

# **ABSTRACT**

Microblogging today has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life everyday in popular websites such as Twitter, Tumblr and Facebook. Thus, companies and media organisation are increasingly seeking ways to mine these social media for information about what people think about their companies and products. Political parties may be interested to know if people support their program . Social organizations may ask people's opinion on current debates. All this information can be obtained from microblogging services, as their users post their opinions on many aspects of their life regularly. In this work, present a method which performs classification of tweets sentiment in Twitter. an model which can determine the sentiment of a tweet.

# INDEX

<b>Sr. No</b>	<b>TOPIC</b>	<b>Pg. No.</b>
<b>1.</b>	<b>SYNOPSIS</b>	<b>1</b>
<b>2.</b>	<b>OBJECTIVE &amp; SCOPE</b>	<b>2</b>
<b>3.</b>	<b>LETRATURE REVIEW</b>	<b>3</b>
<b>4.</b>	<b>INTRODUCTION TO THE PROJECT</b>	<b>6</b>
<b>5.</b>	<b>ANALYSIS OF THE PROJECT</b>	<b>7</b>
<b>6.</b>	<b>METHODOLOGY AND IMPLEMENTATION</b>	<b>8</b>
<b>7.</b>	<b>TESTING</b>	<b>31</b>
<b>8.</b>	<b>FUTURE SCOPE</b>	<b>33</b>
<b>9.</b>	<b>CONCLUSION</b>	<b>33</b>
<b>10.</b>	<b>APPENDIX</b>	<b>34</b>
<b>11.</b>	<b>SCREEN SHOTS</b>	<b>40</b>
<b>12.</b>	<b>REFERENCE</b>	<b>43</b>

# SYNOPSIS

## Hardware Requirements:

- A computer with Internet Connection
- Display device that should support 32-bit color scheme.
- Minimum 1 GB of RAM.

## Software Requirements:

- Anaconda Python ide

There are some general library requirements for the project and some which are specific to individual methods. The general requirements are as follows.

- numpy
- scikit-learn
- scipy
- nltk

# OBJECTIVE

The opinions of others have a significant influence in our daily decision-making process. These decisions range from buying a product such as a smart phone to making investments to choosing a school—all decisions that affect various aspects of our daily life.

Organizations use surveys, opinion polls, and social media as a mechanism to obtain feedback on their products and services. sentiment analysis or opinion mining is the computational study of opinions, sentiments, and emotions expressed in text. The use of sentiment analysis is becoming more widely leveraged because the information it yields can result in the monetization of products and services. By obtaining consumer feedback on a marketing campaign, an organization can measure the campaign's success or learn how to adjust it for greater success. Product feedback is also helpful in building better products, which can have a direct impact on revenue, as well as comparing competitor offerings.

# Scope & LITERATURE REVIEW

Applying sentiment analysis on Twitter is the upcoming trend with researchers recognizing the scientific trials and its potential applications. The challenges unique to this problem area are largely attributed to the dominantly informal tone of the micro blogging.

Pak and Paroubek rationale the use microblogging and more particularly Twitter as a corpus for sentiment analysis. They cited:

- Microblogging platforms are used by different people to express their opinion about different topics, thus it is a valuable source of people's opinions.
- Twitter contains an enormous number of text posts and it grows every day. The collected corpus can be arbitrarily large.
- Twitter's audience varies from regular users to celebrities, company representatives, politicians, and even country presidents. Therefore, it is possible to collect text posts of users from different social and interests groups.
- Twitter's audience is represented by users from many countries.

Parikh and Movassate implemented two Naive Bayes unigram models, a Naive Bayes bigram model and a Maximum Entropy model to classify tweets. They found that the Naive Bayes classifiers worked much better than the Maximum Entropy model could. proposed a solution by using distant supervision, in which their training data consisted of tweets with emoticons. The emoticons served as noisy labels. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM). Their feature space consisted of unigrams, bigrams and POS. The reported that SVM outperformed other models and that unigram were more effective as features. Pak and Paroubek have done similar work but classify the tweets as objective, positive and negative. In order to collect a corpus of objective posts, they retrieved text messages from Twitter accounts of popular

newspapers and magazine, such as “New York Times”, “Washington Posts” etc. Their classifier is based on the multinomial Naïve Bayes classifier that uses N-gram and POS-tags as features. Barbosa too classified tweets as objective or subjective and then the subjective tweets were classified as positive or negative. The feature space used included features of tweets like retweet, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS of words.

Mining for entity opinions in Twitter, Batra and Rao used a dataset of tweets spanning two months starting from June 2009. The dataset has roughly 60 million tweets. The entity was extracted using the Stanford NER, user tags and URLs were used to augment the entities found. A corpus of 200,000 product reviews that had been labeled as positive or negative was used to train the model. Using this corpus the model computed the probability that a given unigram or bigram was being used in a positive context and the probability that it was being used in a negative context. Bifet and Frank used Twitter streaming data provided by Firehouse, which gave all messages from every user in real-time. They experimented with three fast incremental methods that were well-suited to deal with data streams: multinomial naive Bayes, stochastic gradient descent, and the Hoeffding tree. They concluded that SGD-based model, used with an appropriate learning rate was the best.

Agarwal approached the task of mining sentiment from twitter, as a 3-way task of classifying sentiment into positive, negative and neutral classes. They experimented with three types of models: unigram model, a feature based model and a tree kernel based model. For the tree kernel based model they designed a new tree representation for tweets. The feature based model that uses 100 features and the unigram model uses over 10,000 features. They concluded features that combine prior polarity of words with their parts-of-speech tags are most important for the classification task. The tree kernel based model outperformed the other two.

The Sentiment Analysis tasks can be done at several levels of granularity, namely, word level, phrase or sentence level, document level and feature level. As Twitter allows its users to share short pieces of information known as “tweets” (limited to 140 characters), the word level granularity aptly suits its setting. Survey through the literature substantiates that the methods of automatically annotating sentiment at the word level fall into the following two categories:

- dictionary-based approaches
- corpus-based approaches.

Further, to automate sentiment analysis, different approaches have been applied to predict the sentiments of words, expressions or documents. These include Natural Language Processing (NLP) and Machine Learning (ML) algorithms. In our attempt to mine the sentiment from twitter data we introduce a hybrid approach which combines the advantages of both dictionary & corpus based methods along with the combination of NLP & ML based techniques.

# INTRODUCTION

## Problem Statement

Twitter is a popular social networking website where members create and interact with messages known as “tweets”. This serves as a mean for individuals to express their thoughts or feelings about different subjects. Various different parties such as consumers and marketers have done sentiment analysis on such tweets to gather insights into products or to conduct market analysis.

Attempt is to conduct sentiment analysis on “tweets” using various different machine learning algorithms to classify the polarity of the tweet where it is either positive or negative. If the tweet has both positive and negative elements, the more dominant sentiment should be picked as the final label.

Using the dataset which was crawled and labeled positive/negative. The data provided comes with emoticons, usernames and hashtags which are required to be processed and converted into a standard form. also need to extract useful features from the text such unigrams and bigrams which is a form of representation of the “tweet”.

Various machine learning algorithms to conduct sentiment analysis using the extracted features. However, just relying on individual models did not give a high accuracy so picking the top few models to generate a model ensemble. Ensembling is a form of meta learning algorithm technique where we combine different classifiers in order to improve the prediction accuracy.

## Data Description

The data given is in the form of a comma-separated values files with tweets and their corresponding sentiments.

The training dataset is a csv file of type tweet\_id,sentiment,tweet where the tweet\_id is a unique integer identifying the tweet, sentiment is either 1 (positive) or 0 (negative), and tweet is the tweet enclosed in "".

Similarly, the test dataset is a csv file of type tweet\_id,tweet.

The dataset is a mixture of words, emoticons, symbols, URLs and references to people.  
Words



train_dataset	Total	Unique	Average	Max	Positive	Negative
Tweet	509999	-	-	-	298468	211531
User mentions	517993	-	1.0157	12	-	-
Emoticons	5643	-	0.0111	5	5077	566
Urls	15745	-	0.0309	4	-	-
Unigrams	6156858	128005	12.0723	40	-	-
Bigrams	5649572	1304742	11.0776	-	-	-

Table 1: Statistics of preprocessed train dataset

test_dataset	Total	Unique	Average	Max	Positive	Negative
Tweet	498	-	-	-	0	0
User mentions	124	-	0.249	4	-	-
Emoticons	41	-	0.0823	2	27	14
Urls	131	-	0.2631	1	-	-
Unigrams	6214	1948	12.4779	30	-	-
Bigrams	5716	4733	11.4779	-	-	-

Table 2: Statistics of preprocessed test dataset

and emoticons contribute to predicting the sentiment, but URLs and references to people don't. Therefore, URLs and references can be ignored. The words are also a mixture of misspelled words, extra punctuations, and words with many repeated letters. The tweets, therefore, have to be preprocessed to standardize the dataset.

The provided training and test dataset have 509999 and 498 tweets respectively. Preliminary statistical analysis of the contents of datasets, after preprocessing as shown in tables 1 and 2.

# Methodology and Implementation

## Technology Used:

### Python 2.7

It is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

### Numpy

Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the *rank* of the array; the *shape* of an array is a tuple of integers giving the size of the array along each dimension.
- Numpy offers several ways to index into arrays.
- Numpy provides a large set of numeric datatypes that you can use to construct arrays. Numpy tries to guess a datatype when you create an array, but functions that construct arrays usually also include an optional argument to explicitly specify the datatype.
- Basic mathematical functions operate elementwise on arrays.
- Broadcasting is a powerful mechanism that allows numpy to work with arrays of different shapes when performing arithmetic operations.

## **scikit-learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

- **NumPy**: Base n-dimensional array package
- **SciPy**: Fundamental library for scientific computing
- **Matplotlib**: Comprehensive 2D/3D plotting
- **IPython**: Enhanced interactive console
- **Sympy**: Symbolic mathematics
- **Pandas**: Data structures and analysis

Extensions or modules for SciPy are conventionally named SciKits. As such, the module provides learning algorithms and is named scikit-learn.

The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as easy of use, code quality, collaboration, documentation and performance.

Although the interface is Python, c-libraries are leveraged for performance such as numpy for arrays and matrix operations, LAPACK, LibSVM and the careful use of cython.

## **Scipy**

SciPy is a set of open source (BSD licensed) scientific and numerical tools for Python. It currently supports special functions, integration, ordinary differential equation (ODE) solvers, gradient optimization, parallel programming tools, an expression-to-C++ compiler for fast execution, and others. A good rule of thumb is that if it's covered in a general textbook on numerical computing (for example, the well-known Numerical Recipes series), it's probably implemented in scipy.

## **Nltk**

The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing(NLP).

It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. It also includes graphical demonstrations and sample data sets as well as accompanied by a cook book and a book which explains the principles behind the underlying language processing tasks that NLTK supports.

# Methodology

## Pre-processing

Raw tweets scraped from twitter generally result in a noisy dataset. This is due to the casual nature of people's usage of social media. Tweets have certain special characteristics such as retweets, emoticons, user mentions, etc. which have to be suitably extracted. Therefore, raw twitter data has to be normalized to create a dataset which can be easily learned by various classifiers. We have applied an extensive number of pre-processing steps to standardize the dataset and reduce its size. We first do some general pre-processing on tweets which is as follows.

- Convert the tweet to lower case.
- Replace 2 or more dots (.) with space.
- Strip spaces and quotes (" and ') from the ends of tweet.
- Replace 2 or more spaces with a single space.

handling special twitter features as follows.

### URL

Users often share hyperlinks to other webpages in their tweets. Any particular URL is not important for text classification as it would lead to very sparse features. Therefore, we replace all the URLs in tweets with the word URL. The regular expression used to match URLs is ((www\.|(\S+)|(https?:\/\/(\S+))).

### User Mention

Every twitter user has a handle associated with them. Users often mention other users in their tweets by @handle. We replace all user mentions with the word USER\_MENTION. The regular expression used to match user mention is @(\S)+.

### Emoticon

Users often use a number of different emoticons in their tweet to convey different emotions. It is impossible to exhaustively match all the different emoticons used on social media as the number is ever increasing. However, we match some common emoticons which are used very frequently. We replace the matched emoticons with either EMO\_POS or EMO\_NEG depending on whether it is conveying a positive or a negative emotion. A list of all emoticons matched by our method is given in table 3.

Emoticon(s)	Type	Regex	Replacement
:), : ), :-), (:, ( :, (-:, :')	Smile	(:\s?\) :-\) \(\s?: \(-: :\'\))	EMO_POS
:D, : D, :-D, xD, x-D, XD, X-D	Laugh	(:\s?D :-D x-?D X-?D)	EMO_POS
;-), ;), ;-D, ;D, (;, (-;	Wink	(:\s?\( :-\( \)\s?: \)\-:)	EMO_POS
<3, :*	Love	(<3 :\*)	EMO_POS
:-(), : (, :(), :), )-:	Sad	(:\s?\( :-\( \)\s?: \)\-:)	EMO_NEG
:(, :'(, :"(	Cry	(:,\( :\'\( :"\()	EMO_NEG

Table 3: List of emoticons matched by method

## Hashtag

Hashtags are unspaced phrases prefixed by the hash symbol (#) which is frequently used by users to mention a trending topic on twitter. We replace all the hashtags with the words with the hash symbol. For example, #hello is replaced by hello. The regular expression used to match hashtags is #(\S+).

## Retweet

Retweets are tweets which have already been sent by someone else and are shared by other users. Retweets begin with the letters RT. We remove RT from the tweets as it is not an important feature for text classification. The regular expression used to match retweets is \brt\b.

After applying tweet level pre-processing, we processed individual words of tweets as follows.

- Strip any punctuation [!'?!,.():;:] from the word.
- Convert 2 or more letter repetitions to 2 letters. Some people send tweets like *I am soooooo happpppy* adding multiple characters to emphasize on certain words. This is done to handle such tweets by converting them to *I am soo happy*.
- Remove - and '. This is done to handle words like t-shirt and their's by converting them to the more general form tshirt and theirs.
- Check if the word is valid and accept it only if it is. We define a valid word as a word which begins with an alphabet with successive characters being alphabets, numbers or one of dot (.) and underscore(\_).

Some example tweets from the training dataset and their normalized versions are shown in table 4.

Raw	misses Swimming Class. <a href="http://plurk.com/p/12nt0b">http://plurk.com/p/12nt0b</a>
Normalized	misses swimming class URL
Raw	@98PXYRochester HEYYYYYYYYYY!! its Fer from Chile again
Normalized	USER_MENTION heyy its fer from chile again
Raw	Sometimes, You gotta hate #Windows updates.
Normalized	sometimes you gotta hate windows updates
Raw	@Santiago_Steph hii come talk to me i got candy :)
Normalized	USER_MENTION hii come talk to me i got candy EMO_POS
Raw	@bolly47 oh no :( r.i.p. your bella
Normalized	USER_MENTION oh no EMO_NEG r.i.p your bella

Table 4: Example tweets from the dataset and their normalized versions.

## Porter stemmer

The Porter stemming algorithm (or 'Porter **stemmer**') is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems.

**stemming** is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root.

## Stemming

“I really **liked** this movie!”

“I really **like** this movie!”

cats, catlike, catty -> cat

watching, watched -> watch

liked, liking -> lik

## Feature Extraction

We extract two types of features from our dataset, namely unigrams and bigrams. We create a frequency distribution of the unigrams and bigrams present in the dataset and choose top  $N$  unigrams and bigrams for our analysis.

### n-grams

“this movie will knock your socks off!”

### Unigrams

Probably the simplest and the most commonly used features for text classification is the presence of single words or tokens in the the text. We extract single words from the training dataset and create a frequency distribution of these words. A total of 128005 unique words are extracted from the dataset. Out of these words, most of the words at end of frequency spectrum are noise and occur very few times to influence classification. therefore, only use top  $N$  words from these to create our vocabulary where  $N$  is 15000 for sparse vector classification and 90000 for dense vector classification. frequency distribution follows Zipf’s law which states that in a large sample of words, the frequency of a word is inversely proportional to its rank in the frequency table. This can be seen by the fact that a linear trendline with a negative slope fits the plot of  $\log(Frequency)$  vs.  $\log(Rank)$ . The equation of the trendline is  $\log(Frequency) = -0.78\log(Rank) + 13.31$ .

### Bigrams

Bigrams are word pairs in the dataset which occur in succession in the corpus. These features are a good way to model negation in natural language like in the phrase – *This is not good.*

A total of 1304742 unique bigrams were extracted from the dataset. Out of these, most of the bigrams at end of frequency spectrum are noise and occur very few times to influence classification. We therefore use only top 10000 bigrams from these to create our vocabulary.



## Sparse Vector Representation

Depending on whether or not we are using bigram features, the sparse vector representation of each tweet is either of length 15000 (when considering only unigrams) or 25000 (when considering unigrams and bigrams). Each unigram (and bigram) is given a unique index depending on its rank. The feature vector for a tweet has a positive value at the indices of unigrams (and bigrams) which are present in that tweet and zero elsewhere which is why the vector is sparse. The positive value at the indices of unigrams (and bigrams) depends on the feature type we specify which is one of *presence* and *frequency*.

- *presence* In the case of *presence* feature type, the feature vector has a 1 at indices of unigrams (and bigrams) present in a tweet and 0 elsewhere.
- *frequency* In the case of *frequency* feature type, the feature vector has a positive integer at indices of unigrams (and bigrams) which is the frequency of that unigram (or bigram) in the tweet and 0 elsewhere. A matrix of such term-frequency vectors is constructed for the entire training dataset and then each term frequency is scaled by the inverse-document-frequency of the term (idf) to assign higher values to important terms. The inverse-document-frequency of a term  $t$  is defined as.

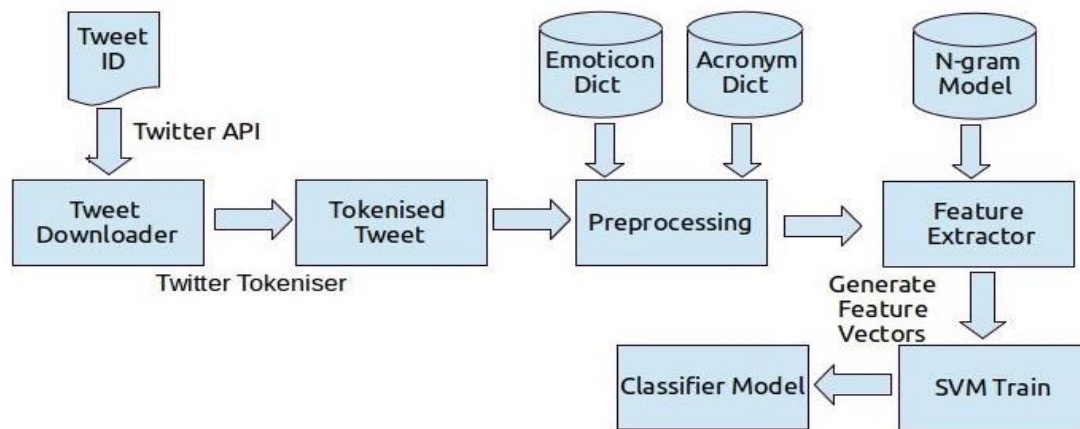
$$idf(t) = \log \left( \frac{1 + n_d}{1 + df(d, t)} \right) + 1$$

where  $n_d$  is the total number of documents and  $df(d, t)$  is the number of documents in which the term  $t$  occurs.

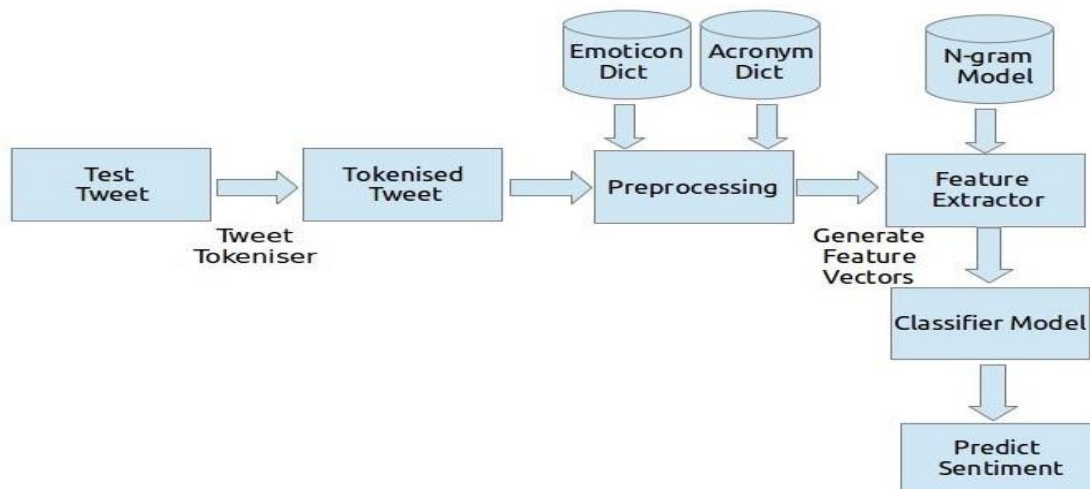
*Handling Memory Issues* Which dealing with sparse vector representations, the feature vector for each tweet is of length 25000 and the total number of tweets in the training set is 509999 which means allocation of memory for a matrix of size 509999×25000. Assuming 4 bytes are required to represent each float value in the matrix, this matrix needs a memory of  $6 \times 10^{10}$  bytes ( $\approx 65$  GB) which is far greater than the memory available in common computer. To tackle this issue, we used **scipy.sparse.lil\_matrix** data structure provided by Scipy which is a memory efficient linked list based implementation of sparse matrices. Also used Python generators wherever possible instead of keeping the entire dataset in memory.

## Dense Vector Representation

For dense vector representation we use a vocabulary of unigrams of size 90000 i.e. the top 90000 words in the dataset. We assign an integer index to each word depending on its rank (starting from 1) which means that the most common word is assigned the number 1, the second most common word is assigned the number 2 and so on. Each tweet is then represented by a vector of these indices which is a dense vector.



Flow Diagram of Training Model



Flow Diagram of Testing Model

# Classifiers

## Baseline

A simple positive and negative word counting method to assign sentiment to a given tweet. We use the Opinion Dataset of positive and negative words to classify tweets. In cases when the number of positive and negative words are equal, we assign positive sentiment. Using this baseline model, we achieve a classification accuracy of 66.24%

## Naive Bayes

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes' Theorem is stated as probability of the event B given A is equal to the probability of the event A given B multiplied by the probability of A upon probability of B

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

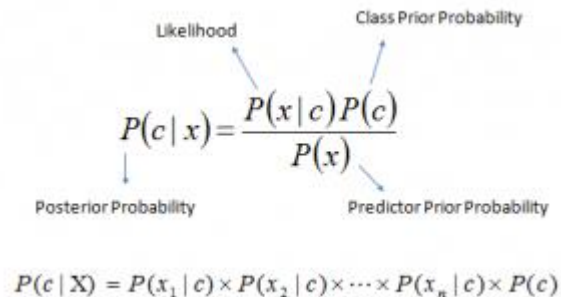
- $P(A|B)$  : conditional probability of occurrence of event A given the event B is true
- $P(A)$  and  $P(B)$  : probabilities of the occurrence of event A and B respectively
- $P(B|A)$  : probability of the occurrence of event B given the event A is true

Bayesian method of probability.

$$\text{Posterior} = \frac{(\text{Likelihood}). (\text{Proposition prior probability})}{\text{Evidence prior probability}}$$

- A is called the proposition and is called the evidence
- P(A) is called the prior probability of proposition and P(B) is called the prior probability of evidence
- P(A|B) is called the Posterior
- P(B|A) is the likelihood

Bayes theorem provides a way of calculating posterior probability  $P(c|x)$  from  $P(c)$ ,  $P(x)$  and  $P(x|c)$ . Look at the equation below:



The diagram shows the equation  $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$  with arrows pointing from labels to the terms. 'Likelihood' points to  $P(x|c)$ , 'Class Prior Probability' points to  $P(c)$ , 'Posterior Probability' points to  $P(c|x)$ , and 'Predictor Prior Probability' points to  $P(x)$ .

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

- $P(c/x)$  is the posterior probability of class (c, target) given predictor (x, attributes).
- $P(c)$  is the prior probability of class.
- $P(x/c)$  is the likelihood which is the probability of predictor given class.
- $P(x)$  is the prior probability of predictor.

## Naive Bayes algorithm working

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
All	5	9
	=5/14	=9/14
	0.36	0.64

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

**Problem:** Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

$$P(\text{Yes} \mid \text{Sunny}) = P(\text{Sunny} \mid \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have  $P(\text{Sunny} \mid \text{Yes}) = 3/9 = 0.33$ ,  $P(\text{Sunny}) = 5/14 = 0.36$ ,  $P(\text{Yes}) = 9/14 = 0.64$

Now,  $P(\text{Yes} \mid \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$ , which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

## Example 2 on sentiments of tweets with naïve bayes

Good: 506

Bad: 507

Goodness:  $506/(506+507) = 0.5$

Badness:  $507/(506+507) = 0.5$

Good: 10

Bad: 14

Goodness:  $10/(14+10) = 0.41$

Badness:  $14/(14+10) = 0.59$

"it's rather like a lifetime special -- pleasant , sweet and forgettable . "

Good: 15

Bad: 6

Goodness:  $15/(6+15) = 0.71$

Badness:  $6/(6+15) = 0.29$

Good: 46

Bad: 22

Goodness:  $46/(46+22) = 0.68$

Badness:  $22/(46+22) = 0.32$

"it's rather like a lifetime special -- pleasant , sweet and forgettable . "



	#GOOD	#BAD	GOODNESS	BADNESS
it's	506	507	0.5	0.5
rather	42	63	0.4	0.6
like	242	396	0.61	0.39
a	3446	3112	0.53	0.47
lifetime	3	5	0.38	0.62
special	29	40	0.42	0.58
pleasant	15	6	0.71	0.29
sweet	46	22	0.68	0.32
and	3198	2371	0.57	0.43
forgettable	10	14	0.42	0.58

"BAG OF WORDS" model



SUM: 5.22 4.8

So we should classify  
this as a **POSITIVE** review!

Naive Bayes Classifier for sentiment analysis of tweets

## Pros and Cons of Naive Bayes

### *Pros:*

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

### *Cons:*

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

## Applications of Naive Bayes Algorithms

- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

## Building a basic model using Naive Bayes in Python

Again, scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of Naive Bayes model under scikit learn library:

- **Gaussian:** It is used in classification and it assumes that features follow a normal distribution.
- **Multinomial:** It is used for discrete counts. For example, let's say, we have a text classification problem. Here we can consider bernoulli trials which is one step further and instead of "word occurring in the document", we have "count how often word occurs in the document", you can think of it as "number of times outcome number  $x_i$  is observed over the  $n$  trials".
- **Bernoulli:** The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with 'bag of words' model where the 1s & 0s are "word occurs in the document" and "word does not occur in the document" respectively.

## SVM

SVM, also known as **support vector machines**, is a non-probabilistic binary linear classifier. For a training set of points  $(x_i, y_i)$  where  $x$  is the feature vector and  $y$  is the class, we want to find the maximum-margin hyperplane that divides the points with  $y_i = 1$  and  $y_i = -1$ . The equation of the hyperplane is as follow

$$w \cdot x - b = 0$$

We want to maximize the margin, denoted by  $\gamma$ , as follows

$$\text{Max } \gamma, s.t. \forall i, \gamma \leq y_i(w \cdot x_i + b)$$

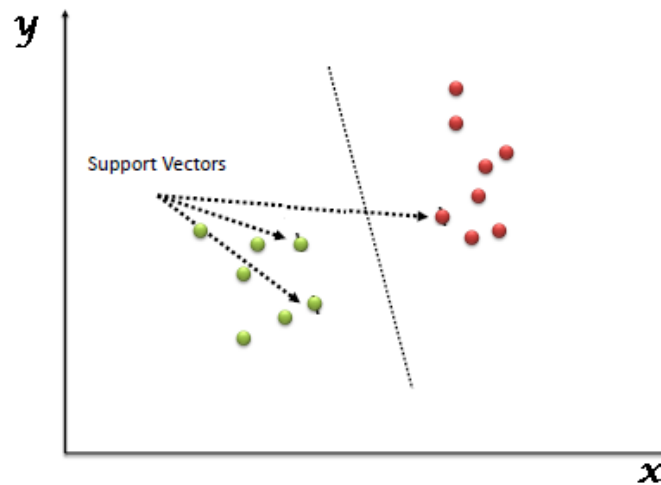
$$w, \gamma$$

in order to separate the points well.

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in  $n$ -dimensional space (where  $n$  is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by



finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

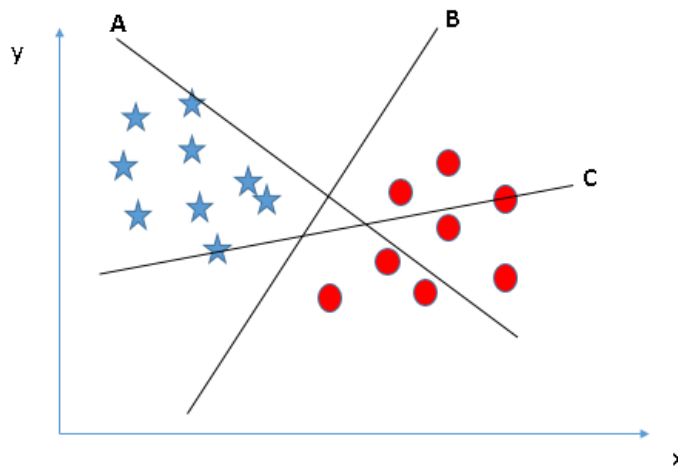
You can look at definition of support vectors and a few examples of its working here.

## How does it work?

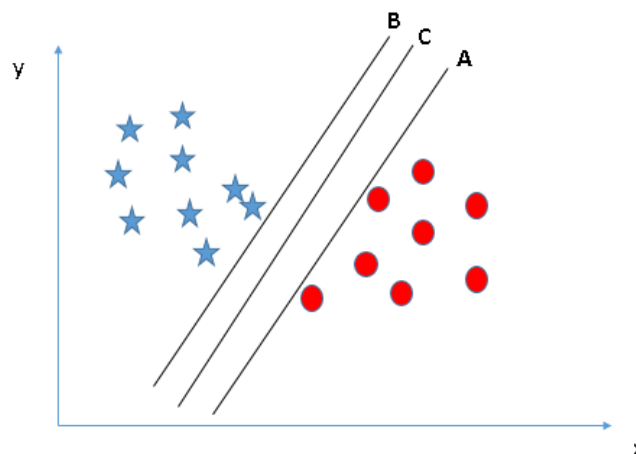
Above, we got accustomed to the process of segregating the two classes with a hyper-plane. Now the question is “How can we identify the right hyper-plane?

Let's understand:

- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.

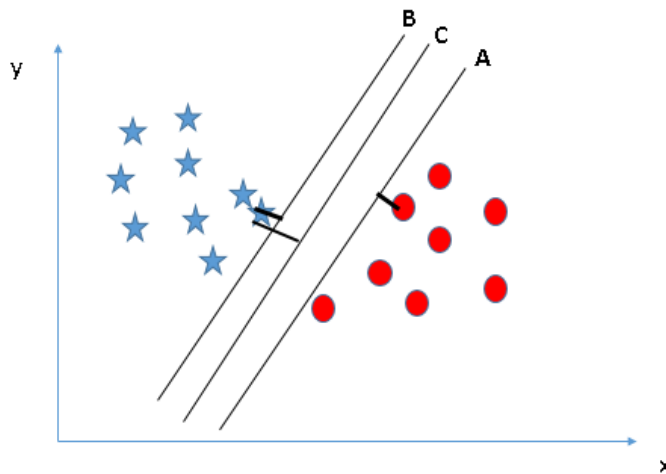


- You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.
- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?



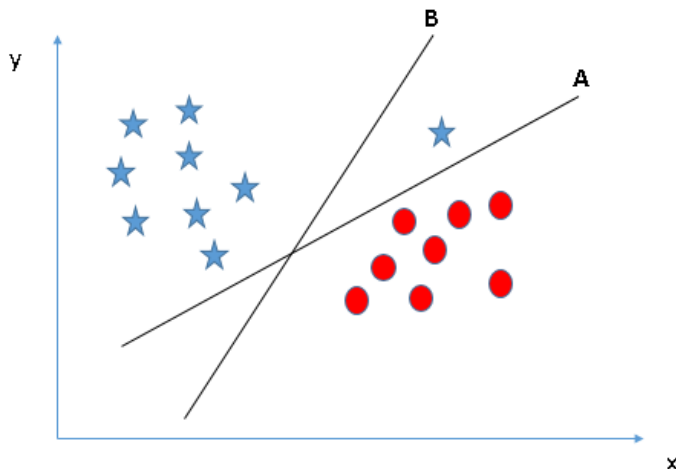
Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**.

Let's look at the below snapshot:



Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

- **Identify the right hyper-plane (Scenario-3):**Hint: Use the rules as discussed in previous section to identify the right hyper-plane



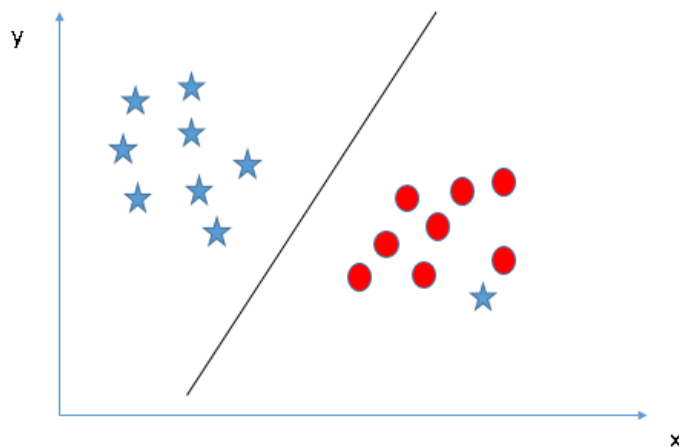
Some of you may have selected the hyper-plane B as it has higher margin compared to A. But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is A.

- **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of star lies in the territory of

other(circle) class as an outlier.

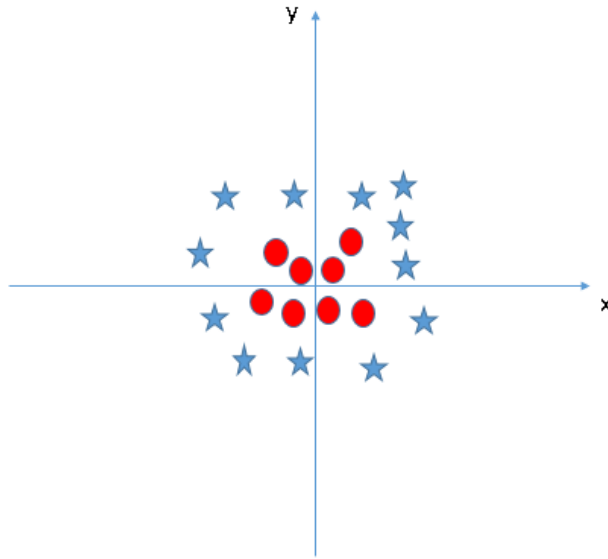


- As I have already mentioned, one star at other end is like an outlier for star class. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.



- **Find the hyper-plane to segregate to classes (Scenario-5):** In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear

hyper-plane.



- SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature  $z = x^2 + y^2$ . Now, let's plot the data points on axis x and z:

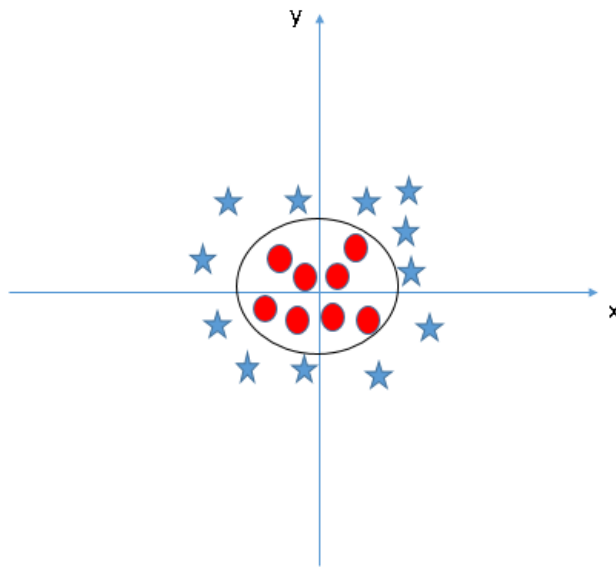


In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

In SVM, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, SVM has a technique called the **kernel trick**. These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then find out the process to separate the data based on the labels or outputs you've defined.

When we look at the hyper-plane in original input space it looks like a circle:



## Pros and Cons associated with SVM

### Pros:

- It works really well with clear margin of separation
- It is effective in high dimensional spaces.
- It is effective in cases where number of dimensions is greater than the number of samples.
- It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

### Cons:

- It doesn't perform well, when we have large data set because the required training time is higher

- It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
- SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

# TESTING

- Testing is an investigation conducted to provide information about the quality of the project.
- Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding bugs.
- Testing can also be stated as the process of validating and verifying that a software program/application/product:
  - meets the business and technical requirements that guided its design and development.
  - works as expected and can be implemented with the same characteristics.
- Different software development models will focus the test effort at different points in the development process.
- A primary purpose of testing is to detect failures so that defects may be discovered and corrected.
- There are many approaches to testing.
  - Reviews, walkthroughs, or inspections are considered as static testing
  - Actually executing programmed code with a given set of test cases is referred to as dynamic testing.



- Dynamic testing takes place when the program itself is used for the first time.
- Software testing is used in association with verification and validation:
  - Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.
  - Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

**Baseline**

For a baseline, we use a simple positive and negative word counting method to assign sentiment to a given tweet. We use the Opinion Dataset of positive and negative words to classify tweets. In cases when the number of positive and negative words are equal, we assign positive sentiment. Using this baseline model, we achieve a classification accuracy of 66.24%

## Naive Bayes

MultinomialNB from `sklearn.naive_bayes` package of *scikit-learn* for Naive Bayes classification. also used Laplace smoothed version of Naive Bayes with the smoothing parameter  $\alpha$  set to its default value of 1. sparse vector representation for classification and ran experiments using both *presence* and *frequency* feature types. We found that *presence* features outperform *frequency* features because Naive Bayes is essentially built to work better on integer features rather than floats. also observed that addition of bigram features improves the accuracy. We obtain a best validation accuracy of 78.62% using Naive Bayes with *frequency* of unigrams and bigrams. A comparison of accuracies obtained on the validation set using different features is shown in table 5

## SVM

Using SVM classifier available in `sklearn`. the C term to be 0.1. C term is the penalty parameter of the error term. In other words, this influences the misclassification on the objective function. We run SVM with both Unigram as well Unigram + Bigram. We also run the configurations with frequency and presence. The best result was 80.12 which came the configuration of frequency and Unigram + Bigram.

Results for non stemmed tweets				
Algorithms	Presence		Frequency	
	Unigrams	Unigrams+Bigrams	Unigrams	Unigrams+Bigrams
Baseline	66.24			
Naive Bayes	77.1863	77.9608	75.9882	78.6235
SVM	77.8275	79.2490	77.9745	80.1235

Table 5: Comparison of various classifiers which are not stemmed in preprocess

Results for stemmed tweets				
Algorithms	Presence		Frequency	
	Unigrams	Unigrams+Bigrams	Unigrams	Unigrams+Bigrams
Baseline	64.51			
Naive Bayes	76.8333	78.1902	75.7765	78.3627
SVM	77.7000	79.9647	77.7627	79.8020

Table 5: Comparison of various classifiers which are stemmed in preprocess

## Future Scope

## Conclusion

### Summary of achievements

The provided tweets were a mixture of words, emoticons, URLs, hastags, user mentions, and symbols. Before training the pre-process the tweets to make it suitable for feeding into models.

implemented several machine learning algorithms like Naive Bayes, SVM, to classify the polarity of the tweet. We used two types of features namely unigrams and bigrams for classification and observes that augmenting the feature vector with bigrams improved the accuracy. Once the feature has been extracted it was represented as either a sparse vector or a dense vector. It has been observed that *presence* in the sparse vector representation recorded a better performance than *frequency*.

And classifies using SVM performed slightly better finally the predictions of 2 of models achieving an accuracy of 80.12 %.

### Future directions

- *Handling emotion ranges*: We can improve and train our models to handle a range of sentiments. Tweets don't always have positive or negative sentiment. At times they may have no sentiment i.e. neutral. Sentiment can also have gradations like the sentence, *This is good*, is positive but the sentence, *This is extraordinary*. is somewhat more positive than the first. We can therefore classify the sentiment in ranges, say from -2 to +2.
- *Using symbols*: During our pre-processing, we discard most of the symbols like commas, full-stops, and exclamation mark. These symbols may be helpful in assigning sentiment to a sentence.
- we can also implement several other machine learning algorithms like Maximum Entropy, Decision Tree, Random Forest, XGBoost, Multi-Layer Perceptron, Recurrent Neural networks and Convolutional Neural Networks to classify the polarity of the tweet.

## Appendix

# User Manual

The training dataset is expected to be a csv file of type tweet\_id,sentiment,tweet where the tweet\_id is a unique integer identifying the tweet, sentiment is either 1 (positive) or 0 (negative), and tweet is the tweet enclosed in "".

## Test.csv

```

1,0,          is so sad for my APL friend.....,
2,0,          I missed the New Moon trailer...,
3,1,          omg its already 7:30 :O,
4,0,          .. Omgaga. Im sooo im gunna CRy. I've been at this dentist since
11,. I was suposed 2 just get a crown put on (30mins)...,
5,0,          i think mi bf is cheating on me!!!      T_I,
6,0,          or i just worry too much?              ,
7,1,          Juuuuuuuuuuuuuuuuusssst Chillin!!,,
8,0,          Sunny Again           Work Tomorrow :-|       TV Tonight,
9,1,          handed in my uniform today . i miss you already,
10,1,         hmmm.... i wonder how she my number @-),
11,0,         I must think about positive..,
12,1,         thanks to all the haters up in my face all day! 112-102,
13,0,         this weekend has sucked so far,
14,0,         jb isnt showing in australia any more!,
15,0,         ok thats it you win.,
16,0,         &lt;----- This is the way i feel right now...,
17,0,"        awwhhe man.... I'm completely useless rt now. Funny, all I can do is

```

Similarly, the test dataset is a csv file of type tweet\_id,tweet. Please note that csv headers are not expected and should be removed from the training and test datasets.

**train.csv**

- 1,"@stellargirl I looooooovvvvvveee my Kindle2. Not that the DX is cool, but the 2 is fantastic in its own right."
- 2,Reading my kindle2... Love it... Lee childs is good read.
- 3,"Ok, first assesment of the #kindle2 ...it fucking rocks!!!"
- 4,@kenburbary You'll love your Kindle2. I've had mine for a few months and never looked back. The new big one is huge! No need for remorse! :)
- 5,@mikefish Fair enough. But i have the Kindle2 and I think it's perfect :)
- 6,@richardebaker no. it is too big. I'm quite happy with the Kindle2.
- 7,Fuck this economy. I hate aig and their non loan given asses.
- 8,Jquery is my new best friend.
- 9,Loves twitter
- 10,how can you not love Obama? he makes jokes about himself.
- 11,Check this video out -- President Obama at the White House Correspondents' Dinner <http://bit.ly/IMXUM>
- 12,"@Karoli I firmly believe that Obama/Pelosi have ZERO desire to be civil. It's a charade and a slogan, but they want to destroy conservatism"
- 13,"House Correspondents dinner was last night whoopi, barbara & sherri went, Obama got a standing ovation"
- 14,Watchin Espn..Jus seen this new Nike Commerical with a Puppet Lebron..sh\*t was hilarious...LMAO!!!
- 15,"dear nike, stop with the flywire. that shit is a waste of science. and ugly. love, @vincentx24x"
- 16,"#lebron best athlete of our generation, if not all time (basketball related) I don't want to get into inter-sport debates about \_\_1/2"
- 17,I was talking to this guy last night and he was telling me that he is a die hard Spurs fan. He also told me that he hates LeBron James.
- 18.i love lebron. <http://bit.ly/PdHUr>

## Preprocessing

1. Run preprocess.py <raw-csv-path> on both train and test data. This will generate a preprocessed version of the dataset.

## Preprocessing and stemming of both dataset csv (Test,Train)

**test-processed-stemmed.csv**

```
1,user_ment i loovve my kindle2 not that the dx is cool but the is fantast in  
it own right  
2,read my kindle2 love it lee child is good read  
3,ok first asses of the kindle2 it fuck rock  
4,user_ment youll love your kindle2 ive had mine for a few month and never look  
back the new big one is huge no need for remors emo_po  
5,user_ment fair enough but i have the kindle2 and i think it perfect emo_po  
6,user_ment no it is too big im quit happi with the kindle2  
7,fuck thi economi i hate aig and their non loan given ass  
8,jqueri is my new best friend  
9,love twitter  
10,how can you not love obama he make joke about himself  
11,check thi video out presid obama at the white hous correspond dinner url  
12,user_ment i firmlly believ that have zero desir to be civil it a charad and a  
slogan but they want to destroy conservat  
13,hous correspond dinner wa last night whoopi barbara sherri went obama got a  
stand ovat  
14,watchin espn ju seen thi new nike commer with a puppet lebron wa hilari lmao  
15,dear nike stop with the flywir that shit is a wast of scienc and ugly love  
user_ment  
16,lebron best athlet of our gener if not all time basketbal relat i dont want  
to get into intersport debat about  
17,i wa talk to thi guy last night and he wa tell me that he is a die hard spur  
fan he also told me that he hate lebron jame  
18,i love lebron url  
19,user_ment lebron is a beast but im still cheer the a til the end  
20.user ment lebron is the boss
```

## Train-processed-stemmed.csv

1,0,is so sad for my apl friend  
2,0,i miss the new moon trailer  
3,1,omg it already o  
4,0,ongaga im soo im gunna cri ive been at thi dentist sinc i wa supos just get  
a crown put on  
5,0,i think mi bf is cheat on me t\_t  
6,0,or i just worri too much  
7,1,juusst chillin  
8,0,sunni again work tomorrow tv tonight  
9,1,hand in my uniform today i miss you already  
10,1,hmm i wonder how she my number user\_ment  
11,0,i must think about posit  
12,1,thank to all the hater up in my face all day  
13,0,thi weekend ha suck so far  
14,0,jb isnt show in australia ani more  
15,0,ok that it you win  
16,0,thi is the way i feel right now  
17,0,awhh man im complet useless now funni all i can do is twitter url  
18,1,feel strang fine now im gonna go listen to some semison to celebr  
19,0,huge roll of thunder just now so scari  
20,0,i just cut my beard off it onli been grow for well over a year im gonna  
start it over user\_ment is happi in the meantime  
21,0,veri sad about iran  
22,0,wompp wompp  
23,1,your the onli one who can see thi caus no one els is follow me thi is for  
you becaus your pretti awesome  
24,0,lmao i am with a new blog tweet on weeb and my cats shut down

2. Run stats.py <preprocessed-csv-path> where <preprocessed-csv-path> is the path of csv generated from preprocess.py. This gives general statistical information about the dataset and will produce two pickle files which are the frequency distribution of unigrams and bigrams in the training dataset.

```
dhaval@ubuntu:~/myproject/tsa$ python pretrain.py train.csv
Processing 509999/509999
Saved processed tweets to: train-processed.csv
dhaval@ubuntu:~/myproject/tsa$ python stats.py train-processed.csv
Processing 509999/509999
Calculating frequency distribution
Saved uni-frequency distribution to train-processed-freqdist.pkl
Saved bi-frequency distribution to train-processed-freqdist-bi.pkl

[Analysis Statistics]
Tweets => Total: 509999, Positive: 298468, Negative: 211531
User Mentions => Total: 517993, Avg: 1.0157, Max: 12
URLs => Total: 15745, Avg: 0.0309, Max: 4
Emojis => Total: 5643, Positive: 5077, Negative: 566, Avg: 0.0111, Max: 5
Words => Total: 6156858, Unique: 128005, Avg: 12.0723, Max: 40, Min: 0
Bigrams => Total: 5649572, Unique: 1304742, Avg: 11.0776
dhaval@ubuntu:~/myproject/tsa$ █
```

```
dhaval@ubuntu:~/myproject/tsa$ python pretest.py test.csv
Processing 498/498
Saved processed tweets to: test-processed.csv
dhaval@ubuntu:~/myproject/tsa$ python stats.py test-processed.csv
Processing 498/498
Calculating frequency distribution
Saved uni-frequency distribution to test-processed-freqdist.pkl
Saved bi-frequency distribution to test-processed-freqdist-bi.pkl

[Analysis Statistics]
Tweets => Total: 498, Positive: 0, Negative: 0
User Mentions => Total: 124, Avg: 0.2490, Max: 4
URLs => Total: 131, Avg: 0.2631, Max: 1
Emojis => Total: 41, Positive: 27, Negative: 14, Avg: 0.0823, Max: 2
Words => Total: 6214, Unique: 1948, Avg: 12.4779, Max: 30, Min: 2
Bigrams => Total: 5716, Unique: 4733, Avg: 11.4779
dhaval@ubuntu:~/myproject/tsa$
```

Above screenshots show preprocessing and Statistic result of train and test data  
also genrating pickle of unigram and bigram frequency disturbution

A Pkl file is a file created by pickle, a Python module that enables objects to be serialized to files on disk and deserialized back into the program at runtime. It contains a byte stream that represents the objects.

The process of serialization is called "pickling," and deserialization is called "unpickling." A PKL file is pickled to save space when being stored or transferred over a network then is unpickled and loaded back into program memory during runtime. The PKL file is created using Python pickle and the dump() method and is loaded using Python pickle and the load() method.

After the above steps, you should have four files in total: <preprocessed-train-csv>, <preprocessed-test-csv>, <freqdist>, and <freqdist-bi> which are preprocessed train dataset, preprocessed test dataset, frequency distribution of unigrams and frequency distribution of bigrams respectively.

Values of USE\_BIGRAMS and FEAT\_TYPE can be changed to obtain results using different types of features.

## Baseline

3. Run baseline.py. With TRAIN = True it will show the accuracy results on training dataset.
4. With TRAIN= False (testing) it will predict the sentiment and save them in file named baseline. Csv

```
dhaval@ubuntu:~/myproject/tsa$ python baseline.py
Correct = 66.24%
dhaval@ubuntu:~/myproject/tsa$ python baseline.py

Prediction Saved to baseline.csv
```

## Naive Bayes

5. Run naivebayes.py. With TRAIN = True it will show the accuracy results on 10% validation dataset.

```
dhaval@ubuntu:~/myproject/tsa$ python naivebayes.py
Generating feature vectors
Processing 509999/509999

Extracting features & training batches

Testing
Processing 1/1
Correct: 40010/51000 = 78.4510 %
dhaval@ubuntu:~/myproject/tsa$
```

6. With TRAIN = False (testing) it will predict the sentiment and save them in file named naivebayes. Csv

```
dhaval@ubuntu:~/myproject/tsa$ python naivebayes.py
Generating feature vectors
Processing 509999/509999

Extracting features & training batches

Testing
Generating feature vectors
Processing 498/498

Predicting batches
Processing 1/1
Saved to naivebayes.csv
dhaval@ubuntu:~/myproject/tsa$ █
```



## SVM

7. Run svm.py. With TRAIN = True it will show the accuracy results on 10% validation dataset.

```
dhaval@ubuntu:~/myproject/tsa$ python svm.py
Generating feature vectors
Processing 509999/509999

Extracting features & training batches

Testing
Processing 1/1
Correct: 40703/51000 = 79.8098 %
dhaval@ubuntu:~/myproject/tsa$ █
```

8. With TRAIN = False (testing) it will predict the sentiment and save them in file named svm.csv

```
dhaval@ubuntu:~/myproject/tsa$ python svm.py
Generating feature vectors
Processing 509999/509999

Extracting features & training batches

Testing
Generating feature vectors
Processing 498/498

Predicting batches
Processing 1/1
Saved to svm.csv
```

# ScreenShots(Results)

## Baseline model prediction

USER_MENTION i loovvee my kindle2 not that the dx is cool but the is fantastic in its own right	1
reading my kindle2 love it lee child's is good read	1
ok first assesment of the kindle2 it fucking rocks	1
USER_MENTION youll love your kindle2 ive had mine for a few months and never looked back the new big one is huge no need for remorse EMO_POS	1
USER_MENTION fair enough but i have the kindle2 and i think its perfect EMO_POS	1
USER_MENTION no it is too big im quite happy with the kindle2	1
fuck this economy i hate aig and their non loan given asses	0
jquery is my new best friend	1
loves twitter	1
how can you not love obama he makes jokes about himself	1
check this video out president obama at the white house correspondents dinner URL	1
USER_MENTION i firmly believe that have zero desire to be civil its a charade and a slogan but they want to destroy conservatism	0
house correspondents dinner was last night whoopi barbara sherr went obama got a standing ovation	1
watchin espn jus seen this new nike commerical with a puppet lebron was hilarious lmao	1
dear nike stop with the flywire that shit is a waste of science and ugly love USER_MENTION	0
lebron best athlete of our generation if not all time basketball related i dont want to get into intersport debates about	1
i was talking to this guy last night and he was telling me that he is a die hard spurs fan he also told me that he hates lebron james	0
i love lebron URL	1
USER_MENTION lebron is a beast but im still cheering the a til the end	1
USER_MENTION lebron is the boss	1
USER_MENTION lebron is a hometown hero to me lol i love the lakers but lets go cavs lol	1
lebron and zydrunas are such an awesome duo	1
USER_MENTION lebron is a beast nobody in the nba comes even close	1
downloading apps for my iphone so much fun EMO_POS there literally is an app for just about anything	1
good news just had a call from the visa office saying everything is fine what a relief i am sick of scams out there stealing	1
URL awesome come back from USER_MENTION via USER_MENTION	1
in montreal for a long weekend of much needed	1
booz allen hamilton has a bad ass homegrown social collaboration platform way cool ttiv	1
customer innovation award winner booz allen hamilton URL	1
USER_MENTION i current use the nikon d90 and love it but not as much as the canon i chose the d90 for the video feature my mistake	1
need suggestions for a good ir filter for my canon got some pls dm	1
USER_MENTION i just checked my google for my business blip shows up as the second entry huh is that a good or ba URL	1
USER_MENTION google is always a good place to look shouldve mentioned i worked on the mustang my dad USER_MENTION	1
played with an android google phone the slide out screen scares me i would break that fucker so fast still prefer my iphone	1
us planning to resume the military tribunals at Guantanamo Bay only this time those on trial will be aig execs and chrysler debt holders	0
omg so bored my tattoos are so itchy help aha	0
in itchy and miserable	0

## Naïve bayes prediction

USER_MENTION i loovvee my kindle2 not that the dx is cool but the is fantastic in its own right	1
reading my kindle2 love it lee child's is good read	1
ok first assesment of the kindle2 it fucking rocks	1
USER_MENTION youll love your kindle2 ive had mine for a few months and never looked back the new big one is huge no need for remorse EMO_POS	1
USER_MENTION fair enough but i have the kindle2 and i think its perfect EMO_POS	1
USER_MENTION no it is too big im quite happy with the kindle2	1
fuck this economy i hate aig and their non loan given asses	0
jquery is my new best friend	1
loves twitter	1
how can you not love obama he makes jokes about himself	1
check this video out president obama at the white house correspondents dinner URL	1
USER_MENTION i firmly believe that have zero desire to be civil its a charade and a slogan but they want to destroy conservatism	1
house correspondents dinner was last night whoopi barbara sherri went obama got a standing ovation	1
watchin espn jus seen this new nike commerical with a puppet lebron was hilarious lmao	1
dear nike stop with the flywire that shit is a waste of science and ugly love USER_MENTION	0
lebron best athlete of our generation if not all time basketball related i dont want to get into intersport debates about	0
i was talking to this guy last night and he was telling me that he is a die hard spurs fan he also told me that he hates lebron james	0
i love lebron URL	1
USER_MENTION lebron is a beast but im still cheering the a til the end	0
USER_MENTION lebron is the boss	1
USER_MENTION lebron is a hometown hero to me lol i love the lakers but lets go cavs lol	1
lebron and zydrunas are such an awesome duo	1
USER_MENTION lebron is a beast nobody in the nba comes even close	1
downloading apps for my iphone so much fun EMO_POS there literally is an app for just about anything	1
good news just had a call from the visa office saying everything is fine what a relief i am sick of scams out there stealing	1
URL awesome come back from USER_MENTION via USER_MENTION	1
in montreal for a long weekend of much needed	1
booz allen hamilton has a bad ass homegrown social collaboration platform way cool ttiv	1
customer innovation award winner booz allen hamilton URL	1
USER_MENTION i current use the nikon d90 and love it but not as much as the canon i chose the d90 for the video feature my mistake	1
need suggestions for a good ir filter for my canon got some pls dm	1
USER_MENTION i just checked my google for my business blip shows up as the second entry huh is that a good or ba URL	1
USER_MENTION google is always a good place to look shouldve mentioned i worked on the mustang my dad USER_MENTION	1
played with an android google phone the slide out screen scares me i would break that fucker so fast still prefer my iphone	0
us planning to resume the military tribunals at quantanamo bay only this time those on trial will be aig execs and chrysler debt holders	1
mg so bored my tattoos are so itchy help aha	0

## SVM prediction

{USER_MENTION i loovvee my kindle2 not that the dx is cool but the is fantastic in its own right	1
{reading my kindle2 love it lee childs is good read	1
{ok first assesment of the kindle2 it fucking rocks	1
{USER_MENTION youll love your kindle2 ive had mine for a few months and never looked back the new big one is huge no need for remorse EMO_POS	1
{USER_MENTION fair enough but i have the kindle2 and i think its perfect EMO_POS	1
{USER_MENTION no it is too big im quite happy with the kindle2	1
{fuck this economy i hate aig and their non loan given asses	0
{jquery is my new best friend	1
{loves twitter	1
{how can you not love obama he makes jokes about himself	1
{check this video out president obama at the white house correspondents dinner URL	1
{USER_MENTION i firmly believe that have zero desire to be civil its a charade and a slogan but they want to destroy conservatism	0
{house correspondents dinner was last night whoopi barbara sherri went obama got a standing ovation	0
{watchin espn jus seen this new nike commerical with a puppet lebron was hilarious lmao	1
{dear nike stop with the flywire that shit is a waste of science and ugly love USER_MENTION	0
{lebron best athlete of our generation if not all time basketball related i dont want to get into intersport debates about	1
{i was talking to this guy last night and he was telling me that he is a die hard spurs fan he also told me that he hates lebron james	0
{i love lebron URL	1
{USER_MENTION lebron is a beast but im still cheering the a til the end	0
{USER_MENTION lebron is the boss	1
{USER_MENTION lebron is a hometown hero to me lol i love the lakers but lets go cavs lol	1
{lebron and zydrunas are such an awesome duo	1
{USER_MENTION lebron is a beast nobody in the nba comes even close	0
{downloading apps for my iphone so much fun EMO_POS there literally is an app for just about anything	1
{good news just had a call from the visa office saying everything is fine what a relief i am sick of scams out there stealing	1
{URLawesome come back from USER_MENTION via USER_MENTION	1
{in montreal for a long weekend of much needed	1
{booz allen hamilton has a bad ass homegrown social collaboration platform way cool ttiv	1
{customer innovation award winner booz allen hamilton URL	1
{USER_MENTION i current use the nikon d90 and love it but not as much as the canon i chose the d90 for the video feature my mistake	1
{need suggestions for a good ir filter for my canon got some pls dm	1
{USER_MENTION i just checked my google for my business blip shows up as the second entry huh is that a good or ba URL	1
{USER_MENTION google is always a good place to look shouldve mentioned i worked on the mustang my dad USER_MENTION	1
{played with an android google phone the slide out screen scares me i would break that fucker so fast still prefer my iphone	0
{us planning to resume the military tribunals at Guantanamo bay only this time those on trial will be aig execs and chrysler debt holders	1
{img so bored my tattoos are so itchy help aha	0