iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.43806

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Analysis of Big Data Methodology for Pattern Recognition

Dr. D. R. Ingle[1], Mr. Samirkumar R. Waghmare[2]

Abstract: Sentiment Analysis (SA) is an ongoing field of research in text mining field. SA is the computational treatment of opinions, sentiments and subjectivity of text. This survey paper tackles a comprehensive overview of the last update in this field. Many recently proposed algorithms' enhancements and various SA applications are investigated and presented briefly in this survey. These articles are categorized according to their contributions in the various SA techniques. The related fields to SA (transfer learning, emotion detection, and building resources) that attracted researchers recently are discussed.

I. INTRODUCTION

Sentiment analysis, also refers as opinion mining, is a sub-AI task where we need to figure out which is the overall opinion of a given record. Utilizing AI strategies and normal language handling we can extricate the abstract data of a report and attempt to group it as indicated by its extremity like positive, unbiased or negative. It is a truly valuable examination since we might actually decide the general assessment on a selling object, or anticipate securities exchanges for a given organization like, assuming a great many people figure sure about it, perhaps its financial exchanges will increment, etc. Feeling investigation is a long way from to be settled since the language is extremely mind boggling (objectivity/subjectivity, refutation, jargon, grammar,) yet it is additionally why it is exceptionally fascinating to dealingwith.

In this project I decide to attempt to characterize tweets from Twitter into "positive" or "negative" feeling by building a model in light of probabilities. Twitter is a microblogging site where individuals can talk about their thoughts rapidly and precipitously by sending a tweets restricted by 140 characters. You can straightforwardly address a tweet to somebody by adding the objective sign "@" or take an interest to a subject by adding a hashtag "#" to your tweet. Due to the use of Twitter, it is an ideal wellspring of information to decide the current generally assessment on anything.

A. Context

This project has been done as a part of my course for the MSc Information Technology at Hong Kong University of Science and Technology. Supervised by Dr David Rossiter, I had three months to fulfill the requirements in order to succeed the module. Every three weeks, a meeting was organized to show and report my progress and fix the next objectives.

B. Motivations

Being extremely interested in everything having a relation with the Machine Learning, the independent project was a great occasion to give me the time to learn and confirm my interest for this field. The fact that we can make estimations, predictions and give the ability for machines to learn by themselves is both powerful and limitless in term of application possibilities. We can use Machine Learning in Finance, Medicine, almost everywhere. That’s why I decided to conduct my project around the Machine Learning.

C. Idea

This project was motivated by my desire to investigate the sentiment analysis field of machine learning since it allows to approach natural language processing which is a very hot topic actually. Following my previous experience where it was about classifying short music according to their emotion, I applied the same idea with tweets and try to figure out which is positive or negative.

D. Sources

Because I truly think that sharing sources and knowledges allow to help others but also ourselves, the sources of the project are available at the following link: https://github.com/marclamberti/TwitterEmotionAnalysisFeel free to give me your point of view or ideas for anything you want. I used ipython notebook which is very useful to understand the entire process of my project since you can follow each step with the corresponding code. Hereis the direct link to it:
http://nbviewer.ipython.org/github/marclamberti/TwitterEmotionAnalysis/blob/master/TwitterSentimentAnalysis.ipynb

©IJRASET: All Rights are Reserved

II. LITERATURE REVIEW

Concerning homophily and correspondence investigation, [12] discovered that the top clients by the quantity of adherents are for the most part famous people and broad communications, and the majority of them don't follow back their devotees. A low degree of correspondence had been noticed; 77.9% of clients sets are associated one-way, and just 22.1% of clients have proportional connection between them. [6] Additionally showed a low correspondence in their examination of the devotee chart. Generally, 20% of clients have equal relationship. Both [6], [12] concurred that twitter is a wellspring of data than a long range interpersonal communication. [5] Observed that bloggers spread data more than different classifications like famous people, media, or associations. [3], [19] found that utilizing hashtags in tweets works on the exactness and the presentation of the examination. [5], [13] found that the political hashtags persevered more timeframe than different ones, and that implies a higher recurrence of tweets throughout an extensive stretch oftime.

The timeframe for each hashtag should be predictable. For instance, while creeping political hashtags, each hashtag should be estimated yearly, month to month, week after week, or every day. Not at all like [5]'s aggressive, however imperfect, investigation where the time breaks for the five hashtags in study have different time breaks, in which they estimated #FreeIran and #FreeVenzuela consistently, #25Jan and #OccupyWallSt on everyday schedule, and #SpanishRevolution on month to month premise. Subsequently, it was essential to set a steady time measure, as the point classification was something very similar.

Persuasive clients on twitter may not really be legislators, superstars, or activists, they can be customary clients, on the other hand to [5]'s discoveries. They looked like the movement on twitter to the Pamphleteering activity (an authentic term for somebody who makes or appropriates Leaflets, where flyers used to communicate the essayist's perspectives [21]). In pamphleteering the political activists keep pamphlets since they are the main powerfulindividuals.

Ref. [8] observed that it is simpler to spread instant messages than photograph messages. This implies that clients are concerned more with data sharing than speaking with different clients. Likewise answering to making it known messages was essentially more than conventional messages, that is to say, clients talk about and share data and thoughts towards a particular theme more than participating in discussions. That brought about expanding the organization of clients in making it known occasions.The examinations in [9] ought to have been performed to construe the most dynamic and contributing clients. In any case, it would be profitable on the off chance that they have recognized the retweet rate and the organization geography of the dynamic clients to look at their impact, and to resolve the subject of the connection between being dynamic and being powerful. Also, the techniques in [13] miss the mark on calculated conduct of impact, as the pace of tweets and the date of joining are not signs of being powerful. Likewise, being persuasive in the past doesn't be guaranteed to mean being compelling as of now or future.

Ref. [2], [19] showed that there is no solid connection between the retweet rate and the organization geography as a little level of retweeted endlessly messages with specifies were between interconnected clients. [2] Tracked down that on account of hard-political news (legislative issues, financial aspects, violations, and calamities) hash tags, the retweet rate was higher between interconnected clients. Dissimilar to [2]; [19] found that the organization geography isn't the primary element in breaking down re-tweet ability. Extra investigation in [10] showed that the substance of messages assumed a solid part in the message engendering.

In addition, [18] showed that utilizing the notable "geo-labelled" include in twitter to recognize the extremity of a political up-and-comers in the US should be possible by utilizing the opinion investigation calculations to anticipate the future occasions, for example, the official races results. Contrasting with past methodologies in feeling points, extra discoveries by [20] showed that adding the semantic element delivers better Review (recovered reports) [22] and F-Score (a proportion of a test exactness as it considers both the accuracy and the review of the test to register the score) in pessimistic opinion order [23], see (1), (2), (3). It likewise created better Accuracy (the pertinent records) [24] and F-Score in sure feeling arrangement. [6] Observed that utilizing AI calculations, for example, (Guileless Bayes, Greatest Entropy, and SVM) have more exact outcomes (more than 80%) while preparing the emoji information alongside twitter messages. Utilizing the weighted F-Measure to gauge the exactness of the feeling investigation would aid more precise outcomes. F2 measure weighs review two times however much accuracy and F0.5 weighs accuracy two times as much as review [25]. Ref. [20] Utilized F-Score to quantify the precision of their opinion examination.

III. SENTIMENTANALYSIS

Sentiment analysis, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive, neutral or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling objects, or predict stock markets for a given company

like, if most people think positive about it, possibly its stock markets will increase, and so on.
Sentiment analysis is actually far from to be solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar,...) but it is also why it is very interesting to workingon.
In this project I choose to try to classify tweets from Twitter into “positive” or “negative” sentiment by building a model based on probabilities. Twitter is a micro blogging website where people can share their feelings quickly and spontaneously by sending a tweets limited by 140 characters. You can directly address a tweet to someone by adding the target sign “@” or participate to a topic by adding anhastag “#” to your tweet. Because of the usage of Twitter, it is a perfect source of data to determine the current overall opinion about anything.

A. Data

To gather the data many options are possible. In some previous paper researches, they built a program to collect automatically a corpus of tweets based on two classes, “positive” and “negative”, by querying Twitter with two type of emoticons:

3.1 Happyemoticons, suchas“:)”,“:P”,“:-)”etc.
- Sad emoticons, such as “:(“, “:’(”,“=(“.

Others make their own dataset of tweets my collecting and annotating them manually which very long and fastidious.
Additionally to find a way of getting a corpus of tweets, we need to take of having a balanced dataset,meaningweshouldhaveanequalnumberofpositiveandnegativetweets, butit needsalsotobe large enough. Indeed, more the data we have, more we can train our classifier and more the accuracy willbe.
After many researches, I found a dataset of 1578612 tweets in english coming from two sources: Kaggle and Sentiment140. It is composed of four columns that are Item ID, Sentiment, Sentiment Source and Sentiment Text. We are only interested by the Sentiment column corresponding to our label class taking a binary value, 0 if the tweet is negative, 1 if the tweet is positive and the Sentiment Text columns containing the tweets in a raw format.

	ItemID	Sentiment	SentimentSource	SentimentText
0	1	0	Sentiment140	is so sad for my APL friend.....
1	2	0	Sentiment140	I missed the New Moon trailer...
2	3	1	Sentiment140	omg its already 7:30 :O
3	4	0	Sentiment140	.. 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)...
4	5	0	Sentiment140	i think mi bf is cheating on me!!! T_T
5	6	0	Sentiment140	or i just worry too much?
6	7	1	Sentiment140	Juuuuuuuuuuuuuuussssst Chillin!!
7	8	0	Sentiment140	Sunny Again Work Tomorrow :-| TV Tonight
8	9	1	Sentiment140	handed in my uniform today . i miss you already
9	10	1	Sentiment140	hmmm.... I wonder how she my number @-)

Table 3.1.1: Example of twitter posts annotated with their corresponding sentiment, 0 if it is negative, 1 if it is positive.

In the *Table 3.1.1* showing the first ten twitter posts we can already notice some particularities and difficulties that we are going to encounter during the preprocessing steps.

- The presence of **acronyms** "bf" or more complicated "APL". Does it means apple ? Apple (the company) ? In this context we have "friend" after so we could think that he refers to his smartphoneandsoApple,butwhataboutiftheword"friend"wasnothere?
- The presence of **sequences of repeated characters** such as "Juuuuuuuuuuuuuuuussssst", "hmmmm". In general when we repeat several characters in a word, it is to emphasize it, to increase itsimpact.
- Thepresenceof**emoticons**,":O","T_T",":•|"andmuchmore,giveinsightsaboutuser's moods.
- **Spellingmistakes**and“**urbangrammar**”like"imgunna"or"mi".
- Peoplealsoindicatetheirmoods,emotions,states,betweentwo*suchas*,*\cries*, *hummin*, *sigh*.
- The negation, “can't”, “cannot”, “don't”, “haven't” that we need to handle like: “I don’t like chocolate”, “like” in this case isnegative.

We could also be interested by the grammar structure of the tweets, or if a tweet is subjective/objective and so on. As you can see, it is **extremely complex** to deal with languages and even more when we want to analyse text typed by users on the Internet because people don’t take care of making sentences that are grammatically correct and use a ton of acronyms and words that are more or less english in our case.

We can visualize a bit more the dataset by making a chart of how many positive and negative tweets does it contains,

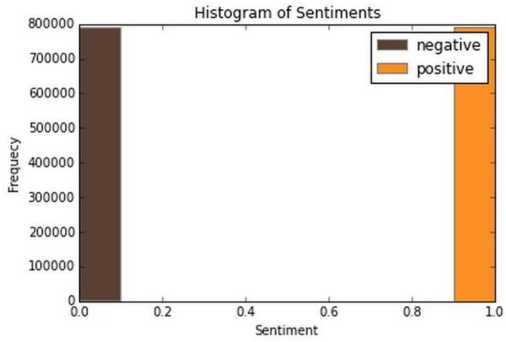


Figure 4.1.1: Histogram of the tweets according to their sentiment

We have exactly 790177 positive tweets and 788435 negative tweets which signify that the dataset is well•balanced. There is also no duplicates.Finally, let's recall the Twitter terminology since we are going to have to deal with in the tweets:

- Hashtag: A hashtag is any word or phrase immediately preceded by the # symbol. Whenyouclick onahashtag,you'llseeotherTweetscontainingthesamekeywordortopic.
- @username: A username is how you're identified on Twitter, and is always preceded immediately by the @ symbol. For instance, Katy Perry is@katyperry.
- MT: Similar to RT (Retweet), an abbreviation for "Modified Tweet." Placed before the Retweeted text when users manually retweet a message with modifications, for example shortening aTweet.
- Retweet:RT,ATweetthatyouforwardtoyourfollowersisknownasaRetweet.Oftenusedto pass along news or other valuable discoveries on Twitter, Retweets always retain original attribution.
- Emoticons:Composedusingpunctuationandletters,theyareusedtoexpressemotions concisely, ":) :)...".

Now we have the corpus of tweets, we need to use other resources to make easier thepre•processing step.

B. Resources

In order to facilitate the pre•processing part of the data, we introduce five resources which are,

1) A stop word dictionary corresponding to words which are filtered out before or after processingofnaturallanguagedatabecausetheyarenotusefulinourcase.
2) A positive and negative word dictionaries given the polarity (sentiment out•of•context) of words.
3) Anegativecontractionsandauxiliariesdictionary whichwillbeusedtodetectnegation in a given tweet such as "don't", "can't", "cannot",etc.

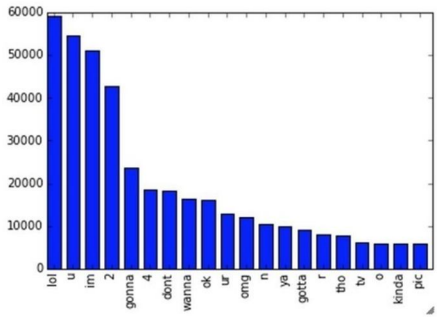
The introduction of these resources will allow to uniform tweets and remove some of their complexities with the acronym dictionary for instance because a lot of acronyms are used in tweets. The positive and negative word dictionaries could be useful to increase (or not) the accuracy score of the classifier. The emoticondictionaryhasbeenbuiltfromwikipediawitheachemoticonannotatedmanually.Thestopword dictionary contains 635 words such as "the", "of", "without". Normally they should not be useful for classifying tweets according to their sentiment but it is possible that theyare. Also we use Python 2.7 (https://www.python.org/) which is a programming language widely used in data science and scikit•learn (http://scikit•learn.org/) a very complete and useful library for machine learning containing every techniques, methods we need and the website is also full of tutorials well•explained. With Python, the libraries, Numpy (http://www.numpy.org/) and Panda (http://pandas.pydata.org/) for manipulating data easily and intuitively are just essential.

- *HTMLentities*

HTML entities are characters reserved in HTML. We need to decode them in order tohave characters entities to make them understandable. The case is something that can appears useless but in fact it is really important for distinguish proper noun and other kind of words. Indeed: "General Motor" is the same thing that "general motor", or "MSc" and "msc". So reduce all letters to lowercase should be normally done wisely. In this project, for simplicity we will not take care of that since we assume that it should not impact too much the classifier's performance.

- *Acronyms*

We replace all acronyms with their translation. An acronym is an abbreviation formed from the initial components in a phrase or a word. Usually these components are individual letters (as in NATO or laser) or parts of words or names (as in Benelux). Many acronyms are used in our data set of tweets as you can see in the following barchart.
At this point, tweets are going to be tokenized by getting rid of the punctuation and using split in order to do the process really fast. We could use nltk.tokenizero but it is definitely much much slower (also much more accurate).



As you can see, “lol”, “u”, “im”, “2” are really often used by users. The table below shows the top 20 acronyms with their translation and their count.

- *Negation*

We replace all negation words such as “not”, “no”, “never” by the tag **||not||** using the negation dictionary in order to take more or less of sentences like "I don't like it". Here like should not be considered as positive because of the "don't" before. To do so we will replace "don't" by ||not|| and the word like will not be counted as positive. We should say that each time a negation is encountered, the words followed by the negation word contained in the positive and negative word dictionaries will be reversed, positive becomes negative, negative becomes positive, we will do this when we will try to find positive and negative words.

$$P(C = c|D = d) = \frac{P(D = d|C = c)P(C = c)}{P(D = d)}$$

Figure 3.3.8.2: A tweet after processing negation words.

- *Sequence of Repeated Characters*

Now, we replace all sequences of repeated characters by two characters (e.g: "helloooo"= "helloo") to keep the emphasized usage of the word.

Table 3.3.9.1: Tweets before processing sequences of repeated characters.

	ItemID	Sentiment	SentimentSource	SentimentText
1578604	1578620	1	Sentiment140	[zzzz, no, work, tomorrow, yayyy]
1578605	1578621	1	Sentiment140	[zzzzz, time, tomorrow, will, be, a, busy, day, for, serving, loving, people, love, you, all]
1578606	1578622	0	Sentiment140	[zzzzz, want, to, sleep, but, at, sister's, in, laws's, house]
1578607	1578623	1	Sentiment140	[zzzzzz, finally, night, tweeters]
1578608	1578624	1	Sentiment140	[zzzzzzz, sleep, well, people]
1578609	1578625	0	Sentiment140	[zzzzzzzzz, wait, no, i, have, homework]
1578610	1578626	0	Sentiment140	[zzzzzzzzzzz, whatever, what, am, i, doing, up, again]
1578611	1578627	0	Sentiment140	[zzzzzzzzzzzzzzzz, i, wish]

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

C. Machine Learning

Once we have applied the different steps of the preprocessing part, we can now focus on the machine learning part. There are three major models used in sentiment analysis to classify a sentence into positive or negative: SVM, Naive Bayes and Language Models (N•Gram). SVM is known to be the modelgivingthebestresultsbutinthisprojectwefocusonlyonprobabilisticmodel thatareNaiveBayes and Language Models that have been widely used in this field. Let's first introduce the Naive Bayes model which is well•known for its simplicity andefficiency for textclassification.

- NaiveBayes

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive)independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.

Maximum•likelihood training can be done by evaluating a closed•form expression (mathematical expression that can be evaluated in a finite number of operations), which takes linear time.

It is based on the application of the Baye's rule given by the following formula:

In our case, atweet$d$ is represented by a vectorof$K$ attributes such as $d = (w_1, w_2, ...,w_k)$.

Computing $P(d|c)$ is not trivial and that's why the Naive Bayes introduces the assumption that all of the feature values w_j are independent given the category label c . That is, for $i \neq j$,w_i and $w_j$are conditionally independent given the category label c . So the Baye's rule can be rewrittenas,

$$c* = arg\max_{c \in C} P(c|d)$$

$$c* = arg\max_{c \in C} \left\{ P(c) \times \frac{\prod_{j=1}^{K} P(w_j|c)}{P(d)} \right\}$$

$$c* = arg\max_{c \in C} \left\{ P(c) \times \prod_{j=1}^{K} P(w_j|c) \right\}$$

Formula 3.4.1.3: Baye's rule rewritten

Basedonthisequation,maximumaposterior(MAP)classifiercanbeconstructingbyseekingtheoptimal category which maximizes the posterior $P(c|d)$:

Formula 3.4.1.4: Classifier maximizing the posterior probability P(c|d)

Notethat$P(d)$isremovedsinceitisaconstantforeverycategory$c$.Therareseveralvariantsof Naive Bayes classifiers thatare:

- The **Multi•variate Bernoulli Model**: Also called binomial model, useful if our feature vectors are binary (e.g 0s and 1s). An application can be text classification with bag of words model where the 0s 1s are "word does not occur in the document" and "word occurs in the document"respectively.
- The Multinomial Model: Typically used for discrete counts. In text classification, we extend the Bernoulli model further by counting the number of times a word w_i appears over the number of words rather than saying 0 or 1 if word occurs ornot.
- the Gaussian Model: We assume that features follow a normal distribution. Instead of discrete counts, we have continuousfeatures.

For text classification, the most used considered as the best choice is the Multinomial Naive Bayes.

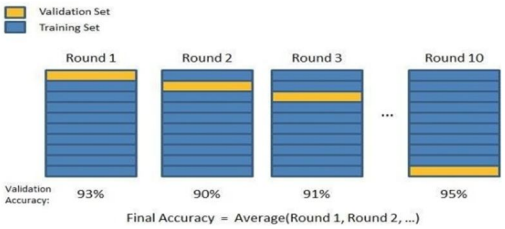
Thepriordistribution$P(c)$canbeusedtoincorporateadditionalassumptionsabouttherelative frequencies of classes. It is computedby:

$$P(c) = \frac{N_i}{N}$$

Formula 3.4.1.5: Prior distribution $P(c)$

©IJRASET: All Rights are Reserved

where N is the total number of training tweets and $N_i$is the number of training tweets in class c
The likelihood $P(w_j|c)$ is usually computed using the formula:



Formula 3.4.1.6: Likelihood P (wj|c)

where*count*(w_j, c) is the number of times that word w_j occurs within the training tweets of class
c , and $|V|$ = $\sum w_j$ the size of the vocabulary. This estimation uses the simplest smoothing method to solve the zero•probability problem, that arises when our model encounters a word seen in the test set but not in the training set, Laplace or add•one since we use 1 as constant. We will see that Laplace smoothing method is not really effective compared to other smoothing methods used in language models.

- *Baseline*

In every machine learning task, it is always good to have what we called a baseline. It often a "quick and dirty" implementation of a basic model for doing the first classification and based on its accuracy, try to improve it.
We use the Multinomial Naive Bayes as learning algorithm with the Laplace smoothing representing the classic way of doing text classification. Since we need to extract features from our data set of tweets, we use the **bag of words model** to represent it.
The bag of words model is a simplifying representation of a document where it is represented as a bag of its words without taking consideration of the grammar or word order. In text classification, the count (number of time) of each word appears is a document is used as a feature for training the classifier.
Firstly, we divide the data set into two parts, the training set and the test set. To do this, we first shuffle the data set to get rid of any order applied to the data, then we from the set of positive tweets and the set of negative tweets, we take 3/4 of tweets from each set and merge them together to make the training set. The rest is used to make the test set. Finally the size of the training set is 1183958 tweets and the test set is 394654 tweets. Notice that they are balanced and follow the same distribution of the initial data set.
Once the training set and the test set are created we actually need a third set of data called the validation set. It is really useful because it will be used to validate our model against unseen data and tune the possible parameters of the learning algorithm to avoid under fitting and over fitting for example.
We need this validation set because our test set should be used only to verify how well the model will generalize. If we use the test set rather than the validation set, our model could be overly optimistic and twist the results.

To make the validation set, there are two main options:

- Splitthetrainingsetintotwoparts(60%,20%)witharatio2:8whereeachpartcontainsanequal distribution of example types. We train the classifier with the largest part, and make prediction with the smaller one to validate the model. This technique works well but has the disadvantage of our classifier not getting trained and validated on all examples in the data set (without counting the testset).

$$P(w_j|c) = \frac{1 + count(w_j, c)}{|V| + N_i}$$

- The **K•foldcross•validation**. We split the data set into k parts, hold out one, combine the others and train on them, then validate against the held•out portion. We repeat that process k times (each fold), holding out a different portion each time. Then we average the score measured for each fold to get a more accurate estimation of our model'sperformance.

We split the training data into 10 folds and cross validate on them using scikit•learn as shown in the figure 3.4.2.1 above. The number of K•folds is arbitrary and usually set to 10 it is not a rule. In fact, determine the best K is still an unsolved problem but with lower K: computationally cheaper, less variance, more bias. With large K: computationally expensive, higher variance, lower bias.

We can now train the naive bayes classifier with the training set, validate it using the hold out part of data taken from the training set, the validation set, repeat this 10 times and average the results to get the final accuracy which is about **0.77** as shown in the screen results below,

```
Total tweets classified: 1183958
Score: 0.77653600187
Confusion matrix:
[[465021 126305]
 [136321 456311]]
```

Notice that to evaluate our classifier we two methods, the F1 score and a confusion matrix. The **F1 Score** can be interpreted as a weighted average of the precision and recall,where an F1 score reaches its best value at 1 and worst score at 0. It a measure of a **classifier's accuracy**. The F1 score is given by the following formula,

$$F1 = \frac{2 \times (precision \times recall)}{(precision + recall)}$$

Formula 3.4.2.1: F1 score

where the precision is the number of true positives (the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class,

$$Precision = \frac{TP}{TP + FP}$$

and the recall is the number of true positives divided by the total number of elements that actually belong to the positive class,
A precision score of 1.0 means that every result retrieved was relevant (but says nothing about whether all relevant elements were retrieved) whereas a recall score of 1.0 means that all relevant documents were retrieved (but says nothing about how many irrelevant documents were also retrieved).
There is a trade•off between precision and recall where increasing one decrease the other and we usually use measures that combine precision and recall such as F•measure or MCC.
A confusion matrix helps to visualize how the model did during the classification and evaluate its accuracy. In our case we get about 156715 false positive tweets and 139132 false negative tweets. It is "about" because these numbers can vary depending on how we shuffle our data for example.

		Predicted Class: Yes	Predicted Class: No
Actual Class	Yes	TP	FN
	No	FP	TN

Notice that we still didn't use our test set, since we are going to tune our classifier for improving its results.
The confusion matrix of the naive bayes classifier can be expressed using a color map where dark colors represent high values and light colors represent lower values as shown in the corresponding color map of the naive bayes classifier below,

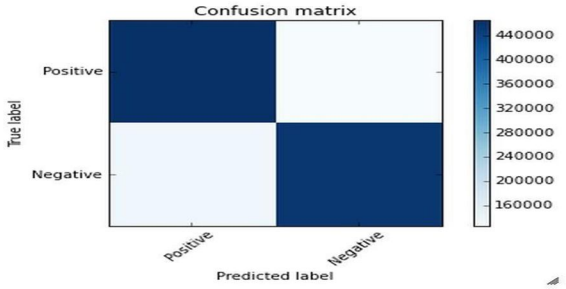


Figure 3.4.2.4: Colormap of the confusion matrix related to the naive bayes classifier used.

Hopefully we can distinguish that the number of true positive and true negative classified tweets is higher than the number of false and positive and negative tweets. However from this result we try to improve the accuracy of the classifier by experimenting different techniques and we repeat the same process using the k•fold cross validation to evaluate its averaged accuracy.

- *Improvements*

From the baseline, the goal is to improve the accuracy of the classifier, which is 0.77, in order to determine better which tweet is positive or negative. There are several ways of doing this and we present only few possible improvements (or not).
First we could try to removed what we called, stop words. Stop words usually refer to the most common words in the English language (in our case) such as: "the", "of", “to” and so on.
They do not indicate any valuable information about the sentiment of a sentence and it can be necessary to remove them from the tweets in order to keep only words for which we are interested. To do this we use the list of 635 stopwords that we found. In the table below, you can see the most frequent words in the data set with their counts,
We can derive from the table, some interesting statistics like the number of times the tags used in the pre•processing step appear,

```
Total tweets classified: 1183958
Score: 0.773106857186
Confusion matrix:
[[462537 128789]
 [138039 454593]]
```

Figure 3.4.3.1: Tags in the data set with their corresponding count.

Recall that ||url|| corresponds to the URLs, ||target|| the twitter usernames with the symbol “@” before, ||not||replacesthenegationwords,||pos||and||neg||replacethepositiveandnegativesmileyrespectively. After removing the stop words we get the resultsbelow,Compared to the previous result, we lose 0.02 in accuracy and the number of false positive goes from 126305 to 154015 . We conclude that stop words seem to be useful for our classification task and remove them do not represent animprovement.
We could also try to stem the words in the data set. **Stemming** is the process by which endings are removed from words in order to remove things like tense or plurality. The stem form of a word could not exist in a dictionary (different from Lemmatization). This technique allows to unify words and reduce the dimensionality of the dataset. It's not appropriate for all cases but can make it easier to connect together tenses to see if you're covering the same subject matter.It is faster than **Lemmatization** (remove inflectional endings only and return the base or dictionary form of a word, which is known as the lemma). Using the library NLTK which is a library in Python specialized in natural language processing, we get the following results after stemming the words in the dataset,

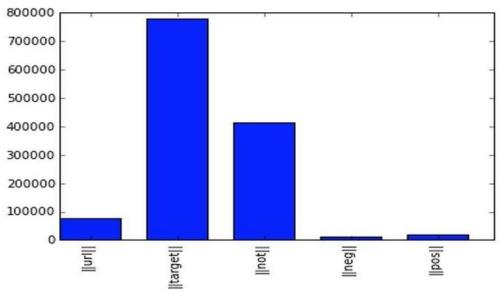


Figure 3.4.2.2: Result of the naive bayes classifier after stemming.

We actually lose 0.002 in accuracy score compared to the results of the baseline. We conclude that stemming words does not improve the classifier’s accuracy and actually do not make any sensible changes.

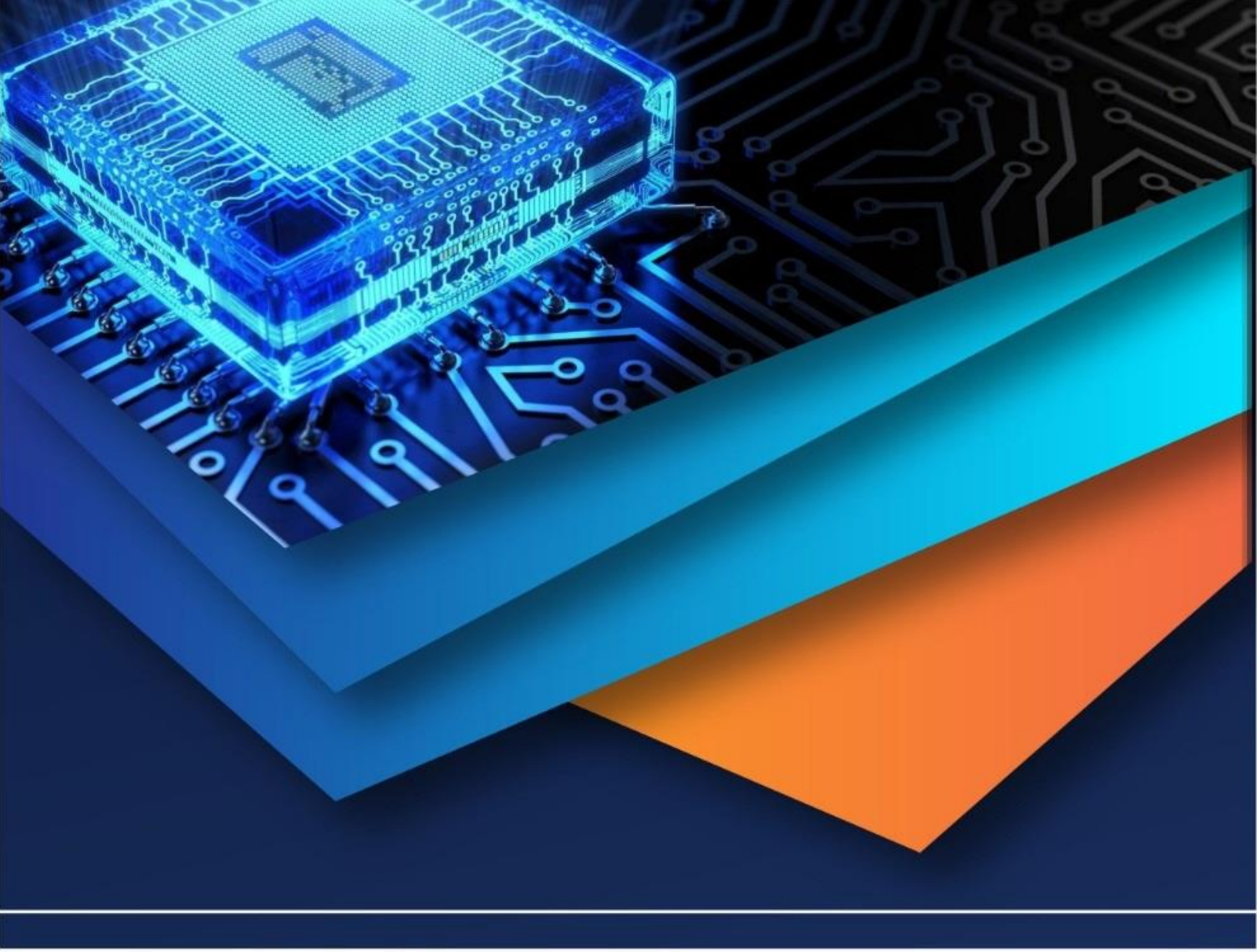
- *Language Models*

Let’s introduce language models to see if we can have better results than those for our baseline. Language models are models assigning **probabilities to sequence of words**. Initially, they are extensively used in speech recognition and spelling correction but it turns out that they give good results in text classification. The quality of a language model can be measured by the empirical perplexity (orentropy) using:

REFERENCES

[1] AlexanderPak,PatrickParoubek.2010,TwitterasaCorpusforSentimentAnalysisandOpinion Mining.
[2] AlecGo,RichaBhayani,LeiHuang.TwitterSentimentClassificationusingDistant Supervision.
[3] JinBai,Jian•YunNie.UsingLanguageModelsforTextClassification.
[4] ApoorvAgarwal,BoyiXie,IliaVovsha,OwenRambow,RebeccaPassonneau. Sentiment Analysis of TwitterData.
[5] FuchunPeng.2003,AugmentingNaiveBayesClassifierswithStatisticalLanguageModels
[6] Mikalai Tsytsarau, Themis PalpanasSurvey on mining subjective data on the web
[7] Data Min KnowlDiscov, 24 (2012), pp. 478-514Wilson T, Wiebe J, Hoffman P. Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of HLT/EMNLP; 2005.
[8] B. LiuSentiment analysis and opinion mining
[9] Synth Lect Human Lang Technol (2012)
[10] Liang-Chih Yu, Jheng-Long Wu, Pei-Chann Chang, Hsuan-Shou Chu
[11] Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news
[12] Knowl-Based Syst, 41 (2013), pp. 89-97
[13] Michael Hagenau, Michael Liebmann, Dirk Neumann. Automated news reading: stock price prediction based on financial news using context-capturing features. DecisSuppSyst; 2013.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)