Project Title: Sentiment Analysis of New Apple Products

Team Members: Shannon Tee, Tianze Lyu, Zhejie Cui

Problem Definition

Apple announced its next generation of iPhone and Apple Watch products on September 12, 2018. To gauge the population's response to these products, Twitter data can be utilized to determine if the sentiment was positive, neutral or negative. At a high-level, sentiment analysis was performed on tweets related to the new Apple products to evaluate the overall attitude and opinion towards these products. Next, more in-depth sentiment analysis was conducted to analyze the sentiment for each product to hopefully answer the following sub-questions:

- What is the sentiment for each individual product? Was there one product that got a more positive response than the rest?
- What was the sentiment trend in the month following the release for each product? Did it slowly become more positive or negative? Were there any spikes?

The products analyzed were iPhone XS, iPhone XS Max, and Apple Watch Series 4. The iPhone XR was excluded as its pre-order and shipping date differed from the other three products. The three products in scope for this project were expected to ship out on September 21, 2018. By performing sentiment analysis on Twitter data, the objective was to examine the sentiment for new Apple products launched in September 2018.

Description of Background

Sentiment Analysis allows us to identify and extract opinions from messages by calculating the polarity to determine if the message was favorable, unfavorable, or neutral, and to what degree. When performing it on Twitter data, it provides the ability to extract insights from social data to demonstrate the true opinion that the population has. This can be valuable for a company in many ways such as identifying problems that customers are having, getting feedback on new products, and determining customer satisfaction towards the company. With this knowledge, companies can quickly understand customers' opinions and react accordingly. For example, if the sentiment analysis demonstrated that the population had a negative response due to a bug in a new product, the company could respond by immediately fixing the issue to minimize the negative impact it has on their customers. Furthermore, understanding their customers' opinions allows companies to strategically make decisions and plan for future products. More specifically to our project, if the results demonstrated that one Apple product had a significantly more negative sentiment than the other products, Apple could decide to focus their efforts on that specific product to figure out why the population has a negative opinion towards it and how to improve it for future releases. Overall, sentiment analysis can be really beneficial for a company to identify what their customers are thinking, wanting, and needing, and to use those insights to determine what steps to take next.

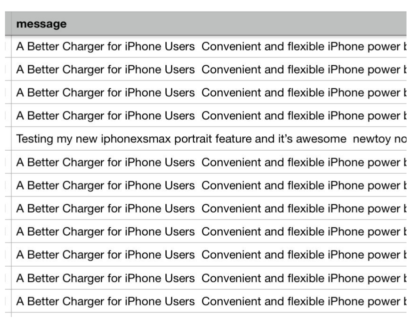
Description of Dataset

Using the Tweepy API, data was collected from October 1 to October 31, 2018. 2000 English tweets were collected each day for each product, and retweets were filtered away before storing the data into CSV files by removing tweets that started with 'RT'. While collecting data, the Twitter Development documentation was referenced to change the message collection to extended mode to ensure the message would not be cut off at 140 characters [1]. The tweet data stored was: message, date, tweet ID, user screen name, language, source, location, follower count, friend count, retweeted, and retweeted count. A 15" Macbook Pro (2018) with a 2.8 GHz Intel Core i7 CPU and 16GB main memory size was used to collect the tweets. The performance was approximately 7 seconds to collect 2000 tweets. At the end of the data collection stage, there was a total of 74,692 tweets stored in our CSV files. From an initial glance, there were some inconsistencies in our data which could affect the quality of the results. Some of these include: repetitive tweets, giveaway messages, and irrelevant messages (such as those that only contained the words "apple" or "iPhone"). The dataset was cleaned using various methods described in the next section to improve its quality before performing sentiment analysis. The location field was also inconsistent or blank for a large portion of the tweets. Therefore, this field was not used during analysis as it may not have produced the best quality results as it would not have been an accurate representation of all the tweets. All of the data in CSV files can be found here: https://drive.google.com/drive/u/0/folders/1Or9lLZuG_oUq6AzYoAd9UOTEp0bVFBOm

Description of Method Used

Once the data collection was complete, all the data was imported from the CSV files into a pandas dataframe to be processed. All data cleaning and analysis was done using code written in

Python. With the help of the article, “This Is How Twitter Sees The World”, basic data cleaning was performed such as removing punctuation, special characters, emojis, and weblinks as well as converting the message to lowercase and doing lemmatization [2]. Originally, the plan was also to perform stemming and handle negations. However, after conducting stemming using the tutorial at Python Programming, important words necessary for our analysis were negatively affected such as “iphone” and “apple” were changed to “iphon” and “appl” [2]. After initial analysis on negations, there was only 20 “not good” and 10 “not bad” phrases inside all messages, demonstrating that this data cleaning step would have minimal impact on the final results. Therefore, stemming and negation handling were both excluded from data cleaning. In addition to basic cleaning, 6902 tweets related to promotions were filtered which were any messages that started with “Win” or contained “giveaway”. These tweets contributed to 9.2% of the total messages and could impact the final sentiment analysis results as these words tend to have more of positive meaning to them. There were also repetitive rows of tweets that could have resulted from companies or individuals creating multiple accounts to tweet the same message. An example of this can be seen in the image to the left. 2704 rows of repetitive messages (3.6% of the total messages) were removed to ensure these repeats would not cause the results to be more positive or negative. The date of the message was also broken down into month and day columns to be used in the analysis stage. The message was searched for key phrases to associate each tweet to a product. Note that a tweet could be associated with more than one product if multiple products were mentioned in the tweet. The association is as follows:



message
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t
Testing my new iphonexsmax portrait feature and it's awesome newtoy no
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t
A Better Charger for iPhone Users Convenient and flexible iPhone power t

- **Apple Watch:** any message that contained “applewatch4”, “apple watch 4”, “apple watch series 4”, “applewatchseries4”, “applewatch 4”, “apple watch4”, “applewatch series 4”, or “apple watchseries4”
- **iPhone XS:** any message that contained “iphonexs” or “iphone xs” and did not have “max” as the next word
- **iPhone XS Max:** any message that contained “iphone xs max”, “iphonexs max”, “iphonexsmax”, “iphone xsmax”, “xs max”, or “xsmax”

This step would filter out irrelevant tweets such as those that only had “iphone” or “watch”. During this stage, 37452 tweets were dropped, resulting in 27,634 tweets.

Next, the TextBlob library was used to determine the sentiment of each tweet where the polarity was stored as a new column in the pandas dataframe. A 15” Macbook Pro (2018) with a 2.8 GHz Intel Core i7 CPU and 16GB main memory size was used to perform data cleaning and sentiment analysis. The stage with the longest performance was data cleaning which took 9 minutes and 36 seconds to perform, while determining polarity using TextBlob took 14 seconds to perform.

After sentiment analysis was complete, a dictionary was created with each word as the key and the frequency it occured as the value. Words related to the product name were excluded from this dictionary such as apple, watch, iphonexs, iphonexsmax, iphone, xs, and max. Referencing a tutorial from DataCamp, the WordCloud library in Python was used to create a visualization where the most frequent words appear larger, and the less frequent words appear smaller [4].

Basic analysis such as counts and percentages were performed on the final dataset before exporting the pandas dataframe to a CSV file, which was then used to perform the visualization in Excel and Tableau.

Experiment: Analysis Results

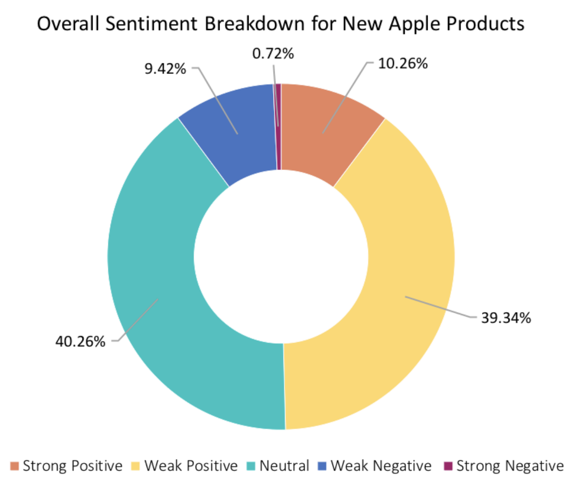
To begin, the overall sentiment analysis for the new Apple products was analyzed as well as the sentiment for each product. The messages were broken down into 5 groups based on its polarity which were:

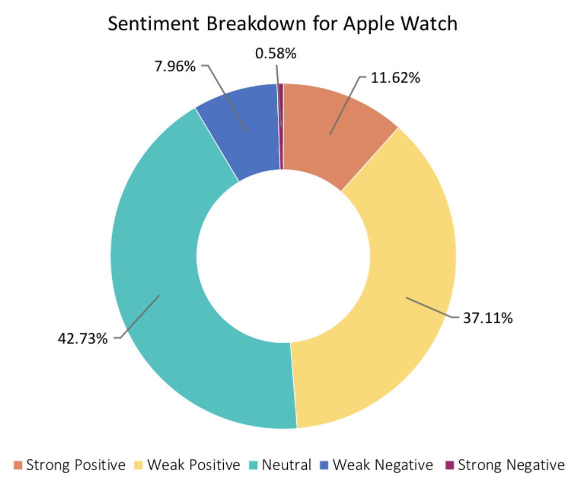
- **Strong Positive:** Polarity > 0.5
- **Weak Positive:** 0.5 >= Polarity > 0
- **Neutral:** Polarity = 0
- **Weak Negative:** 0 > Polarity >= -0.5
- **Strong Negative:** -0.5 > Polarity

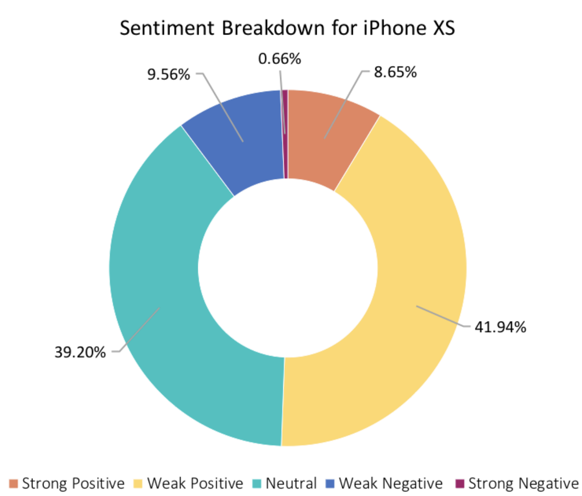
The breakdown of the number of tweets that fell into each group and for each product can be found in the table below:

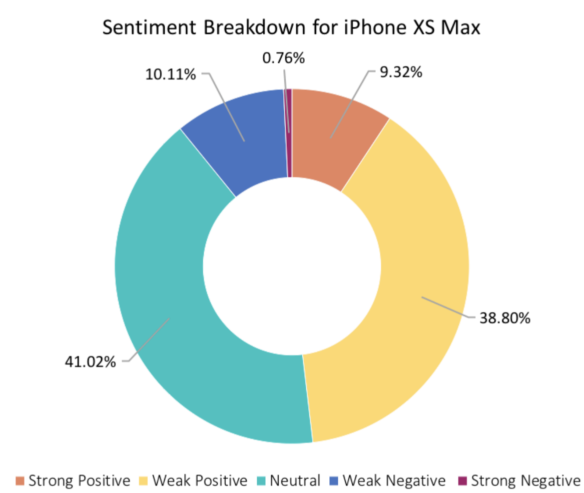
	Final # Tweets	Strong Positive	Weak Positive	Neutral	Weak Negative	Strong Negative
Overall	27,634	2,835	10,870	11,126	2,603	200
Apple Watch	10,339	1,201	3,837	4,418	823	60
iPhone XS	9,991	864	4,190	3,916	955	66
iPhone XS Max	12,946	1,206	5,023	5,310	1,309	98

Since it was difficult to compare the sentiment between different products due to the varying number of total tweets, the percentage of tweets for each group was calculated using the numbers above. The overall sentiment breakdown for new Apple products and the sentiment breakdown for each product is shown in the four pie charts below. These charts were created in Excel.

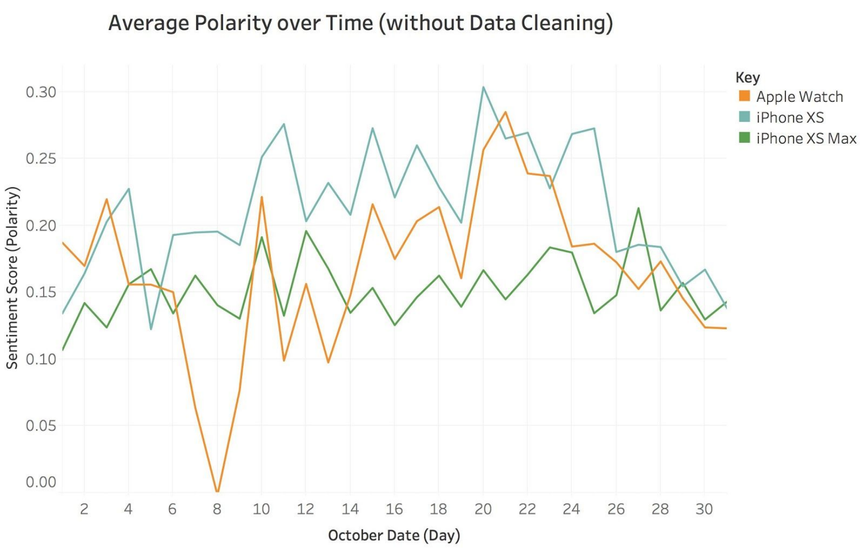


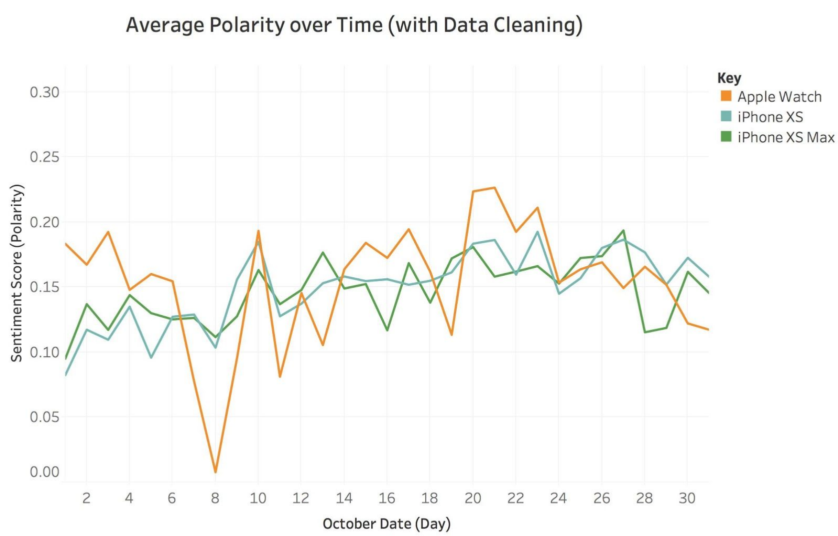






By looking at the charts, the sentiment breakdown was relatively similar for the overall new Apple products and for each Apple products. Majority of the tweets were identified as weak positive or neutral, meaning that most messages had polarity between 0 to 0.5.





Next, in Tableau, the sentiment trend for each product over the month of October was analyzed using a line graph of the average polarity on each day. The plot was also recreated using the same dataset that did not have any cleaning to demonstrate the impact that data cleaning had on the sentiment trend. The sentiment trend demonstrates that the average polarity for each product was positive for the entire month. While the average did increase and decrease a bit over the days, it remained relatively consistent except for the sharp drop in the Apple Watch on October 8. Furthermore, data cleaning did have a large impact on the sentiment trend during the month as data cleaning made the average polarity and trend a lot more similar across the 3 products. The reason for this is likely that there were many giveaway, repetitive, or irrelevant tweets in the dataset prior to performing data cleaning.

Lastly, a WordCloud was created to look at the most frequent words in positive and negative tweets. The WordCloud for positive polarity messages is as shown to the left. For the most part, there were quite a number of positive words that had high frequency such as best, love, better like/liked, worth, and great. The

Word Cloud created for negative polarity messages can be seen on the right. There was not too many negative words, but some that stood out were bug, crashing, hard, and rebooting. Through the analysis conducted using WordCloud, Excel, and Tableau, the visualizations produced helped better understand the population's sentiment for the new Apple products as well as how it changed over time.

Observation and Conclusion

Based on the sentiment breakdown, the overall response to the new Apple products was relatively positive as the majority of the tweets had a weak positive or neutral polarity. Looking at the sentiment breakdown for each product, the percentage of positive, neutral, and negative responses was approximately the same for each product. This demonstrates that there was not one product that had a more positive or negative response than the rest. Surprisingly, the percentage of negative messages was pretty small. This was also reflected in the WordCloud generated from the negative tweets as the most frequent words in those messages did not have a negative meaning such as case, savings, black, new, time, and daylight. While bug and crashing were two obvious negative words in the WordCloud, these are simple fixes that Apple could immediately address to improve customer satisfaction. This also shows the importance of sentiment analysis as it allows the company to identify the issues that their customers are having and respond quickly to minimize the negative impact. In contrast, the most frequent words in positive tweets demonstrated that the population had a good opinion towards the products through words like best, love, better, and like/liked. Camera, screen, and feature also appeared quite frequent in positive messages which could signify that the improved camera and screen are features that customers are enjoying. The sentiment trend for each product shows it was consistently positive throughout the month aside from a dip on October 8th for Apple Watch 4. Apple might find it beneficial to investigate this dip as the cause further potentially could be from a problem customers were having, bad news on the media, the release of another competitors product, or just an outlier. However, digging deeper into its cause could provide Apple with insights about future steps or decisions they should take. This sentiment analysis provided valuable insights that the population had an overall positive/neutral response towards the new Apple products, and demonstrated that there was not one product Apple needed to focus on improving. Furthermore, it provided more detailed feedback to what features the population liked as well as problematic issues that they were having with the products.

References

[1] "Introduction to Tweet JSON - Twitter Developers." Twitter Developers, Twitter, developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json.html.

[2] Wahome, Ronald. "This Is How Twitter Sees The World : Sentiment Analysis Part One." Towards Data Science, 7 Sept. 2018, towardsdatascience.com/the-real-world-as-seen-on-twitter-sentiment-analysis-part-one-5ac2d06b63fb.

[3] "Stemming Words with NLTK." Python Programming Tutorials, pythonprogramming.net/stemming-nltk-tutorial/.

[4] Vu, Duong. "Generating WordClouds in Python." DataCamp, 7 Aug. 2018, www.datacamp.com/community/tutorials/wordcloud-python

Appendix

Function in Code	Input	Runtime Complexity
Data Collection		
Data collection and write to CSV files	n lines of strings in total, which are stored in 93 CSV files, respectively	O(n)
Data Cleaning and Sentiment Analysis		
Iterating through all CSV files and store in Pandas dataframe	n lines of strings	O(n)
Delete messages that are related to giveway	n lines of strings constant m characters in each line	O(m*n)
Remove weblinks	n lines of strings constant m characters in each line	O(m*n)
Delete repetitive messages	n lines of strings constant m characters in each line	O(m*n)
Split message date into month/day columns	n lines of strings constant k characters in each line	O(k*n)
Process message to perform data cleaning	n lines of strings constant k characters in each line.	O(k*n)
Identify what product each message associated to	n lines of strings constant k characters in each line	O(k*n)
Drop rows not associated to a product	3 * n strings, each line stores "True" or "False"	O(n)
Determine polarity of each message	n line of strings	O(n)
Data Processing		
Sentiment analysis	n lines of strings, constant u number of words in each line	O((u^2) * n)
Visualization		
Remove special words	n lines of strings constant k characters in each line	O(k*n)
Append (key: word, value: frequency) pair into dictionary	x number of unique words in total Append: O(x)	O(x)
Sort the dictionary based on word frequency	w number of unique words (w ≤ x)	O(w*log(w))
Create word cloud	x number of unique words	O(x)

The table above shows runtime complexities of each functions we used in the data collection, data cleaning, data processing and visualization. Most functions are linear runtime complexity, only sort function takes $O(x*\log(x))$ runtime complexity. Those functions run sequentially, so the total runtime complexity is $O(c*n + w*\log(w))$, where c is a constant number, n is the total number of lines and w is total number of unique words.