Sentiment Analysis between Product Reviews and Social Media Platforms

SIADS 591/592 Milestone I Project Report by James Mete & Stanley Hung (09-stanhung-jmete)

Motivation

With more consumers tending to purchase online and the important role of reviews in the online purchase decision making process, we think providing a guideline of how to read reviews effectively would be valuable for today's consumers. To do this, we want to investigate how reviews compare for certain categories of products across different platforms.

Namely, we will compare Amazon product reviews with social media platforms, Twitter and Reddit, which are the key platforms where consumers consume reviews. For this project, we will be comparing reviews of a few popular headphone models.

Our three main goals are the following:

- Compare sentiment for the same categories between Amazon, Reddit, and Twitter. One key question could be "Is one platform biased in a certain way?"
- Evaluating the constructiveness of the reviews between platforms. Which platform provides the most constructive feedback? And how constructive is it?
- Evaluating the top features that influence the discussion amongst the platforms. This will require top-keyword analysis. For example, on Amazon the reviews may focus on value for money whereas social media may focus on other aspects such as battery life. One key question here would be "What are the top features that are described in the reviews on each platform?"

Methodology and Analysis Framework

We use some key terminology throughout our research, namely sentiment and features.

Sentiment: This project used a python library named TextBlob[1] to perform sentiment analysis. It defines sentiment into two parts, polarity (how negative or positive) on a scale of -1 to 1 as well as subjectivity (how objective or subjective) on a scale of 0 - 1. We will focus most of our research on polarity, and any references to sentiment refer to polarity unless specified otherwise.

Features: Feature is a function / evaluation aspect of the headphone product. (e.g sound quality). This project defines features by analysing the noun phrase used in a review/comment. For instance, the noun phrase of the review "The sound quality is good!" is "sound quality". The noun phrase will be extracted by using the TextBlob library.

Data Sources

Scope

This project has used three data sources[16], and we have set a few boundaries to limit the scope of the project, primarily because of feasibility and efficiency reasons. These limitations were implemented in all the data collections processes from the data sources listed later. Details as follow:

- Limit the product category to premium true wireless earbuds

- This project selected 8 popular models to analyze based on its number of ratings on Amazon. These models are:
    - Apple Airpods, Apple Airpods Pro, Jabra Elite 65t, Jabra Elite 75t, Jabra Elite 85t, Samsung Galaxy Buds Live, Samsung Galaxy Buds Pro, Samsung Galaxy Buds Plus
- Limited our scope to Amazon reviews / Reddit comments / Twitter tweets in English posted in the year of 2020.

Amazon reviews

In each of Amazon's product pages, it has a section to display consumer's reviews. This project only scraped the reviews of the specific products listed in the Scope section

- This project used a Python web scraping package (Scrapy[2]) to scrape information from the Amazon review page. Location: https://www.amazon.com/product-reviews/{Amazon_Product_ID}
- The output of the web scraper is in csv format
- The key dataset's features included:
    - Reviewer name - the user name of the reviewer
    - Star rating - the rating that associated with each review
    - Country - the country that the review being posted
    - Review title - the title of the review
    - Review text - the complete text of the review
    - Review date - the review post date
- The web scraper retrieved **51,507 Amazon reviews** posted in 2020**.**

Reddit Comments

Reddit is a mainstream discussion platform for people to share their view on specific topics.

- This project access the Reddit comments archive from https://pushshift.io/ [3]
- Pushshift.io provided an API for public access to all Reddit comments
- The API returned the data in JSON format
- The key dataset's features included:
    - body - The comment text body
    - created_utc - time posted to Reddit
- Retrieved **148,344 reddit comments** that posted in 2020, based on the search keywords - "galaxy bud" for Samsung Galaxy Buds related comments, "airpod" for Apple Airpods related comments, and "jabra elite" for Jabra Elite Earbuds related comments

Twitter Tweets

Although Twitter has an API, the student version is limited to searching only 7 days in the past, so we instead used a large but raw archive of 2020 tweets we found online. The archive is not specific to our products, thus requiring a heavy amount of processing and data mining to find relevant tweets.

- We used a large archive of tweets totalling over 809 GB from archive.org.[4]
- The archive contains a complex nested file structure involving numerous forms of compressed files & folders eventually resulting in 480,451 .json.bz2 files that contained the tweets.
- Each .json.bz2 file is a compressed JSON file representing a minute of 2020. Each minute contains on average 6000 rows / tweets, **totalling ~2.88 billion tweets**.[5]
- The key features of the dataset were:
    - text - The tweet text body
    - timestamp_ms - Time posted to Twitter
- We retrieved **45,200 tweets** posted in 2020 relevant to our chosen products **after data mining**.

Data Manipulation

Due to the limitations set by different data sources, manipulation is required. Furthermore, we used numerous different environments such as local jupyter notebooks, terminal scripts, and Google Colab to help work together as well as take advantage of cloud processing. Also, we discarded any columns containing personal information such as username to help avoid ethical issues of unintentional harm.

Amazon reviews

This project used a web scraper to scrape product reviews from Amazon, web scraping is always a challenging task because Amazon tries its best to prevent a web scraper overloading their server or causing intentional harm to its service. This project specifically fine-tuned the web scraper to deal with the anti-scraping mechanisms. Data manipulation is also required to handle the scraped data.[6]

Data Collection - Scrapping

This project used a web scraping package (Scrapy) to build a web spider for scraping the Amazon site. Below are the anti-scraping mechanisms we discovered so far and the solutions this project implemented

- To avoid Amazon web redirecting the spider request to a Captcha page for human verification, the web scraper scrapes the Amazon web using the umich VPN connection and also randomly rotates the request headers to make every request a human-like request.
- Amazon product review page has a limitation of 500 pages of reviews (each page contains 10 reviews only) at max. Amazon review page provides a few sorting and filtering options. By constructing these options with different combinations, the Amazon review page will return a different set of top 5,000 reviews. By doing so, the web scraper was able to maximise the amounts of scrapped reviews. For instance, the web scraper scrapped 11,594 Airpods reviews out of 15,000 in total, a far better result than the original 5000 review limit.

Data Cleaning

As mentioned in the second bullet point in the above data collection section, the scraper scraped a different set of “top 5,000 reviews” of a single product. To remove possible duplicated reviews, this project used the Pandas concatenate and duplicated function to remove duplicate records. Detailed below:

- Use pandas.concat() to concatenate the multiple set of “top 5,000 reviews” df (belong to the same product) into one DataFrame
- Use df.duplicated() to create a mask layer to flag out duplicate records.
- Then select all non-duplicate records by using df[~df.duplicated()], and create a new dataframe

Reddit Comments

The Pushshift API provides 3 endpoints - comment, submission, and subreddit.

Data Collection - API Endpoints

Given the nature of Reddit as a discussion forum, this project can’t use a simple query term to retrieve comments. For instance, searching comments by using a query term of “Airpod” will only return the comments that have the word “Airpod” in it. It will likely not return the comments that reply to a post which is related to “Airpods”, because some comments may not always have the term - “Airpods” in it. They may just use “it” to refer to the product. To tackle this problem, this project constructed 3 queries to cover as many relevant comments as possible. Detailed below:

1. **Universal comment search using the comment endpoint.** The Pushshift comment endpoint will search through all comments under any posts in the archive that matched the query parameters. This project uses a keyword query string (“galaxy bud” / “airpod” / “jabra elite”) and time parameter, to retrieve all comments in the year of 2020 that have the keyword in it.
2. **Universal submission search using the submission endpoint.** The Pushshift submission endpoint will search through all submissions in the archive that matched the query parameters. On reddit, "submission" refers to the post where comments are not included. A submission only consists of the title of the post and the body text of the post. This project uses a keyword query string (“galaxy bud” / “airpod” / “jabra elite”) and time parameter (limited to the post in the year of 2020) to retrieve all submissions in the year of 2020 that have the keyword in its title
3. **Submission’s comment search using the comment endpoint.** For each submission returned from the universal submission search, this project takes the “submission id” as a parameter to construct a comment endpoint searching query. In this case, the comment endpoint will return all the comments that belong to all submissions (the post) retrieved from the submission endpoint.[7]

Data Cleaning

Similar to the Amazon reviews, the drawback of this approach is duplicated comments between universal comment search and submission’s comment search. We adopt the same cleaning steps as the Amazon Reviews cleaning process to use in cleaning the Reddit dataset.

Twitter Tweets

Twitter was a unique challenge due to the massive scale of the data and uncertainty in whether the datasets we downloaded contained relevant tweets related to our chosen products.

Data Collection - Data Mining

The datasets we collected were folders with a unique structure of nested files. The structure was:

- ❏ Month folder
    - ❏ Day folders as .tar or .zip files
        - ❏ 24 Hour folders per day
            - ❏ 60 Minute files per hour containing tweets compressed as .json.bz2 files.

Extracting the tweets posed a unique challenge due to the massive scale. We needed to consolidate all tweets to make it easier to mine and analyze. We wrote custom functions to perform the following:

1. Loop through the nested structure and decompress the .tar or .zip files.
2. Collect the paths to the minute files per month.
3. Loop through the paths per month and read them into pandas dataframes, filter only English Non-Truncated tweets using available columns, drop unnecessary columns, and finally append the results to a .csv file per month.[8]

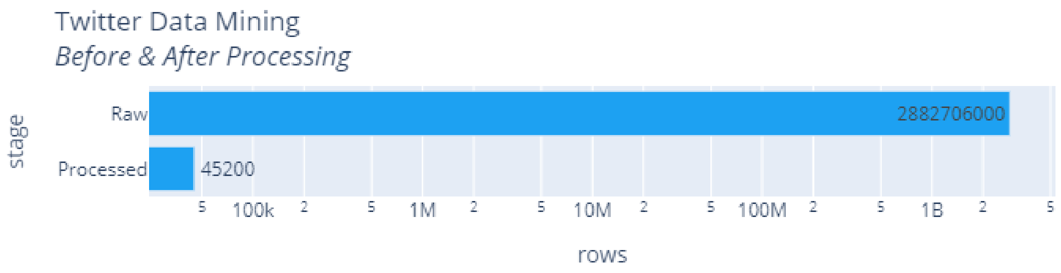
Although some months had less information such as missing some days, we still had to be careful to avoid memory issues. Processing each minute file separately allowed us to avoid running out of RAM.

The resulting 12 .csv files containing english non-truncated tweets were on average 20 GB totalling 242 GB for the entirety of 2020.[9] We chose to separate the processing from the mining due to the large bottleneck of I/O processing related to reading and writing a lot of smaller files which took on average 12 hours per month-folder. We ran our scripts on multiple computers to improve our efficiency similar to a parallel processing cluster.

Data Cleaning

We could use our resulting .csv files to perform data mining and cleaning operations to extract relevant tweets. We wrote custom functions to perform the following steps:

1. Utilized Spark[10] to read and process the large .csv files of raw tweets.
2. Specified data types in Spark to improve memory efficiency and reduce errors.
3. Utilized Spark SQL multiple times to query tweets into a relevant dataframe.
4. Created a udf to perform regex matches to find tweets that contained combinations of our products. For example, the words galaxy & bud anywhere in the sentence in either order.
5. Our UDF presents us with a dict of boolean values for whether the products are in the tweet. We then use that dict to create separate boolean columns that can aid in filtering later on.
6. We filter our resulting dataframe to only show rows that contain at least one of our products.
7. We collect the resulting rows using spark into a list which can then be examined and exported to a .csv file using pandas. This process creates a processed .csv per month.[11]
8. We wrote a function to read all the processed files into a single dataframe to be used for analysis.



Missing Data

Most of the data we sourced contained our key features of interest, namely the main body of text and the date. However, there were some cases where we would drop missing values or fill them in using pandas functions. For example, we may drop rows that have missing text. In terms of filling missing data, the value to use for the fill depends on the situation. A large majority of our data is text based which makes it difficult to fill. We often used fill to remove NaN values with values such as False or 0 which helped us reduce errors despite being similar.

One challenging area was in terms of uncertainty analysis due to Feb for Jabra tweets only having 1 row of data which meant we could not calculate a standard deviation to use in terms of uncertainty analysis. We chose to use the average std of Jabra in 2020 to help fill in the table but this does not avoid the biased starting point of the uncertainty bounds. This topic requires more data collection and investigation.

Joining dataframes and further manipulation

In this project, we have 3 main dataframes - Twitter tweets / Reddit Comments / Amazon Reviews, each dataframe has its own set of column names. This project further manipulated and combined the data to create 2 master dataframes for analysis. (Sentiment Analysis Dataframe & Feature Analysis Dataframe)[12]

Sentiment Analysis Dataframe

The Sentiment Analysis dataframe is a master dataframe that combines all 3 main dataframes (Amazon, Reddit, Twitter) together and calculates a sentiment score for further analysis. The Pandas concatenate function is used to combine the dataframes. However, the concatenate function requires a consistent column name across all dataframes. Therefore, a series of steps are required beforehand:

1. **Slice and get the relevant columns to minimise the memory usage and maximise the processing efficiency**
2. **Standardize product labelling scheme / format** - Twitter and Reddit dataframe used a similar product labelling approach to indicate which product is related to specific comment / tweet. They have 3 product labeling columns (“airpods”, “galaxybuds”, “jabra”) with the boolean dtype, which means an airpods related comment/tweet would have True value in the “airpods” column if present. However, the Amazon dataframe uses a different approach, it only has one string type column (named “product”) to indicate the product label. The Twitter/Reddit dataframe is converted to the Amazon product labelling approach by using pandas .melt() function.
3. **Convert datetime column to pandas datetime object**
4. **Add a “platform” column to indicate the social media platform it belong to**
5. **Standardise the column names across dataframes before the concatenate operation.**
6. **Standard product names and columns allows us to easily combine the dataframes by concatenating them together in pandas effectively joining them.**

Feature Analysis Dataframe

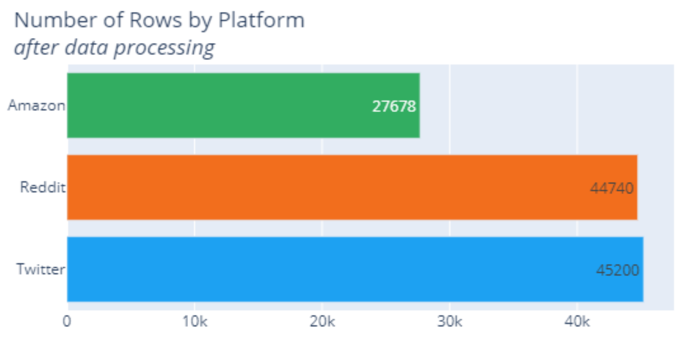
The Feature Analysis dataframe is a master dataframe that contains the noun phrases extracted from all 3 main dataframes (Amazon, Reddit, Twitter) together and calculates sentiment score for further analysis. Pandas Concatenate function is used to combine the dataframes. This project adopted the same manipulation step as the Sentiment Analysis Dataframe manipulation steps. In addition, a series of extra manipulation steps were also implemented. Detailed as follow:

1. **Noun phrase extraction** - This project applied the TextBlob function to convert the review/comment text into a “Blob object”, then extracted the noun phrases by assessing the noun_phrases property of the “Blob object”, finally stored the list of noun phrases in a separate column named “Noun_Phrase”.
2. **Noun Phrase sentiment calculation** - This project references the sentence level sentiment score. For example, the noun phrase of a comment “The sound quality is good!” is “sound quality”, which is a neutral term. Therefore, the sentiment score of the sentence is used instead. After the above steps, a separate column (named Noun_Phrase) was created in each main dataframe with a list of (noun phrase, sentiment score) tuple stored in the column.
3. **List Explode** - The Noun_Phrase column then passed into pandas explode function (Dataframe.explode()) to transform the tuples list to rows with each row containing a single tuple.
4. **Feature Categorisation** - It is common that people use slightly different phrases to describe the same thing due to typo or personal behavior. Python difflib library is used to find similar noun phrases. The top 200 unique noun phrases were analyzed to extract meaningful noun phrases and created a feature list (a list of meaningful unique noun phrases) with each unique noun phrase as category name. Then used the difflib get_close_matches() function to find all similar text and assign it the same feature category name in the new column named “feature_category”. These steps were all wrapped in a custom feature_categorization() function to find the closest match and assign the new category name.
5. **Column separation** - The explode Noun_Phrase column was also separated into 2 columns by unpacking the tuple into a noun phrase and sentiment score column separately.

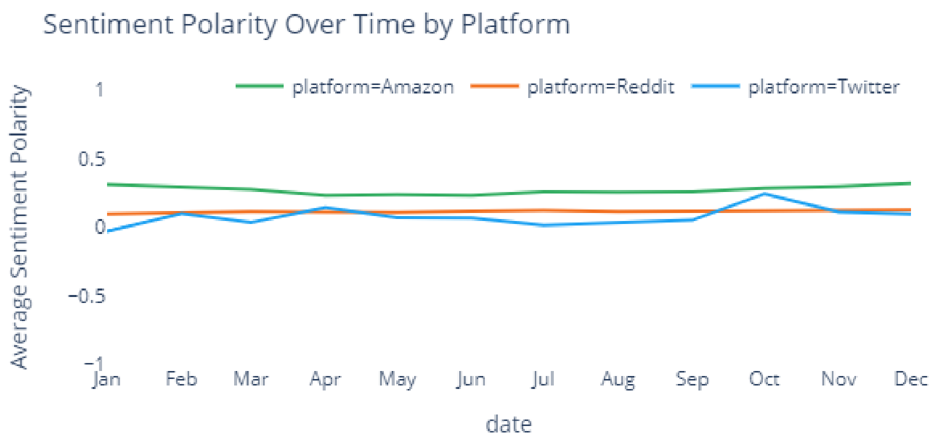
Analysis

Platform Analysis

We were left with **117,618 rows of data** to analyze spread throughout the three platforms after our data processing and cleaning steps. One noticeable difference is that Amazon has only 60% of the number of rows as the social platforms.

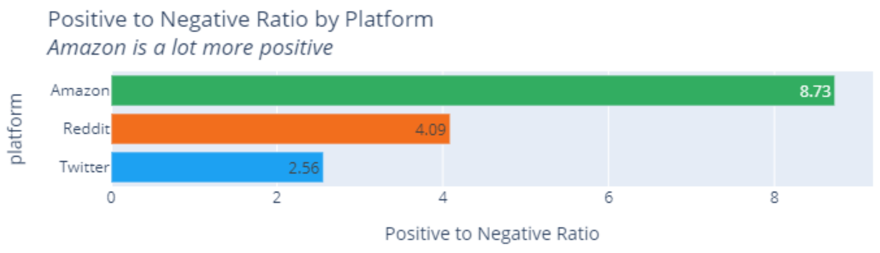


Sentiment Polarity

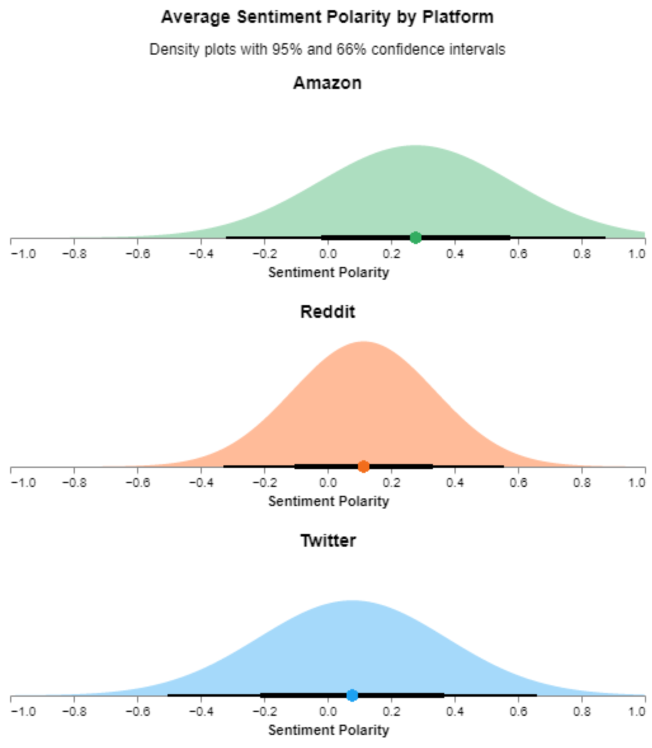


Our research shows a relatively flat relationship over time in regards to our three main platforms in relation to our chosen products. We can notice that **Amazon scores at a higher level of sentiment polarity at a consistent rate of roughly 0.3.** Reddit tends to be flat and closer to neutral sentiment of 0. Twitter is interesting because it starts out negative and peaks at nearly the same level of positivity as Amazon in October before falling back to neutral levels similar to Reddit by December. Further investigation of this spike is desired.

The above visualization and analysis does not include the uncertainty bounds largely to make it easier to read and aid in effectiveness. However, our research does show high levels of uncertainty regarding the sentiment means. To better understand this, we can take a look at the average sentiment per platform in 2020 to show how pure ratios or scores may be deceiving.



Amazon's positive to negative ratio is double Reddit's, and nearly 4 times as much as Twitter. In terms of average sentiment, Amazon continues to show a slight positive edge. This may give the impression that Amazon is truly more positive than the other platforms. However, our sample is merely a fraction of the overall amount of comments, tweets, and reviews that are possible. Thus, our sample (and our means based on it) have a degree of uncertainty compared to a true population. Furthermore, platforms may have external factors biasing results such as purchasers of products on Amazon may lean positive because they are invested in the product and actively bought it.



We performed uncertainty analysis throughout our research by calculating the 95% and 66% confidence intervals of the means, as well as plotted density curves to help visualize the uncertainty in our calculated means per platform. We created custom functions such as create_density_charts_platforms() to create the relevant confidence intervals, density information, and then automate the visualization creation.

By doing so, we can see a high level of uncertainty in our means per platform with wide curves and a lot of overlap between the confidence intervals of each platform. **The uncertainty underlying these means suggests a lack of meaningful difference between the means for both the overall platforms as well as the underlying features.**[13] Further investigation and a more rigorous statistical analysis involving more data as well as a calculation of the confidence intervals and p-values of the differences between the means would help determine if there are statistically significant differences between the means.

Feature Analysis

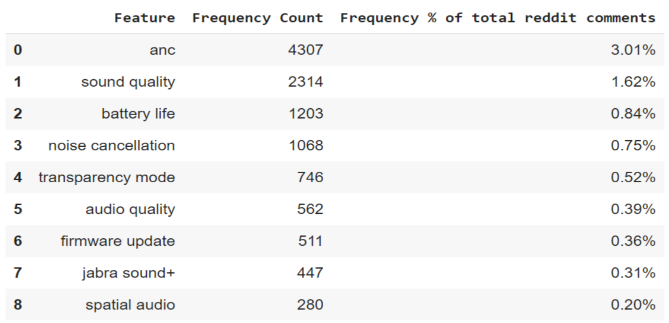
This analysis will look at the most used noun phrase on each platform and compare its sentiment score distribution between different platforms.

Most discussed features (Noun Phrases)

The most discussed noun phrases are…

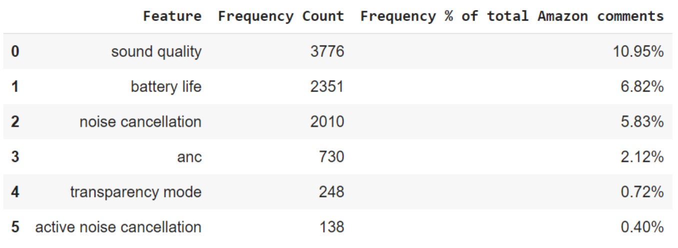
- Twitter
  No meaningful feature related noun phrases were found in the top 200 noun phrase list.

- Reddit
  - Top 3 - anc, sound quality, battery life
  - Some most discussed features are similar, these will be grouped together for sentiment analysis by using the feature categorization function



	Feature	Frequency Count	Frequency % of total reddit comments
0	anc	4307	3.01%
1	sound quality	2314	1.62%
2	battery life	1203	0.84%
3	noise cancellation	1068	0.75%
4	transparency mode	746	0.52%
5	audio quality	562	0.39%
6	firmware update	511	0.36%
7	jabra sound+	447	0.31%
8	spatial audio	280	0.20%

- <u>Amazon</u>
    - Top 3 - sound quality, battery life, and noise cancellation
    - Some most discussed features are similar, these will be grouped together for sentiment analysis by using the feature categorization function

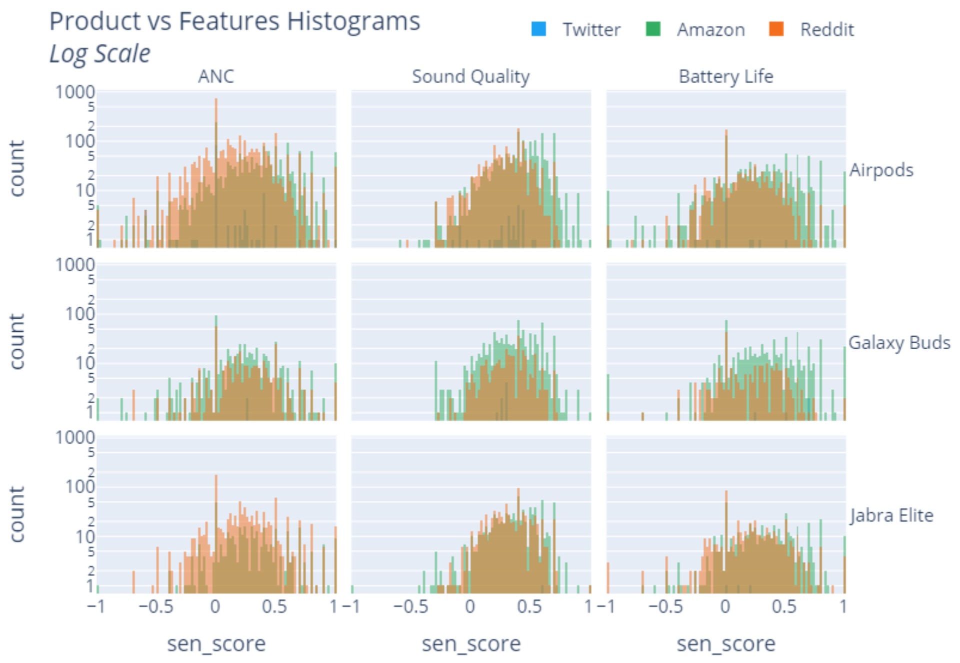


	Feature	Frequency Count	Frequency % of total Amazon comments
0	sound quality	3776	10.95%
1	battery life	2351	6.82%
2	noise cancellation	2010	5.83%
3	anc	730	2.12%
4	transparency mode	248	0.72%
5	active noise cancellation	138	0.40%

In conclusion, the most important features (top 3) are sound quality, battery life, and noise cancellation when evaluating true wireless earbuds on Amazon and Reddit. ANC (Active Noise Cancellation) were discussed more often on Reddit than Amazon.

Feature Sentiment Analysis

The below visualizes the sentiment score distribution of a feature about a specific product by platform. Please keep in mind It is usual that Reddit has a higher frequency count on most of the features due to its discussion forums nature.



A few interesting findings:

- Considering the nature of Reddit as a discussion forum, it should have a higher frequency on every feature. It is interesting to see that the overall discussion about Galaxy Buds's feature has a higher frequency on Amazon than Reddit.
- Airpods sound quality and battery life tend to be more positively discussed on Amazon. This kind of bias does not appear in Galaxy Buds and Jabra Elite buds.
- The distribution of the sentiment score is similar between Galaxy Buds and Jabra Elite across all 3 key features, regardless of the frequency count.
- Certain features evoke stronger emotional responses than others and differ by platform.[14]

Conclusion & Next Steps

- Although Twitter has the highest number of comments (tweets) amongst all platforms, it is less constructive due to the small amount of features found in the relevant tweets.
- Amazon scores higher than both social platforms in terms of average positive sentiment polarity, as well as a much larger positive to negative ratio.
- Reviews on Amazon are biased towards Airpods positively in regards to "sound quality" and "battery life", this phenomenon does not appear for Galaxy Buds and Jabra Elite buds.
- It is recommended that headphone intenders should look for reviews from both Amazon and Reddit due to their more constructive nature, instead of Twitter. However, do keep in mind that a product-level bias exists on different platforms and consume reviews cautiously.
- As a next step, it is recommended to expand the analysis scope to more products and try to improve the reliability of this project. Also, Textblob is a simplified language processing toolkit and this project did not test the accuracy of this library. There are other language processing libraries like Spacy or deep learning NLP models which can provide advanced capabilities such as higher accuracy in terms of sentiment and feature extraction. It is worth trying to compare other languages processing libraries and evaluate which one is best for this project.
- Further investigation of subjectivity and how it relates to polarity.[15]
- Deeper statistical analysis including interactive visualizations and web deployment.
- The uncertainty underlying all aspects of our research warrants further investigation with a larger sample size and scope to make more definite conclusions about bias in terms of sentiment.

Statement Of Work

We worked together on nearly every step of the project with multiple calls, chats, and even live pair-coding sessions together. In terms of focus, we divided the work largely into the following:

- **James Mete:** Twitter Processing, Sentiment & Uncertainty Analysis, Visualizations
- **Stanley Hung:** Amazon & Reddit Processing, Feature Extraction & Analysis

References

1. Loria, S. (2018). textblob Documentation. *Release 0.15*, *2*.
2. Kouzis-Loukas, D. (2016). *Learning Scrapy*. Packt Publishing Ltd.
3. Pushshift Reddit API. (2021), pushshift.io. Retrieved May 2021.
4. Archive Team Twitter Dataset. (2020), Archive.org. Retrieved May 2021.
5. Mete, J.. (2021), Twitter - Number of Minute Files. (Google Sheet).
6. Hung, S. (2021), Milestone1_Amazon_web_scraper.ipynb (Google Drive)
7. Hung, S. (2021), Milestone1_Reddit_Pushshift_API_access.ipynb (Google Drive)
8. Mete, J. (2021), CreateTwitterCSVs.ipynb (Google Drive)
9. Mete, J., Hung, S. (2021), 01-12.csv files. (Google Drive)
10. Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A. et al. (2016). Apache spark: a unified engine for big data processing. *Communications of the ACM*, *59*(11), 56–65.
11. Mete, J. (2021), Milestone1ProcessTweets.ipynb (Google Colab)
12. Mete, J., Hung, S. (2021), Milestone1-Analysis.ipynb (Google Colab)
13. Mete, J., Hung, S. (2021), Feature Polarity Density Charts.png (Google Drive)
14. Mete, J., Hung, S. (2021), All Comments - Radar Clean.png (Google Drive)
15. Mete, J., Hung, S. (2021), Polarity vs Subjectivity by Platform (Google Drive)
16. Mete, J., Hung, S. (2021), 100 Records File Samples (Google Drive)