Twitter Sentiment Analysis

2022 COEN 240 Machine Learning

Professor: Alex Sumarsono

Report by:

Sourabh Deshmukh W1648445

Sarthak Patel W1650954

Index

Abstract

The project tries to classify the real time tweets in two segments: positive and negative. Twitter is a micro blogging social networking site which enables individuals and organizations to express their views in the form of tweets. Twitter is used by Politicians, Actors, Business organizations, Activists to express their views. Numerous applications need the analysis of public mood, including businesses attempting to evaluate the market response to their products, the prediction of political decisions, and the analysis of socioeconomic phenomena like stock exchange. The goal of this project is to create a practical classifier that can accurately and automatically identify the sentiment of a twitter stream.

Motivation and Environment

We decided to work with twitter rather than articles, blogs and other media networks because it accurately represents public emotions and opinions. There are very few people who write blogs and articles and the comments on such posts are even skewed to form a sizable sample size for data while other social media networks are extensively flooded with advertisements. Public emotion is highly crucial in macro-scale socioeconomic. Businesses, for instance, might use this information to evaluate the causes of regionally diverse responses and sell their products more effectively by looking for relevant solutions, such as the establishment of suitable market segments.

- **Tech Stack**

We have used Python3 programming language along with libraries like Pickle, Seaborn, Numpy, Pandas, Tweepy, Wordcloud, Matplotlib, ntlk, sci-kit learn. We used the Google Colab notebook for data analysis and model assessment.

Dataset & Algorithms

- **Data Acquisition**

Kaggle is an online community of data scientists and machine learning practitioners. It hosts thousands of datasets and often one of the best data sources for training machine learning models. Users can also train their models on the portal and share it people. We have used the Sentiment140 dataset from Kaggle to train our machine learning model.

- **Data Description**

The dataset consists of total 1.6 million tweets consisting the attributes of Target, ids, Date, Flags, Username and Text.

1. target: polarity of a tweet (*0* = negative, *2* = neutral, *4* = positive)
2. ids: The id of a tweet (3278)
3. date: the date it was tweeted (*Sun May 17 12:43:21 UTC 2011*)
4. flag: The query (*lyx*). If there is no query, then this value displays NO_QUERY.
5. user: username of the tweeter (*elonmusk*)
6. text: text of a tweet (*Lyx is hotl*)

- **Data Preprocessing**

We aim to predict the sentiment of tweets, whether being positive or negative. Before training the model with the dataset, we do some basic data exploration to gauge the overall data. We examine the features to determine their type, remove null values and normalize the features.

	sentiment	ids	date	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

Fig1 Dataset

As we can see in fig1 there are many columns in the dataset which are not needed to train our model. The only data columns required for our project are sentiment and text, so we maintain those columns and discard the rest.

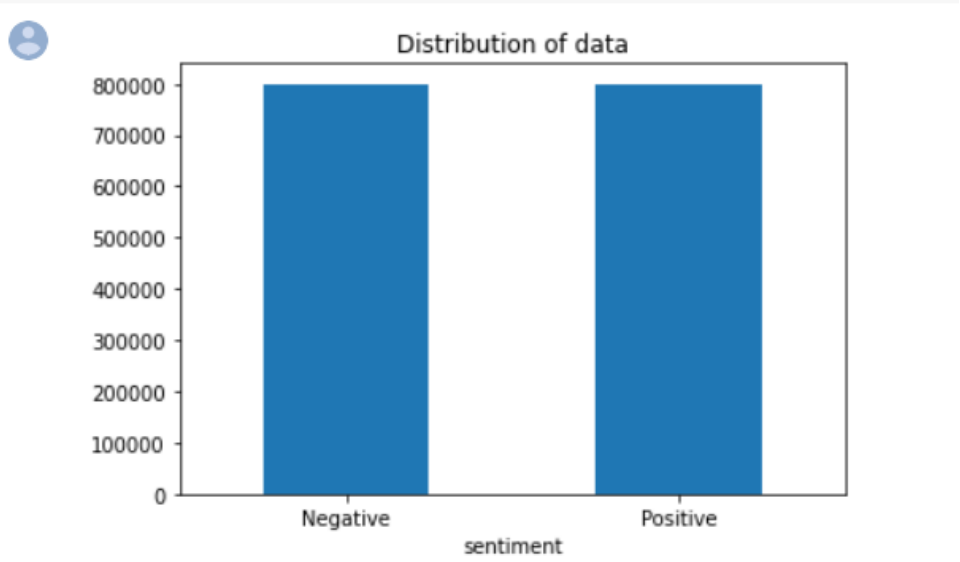

Fig 2 Dataset subsets

In the original dataset the polarity of tweets was annotated with 0,2 and 4 which stood for negative, neutral and positive respectively. However; we are changing the target values to 0 and 1 for negative and positive respectively which is depicted in fig2.

We have further processed data by removing hyperlinks and retweets from the feed, replacing emoji's with relevant keywords any by removing mentions(@) and hashtags and removing stop words – stop words are words like is, are, this, there, they, which, from, etc.

- ## Data Analysis

Here we used word cloud to plot the tweets on a graph for both positive and negative tweets from dataset. A word cloud is a powerful tool for text processing and may be created from nearly any data source, such as internet reviews, consumer surveys, news stories, and so on. Here, a word cloud is a graphical depiction of your published Tweets. The larger a term appears in the cloud, the more frequently it appears in your tweets. As a result, the largest word in your tweets is the most influential term. They are fascinating because they allow you to quickly and easily analyze your Twitter data and obtain an overview of what's being referenced most frequently in your feed.

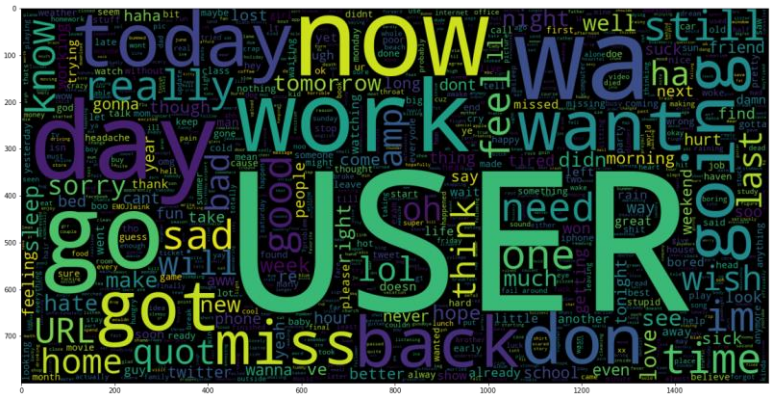

Fig 3 Word Cloud featuring Negative tweets

Fig 4 Word Cloud featuring positive tweets

The data was split in two subsets: 95% was used for training the model and remaining 5% of the dataset was used for testing the model. The training set is the set of data that was utilized to train the model. The model should watch and learn from the training set, and any of its parameters should be optimized. The testing set is the piece of data examined in the final model and compared to the preceding data sets. The testing set serves as an assessment of the final mode and algorithm.

- ## TF-IDF Vectoriser

A Bag of words (BoW) is a broader technique that turns text into a feature vector by counting the occurrences of words in a document. It disregards the significance of words. Term frequency — Inverse document frequency (TFIDF) is, on the other hand, based on the Bag of Terms (BoW) model, which expected to analyze into the less and more important words in a text. The value of a word in a text is extremely important in information retrieval.

Term Frequency: The term frequency (TF) of a word (w) in a text is measured (d). The ratio of a word's occurrence in a text to the total number of words in a document is defined as the TF.

Because the corpus documents are of various lengths, the denominator term in the formula is to normalize.

$$TF\ (w, d) = \frac{occurence\ of\ w\ in\ document\ d}{total\ numbr\ of\ words\ in\ document\ d}$$

Inverse Document Frequency (IDF): It is an assessment of a word's significance. Term frequency (TF) doesn't at all take into account the significance of terms. Some words, such as 'from,' 'is,' and so on, are commonly used yet have little meaning. Each word in the corpus D is given a weightage by IDF depending on its frequency. IDF of a word (w) is defined as

$$IDF\ (w, D) = log\left(\frac{Total\ number\ of\ documents\ (N)\ in\ corpus\ D}{number\ of\ documents\ containing\ w}\right)$$

TF-IDF: The fundamental idea behind TF-IDF is that a phrase's importance is inversely related to its frequency throughout texts. The term frequency (TF) represents how frequently a word appears in a document, whereas the phrase rarity (IDF) represents the proportional uniqueness of a term in the document set. We can get our final TF-IDF value by product of these numbers together.

$$TFIDF\ (w, d, D) = IDF\ (w, D) * TF\ (w, d)$$

Evaluating the models

Here we are training a model for binary classification of data, whether the tweet being positive or negative. There are various machine learning algorithms for binary classification of data such as decision trees, k-nearest neighbor, Support Vector Machines, Naive Bayes, etc. Here we have selected 3 models for reference to select our model based on accuracy and speed by comparing the given models.

Bernoulli Naïve Bayes: The Bayes Theorem used for calculating conditional probabilities is the foundation of the Naive Bayes algorithm, which is utilized in a broad range of classification problems. It is given by:

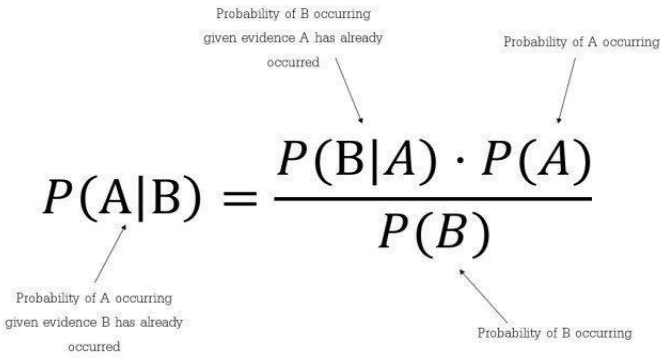

Fig 5 Naïve Bayes formula

Bernoulli Naïve Bayes is a model applied on binary classification where the feature is independent of each other and they are not correlated. This was quicker compared to most other algorithms and it could also work on a small size of dataset.

LinearSVC: Linear Support Vector Machine (Linear SVC) is an algorithm that attempts to find a hyperplane to maximize the distance between classified samples. Based on testing this algorithm on the dataset, it provides slightly better accuracy than Bernoulli Naïve Bayes.

Logistic Regression: In logistic regression, a probabilistic method is applied. Logistic regression delivers the best projections when using the maximum likelihood strategy. A sigmoid is a mathematical representation with the property of being able to accept any real value between -∞ and +∞ and convert it to a real value between 0 and 1. So in the project, if the sigmoid value is more than 0.5, we consider it positive; if it is below 0.5, we consider it negative.

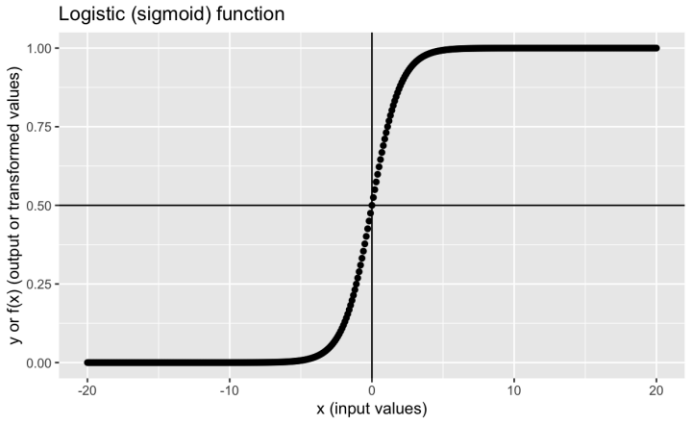

Fig 6 Sigmoid Function

$$S(z) = \frac{1}{1 + e^{-z}}$$

The vectored form of cost function is given by

$$J(\theta) = -\frac{1}{m}(y^T log\,(h) + (1 - y)^T log\,(1 - h))$$

Confusion Matrix

Confusion matrix is a performance matrix for machine learning classification problems with two or more classes as output. It is a table with four distinct (False Positive, False Negative, True Negative, True Positive) projected and actual values. The accuracy of Logistic Regression Model was higher than other two models. So, we proceeded to test the model with Logistic Regression.

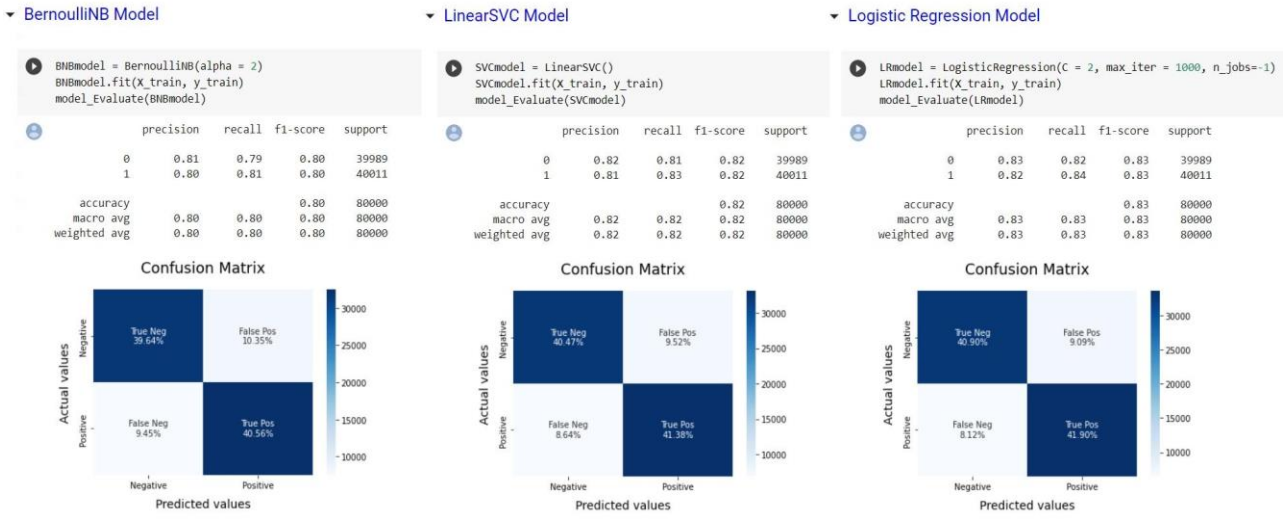

Fig 7 Confusion Matrix of different Models

Conclusion and Future Work

We decided to work on our model with Logistic Regression based on our results of accuracy we received from the tests. The Vectorization process TF-IDF used here takes into account the number of times a single word appears in the tweet and the importance of the words. However, it does not eliminate the possibility of synonymous words in a tweet.

References

1. https://www.kaggle.com/datasets/kazanova/sentiment140
2. https://monkeylearn.com/blog/twitter-word-cloud/
3. https://towardsdatascience.com/text-vectorization-term-frequency-inverse-document-frequency-tfidf-5a3f9604da6d
4. https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/
5. https://www.analyticsvidhya.com/blog/2021/06/support-vector-machine-better-understanding/
6. https://medium.com/nerd-for-tech/twitter-sentiment-analysis-using-logistic-regression-ff9944982c67
7. https://numpy.org/doc/
8. https://pandas.pydata.org/docs/
9. https://seaborn.pydata.org/
10. https://docs.python.org/3/library/pickle.html
11. https://www.kaggle.com/code/stoicstatic/twitter-sentiment-analysis-for-beginners
12. https://www.analyticsvidhya.com/blog/2021/04/beginners-guide-to-logistic-regression-using-python/