iPhone 12 Tweet Sentiment Analysis Using Tweepy API

IS 684: Web Mining (Fall 2020)

Akshay Shirsat
Pace University

Harsh Modi
Pace University

Hiral Parmar
Pace University

Shreshth Rathod
Pace University

ABSTRACT

Data Science is evolving at an astonishing pace. We all remember the times where we would take feedbacks from users about a particular product, and based on the feedback, which would take days, we then made the necessary changes to improve our product. This, however, came with certain limitations. The feedbacks generally were limited to few hundreds. As the number of feedbacks from the users increases, and by the time all those feedbacks are analyzed, a lot of time would be wasted to study what most of the users like or dislike about the product. These times were not so long back that one will find it hard to recollect.

The field of Data Science and analytics since then has evolved so much, that it can do all of the above-mentioned process within a day. But can it do the same analysis with thousands of feedbacks? Maybe not thousands, but millions. But getting all those number of feedbacks will be difficult and time consuming? Maybe, if it was 90's, but since we are in 2020, it shouldn't take more than half a day or if you have a powerful machine, the time will further be reduced substantially. This project highlights at showing how one can gather such huge amount of user data related to a product and how efficiently can it analyze such overwhelming amount of data in seconds.

KEYWORDS

iphone, design, screen, camera, battery, charger, tweepy, LDA, NMF, ABSA

INTRODUCTION

iPhone 12 tweet analysis aims at analyzing tweets that users have made in response to the newly launched iPhone 12 by Apple company. The goal is to study what the users think about the product. Twitter, being a platform to opinions and experiences of users, and to communicate directly with concerned people, was chosen as a good source to extract information about iPhone 12.

Twitter is a huge platform, extracting tweets directly from twitter is not possible for security purposes and to maintain user privacy. To obtain tweets for this project, all the project members had created a developer's account which allotted a unique key and token pair for each account. While extracting the tweets, for 20000 tweets, we could extract only 5000-6000 tweets or at times run into session time out. For which we decided that each member will run the extraction code twice and all the tweets will be joined to form a single dataset. A dataset of 20000 unique tweets and its related information was built. All the tweets specific to the latest iPhone 12 product were targeted. After extracting tweets multiple times, it was observed that many of the tweets were not in English language. It was then decided to extract tweets which are in English language. English, since is a preferred language, was chosen for better understandability and to perform the required analysis.

Various models were used, some came with certain drawbacks, but finding out the right model that will suit best to the dataset was selected for final analysis. LDA and NMF unfortunately did not yield better results. So, ABSA model was chosen which gave us our final expected analysis.

TWITTER APPLICATION PROGRAMING INTERFACE (API)

As mentioned earlier, developer account was created to use twitter API to get access to all the tweets for web scrapping purpose. The developer's account gets approved by twitter in 2-3 business days. The function of the API is to take the user's request response to the application and sends application response back to the user. This facilitates interaction of user with the application for various use cases.

The twitter API allows the user to read specific tweets, for this project we scrapped tweets specific to iPhone 12 only. Along with the tweets, one can also scrape information about the user, like the username, location of the tweet, date of the tweet, likes and comments on the tweet, and etc. All this information can help us to perform better analysis for the selected product. The API is a great way to get all the views(tweets) of the user, in much more numbers, study it and reach a better conclusion using various models.

The twitter API, however, does come with certain limitations. We had created a developer's account, which is a free licensed version. The limitation of this version is that it will allow you to extract fewer tweets. Further, the tweets will be extracted only up to a particular date or for a period and not all the related tweets as expected. The paid version of the API will give full access to unlimited tweets and allow you to scrape data without any restrictions. Since the paid API is too expensive, that's why we chose to go with the free version for our project.

DATA SELECTION

Tweets related to iPhone 12 were scraped from Twitter using an open-source python package known as Tweepy API. Attributes extracted from the tweets are

- **tweet_id**: Unique number for each user
- **Username**: Username of the user
- **User_description**: Description of user
- **User_followers**: Number of followers per user
- **User_following**: Number of followings per user
- **User_location**: Location of user
- **Date_Time**: Date & time of the tweet
- **tweet**: Text of the tweet
- **Hashtags**: Hashtags extracted from the tweets
- **Likes**: Number of likes on the tweet
- **Retweet_count**: Number of retweets for the tweet
- **source**: Device from which the tweet was posted
- **tweet_place**: Location of the tweet when posted
- **tweet_is_quote_status**: Retweet with a comment
- **tweet_language**: Language of the tweet
- **tweet_coordinates**: Location coordinates of tweet

DATA WRANGLING

Hashtags were pre-processed in order to remove the attached index numbers while scrapping the tweets.

Pre-processing of tweets was conducted in various steps as mentioned below:

- Transformation of tweet to lower case.
- URL's, numbers and special characters were excluded.
- Expansion of English language contractions.
- Finally, tokenizing the tweets and applying lemmatization technique on it.

EXCLUSION CRITERIA

Due to readability constraint, tweets which were only in English language were considered. Moving further, tweets were considered before the product was launched for users and due to the limitations of open source Tweepy API we were able to scrape tweets within a range of 7 days only. Session time-out error occurred if we tried to scrape large number of tweets, so we had to scrape multiple times with a smaller number in order to obtain large number of tweets.

DESCRIPTIVE ANALYSIS

Firstly, top 3 sources from where tweets were generated were taken into consideration to observe the impact of sources amongst the tweets.

1. Sentiment Polarity Distribution from iPhone devices

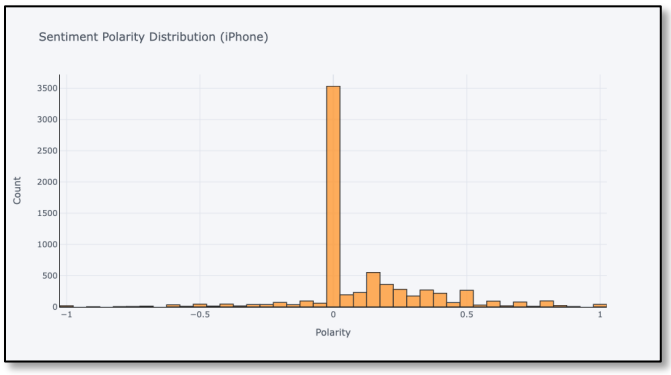

FIGURE 1

BAR PLOT OF POLARITY DISTRIBUTION VERSUS NUMBER OF TWEETS FROM IPHONE DEVICES.

- NUMBER OF NEUTRAL TWEETS IS 17.27% (3468/20047) OF THE TOTAL DATASET.
- NUMBER OF POSITIVE TWEETS IS 15.17% (3047/20047) OF THE TOTAL DATASET.
- NUMBER OF NEGATIVE TWEETS IS 2.90% (584/20047) OF THE TOTAL DATASET.

2. Sentiment Polarity Distribution from Android devices

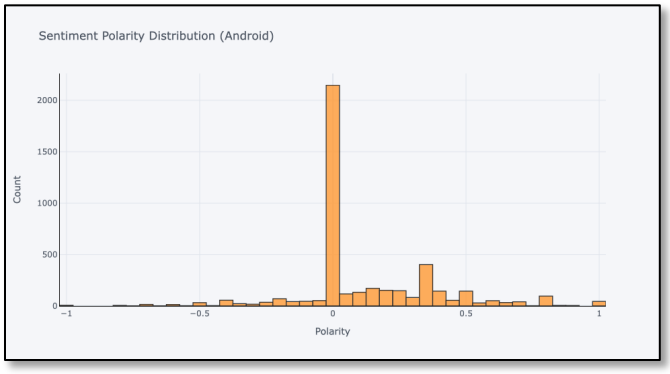

FIGURE 2

BAR PLOT OF POLARITY DISTRIBUTION VERSUS NUMBER OF TWEETS FROM ANDROID DEVICES.

- NUMBER OF NEUTRAL TWEETS IS 10.61% (2132/20047) OF THE TOTAL DATASET.
- NUMBER OF POSITIVE TWEETS IS 9.25% (1859/20047) OF THE TOTAL DATASET.
- NUMBER OF NEGATIVE TWEETS IS 2.12% (427/20047) OF THE TOTAL DATASET.

3. Sentiment Polarity Distribution from Web App

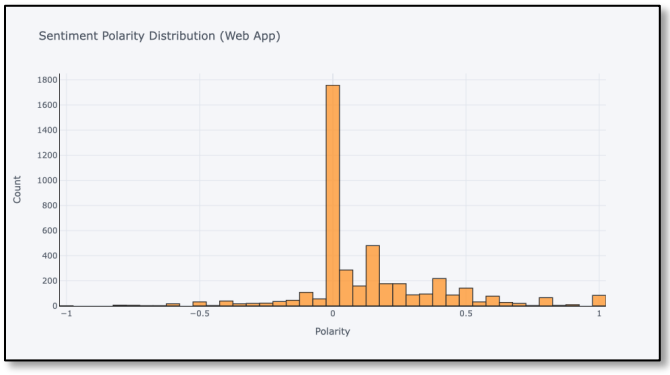

FIGURE 3

BAR PLOT OF POLARITY DISTRIBUTION VERSUS NUMBER OF TWEETS FROM WEB APP.

- NUMBER OF NEUTRAL TWEETS IS 8.49% (1706/20047) OF THE TOTAL DATASET.
- NUMBER OF POSITIVE TWEETS IS 11.24% (2257/20047) OF THE TOTAL DATASET.
- NUMBER OF NEGATIVE TWEETS IS 2.18% (439/20047) OF THE TOTAL DATASET.

EXPLORATORY ANALYSIS

1. Hashtags WordCloud

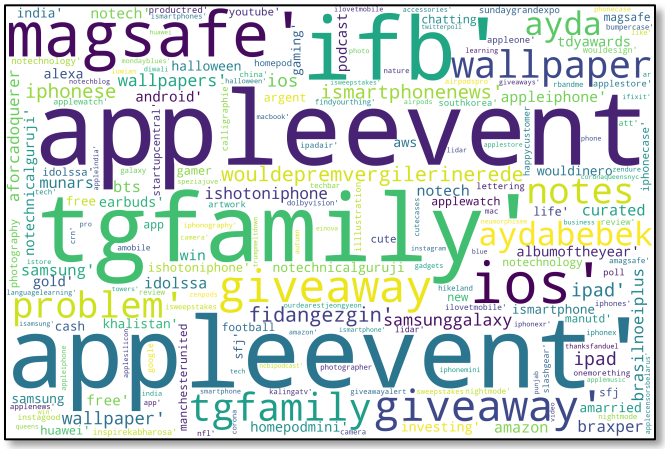

FIGURE 4

WORDCLOUD FOR POPULAR HASHTAGS IN THE TWEETS

2. User Statistics for Number of followers of each user

```
###90-100 percentile
for i in range(0,11):
    print(90+i, 'percentile value is', np.percentile(dataframe['User_followers'], 90+i))
```

```
90 percentile value is 15909.800000000007
91 percentile value is 18170.440000000002
92 percentile value is 21631.0
93 percentile value is 25688.0
94 percentile value is 29175.96
95 percentile value is 38113.8
96 percentile value is 55662.19999999962
97 percentile value is 96904.92
98 percentile value is 508032.83999999816
99 percentile value is 605976.3200000001
100 percentile value is 10557745.0

90% users having 15909 followers or fewer followers
```

FIGURE 5

- 90% OF USERS HAVE 15909 FOLLOWERS OR LESS.
- HIGHER THE FOLLOWERS, MORE WILL BE THE REACH OF THE TWEETS TO OTHER USERS.

3. User Statistics for Number of followings by each user

```
###90-100 percentile
for i in range(0,11):
    print(90+i, 'percentile value is', np.percentile(dataframe['User_following'], 90+i))
```

```
90 percentile value is 2707.600000000046
91 percentile value is 3066.0
92 percentile value is 3359.0
93 percentile value is 3572.1200000000026
94 percentile value is 4095.9199999999983
95 percentile value is 4733.399999999998
96 percentile value is 5044.0
97 percentile value is 7019.0
98 percentile value is 9202.0
99 percentile value is 12357.48
100 percentile value is 405977.0

90% users are having 2707 or fewer followings
```

FIGURE 6

- 90% OF USERS HAVE 2707 FOLLOWING OR LESS.
- HIGHER THE FOLLOWING, MORE WILL BE EXPOSURE TO TWEETS ON IPHONE 12 BY OTHER USERS, AND THUS MORE WILL BE THE CHANCES OF RETWEETING.

4. Source of Tweets

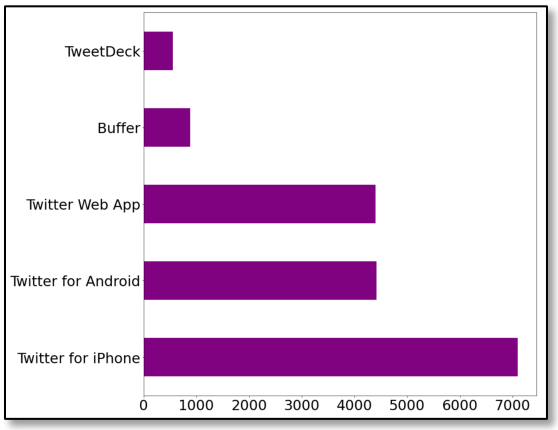

FIGURE 7

COUNT PLOT OF NUMBER OF TWEETS FROM VARIOUS SOURCES

- THESE ARE THE TOP 5 PLATFORMS FROM WHICH THE TWEETS WERE GENERATED
- IT CAN BE OBSERVED THAT MOST OF THE TWEETS WERE GENERATED FROM IPHONE

5. Overall Polarity of the Tweets

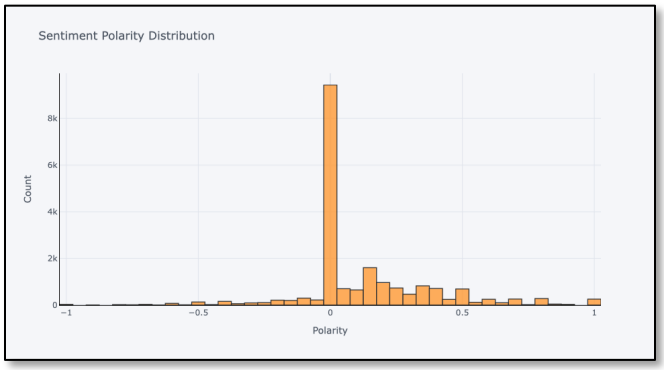

FIGURE 8

BAR PLOT FOR POLARITY OF TWEETS

- THIS IS THE GENERAL POLARITY OF ALL THE TWEETS FROM ALL THE SOURCES

6. Word Count of Tweets

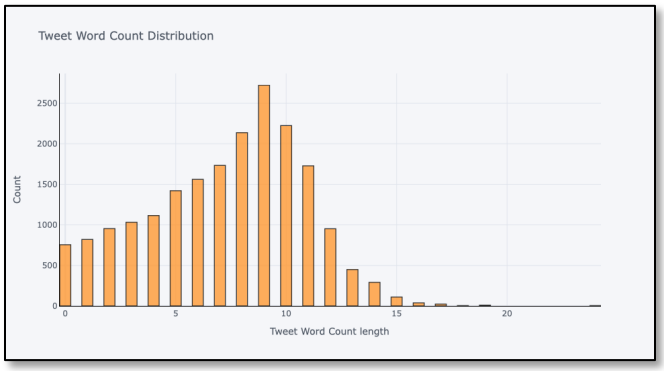

FIGURE 10

BAR PLOT FOR WORD COUNT OF TWEETS

- MOST OF THE TWEETS' LENGTH LIES BETWEEN 7 & 11.
- AS A RESULT, THESE TWEETS DO NOT YIELD GOOD RESULTS WHEN FED INTO THE MODELS.

7. WordCloud for Top 100 Positive words

FIGURE 11
WORDCLOUD FOR TOP 100 POSITIVE WORDS

8. WordCloud for Top 100 Negative words

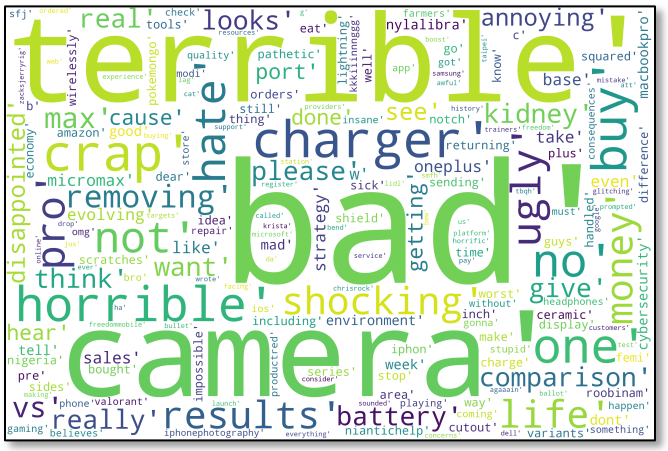

FIGURE 12
WORDCLOUD FOR TOP 100 NEGATIVE WORDS

MODEL CREATION

1. Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. LDA has made a big impact in the fields of natural language processing and statistical machine learning and has quickly become one of the most popular probabilistic text modelling techniques in machine learning.

Intuitively in LDA, documents exhibit multiple topics. In text pre-processing, we exclude punctuation and stop words (such as, “if”, “the”, or “on”, which contain little topical content). Therefore, each document is regarded as a mixture of corpus-wide topics. A topic is a distribution over a fixed vocabulary. These topics are generated from the collection of documents. For example, the sports topic has word "football", "hockey" with high probability and the computer topic has word “data”, “network” with high probability. Then, a collection of documents has probability distribution over topics, where each word is regarded as drawn from one of those topics. With this document probability distribution over each topic, we will know how much each topic is involved in a document, meaning which topics a document is mainly talking about.

	Topic 0 words	Topic 0 weights	Topic 1 words	Topic 1 weights	Topic 2 words	Topic 2 weights	Topic 3 words	Topic 3 weights	Topic 4 words	Topic 4 weights	Topic 5 words	Topic 5 weights
0	could	394.1	series	495.9	blue	349.1	buy	606.6	ios	452.1	technicalguruji	569.1
1	away	375.1	glass	446.4	go	329.2	not	604.9	via	446.0	charger	491.1
2	enter	364.5	camera	391.0	available	296.9	screen	368.7	review	428.9	tgfamily	298.1
3	giving	336.1	not	301.6	video	273.3	know	247.6	youtube	319.3	giveaway	268.0
4	tweet	333.1	dome	285.1	order	255.3	people	211.6	appleevent	246.6	magsafe	242.7
5	supcase	303.1	every	218.1	vs	251.4	use	211.0	latest	188.6	time	237.8
6	increase	293.1	better	196.3	pre	245.5	available	209.7	case	184.0	battery	229.4
7	atl	285.1	free	178.6	today	200.4	store	202.5	wallpapers	173.1	one	211.2
8	wins	281.1	product	168.2	brand	194.2	cases	184.7	check	172.8	device	206.3
9	everyday	280.1	finally	138.9	unboxing	170.5	protector	175.1	start	151.0	really	157.6

FIGURE 13
TOPICS EXTRACTED FROM LDA MODEL

2. Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) is an unsupervised technique so there is no labeling of topics that the model will be trained on. The way it works is that NMF decomposes (or factorizes) high-dimensional vectors into a lower-dimensional representation. These lower-dimensional vectors are non-negative which also means their coefficients are non-negative.

	Topic 0 words	Topic 0 weights	Topic 1 words	Topic 1 weights	Topic 2 words	Topic 2 weights	Topic 3 words	Topic 3 weights	Topic 4 words	Topic 4 weights	Topic 5 words	Topic 5 weights
0	win	3.7	away	2.9	glass	6.2	gt	6.5	amp	8.7	follow	3.3
1	price	3.6	enter	2.9	dome	4.0	lt	4.2	life	0.8	something	3.0
2	shop	3.5	could	2.9	series	3.6	samsunggalaxy	1.0	like	0.8	different	3.0
3	endless	3.4	giving	2.8	camera	2.0	ipad	1.0	pretty	0.7	retweets	2.8
4	found	3.4	tweet	2.8	every	1.7	page	1.0	dark	0.7	likes	2.8
5	slash	3.4	supcase	2.8	product	1.7	top	0.9	podcast	0.7	completely	2.8
6	fantastic	3.4	atl	2.8	ez	1.6	bio	0.9	parents	0.7	ifb	2.8
7	bargain	3.4	wins	2.8	protector	1.6	ios	0.9	neil	0.7	seen	1.1
8	wonders	3.3	everyday	2.7	brand	1.3	public	0.8	married	0.7	mu	0.9
9	free	0.1	increase	2.7	hot	1.1	download	0.7	chatting	0.7	like	0.2

FIGURE 14
TOPICS EXTRACTED FROM NMF MODEL

Using the original matrix (A), NMF will give you two matrices (W and H). W is the topics it found, and H is the coefficients (weights) for those topics. In other words, A is articles by words (original), H is articles by topics and W is topics by words.

3. Aspect Based Sentiment Analysis

Aspect-based sentiment analysis (ABSA) aims at identifying the sentiment polarity towards the specific aspect in a sentence. A target aspect refers to a word or a phrase describing an aspect of an entity. For example, in the sentence “The decor is not special at all, but their amazing food makes up for it”, there are two aspect

terms “decor” and “food”, and they are associated with negative and positive sentiment respectively.

Noun	Adjective	Polarity	Subjectivity
iphone	quick, amazing, powerful, awesome	0.2177	0.5233
design	ancestral, reminiscent, familiar, rugged	0.3266	0.6244
screen	aesthetic, strong, oled, beautiful	0.3349	0.7195
camera	stellar, elite, clear, dslr, wide	0.1189	0.5417
battery	new, decent, efficient, improved	0.1475	0.5064
charger	wireless, unique, expensive, fast	0.1149	0.5118

TABLE 1

Noun – Adjective pair extracted using ABSA Model

- These adjectives represent the sentiments for the associated nouns
- The polarity represents the nature of adjectives to the nouns
- The subjectivity lies between [0,1], as it tends towards 1 it is more of an opinion than factual information.

CONCLUSION

It was observed post analysis that majority of the tweets extracted were either positive or neutral. In explanation to this, tweets were scrapped before the product was launched publicly. So, it might be that people were positive about the new product and expecting that the product will be good.

We believe that most of the tweets are positive and neutral, and this might change once the users get a hands-on experience and be able to relate to the new features and its practical use. If it doesn’t meet the expectations of the user, they might tweet.

REFERENCES

1. Analyze the Sentiment of Tweets from Twitter Data and Tweepy in Python. https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/analyze-tweet-sentiment-in-python/
2. Building a Twitter Sentiment Analysis in Python. https://www.pluralsight.com/guides/building-a-twitter-sentiment-analysis-in-python
3. Twitter Sentiment Analysis with Machine Learning. https://monkeylearn.com/blog/sentiment-analysis-of-twitter/
4. How to Do Sentiment Analysis on a Twitter Account. https://medium.com/better-programming/twitter-sentiment-analysis-15d8892c0082
5. Latent Dirichlet Allocation https://ai.stanford.edu/~ang/papers/jair03-lda.pdf https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
6. LDA Topic Modeling: An Explanation. https://towardsdatascience.com/lda-topic-modeling-an-explanation-e184c90aadcd
7. Topic Modeling Articles with NMF https://towardsdatascience.com/topic-modeling-articles-with-nmf-8c6b2a227a45
8. Implementing Aspect Based Sentiment Analysis using Python. https://medium.com/analytics-vidhya/aspect-based-sentiment-analysis-a-practical-approach-8f51029bbc4a
9. Aspect-Based Opinion Mining (NLP with Python) https://medium.com/@pmin91/aspect-based-opinion-mining-nlp-with-python-a53eb4752800