that people have more negative things to talk about. From the figure below, we found out that the top 4 categories of nothing phone (1) that most people have bad things to say about is Features, Design, Price, and Performance. Which will be the 4 categories we choose to focus more on when deducing our final design requirements.

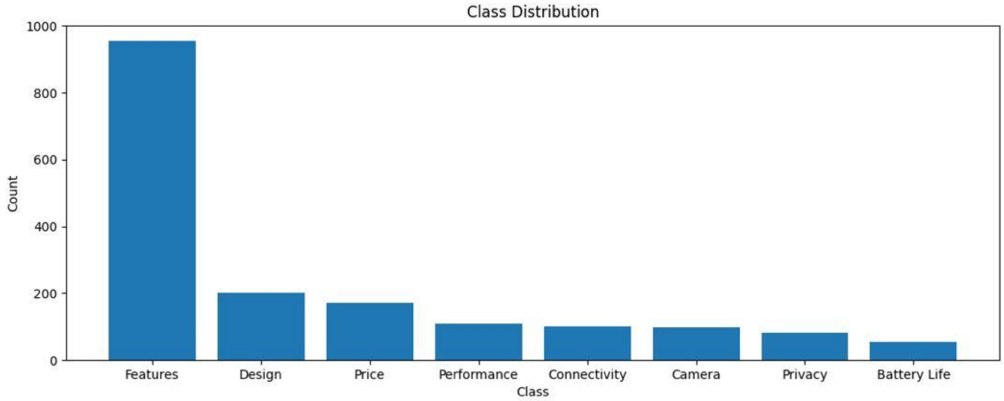


Figure : Distribution of negative comments from Sentiment analysis

As can be seen by the above figure, most of the negative comments are dissatisfied about the Features of Nothing Phone (1), due to the overwhelming negative comments on this specifications as compared to other specifications, we decided to focus on designing improvements for the features, design, price and performance of the Nothing Phone(1) which are the top 4 classes that received the most negative comments.

3.7 WordCloud Visualization

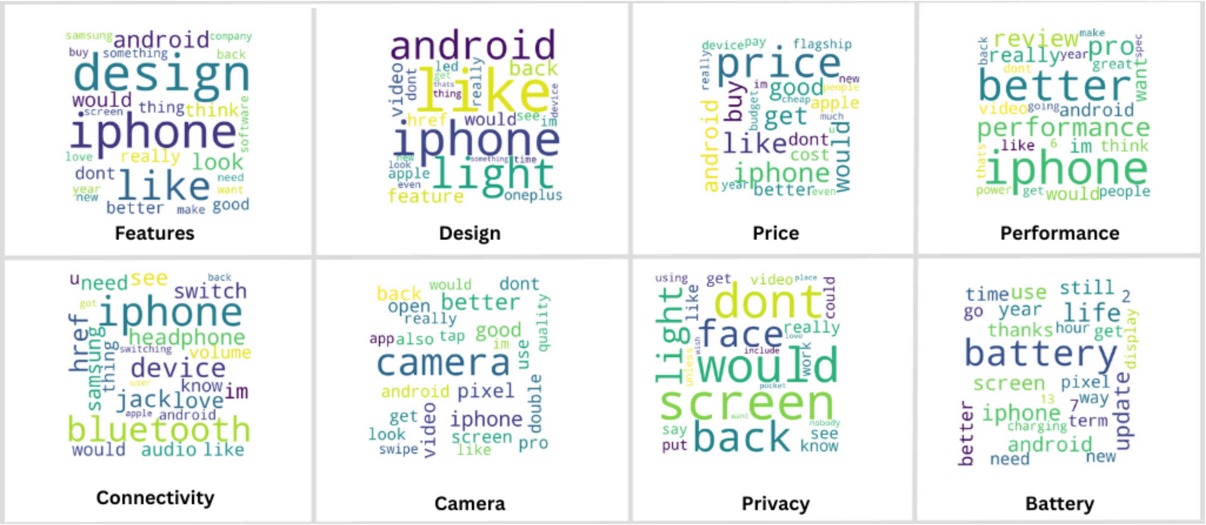


Figure: Wordcloud on Sentiment Analysis

We ran all the negative comments we obtained into the nltk Wordcloud text analysis to get a visualisation of the most common words that appeared for every category, where the larger the word, the bigger the letter size.

Essentially, for each class we counted how many times each word appears, when we take the top 25 highest frequency words and display it in a Wordcloud.

3.5 Extracted comments for final evaluation

After we successfully obtained the words that appeared the most for each class of comets as seen as the Wordcloud figure above. We thought that these words will not give us the full understanding of what these reviews are truly talking about.

So we track back a little to get the comments data that has the top 3 words that appeared the most in the comments. How it works is that we found out what are the 3 highest appearance frequency words, then we wrote the code in such a way that for all comments that have any of the 3 top words, we put it in a new dataframe and export it as a csv file.

Lastly as final evaluation, we analyse each comment in the csv file and finally deduce our design requirements.

Design Opportunities and their Justifications

1. Finding Nothing Phone's own identity in the smartphone market

There are many comments on how the Nothing Phone (1) is highly similar in terms of looks to the iPhone, with some people even calling it a cheap version of the iPhone. As seen in the Wordcloud image for 'Features', "iphone" is one of the top 3 words that were mentioned in the comments. The fact that most comments referred to the iPhone just goes to show that the Nothing Phone (1) does not have a trademark of its own that people can instantly link to the Nothing company immediately upon seeing one of its products. For example, the iPhones are recognised for their flat edged bezel, while the Samsung Galaxy series are known for their curved edged screens. If the Nothing Phone (1) has a feature unique to the brand, people will stop comparing the Nothing Phone to other phone brands just based on how it looks at first glance.

2. Different sub models with different specifications

There were people who commented on how they want higher processing power while others mentioned that having a smaller battery and charging a smaller price would make the Nothing Phone (1)'s price range be a good mid range price for a phone. So by having different sub models, there can be different prices charged for different combinations of specifications that will tailor to different user needs.

3. New backing for the phone

Concerns were raised about the waterproofing for the phone, and there were also complaints about the backlights either being excessive or useless. One commenter mentioned that instead of having a backlight, there could be a stronger back casing that does not shatter. Hence, the Nothing Phone (1) can have a completely new backing where it takes all these comments into account. In other words, balancing the functionality and practicality of the design together with the aesthetics of it.

Reflections

1. Time taken:it take very long time to run SA to dataset, requires desktop/ stronger computer that can run for hours
2. Irrelevancy: Despite multiple cleaning, there would still be comments that are left behind that are irrelevant to our project goals
3. Detection is irrelevant, here, as our product is called "Nothing Phone 1" it leaves us with a lot of 'phone' and 'nothing' in the word clouds, where nothing can be used to refer to the phone, it can also refer to criticism of the product.

Summary of lesson learnt

To conclude our report, we will be sharing a summary of our lessons learnt when doing this project, and would we have done differently if we ever come across a similar project again.

1. Deeper research on each model’s algorithm.

In our project, we rely heavily on existing models and what we did essentially is only to apply them without cleaned data. While we did some experiments and had our own interpretation of why certain models are better and certain models are worse. A lesson learnt is to do more thorough research on the models that may be useful to our project, and study its algorithm answer the question of, will it take longer to run?, how is it classifying certain things?, how does it deal with bad and irrelevant data?, is it really suitable for comments/reviews classification or is it actually more for essays or articles?

2. Data cleaning is not a linear process, but more of a trial and error process.

When we were doing our data cleaning, it was an iterative process. Reason being it was a constant process of us completing one part of the cleaning, then checking manually on the data itself to identify what could be done better to the data. Take for example when we realised that there were a lot of random 39s in our dataset which needed to be removed, when we realised one word comments were invaluable for our study, and when we found out that some comments were repeated or non english comments. It was an iterative process of checking the data, finding out what was wrong with it, and coming with the code to clean it more thoroughly. So do not expect the cleaning data process to be a one shot finished process.

3. Different sources for different products

For different products that you are interested to do more research on, there will be different websites that have more data on certain products. For our project, we manually checked which were the best sources to get data about the nothing phone (1) from and landed with reddit, youtube, and top 10 google websites. So it is important to note that for different products, we cannot use the same sources as some products may be more popular in some sources and some products may be less popular in some sources. Some of the key factors we identify with are because some social sources, for example youtube/facebook/instagram, have very different users. For example facebook users are normally older people, instagram users are normally more of a teenage person. Therefore the topics that the users discuss in both these apps will be very different. So it is very important to do research on the product of interest to find out which are the sources you can achieve the best data and result out of it.