Twitter Sentiment Analysis

Affan Ali Hasan Khan - 50432243, Khushal Sharma - 50441869
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260

1 Abstract

A significant amount of data is generated as well as being made available to internet users thanks to the development and growth of online technologies. The internet has developed into a forum for online education, idea sharing, and opinion exchange. Social networking services like Twitter, Facebook, and Google+ are quickly gaining popularity as a result of the ability for users to share and express their opinions on many subjects, engage in conversation with various communities, and broadcast messages globally.The study of sentiment in Twitter data has received a lot of attention.The major aim of this project is sentiment analysis of twitter data, which is useful for analyzing information in tweets when opinions are very unstructured, varied, and occasionally neutral. In this project, we present a comparative analysis, assessment metrics, and existing methods for opinion mining, such as lexicon-based methods and machine learning methods. We present research on twitter data streams using a variety of machine learning methods, including Bernoulli Naive Bayes, SVM (Support Vector Machine), Logistic Regression, and Neural Network.

2 Introduction

Nowadays, people communicate their views and beliefs differently thanks to the internet. Nowadays, it is done primarily through blog postings, internet forums, websites that offer product reviews, social media, etc. Millions of individuals today use social networking sites like Facebook, Twitter, Google Plus, and others to share their thoughts on daily life and express their emotions. We receive interactive media from online communities where users can use forums to inform and persuade others.Tweets, status updates, blog posts, comments, reviews, and other types of social media content produce a lot of sentiment-rich data. Additionally, social media gives businesses a chance by offering them a platform to engage with their customers for advertising. People heavily rely on user-generated content from the internet when making decisions. For instance, if someone wants to purchase a good or use a service, they will research it online and discuss it on social media before making a choice. The volume of user-generated content is too great for a typical user to process. Since this must be automated, several different sentiment analysis approaches are employed.

Before a user purchases a product, sentiment analysis (SA) lets them know if the product's information is good or not. Marketers and businesses use this analysis data to learn more about their goods or services so that they can cater to the needs of the customer.

The fundamental goals of textual information retrieval strategies are to process, search for, or examine the factual material that is already there. Even if

Preprint. Under review.

facts have an objective component, some other literary contents exhibit subjective traits.Sentiment Analysis's fundamental components—opinions, sentiments, assessments, attitudes, and emotions—are primarily represented by these contents (SA). In large part because of the enormous expansion in the amount of information available online from sources like blogs and social networks, it presents many challenging chances to design new applications. For instance, by using SA and taking into account factors such as positive or negative attitudes about the goods, recommendations of items proposed by a recommendation system can be predicted.

3 Sentiment Analysis

Sentiment analysis is a procedure that uses Natural Language Processing (NLP) to automatically mine attitudes, opinions, perspectives, and emotions from text, audio, tweets, and database sources. In a sentiment analysis, opinions in a text are categorized into "positive" and "negative" categories. Subjectivity analysis, opinion mining, and assessment extraction are other names for it.

Although the terms "opinion," "sentiment," "view," and "belief" are frequently used interchangeably, they have different meanings.

- Opinion: A conclusion open to dispute (because different experts have different opinions)
- View: subjective opinion
- Belief: deliberate acceptance and intellectual assent
- Sentiment: opinion representing one's feelings

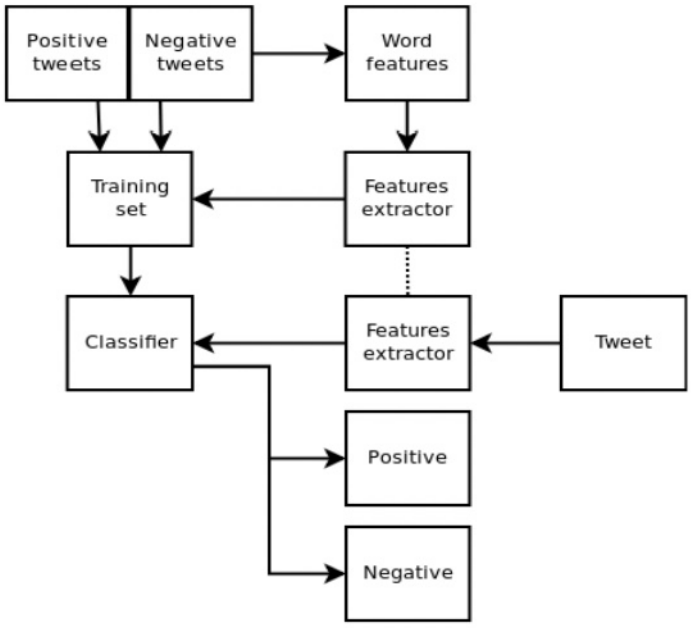


Figure 1: Sentiment Analysis Architecture

4 Pre-processing of the datasets

A tweet includes numerous perspectives about the data that are presented by various people in various ways. The Twitter dataset utilized in this study is already divided into two categories, negative and positive polarity, making it simple to perform a sentiment analysis on the data and see how different variables affect sentiment. Polarity in the raw data makes it particularly prone to redundancy and inconsistency.

Preprocessing of datasets and tweet include following points:

- Removing unused features from the dataset.
- Removing stopwords from tweets using NLTK.
- Removing punctuations from the tweets.
- Removing repeating characters from the tweets.
- Removing URL's from the tweets.
- Removing numbers from the tweets.

Table 1: Publicly Available Datasets For Twitter

HASH	Tweets	http://demeter.inf.ed.ac.uk	31,861 Pos tweets 64,850 Neg tweets,125,859 Neu tweets
EMOT	Tweets and Emoticons	http://twittersentiment.appspot.com	230,811 Pos and 150,570 Neg tweets
ISIEVE	Tweets	www.i-sieve.com	1,520 Pos tweets,200 Neg tweets, 2,295 Neu tweets
Columbia univ.dataset	Tweets	Email: apoorv@cs.columbia.edu	11,875 tweets
Patient dataset	Opinions	http://patientopinion.org.uk	2000 patient opinions
Sentiment140	Tweets and URIs	https://www.kaggle.com/datasets/kazanova/sentiment140	800K Pos and 800K Neg Tweets

5 Feature Extraction

The preprocessed dataset has a lot of unique characteristics. We take the features from the processed dataset using the feature extraction approach. Later, using models like unigram and bigram, this characteristic is utilized to compute the positive and negative polarity of a sentence, which is helpful for gauging people's opinions. In order to process text or documents, machine learning approaches need to be able to express its main aspects. These important characteristics are regarded as feature vectors that are applied to the classification task. There are number of features that have been reported in literature, for this project we are using:

- Words And Their Frequencies: Unigrams, bigrams and n-gram models with their frequency counts are considered as features. There has been more research on using word presence rather than frequencies to better describe this feature.

After preprocessing the dataset, TfidfVectorizer was used to convert the text to a suitable numeric input format for the machine learning models. TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.

The key intuition motivating TF-IDF is the importance of a term is inversely related to its frequency across documents.TF gives us information on how often a term appears in a document and IDF gives us information about the relative rarity of a term in the collection of documents. By multiplying these values together we can get our final TF-IDF value. The higher the TF-IDF score the more important or

$$tf\ idf\ (t, d, D) = tf\ (t, d)\ .\ idf\ (t, D)$$

relevant the term is; as a term gets less relevant, its TF-IDF score will approach 0. As we can see, TF-IDF can be a very handy metric for determining how important a term is in a document. But how is TF-IDF used? There are three main applications for TF-IDF. These are in machine learning, information retrieval, and text summarization/keyword extraction.

5.1 Using TF-IDF in machine learning natural language processing

When working with textual data or any natural language processing (NLP) activity, a sub-field of ML/AI that deals with text, that data must first be converted to a vector of numerical data using a procedure known as vectorization. Machine learning techniques frequently use numerical data. In order to do TF-IDF vectorization, you must first determine the TF-IDF score for each word in your corpus in relation to the given document (see the example documents "A" and "B" in the figure below). As a result, each document in your corpus would have its own vector, and each word in the entire collection of documents would have a TF-IDF score in the vector. Once you have these vectors you can apply them to various use cases such as seeing if two documents are similar by comparing their TF-IDF vector using cosine similarity.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

Figure 2: A = “The car is driven on the road”; B = “The truck is driven on the highway”

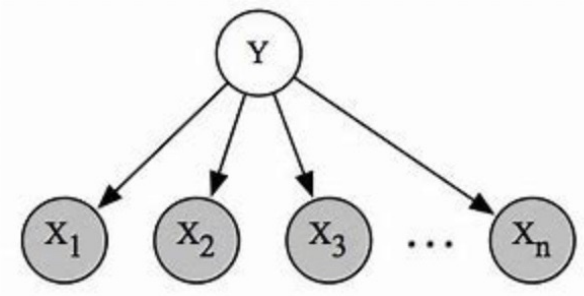
6 Classification models

For this project we used 4 different types of machine learning models to compare and choose the best one as follow:

6.1 Bernoulli Naive Bayes

Naive Bayes works works on the Bayes theorem of probability to predict the class of unknown data sets. Bayes theorem describes the probability of an event based on the prior knowledge or other certain known probabilities of that event.

Naive Bayes classifier assumes that the presence of a particular feature that is present in a class is not related to any other feature in that class.



As was just mentioned, y is a collection of n features where each x in a class of y is distinct from the others. Currently, supervised learning algorithms utilized for classification belong to the Naive Bayes family.

$$\underset{\text{Posteriori}}{\uparrow} P(y|X) = \frac{\overset{\text{Likelihood}}{\downarrow} P(X|y) \overset{\text{Prior}}{\downarrow} P(y)}{\underset{\text{Predictor Prior}}{\uparrow} P(X)}$$

Bernoulli Naive Bayes is a part of the family of Naive Bayes, it accepts just binary values. The most basic example is when we determine whether or not a word will appear in a document for each value. That model is quite condensed. When counting word frequencies is less crucial, Bernoulli might get more accurate findings. Simply put, we must count each value for the binary term occurrence features, which determine if a word appears in a document or not. Instead of determining a word's frequency within the manuscript, these attributes are utilized.

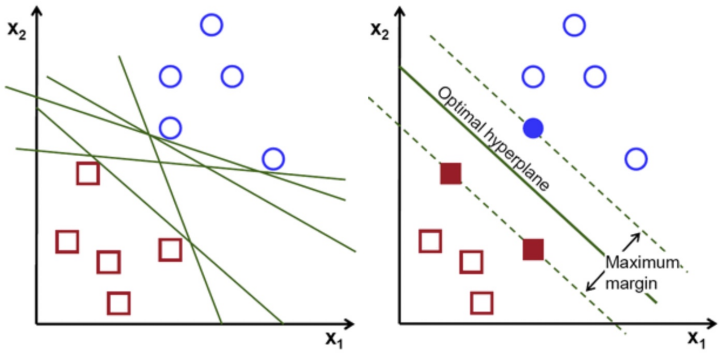
In layman's terms, the Bernoulli distribution has two possibilities that are mutually exclusive: P(X=1)=p or P(X=0)=1-p. Although the BernoulliNB theory allows for several features, each one is supposed to be a boolean binary valued variable. As a result, samples for this class must be represented as binary-valued feature vectors. Any other type of data will cause a BernoulliNB instance to binarize the input.

The decision rule for Bernoulli naive Bayes is based on

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

6.2 SVM (Support Vector Machine)

Finding a hyperplane in an N-dimensional space (N is the number of features) that categorizes the data points clearly is the goal of the support vector machine algorithm.



Possible hyperplanes

To separate the two classes of data points, there are many possible hyperplanes that could be chosen.Support vectors are data points that are closer to the hyperplane and have an impact on the hyperplane's position and orientation. By utilizing these support vectors, we increase the classifier's margin. The hyperplane's location will vary if the support vectors are deleted. These are the ideas that aid in the development of our SVM.

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \qquad c(x, y, f(x)) = (1 - y * f(x))_+$$

Hinge loss function (function on left can be represented as a function on the right)

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value.

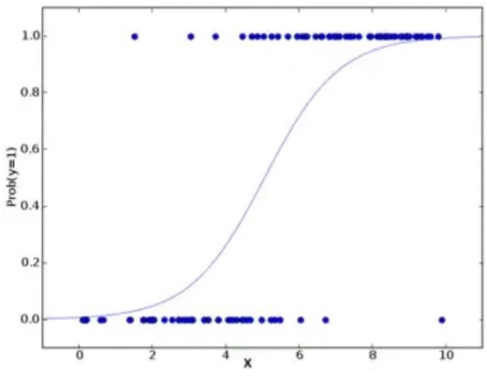
6.3 Logistic Regression

Logistic Regression is one of the basic and popular algorithms to solve a classification problem. It is named 'Logistic Regression' because its underlying technique is quite the same as Linear Regression. The term "Logistic" is taken from the Logit function that is used in this method of classification.

Logistic Regression uses the Sigmoid function. An explanation of logistic regression can begin with an explanation of the standard logistic function. The logistic function is a Sigmoid function, which takes any real value between zero and one. It is defined as

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

The graph of the sigmoid function is given as:



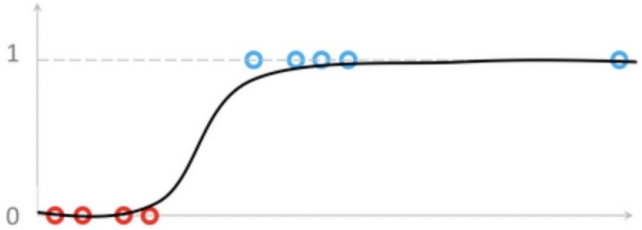
Let's consider t as a linear function in a univariate regression model.

$$t = \beta_0 + \beta_1 x$$

So the Logistic Equation will become

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

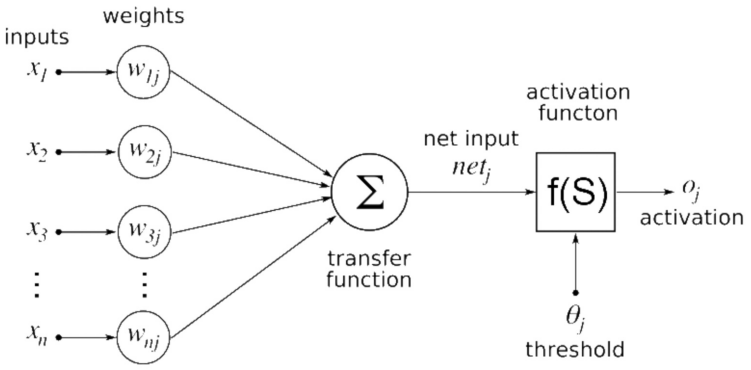
Now, when the logistic regression model comes across an outlier, it will take care of it.



6.4 Neural Network

Artificial neural networks function similarly to the biological systems that served as their inspiration. The connections between the neurons can be viewed as weighted directed graphs, with the neurons serving as the nodes and the connections between the neurons serving as the edges. A neuron's processing component receives numerous signals (both from other neurons and as input signals from the external world). Sometimes, signals are altered at the receiving synapses, and the processing element adds the weighted inputs. Input from one neuron is sent to another if it reaches the threshold, and the cycle continues.

The strength of the connections between the neurons is typically represented by the weights. To acquire the required result for the problem that was defined, the activation function is a transfer function that is applied. In the instance of a binary classifier, let's say the intended output is either zero or one. The activation function can be the sigmoid function. There are several activation functions, but a few



examples include ReLU, linear regression, logistic regression, identity function, binary sigmoid, bipolar sigmoid, bipolar, binary, and bipolar tangent. Through learning processes, artificial neural networks are particularly created for a certain function like binary classification, multi class classification, pattern recognition, and so forth. Both neural networks' synaptic connection weights change as they learn new information.

7 Models comparison

All the above mentioned models gives satisfactory results, below is the results as

Table 2: Models comparison

Model	Accuracy(percentage)	ROC curve area
Bernoulli Naive Bayes	80	80
SVM (Support Vector Machine)	82	82
Logistic Regression	83	83
Neural Network	82	82

As seen from above logistic regression gives the best results, although there's not much difference in accuracy as compared to the other models.

References

[1] Vishal A. Kharde, S.S. Sonawane. Sentiment Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications*, 139, 2016.