Twitter Sentiment Analysis

Introduction

Twitter, Inc. is a real-time social media platform that allows people to express themselves publicly. It connects users to people, information, ideas, opinions, and news through a network. Live commentary, live connections, and live conversations are among the company's services.

Twitter is a microblogging and social networking service based in the United States, where users can send and receive messages known as "tweets." It is a rapidly growing service with over 200 million registered users, 100 million of whom are active users, and half of whom log on daily, resulting in nearly 250 million tweets per day.

Because of the large number of people who use Twitter, the tweets reflect public opinion. Many applications require analyzing public sentiment, including firms attempting to determine the market response to their products, political election forecasting, and socioeconomic phenomena such as stock exchange forecasting.

The goal of this project is to create a functional classifier that can automatically classify sentiment in a tweet stream. We are given a message; we must decide whether it is positive, negative, or neutral. When sending messages with both positive and negative sentiments, choose the one with the stronger sentiment.

Related Work

Wilson et al. presented a seminal paper on phrase level sentiment analysis in 2005, identifying a new approach to the problem by first classifying phrases according to subjectivity (polar) and objectivity (neutral), and then further classifying the subjectively classified phrases as positive or negative. Many of the objective phrases used prior sentiment bearing words, which resulted in poor classification of especially objective phrases, according to the paper. It claims that using a simple classifier that assumes the word's contextual polarity is simply equal to its prior polarity yields a result of around 48%. This paper's novel classification process, combined with a list of ingenious features that included information about contextual polarity, resulted in a significant improvement in classification performance.

Koulmpis also claims that using emoticons and internet slang words as features yielded positive results. In a tweet, Brody et al. investigates word lengthening as a sign of subjectivity. The study found that the greater the number of instances of a word lengthening, the greater the likelihood of that word being a strong indicator of subjectivity.

Bollen et al. investigate how to categorize tweets based on the mood expressed in them and develop a method for categorizing tweets into six distinct moods: tension, depression, anger, vigor, fatigue, and confusion. They employ an enhanced version of the Profile of Mood States, a

widely used psychometric tool. They created a word dictionary and assigned weights to each of the six mood states, after which each tweet was represented as a vector with these six dimensions. However, there is not much information about how they created their customized lexicon or what method they used.

Data Acquisition:

We have created developer account on Twitter from which we are able to gather tweets around **500,000** Tweets per month using Token Keys. We are gathering as many as Tweets we can based on positive and negative feedback. We are using python to download all the tweets by seeking help from GitHub, Kaggle etc. for codes to access the live data. The data is extracted using keywords or simple word phrases having very little or no individual meaning when used alone. This data is collected through the package called 'tweepy' in Python through which script is run in ample amount of time to save the real time tweets in a comma separated value (csv) format. SampleStream and FilterStream are two ways to access tweets using the API. SampleStream simply provides a small, random sample of all tweets that are being streamed in real time. FilterStream sends tweets that meet a set of criteria. It can filter tweets based on three criteria:

- Tracking/searching for specific keywords in tweets.
- Tracking/searching for specific Twitter users based on their user-ids.
- Tweets originating from a specific location (only for geo-tagged tweets).

Any one of these filtering criteria, or any combination of them, can be specified by a programmer. For our purposes, however, there are no such limitations, so used the SampleStream mode.

Feature Extraction:

Tokenization: It is the process of breaking down a continuous stream of text into words, symbols, and other meaningful elements known as "tokens." Whitespace and/or punctuation characters can be used to separate tokens. It's done this way so that we can examine tokens as individual components of a tweet.

Stemming: It is the process of reducing a derived word to its root or stem in order to normalize the text. A stemmer, for example, would reduce the words "lovable," "loving," to the root word "love."

Parts of Speech Tagging: POS-tagging is the process of tagging each word in a sentence to indicate which grammatical part of speech it belongs to, such as noun, verb, adjective, adverb, coordinating conjunction, and so on.

- POS tag polarity score (Noun, Preposition, Adjectives)

Special characters and Negation.

Count of repetition words, Non-English words, and Acronyms.

Data Cleaning

Firstly, we removed unwanted text patterns and twitter handles(@user). Then, we removed Punctuation, numbers, and special characters.

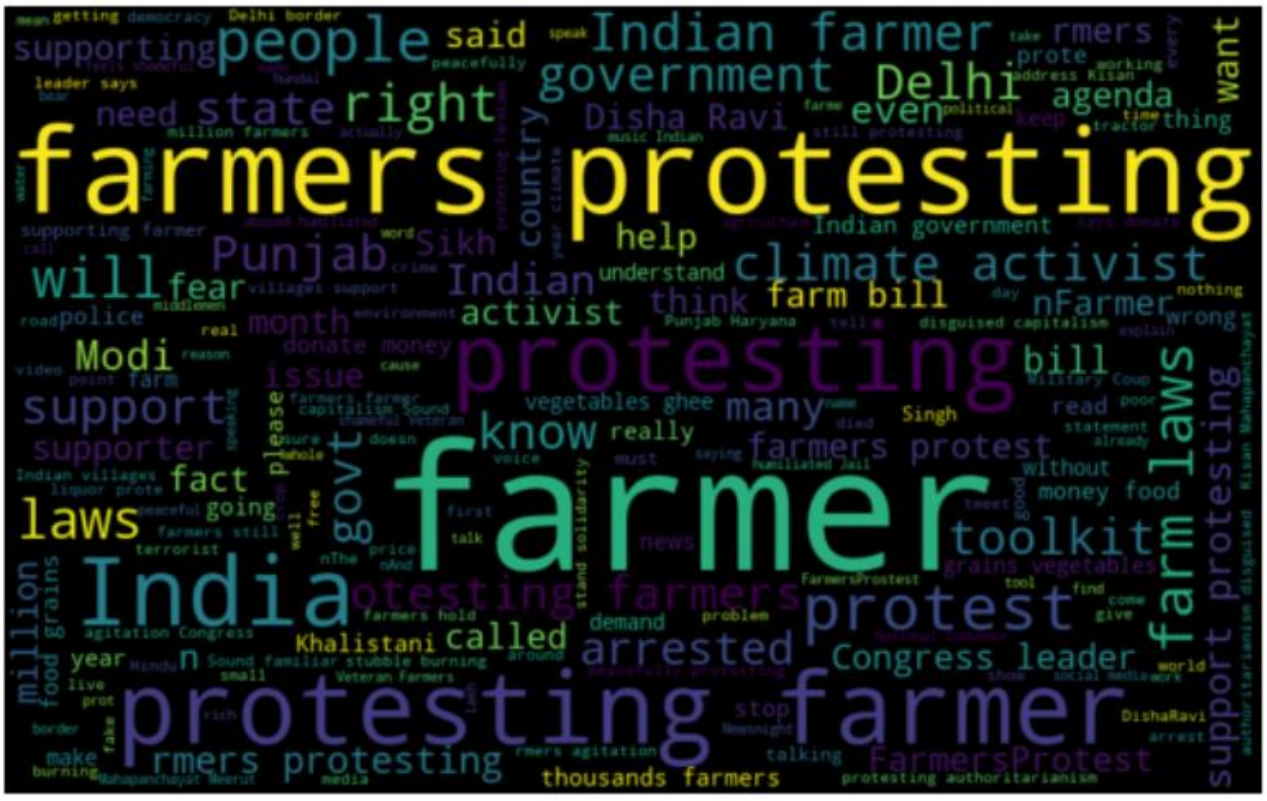
Removing Short words: Here, we remove short words having length less than 3 letters such as hmm, bye.

Stop-words removal: Stop words are a group of extremely common words that, when used in a text, provide no additional information, and are thus considered of no use. “A,” “an,” “the,” “he,” “she,” “by,” “on,” and so on are examples.

Text Normalization: It is the process of converting a text into a standard (canonical) form.The words "gooood" and "gud," for example, can be transformed into "good," which is the canonical form.

Data Visualization

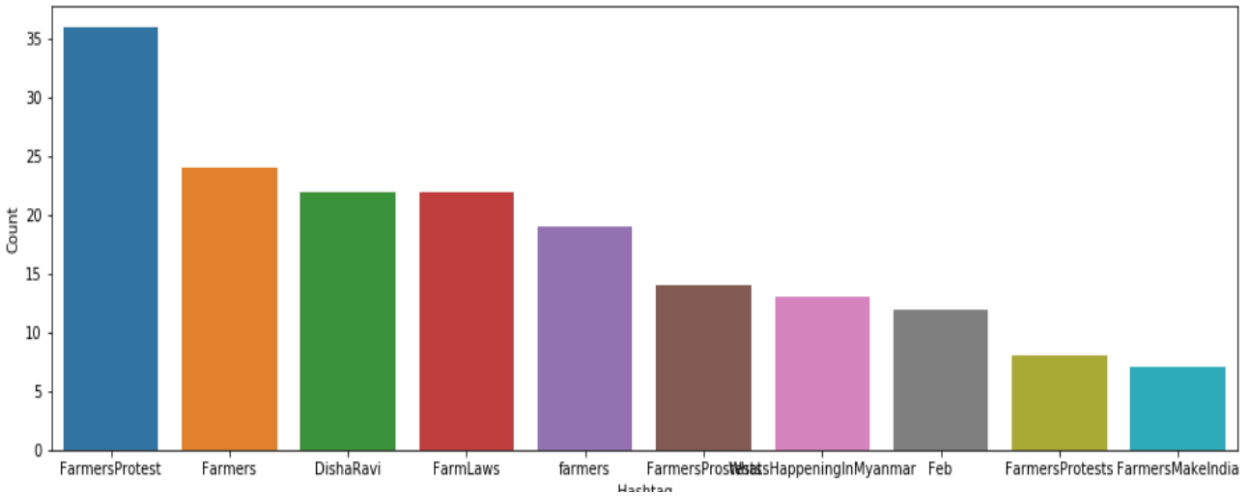
Word Cloud



Here, we have created a word cloud from our data, as our data is about Farmer Protest so Farmer and protest is coming out big and bolder. Farmers and protests are among the most

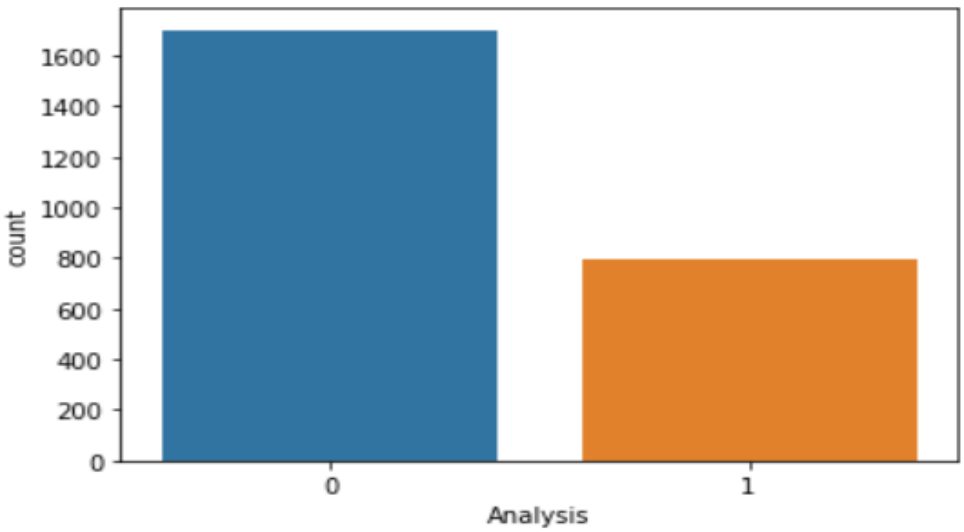
frequently used terms. However, words like Khalistan, modi, congress, and climate activist indicate that the farmers' movement was twisted into other related issues as well, and the political scenario was created in such a way that the whole issue was diversified rather than converged.

Top 10 trending tweets by using Hashtags:



From above representation we can say which hashtags has mostly tweeted. #FarmersProtest is highest as seen followed by #Farmers. While we observe that #farmersProtest is highest in trend but if we observe closely third bar shows that Disha Ravi hashtag which implies that data we have got for observing trends in farmers protest also have various other elements and presence of these kind of elements indicates how the whole movement was turned around to specific person who are not even related to these protests.

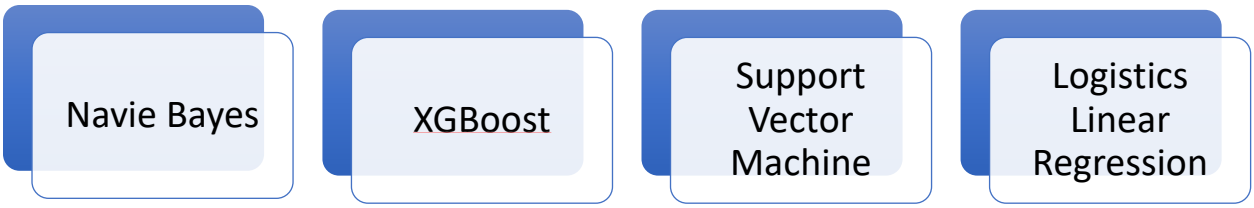
Bar Graph of Positive/Negative Tweets



Here we plotted bar graph of positive and negative tweets. 0 in the blue color is negative tweet and 1 in the orange is positive tweet. From the above graph we can say that our data contains more negative tweet than the positive one.

Methods:

We have used Machine learning models as it provides smart alternatives to analyzing vast volumes of data. By developing fast and efficient algorithms and data-driven models for real-time processing of data, **Machine Learning** can produce accurate results and analysis.



Naïve Bayes:

Naïve Bayes Classifier is one of the simple and most effective **Classification** algorithms which helps in building the fast **machine learning** models that can make quick predictions. Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is a probabilistic **classifier**, which means it predicts based on the probability of an object.

XGBoost:

XGBoost is a gradient boosting-based decision-tree-based ensemble Machine Learning algorithm. Artificial neural networks outperform all other algorithms or frameworks in prediction problems involving unstructured data (images, text, etc.). XGBoost is a parallel tree boosting (also known as GBDT, GBM) algorithm that solves a variety of data science problems quickly and accurately.

Support Vector Machine:

SVM (Support Vector Machine) is a supervised machine learning algorithm that can be used to solve classification and regression problems. However, it is mostly used in classification problems. Support Vectors are simply the co-ordinates of individual observation. SVM models

can categorize new text after being given sets of labelled training data for each category. When attempting to solve a text classification issue.

Logistics Linear Regression

Logistic regression is considered a generalized linear model because the outcome always depends on the sum of the inputs and parameters. Or in other words, the output cannot depend on the product (or quotient, etc.) of its parameters. The logistic model is used to estimate the probability of a specific class or event, such as pass/fail, win/lose, alive/dead, or positive/negative. This can be used to model a variety of events, such as determining whether an image contains a cat, dog, lion, or other animal. Each detected object in the image would be assigned a probability ranging from 0 to 1, with a total of one.

Results:

After Applying all the model. Following are the F1 score accuracy results:

Model	F1- Score
Naïve Bayes	0.82
XGBoost	0.78
SVM	0.81
Logistic Regression	0.8

Here, we are getting maximum accuracy for Naïve Bayes model that is 82% followed by SVM that is 81%. For our farmer protest sentiment analysis Naïve Bayes is the best fit.

Discussion:

The main requirement was to conduct sentiment analysis, which we were very clear about. With that in mind, we began searching for trending topics, eventually settling on Farmer Protest as the most trending. Then we started working with the Twitter API, cleaned up the data, and applied models, and we believe our results are useful.

There were many challenges that we came across like tweet content shared by different people is vastly different, In opinion texts, lexical content alone can be misleading, Intra-textual and sub-sentential reversals, negation, topic change are common in original tweets by people, Unstructured and non-grammatical as people don't follow rules of grammar, lexical resource on a public platform like twitter, Out of Vocabulary words can result in many of impromptu

tweets, Extensive usage of acronyms like asap, lol, afaik and the most important one was to get approval from Twitter Team for the Twitter API.

Conclusion:

We conclude that using even the simplified features and API data, our modelling approach improves accuracy. Sentiment analysis, especially in the context of microblogging, is still in its early stages and far from complete. As a result, we suggest a couple of ideas that we believe are worth pursuing in the future and could lead to even better results. However, there are a few items we would like to think about for future work, which we will discuss in the next segment.

Future Plans:

- We will be applying the same model on other social media data gathered from Facebook and Instagram to compare how these social media channels fare in spreading hatred. we can improve those models by adding extra information like closeness of the word with a negation word.
- We will create a web application that will be conducting real-time sentiment analysis on tweets that matched the user's specific keywords on Twitter. For example, if a user wants to perform sentiment analysis on tweets containing the word "Mccain," he or she can insert the keyword into the web application, and the web application will perform the necessary sentiment analysis and show the results for the user.

Contribution:

Ruchita Tamboli:

- Data visualization.
- Worked on Pre-modelling Stage - bag-of-words and prepared vectors for tweet.
- Worked on implementing classifier for the processed dataset using the techniques of SVM and XGBoost models.
- Compared results using F-1 scores and accuracy plots.

Khushboo Patel:

- Feature Extraction.
- Data Cleaning.
- Worked on Pre-modelling stage - Word2Vec Embedding, Doc2Vec Embedding.
- Worked on modelling using logistic linear regression.

Aayush Seth:

- Creating developer Account for API Access.
- Fetching Raw API data into csv format

- Worked on and filtering of hashtags keywords to prepare tweets data for passing in learning model.
- Worked on modelling using Naïve Bayes.

We all three members worked on the documentation part – Problem Statement, data assessment, Presentation and Report.

References:

For API

- https://developer.twitter.com/en/portal/projects/1358920922967736321/apps/20069377/keys.
- https://www.storybench.org/how-to-collect-tweets-from-the-twitter-streaming-api-using-python/

About Twitter

- https://about.twitter.com/en/who-we-are/our-company
- https://en.wikipedia.org/wiki/Twitter

For Modeling

- https://www.justintodata.com/twitter-sentiment-analysis-python/
- https://www.brandwatch.com/blog/understanding-sentiment-analysis/#:~:text=Sentiment%20analysis%20is%20extremely%20useful,public%20opinion%20behind%20certain%20topics.&text=Being%20able%20to%20quickly%20see,and%20plan%20for%20the%20future
- https://medium.com/swlh/tutorial-gathering-text-data-w-python-twitter-streaming-api-e1007a3b70ef
- https://www.researchgate.net/publication/301335561_Sentiment_Analysis_of_Twitter_Data_A_Survey_of_Techniques