# Project Report

*on*

# Brand's Sentiment Analysis

*Due: 2022-12-08*

## CS 410: Text Information Systems
*Fall 2022*

*by*

**ChengXiang Zhai**

Professor

Department of Computer Science

University of Illinois at Urbana-Champaign

*Submitted By*

**Ayush Khanna**

*akhanna6@illinois.edu*

**Ethan Alberto**

*ethanma3@illinois.edu*

**Jayanth Chandra**

*jc101@illinois.edu*

UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

# Contents

# 1   Project Topic

Perform sentiment analysis on Twitter tweets for a given brand to help companies gain insights on their brand or on product(s).

# 2   Motivation

The customer feedback to any product is very crucial in brand building as it allows to gain an overview of the wider public opinion. **Sentiment Analysis** aka. *opinion mining* is the area which deals with judgments, responses as well as feelings generated from such texts. Sentiment analysis can help companies to automatically read tons of product reviews and extract useful and meaningful information to discover if the customers are really satisfied with their product or not. The next question is how to gain access to users feedback, should the companies have some forms to collect surveys or questionnaires? This sounds a reasonable approach to gain immediate feedback if all of our intended users respond back to survey forms. But in today's word people are more vocal on social media platforms such as Twitter instead of responding back to some survey's. Hence, companies always find a problem of not getting the correct/enough feedback on its products. The motivation of this project is to help companies to gain insights about it's product using Sentiment analysis, POS (Part of Speech) Tagging and Natural Language understanding techniques on Twitter tweets. Twitter is one of the platforms widely used by people to express their opinions and showcase sentiments on various occasions. Using the companies brand or product sentiment details, companies can boost their business on following five areas:

- Sentiment Analysis to improve Customer Service

- Sentiment Analysis to boost the product and services

- How to improve their Marketing Campaigns

- Monitor Brand's Perception

- Track the Sentiments in Real-Time

# 3   Technical Approach

The purpose of this project is to extract tweets for a given Fast-moving consumer goods (FMCG) brand and analyse the sentiment of their consumers. Further natural language analysis is then performed to give the company insights into the demographics and the topics the consumers are tweeting about. The application, designed and implemented as a dashboard, is a fusion of a number of NLP techniques. The aim is to try out and implement the various concepts learned as part of the course, so as to give FMCG companies a better understanding of their customer base, allowing them to exploit user opinions to drive decisions, promote their brand and most importantly, improve customer experience.
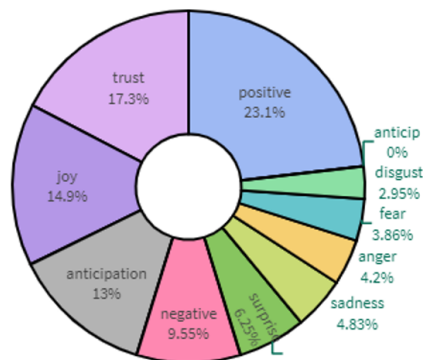
## 3.1    Functional code overview

The project is broken down into several modules depending upon the features implemented. Following are modules created to analyse various aspects of the data:

- **Tweet Scraping:** The first step to build the application is to scrape N tweets for a given brand. Our initial plan was to use the Twitter API. Although the api works as expected, we decided to switch over to use *snscrape*, as it provides users the ability to fetch tweets as well as define a date range that these tweets should be fetched from. The latter was a limitation of the Twitter api.

- **Tweet pre-processing:**   Text pre-processing serves as a key step for any NLP task. It helps remove any unwanted or unimportant text, that may distract the actual NLP task and lower accuracy. The tweets we analyse are subject to various pre-processing steps like lemmetising, removing hyperlinks, emojis, hashtags, punctuations, etc to make the input text ready for analysis.

- **Sentiment analysis:** *TextBlob* is used to perform the sentiment analysis, given the list of tweets. This captures the general sentiment of the consumer base and a polarity score [-1, 1] is extrapolated to define a sentiment for a given tweet, to be either positive, neutral or negative.



***Figure 1:*** *Sentiment Analysis*

- **Emotion analysis:** Sentiment analysis provides a high-level summary of the consumer sentiment. However, we noticed that the boundary between positive and neutral statements is a little fuzzy at times. Emotion analysis gets around this, by providing a more granular intuition of the sentiment. We tried out two available libraries *Text2emotion* and *NRClex* and found that *NRClex* performed optimally for larger data volumes.



***Figure 2:*** *Emotion Analysis*

- **Identify trending topics:** Bag-of-words (BoW) and n-grams of the processed tweets are captured and we use these to plot the most trending topics as a word cloud and as search terms for ranking tweets by sentiment.



***Figure 3:*** *Trending topics*



***Figure 4:*** *Wordcloud of most tweeted words*

- **Satisfaction Score based on Tweets Sentiment:** We define the satisfaction score metric that interpolates the polarity obtained as part of sentiment analysis to define a score between 0 and 10. This translates into a form that is easy to interpret and provides a base for other consumer satisfaction scores that can be calculated.

- **Net Promoter Score (NPS):** NPS serves as a good metric to quantify the overall sentiment of consumers. We use the satisfaction score to obtain the overall NPS score of the brand as well as an hourly and daily trend.

- **Geo-location sentiment mapping:** The *Nominatim geocoder* from the *GeoPy* library is used to capture the coordinate information for the tweet origin, which is then plotted on a map. This enables the FMCG brand to better target areas that have a higher negative sentiment and further accelerate growth in areas with positive consumer feedback.



***Figure 5:*** *Location of Tweet origin based on sentiment*
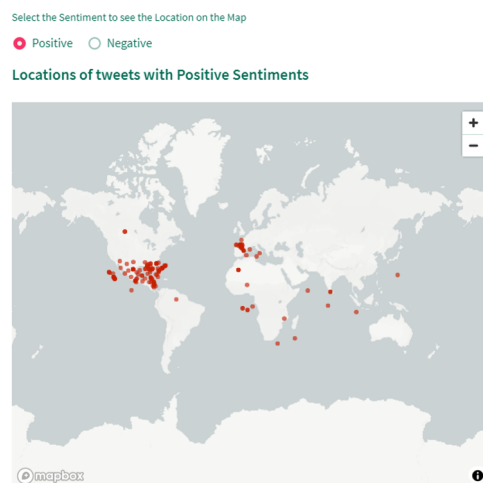
- **BM25 ranking:** The most common bi-grams obtained as part of trending topics are used as query terms to the collection of tweets. The top 10 results are ranked for each sentiment.

- **Miscellaneous trends:** The top retweets and hashtags are also captured.

## 3.2   Software Implementation Details

The application is built entirely using open-source Python frameworks and libraries. Some of the key libraries used, include:

1. **Interface**

   (a) **Streamlit**[1]**:** Streamlit is an open source python based framework for developing and deploying interactive data science dashboards and machine learning models. It is built on top of Python and supports many of the mainstream Python libraries such as matplotlib, plotly and pandas. Figure 6 showcases a snippet of our dashboard in Streamlit.
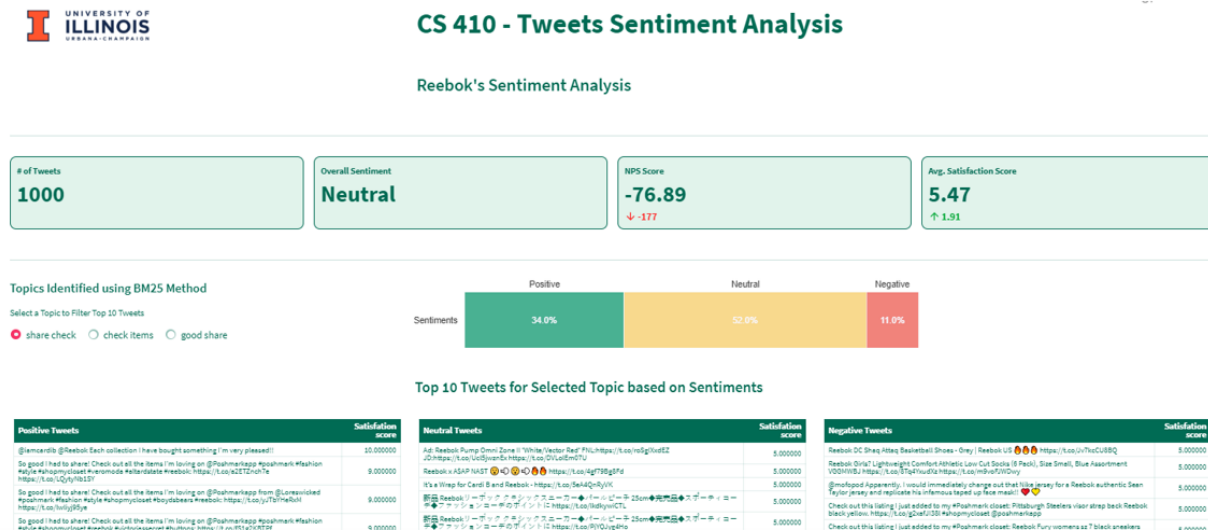


*Figure 6: Dashboard in Streamlit*

2. **Web Scraper**

   (a) **snscrape**[2]**:**   snscrape is a scraper for social networking services (SNS). It scrapes things like user profiles, hashtags, or searches and returns the discovered items, e.g. the relevant posts. Its advantages are that there are no limits to the number of tweets you can retrieve or the window of tweets (that is, the date range of tweets). This means that snscrape allows you to retrieve historical data.

3. **Text Preprocessing**

   (a) **NLTK**[3]**:** NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical

---

[1]https://streamlit.io/

[2]https://github.com/JustAnotherArchivist/snscrape

[3]https://www.nltk.org/

resources such as WordNet, along with a suite of text processing libraries for classification, tokenisation, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

(b) **tweet-preprocessor**[4]**:** Preprocessor is a pre-processing library for tweet data written in Python. This library makes it easy to clean, parse or tokenise tweets. It supports cleaning, tokenising and parsing of URLs, hashtags, mentions, reserved words (RT, FAV), emojis and smileys.

4. **NLP Analysis**

(a) **NRClex**[5]**:** NRCLex measures the emotional affect from a body of text. Affect dictionary contains approximately 27,000 words, and is based on the National Research Council Canada (NRC) affect lexicon and the NLTK library's WordNet synonym sets.

(b) **TextBlob**[6]**:** TextBlob is a python library for Natural Language Processing (NLP). TextBlob actively uses Natural Language ToolKit (NLTK) to achieve its tasks. TextBlob returns polarity and subjectivity of a sentence. Polarity lies between [-1,1], -1 defines a negative sentiment and 1 defines a positive sentiment. Negation words reverse the polarity. Subjectivity quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information.

(c) **rank_bm25**[7]**:** This package provides a collection of algorithms for querying a set of documents and returning the ones most relevant to the query.

(d) **geopy**[8]**:** geopy is a Python client for several popular geocoding web services. It makes it easy for developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.

5. **Data Visualisation**

(a) **Plotly**[9]**:** The plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

(b) **Matplotlib**[10]**:** Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

---

[4]https://pypi.org/project/tweet-preprocessor/
[5]https://pypi.org/project/NRCLex/
[6]https://textblob.readthedocs.io/en/dev/
[7]https://github.com/dorianbrown/rank_bm25
[8]https://geopy.readthedocs.io/en/stable/
[9]https://plotly.com/python/
[10]https://matplotlib.org/

### 3.3    Software Usage Details

Outlined below are the installation and set up instructions for our software.

#### 3.3.1    Install the Docker Image

- Please install **Docker** in your machine:
    - Use this link if you are running the software on Windows: Install on Windows | Docker Documentation
    - Use this link for Mac installation: Install on Mac | Docker Documentation
- Once installed, start Docker Engine in your machine.
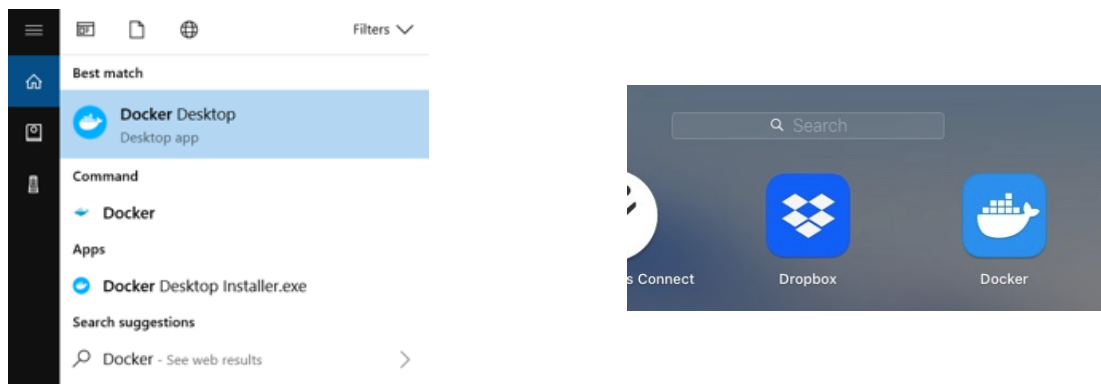
**Figure 7:** *Installing Docker Desktop*

#### 3.3.2    Git Clone and Run Docker Container

1. Clone the github repository from command line or terminal:

    git clone https://github.com/akhanna6/CS410−CourseProject−Team−AEJ.git

2. Go inside CS410-CourseProject-Team-AEJ directory.

    **cd** CS410−CourseProject−Team−AEJ

3. Build the Docker Image *(make a note of . (dot) at the end) – Approx. time to run 2 mins*

    docker build −t brandanalyser .

[Ayushs−MacBook−Air:cs410_final_project ayushkhanna$ docker build −t brandanalyser .

**Figure 8:** *docker build cli command*

4. Run the Docker container

    docker run −p 8501:8501 brandanalyser

```
[Ayushs-MacBook-Air:cs410_final_project ayushkhanna$ docker run -p 8501:8501 brandanalyser
2022-12-04 05:14:46.299 INFO    matplotlib.font_manager: generated new fontManager

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.


  You can now view your Streamlit app in your browser.

  Network URL: http://172.17.0.2:8501
  External URL: http://122.161.83.247:8501
```

**Figure 9:** *docker run cli command*

5. To use the software, open the URL which you see once you type *docker run -p 8501:8501 brandanalyser* in your command line. Alternatively, you can also open it by clicking on URL visible inside PORT(S) in Docker Image.

| | | NAME | IMAGE | STATUS | PORT(S) | STARTED | ACTIONS |
|---|---|---|---|---|---|---|---|
| ☐ | | hardcore_feynman e5f00795dd63 | brandanalyser:latest | Running | 8501:8501 | 15 seconds ag | ▪ ⋮ 🗑 |

**Figure 10:** *brandalalyser docker image*

# 4   Application Limitations

- **Tweets Pre-Processing:** We noticed that when we pull thousands of tweets by going back to several months, the performance is reduced. On an average, it takes around 10-30 seconds to process 1000 tweets from the past 7 days. Keeping the performance stat in mind, we decided to limit the tweets range to go back up till past 3 months with max. 5000 tweets, which ever comes first. We also introduced the Streamlit Cache (@st.cache) to improve the performance.

- **Performance of Geo Locations:** The process of finding Geo Location (longitude and latitude values), using geopy with the *Nominatim* geocoder, takes some time. As it is a free service, it allows an absolute maximum of 1 request per second and does not support batching. This slows down the process quite a bit. Sometimes, we observe timeout errors while running the geopy library. Hence, we run the process on a sentiment basis rather than on the full dataset.

- **NPS score:** We decided to use Net Promoter Score (NPS) as a metric to quantify customer satisfaction. With the low volume of data scraped, there is higher concentration of tweets with a neutral sentiment. This causes the NPS score to drop or show lower values. The NPS score depends largely on promoters i.e. ratings of 9 and 10. Neutral and negative scores behave as detractors.

- **Streamlit Design challenges:** We realized that there are some limitations of Streamlit while styling different widgets. In Streamlit, the column class indicator remains same in the dashboard for all *st.columns* classes having equal number of columns. This results into same styling being applied to all other widgets with similar number of columns. So, we spent sometime in figuring out the final layout of our dashboard.
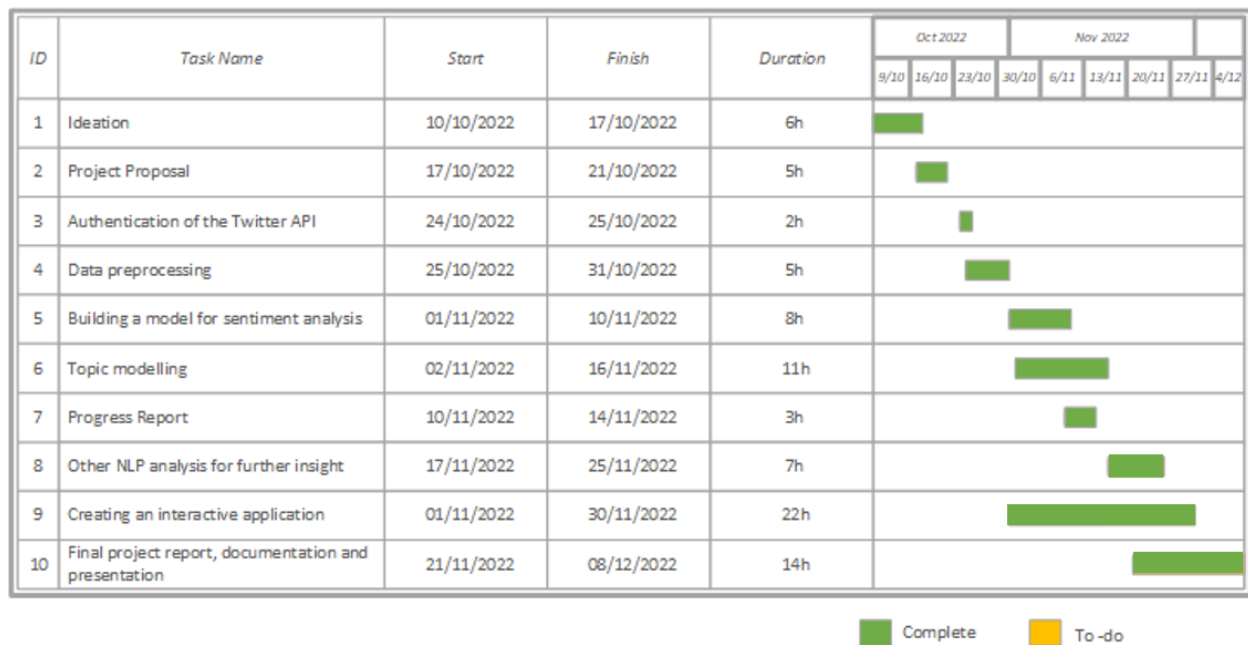
# 5   Future Scope

We have a identified a number of areas that could further be improved and optimised:

- The performance and accuracy of a number of the NLP tasks is directly dependent on the volume of tweets scraped. We currently limit our scraper to only capture a limited number of tweets (1000 for the purpose of the demo), as the performance of the application degrades with an increased size of the dataset. The application can be scaled and optimised through batch and stream processing using technologies like Spark on the cloud.

- We currently use the free geopy API. Hence, as discussed earlier we can only it run once per second for a given location to obtain map coordinates (approx. 6-7 mins for the full dataset). We sometimes see timeout errors while running the geopy library. The performance can be improved in the future by using Google's paid subscription APIs.

- The project can be further expanded to showcase brand results against competitors as well as incorporate customer profile data which can be used to better target different consumer demographics for enhanced data driven decisions.

# 6   Team Members and Contributions

- Ayush Khanna (Net ID: `akhanna6`) - Team Lead

- Ethan Alberto (Net ID: `ethanma3`)

- Jayanth Chandra (Net ID: `jc101`)

All the tasks as defined in the Gantt chart below, were split equally among all team members. This includes experimentation with various techniques, building the various components and creating all the necessary documentation.

| ID | Task Name | Start | Finish | Duration | Oct 2022 | | | | Nov 2022 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 9/10 | 16/10 | 23/10 | 30/10 | 6/11 | 13/11 | 20/11 | 27/11 | 4/12 |
| 1 | Ideation | 10/10/2022 | 17/10/2022 | 6h | | | | | | | | | |
| 2 | Project Proposal | 17/10/2022 | 21/10/2022 | 5h | | | | | | | | | |
| 3 | Authentication of the Twitter API | 24/10/2022 | 25/10/2022 | 2h | | | | | | | | | |
| 4 | Data preprocessing | 25/10/2022 | 31/10/2022 | 5h | | | | | | | | | |
| 5 | Building a model for sentiment analysis | 01/11/2022 | 10/11/2022 | 8h | | | | | | | | | |
| 6 | Topic modelling | 02/11/2022 | 16/11/2022 | 11h | | | | | | | | | |
| 7 | Progress Report | 10/11/2022 | 14/11/2022 | 3h | | | | | | | | | |
| 8 | Other NLP analysis for further insight | 17/11/2022 | 25/11/2022 | 7h | | | | | | | | | |
| 9 | Creating an interactive application | 01/11/2022 | 30/11/2022 | 22h | | | | | | | | | |
| 10 | Final project report, documentation and presentation | 21/11/2022 | 08/12/2022 | 14h | | | | | | | | | |

■ Complete     ■ To-do

*Figure 11: Project plan Gantt Chart*

# 7 References

- https://medium.com/swlh/tweet-sentiment-analysis-using-python-for-complete-beginners-4aeb4456040

- https://medium.com/mlearning-ai/web-scraping-word-cloud-nlp-techniques-amazon-product-review-99c1d44e58e8

- https://www.surveymonkey.com/mp/nps-calculator/