Abby Liu

Final Project Report:
Twitter Sentiment Analysis for Product Analytics

Problem Statement

Apple Inc. recently launched a new accessory called AirTag that can find physical items like bags, wallets and keys. The company wants to gain insights on the general perception of the product and understand how they can improve their product by increasing their strengths and reducing their weaknesses.

By doing a Twitter sentiment analysis, we can monitor emotions in 5000 past conversations regarding the product and its performance on the social media platform Twitter, and we will extract useful information from customer feedback to make more informed decisions in their marketing strategies and campaigns. It will help answering the following questions:

- What do customers think of Apple's new product AirTag?
- How can we use these comments/feedback to improve this product and marketing strategies as well?

Data

The dataset used for the analyses in this project is a set of tweets in English that are pulled from the Twitter API regarding consumer impressions of AirTag that were posted in the 2nd week(May 6th - May 12th) after AirTag released on April 30th, 2021. Features include the date posted, text content, and location of each tweet. Retweets are excluded.

Data Wrangling

The raw dataset contains 5000 rows and 3 columns. I first checked if there were any duplicates and dropped 18 duplicated tweets, then I had 4982 unique tweets left. Next, I performed some data cleaning to my tweet texts, which included removing irrelevant words like "b'" in almost all tweets, removing names mentioned, links, unwanted UTF-8 (Bytes) emoji code, and punctuations in tweets. The cleaned texts were saved to a new column named 'cleaned_tweet', and the original 'tweet' column got dropped. Lastly, I converted all text to lowercase. I also converted the column 'date' to datetime objects and only kept the dates without time for further analysis. I decided to drop the column 'location' after discovering that there were too many missing locations or fake locations such as "everywhere" and "in your couch" in this column.

At this point, I have 4982 rows and 2 columns left in our cleaned data.

Exploratory Data Analysis

Some EDA was done on the cleaned dataset, which consists of the features ‘date’ and ‘cleaned_tweet’.

a) Sentiment Analysis

I first used TextBlob to process my textual data and assign polarity score and sentiment label to each tweet. By creating a pie chart, I could then discover the general perception of AirTag.

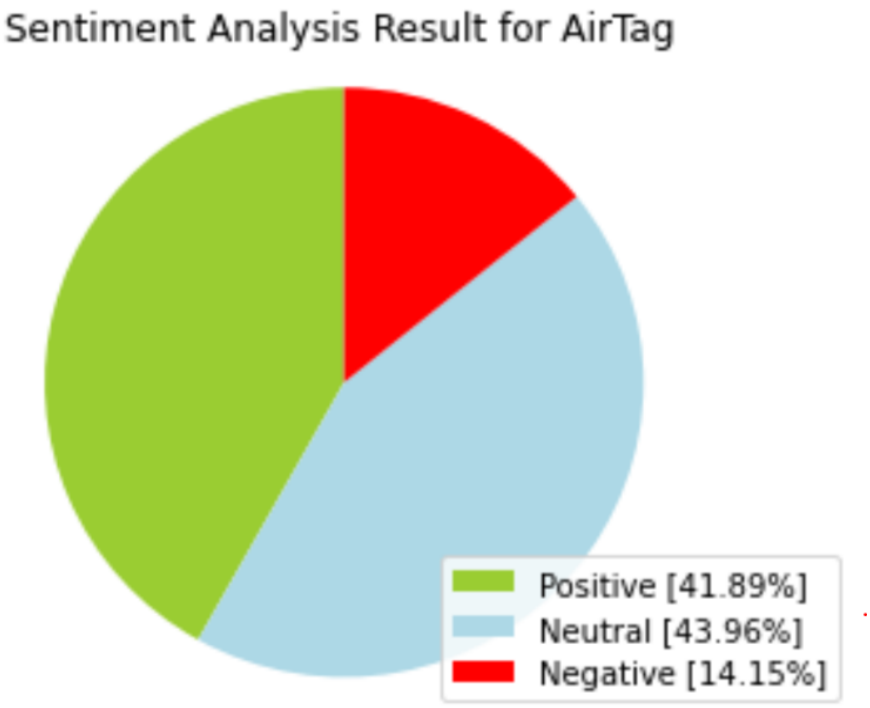

Figure 1: Pie chart for sentiment components

With 41.89% of positive tweets, 43.96% of neutral tweets, and only 14.15% of negative tweets, we can see that the general perception of AirTag was not bad at all.

b) Keywords by Sentiments

To find out what people mentioned the most in positive tweets and negative tweets, respectively, I've generated a word cloud for each category.

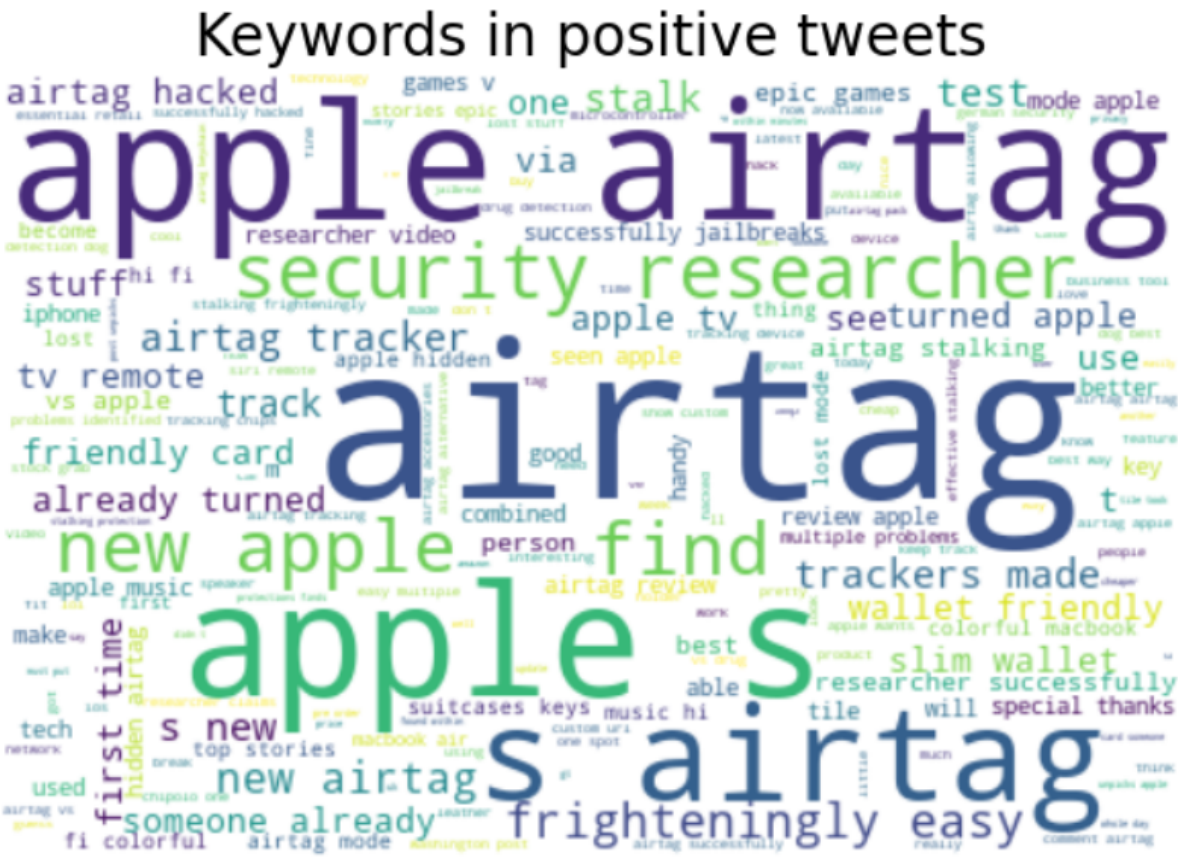

Figure 2: Word cloud for positive tweets

According to the word cloud above, some common words for tweets with positive sentiment include but not limited to "wallet friendly", "security researcher", "new", "friendly card", "frightening easy", and "find".

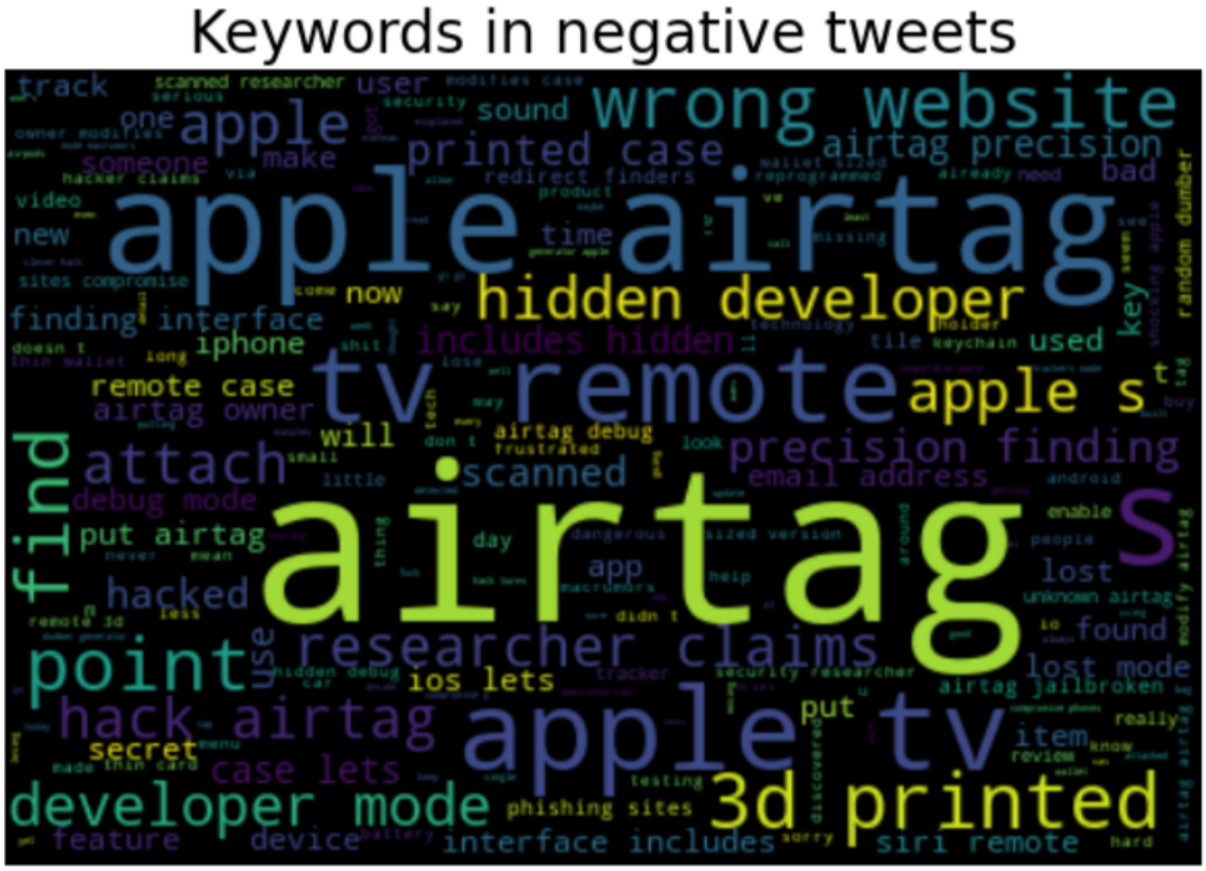

Figure 3: Word cloud for negative tweets

From figure 3, the word cloud for tweets with negative sentiment, we can spot some words like "wrong website", "hidden developer", "3d printed", "hack airtag", and "tv remote". It's not easy to tell what's going on from these keywords, but we can dive deeper into it later by looking at some example tweets.

c) Tweet Length by Sentiment

Let’s look at the density of words used in tweets based on different sentiment by calculating the word counts of tweets.

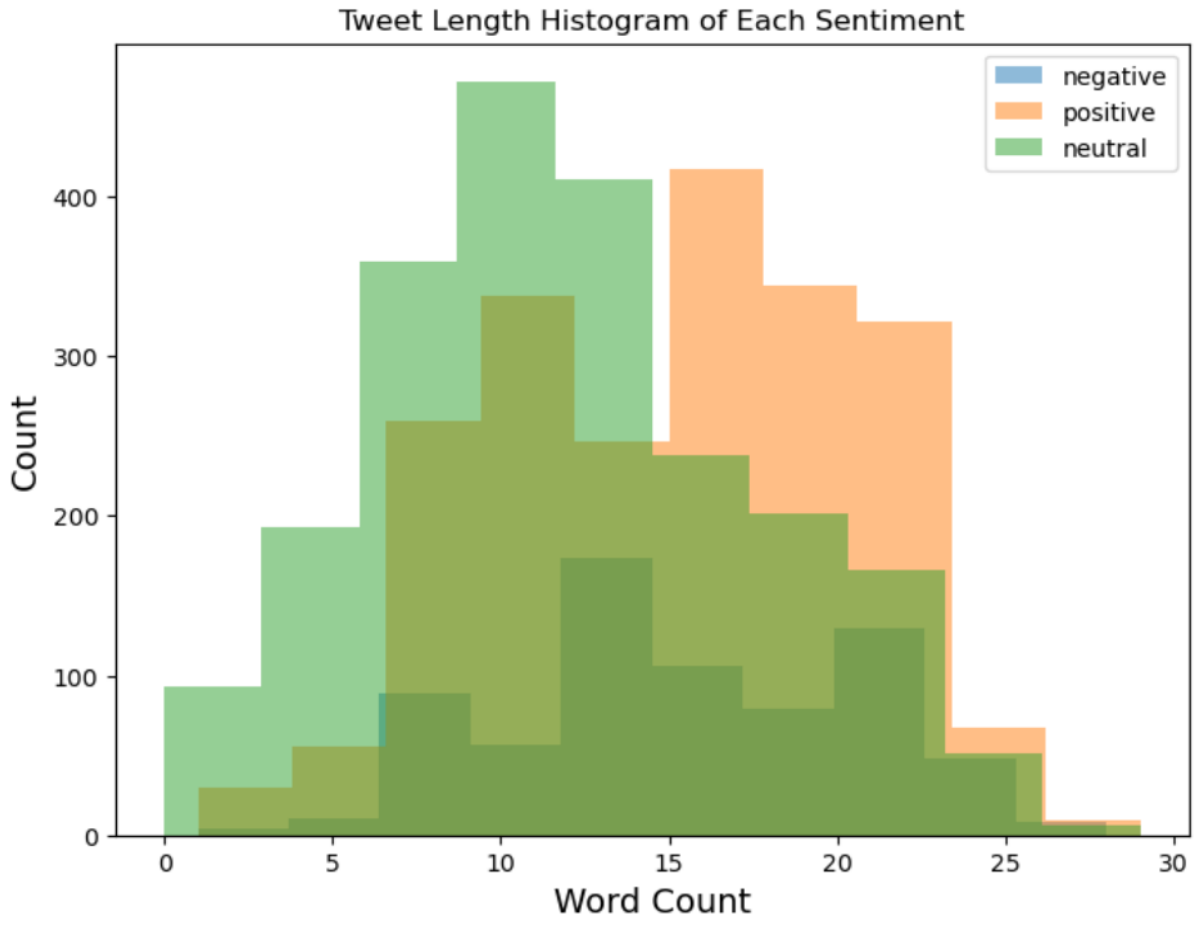

Figure 4: Histogram for tweet lengths varied by sentiments

From figure 4, it seems like neutral tweets are usually shorter, and the tweets with positive or negative emotion are generally longer. It's probably because people tend to have more to say or use more expressions when they have a stronger reaction to the product.

d) Emotion Trends

To determine how the sentiment of AirTag changed overtime, I created a line chart showing how the polarity score changed during the week of 5/6 - 5/12.

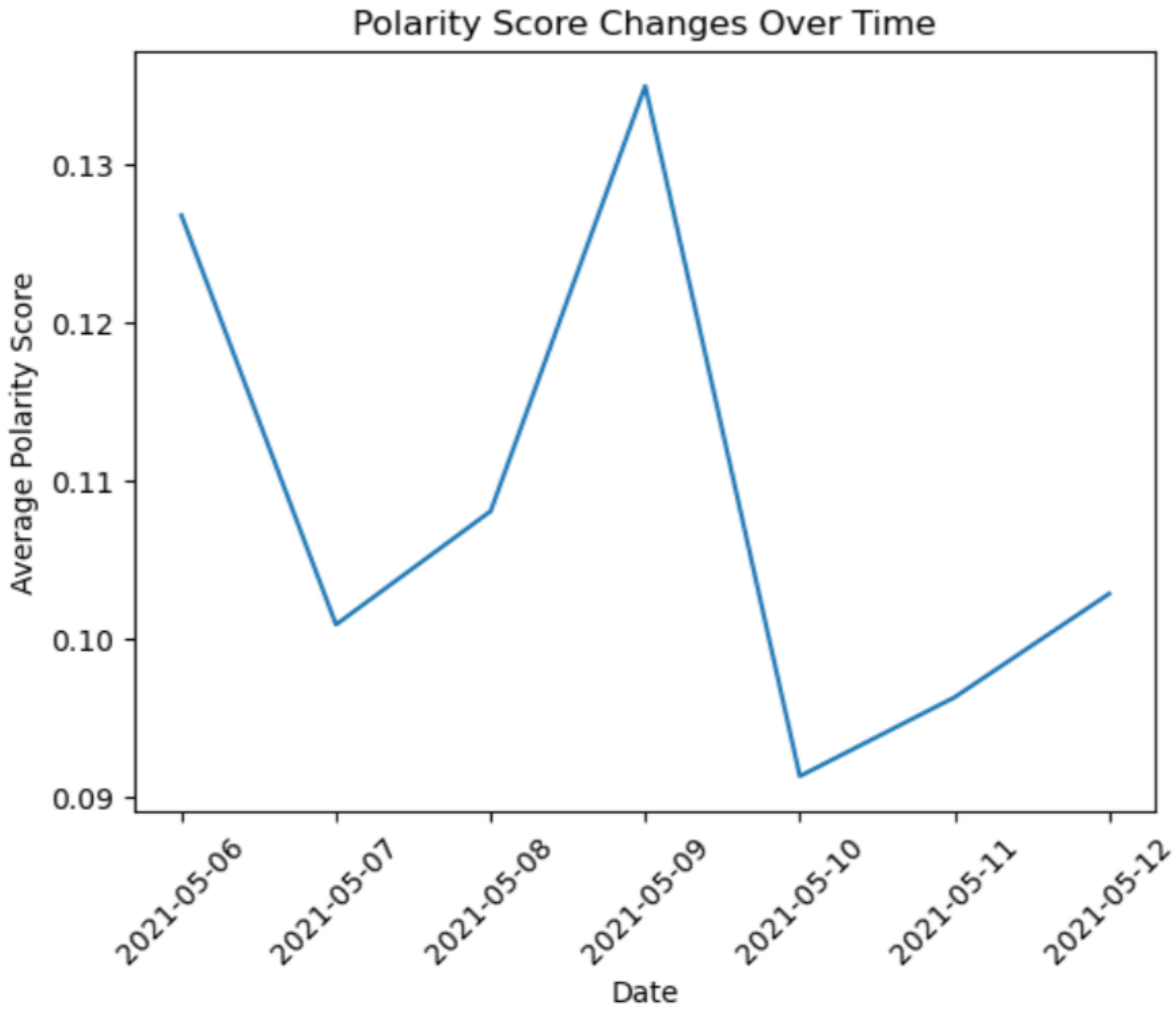

Figure 5: Line chart of average polarity score from 2021-05-06 to 2021-05-12

The curve in figure 5 illustrates that the average polarity score went up and down in the 2nd week after AirTag was released. It arrived at its peak on May 9th with an average polarity score that's higher than 0.13, but then it jumped to its lowest point on the next day with a score of 0.09. However, the changes were not huge, and the overall sentiment was positive at all times.

Modeling

a) Preprocessing and Training

Before starting to build my models, I've performed the following NLP preprocessing steps to the tweet text:

- Tokenization was done to split sentences into tokens.
- Stop words were removed to give more focus to the important information.
- Lemmatization was done to convert words to their meaningful base forms.
- The dataset was vectorized by using the CountVectorizer, which turned the text into numerical data.

b) Model Selection

After these preprocessing steps, I was ready to dive into the modeling section. Since my goal was to detect negative tweets, I had a binary classification problem. To build a negative tweet detector, all tweets with 'positive' and 'neutral' sentiment labels were relabeled as 0, and tweets with 'negative' sentiment labels were relabeled as 1. Here I have tried two classification models: Multinomial Naive Bayes classifier and Logistic Regression Classifier.

Evaluating the performance of a model by training and testing on the same dataset could lead to overfitting. To prevent that, I used the Cross-Validation technique under the 5-fold CV approach. I've first tried using models with their default parameters to classify sentiments, and I've evaluated their performances in terms of accuracy, precision, recall, and f1 using cross-validation. Next, I did hyperparameter tuning for both models separately, and I then had two models with their best parameters for me to compare. The two models are MultinomialNB(alpha=0.1) and LogisticRegression(C=10.5).

I have evaluated each model using recall score for both the training and test data using cross validation. The result indicated that the Multinomial Naive Bayes classifier gave a higher cross-validation recall score than the Logistic Regression

Classifier by 0.13, and the performances on the test set were the same for both models in terms of recall score. The recall score of the Multinomial Naive Bayes model is 0.696, which means that if we use our chosen model for negative tweets detection, it's expected to detect around 69.6% of negative tweets.

Takeaways

Based on my findings, by the second week after AirTag was released, the general perception of AirTag was really good. Only 14.15% of tweets that people posted about it were negative. In addition, by looking at the most common keywords used in those negative tweets, we discovered that people who complained about AirTag were mostly worried about its security, which suggested that it could be easy to hack and could be used for stalking. To improve its product, Apple should focus on fixing the security problem and make AirTag safer to use. On the other hand, keywords like “wallet friendly” and “find” were mentioned a lot in the positive tweets, so these advantages can be emphasized in Apple’s marketing materials to get more people interested in the product.

Besides Apple Inc, other businesses can also use my sentiment analysis as a guide to monitor the performance of their own products; also, the negative tweet detector can not only be used to detect negative tweets, but it can also be used to detect negative comments, negative emails, or negative customer reviews. By collecting the negative reviews and analyzing them, companies can better understand their products and work on improvement more effectively.

Future work

Since the tweets that were pulled from Twitter didn’t come with sentiment labels, I needed to use unsupervised techniques for predicting the sentiment by using knowledge bases, ontologies, databases, and lexicons that have detailed information, specially curated and prepared just for sentiment analysis. I used TextBlob in this project, but there are other options as well. One can try using other popular lexicons like the AFINN lexicon and see how the sentiment labels will come out differently.