



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering (SITE)

TEXT EMOTION DETECTION & SENTIMENT ANALYSIS

By: -

SIDDHARTH DAS (18BIT0379)

SHRUTI VARSHA VENKATRAMAN (18BIT0405)

Under the guidance of

Prof. Balakrushna Tripathy – SITE

SOFT COMPUTING (ITE1015)

FALL SEMESTER 2020

ABSTRACT

Analysis of public information from social media could yield interesting results and insights into the world of public opinions about almost any product, service or personality. Social network data is one of the most effective and accurate indicators of public sentiment. The explosion of Internet has led to increased activity in Podcasting, Blogging, Tagging, Contributing to RSS, Social Bookmarking, and Social Networking.

As a result, there has been an eruption of interest in people to mine these vast resources of data for opinions. Sentiment Analysis or Opinion Mining is the computational treatment of opinions, sentiments and subjectivity of text. In this paper we will be discussing a methodology which allows utilization and interpretation of twitter data to determine public opinions. Developing a program for sentiment analysis is an approach to be used to computationally measure customers' perceptions.

This paper reports on the design of a sentiment analysis, extracting and training a vast number of tweets. Results classify customers' perspective via tweets into positive and negative, which is represented through code implementation of our project.

Problem Definition:

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. Furthermore, with the recent advancements in machine learning algorithms, we are able improve the accuracy of our sentiment analysis predictions.

In this report, we will attempt to conduct sentiment analysis on “tweets” using various different machine learning algorithms. We attempt 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.

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

We use 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 we pick 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. Finally, we report our experimental results and findings at the end.

The problem at hand consists of two subtasks:

- **Phrase Level Sentiment Analysis:**

Given a message containing a marked instance of a word or a phrase, determine whether that instance is positive, negative or neutral in that context.

- **Sentence Level Sentiment Analysis:**

Given a message, decide whether the message is of positive, negative, or neutral sentiment. For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen.

Introduction:

Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people's sentiments towards certain entities. Internet is a resourceful place with respect to sentiment information. From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. Sentiment analysis seems to have strong fundament with the support of massive online data but those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. For example, instead of sharing topic-related opinions, online spammers post spam on forums. Some spams are meaningless at all, while others have irrelevant opinions also known as fake opinions. The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral.

One challenge is to build technology to detect and summarize an overall sentiment. Our project analyzes tweets of the people and categorize them accordingly. Unlike numerical data, textual data is difficult to handle. For one thing, using mathematical models directly on them is not possible. Now, let us formulate a problem statement and see how we can solve it using NLP and some basic machine learning techniques. This project will primarily focus on practical implementation than the theoretical or mathematical understanding behind the techniques used.

We have made a comment detector that the comments are positive or negative. By Positive it means that the user is happy and by negative it means that the user is sad.

We used sentiment analysis to detect this and classification using Naive Bayes, Linear SVM, Logistic Regression, and Random Forest.

LITERATURE REVIEW:-

Aghdam et al.[1], proposed an algorithm to deal with feature selections from various kinds of texts. This algorithm is called 'Ant Colony Optimisation' and it is based on the methods which ants use in real life while searching for food sources. The author mentions the concept on which the algorithm is based, that is how the ants search for the shortest paths in order to reach their food source. The author compares the proposed algorithm with some already existing algorithms like, information gain, chi-square and genetic algorithm, and compares them with the proposed algorithm which showed the results as the proposed algorithm was having a higher accuracy value (Precision= 77.1343, Recall= 79.7546) on average. The dataset used is the Reuters21578. It can be concluded that this algorithm can provide a very optimal and efficient solution with context to the given problem.

Chang et al.[2], proposed a method which deals with the classifying texts having multiple categories or labels automatically. Various methods were used such as mutual information term selection method, weighted indexing technique and a category-sensitive refinement method which was proposed by them. In the proposed method, Reuters-21578 ModeApte` Split Text Collection was used for training a linear classifier. The author has compared the proposed methods with some existing methods of that time like, Rocchio's method, K-nearest neighbours (KNN), regression model, Naive Bayes and Bayesian nets, decision tree, decision rules, etc and it showed that the proposed method uses less time for training the classifiers and less classifiers are required to speedily classify the given text. On experimentation, the micro-averaged BEP of the top 10 categories was 87.8 and micro-averaged BEP of all categories was 81.2 for the proposed method, which is quite higher than the other existing methods of that time.

Thomas et al.[3], proposed a semi-supervised method for text clustering. The paper deals with text data only. The focus is on the text classification under the domain of text mining. There are two phases in it:- a training phase and classification phase. Training involves forming clusters from the labelled text present in the data set (Reuters - 21578) and categorizing them according to the text labels. Then, in the classification phase, a new unlabelled text is taken and its similarity is measured (using SMTP) with the centroid of those text clusters and classified accordingly. The use of SMTP provides better accuracy in similarity measurement of unlabelled text. The main focus is on text classification, therefore no focus is made on dimensionality reduction of the text data; this is something to work on.

This research [4] is a survey based report on various machine learning algorithms used in text classification and its different aspects. The author has mentioned two

approaches for text classification, namely:- rule based approach which is done manually based on some defined rules, and machine learning approach which is automated text classification approach based on learning from example text/documents. But of them, the later one is quicker with higher accuracy. The author has analysed seven different ML approaches to the problem by different author. The report gives a comparative study of these methods with some external ones.

The research by Lin [5] is an effort towards the energy efficiency aspect of text classification process. Several common classifiers are used and their energy cost is measured using three types of datasets:- 20 Newsgroups dataset by Ken Lang, Reuters-21578, Reuter Ltd. RCV1. So, firstly the author has used classifiers like Naive Bayes, SVM and Perceptron as study objects and then the accuracy and energy cost of those classifiers are measured and compared. Another important thing used is parallelization to reduce energy cost, but it'll decrease the average power of CPU. Finally it was obtained that the parallel version of Naive Bayes can achieve a high accuracy and is competitive to SVM and Perceptron and it is simple too. It is also an efficient one.

Sabbah et. al [6], has proposed a new frequency based term weighting scheme for accurate text classification which has outperformed other benchmarked schemes significantly. The author has proposed four weighting schemes namely; mTF, mTFIDF, TFmIDF, and mTFmIDF which will take the amount of missing terms into account calculating the weight of existing terms. Datasets like Reuters-21578 R8, 20Newsgroups, and WebKb were used along with few classifiers:- SVM, KNN, NB, and ELM. Finally the result says that the highest performance goes for a SVM classifier with a micro-average F1 classification performance value of 97%. Also mTF, mTFIDF, and mTFmIDF were considered to be the highest performance schemes on analyzing the results.

Thorsten Joachims [7] in 2005 had proposed the idea of using SVM in categorizing text data. He had shown that how exactly SVM can be beneficial to our current problem. Experiments were performed comparing the efficiency of SVM with other methods/classifiers like Bayes, Rocchio, C4.5 and k-NN . The datasets used were:- "ModApte" split of the Reuters-21578 dataset by David Lewis and Ohsumed corpus by William Hersh. The results of the following experiment highlighted the following properties of text:-high dimensional feature spaces, few irrelevant features and sparse instance vectors; and in all these aspects SVM outperformed the remaining.

Mujtaba, et. al [8], has presented a comparative study in identifying the cause of death (CoD) of any person using techniques from text classification. Data for 8 different CoD was obtained from a forensic autopsy reports collected from a few hospitals. Various schemes of text classification such as feature extraction, term weighting, feature

reduction, etc were used. Finally the classification model obtained was evaluated and based on 70 dataset instances its overall accuracy was 78.25%, precision = 0.781, recall = 0.783, F-measure = 0.782. But the final model was not good enough for new real time application.

Desmet et al. [9], has given us a very good application of text classification techniques which is suicide prevention. The work is based on forums in the language – Dutch. Genetic algorithms were used in order to optimize the model in a better way of feature selection and hyperparameter optimization. They have used keyword filtering along with some machine learning algorithms for obtaining a good precision-recall value. They have focused the experiment on two kind of tasks:- detecting suicide related posts (in Dutch) and severe, high risk content on internet. The results show high precision and minimal noise. Thus, we can get such system; but the only drawback is that it is for Dutch language only.

Shafiabady, et. al [10], came up with an unsupervised clustering approach for training the Support Vector Machines used in text classification. Training a classifier manually is a very tedious job, but if we can use some unsupervised schemes the clustering part becomes easier. A technique is proposed which uses certain unsupervised methods such as self-organizing maps (SOM) and correlation coefficient(CorrCoef) in order to form clusters for unlabelled text data and then use this data to train SVM. This method also eliminates the problem of dimensionality which is referred as Curse of Dimensionality (COD). The author has experimented with 3 methods to achieve good results. Of all the three methods, the 3rd method in which SVM classifier is trained using “all-to-all” Correlation Coefficient (CorrCoef) produces the best result i.e. highest accuracy. Three datasets are used which are Reuters, Webkb and 20 Newsgroups and the accuracy achieved for them in method 3 are 96.98, 94.73 and 99.72 respectively. Thus, this approach can be used in places where expert decision is not available such as pipeline defect prediction or when the clustering task is too much tedious.

Rehman et al. [11], has obtained a new feature ranking method for selection of most relevant terms while classifying text. Selection of most relevant terms is one of the important part of text classification. The new method used by them is called max-min ratio (MMR). When compared with some of the metrics like balanced accuracy measure, information gain, chisquared, Poisson ratio, Gini index, odds ratio, distinguishing feature selector, and normalized difference measure, this new method proved to be more efficient. Six data sets were used for comparison namely WebACE (WAP, K1a, K1b), Reuters (RE0, RE1), and 20 Newsgroups and classifiers used are multinomial naive Bayes (MNB) and support vector machines (SVM).

The research [12] is a survey done by the author Krina Vasa which deals with some statistical and machine learning approaches that can be applied in text classification.

The author has made a survey report for the methods like knearest neighbors, support vector machine, naive Bayesian method, decision tree, rule based classification and neural network. The conclusion of the survey was found as which method provides better performance. So, according to author, The hybrid approach of linear SVM and k-NN provides better accuracy but with the use of kernel function gives better performance than linear SVM.

Viegas et. al [13], has made an attempt to utilize some lazy semi-naive Bayesian strategies effectively and efficiently for text classification. It has been examined whether a correct mix of some alternative NB(Naive Bayes) learning models with distinct feature weighting methods can enhance the efficiency of NB in ADC assignments and then comparing the results with several other supervised algorithms like Nearest-Neighbour classifiers, Support Vector Machines, boosting and some other Bayes algorithms. The experiments findings indicate that a correct mix of learning paradigms and weighting approaches results in several datasets that are similar and even superior to SVMs at reduced cost efficiency. This strategies requires high computational time so they have used GPU parallelization here.

Faraz in his research [14] has presented to us some of the mathematical notations and graphical representations for descriptions of Automatic Text Classification (ATC). A Text Mining Model is also developed which will help to facilitate the design and development of algorithms for Text Categorization (TC) and Automatic Text Classification (ATC) and thus it would enhance the performance of software based on text mining.

Dogan et al. [15], has presented an improved version of inverse gravity method of weighting terms in which two schemes have been highlighted namely SQRT_TF-IGMimp and TF-IGMimp. The performances of these schemes are then compared with some other standard methods like TF-IDFCSDF, TF-IDF, TF-IDF-ICF, TF-PB, TF-RF, TF-IGMimp, TF-IGM, SQRT_TF-IGMimp and SQRT_TF-IGM. Classifiers used are SVM, KNN and NN, and datasets are Reuters-21578, 20 Mini Newsgroups and 20 Newsgroups. The results showed that SQRT_TF-IGMimp is better than all the other schemes in most cases while TF-IGMimp proved to be better than standard TF-IGM.

Lam et al. [16], has proposed four methods for dimensionality reduction based on artificial neural networks and compared each in terms of precision and recall. A three layer feed-forward neural network was trained and then tested by backpropagation which resulted in a good performance as measured by precision and recall. The techniques used are:- (i)The document frequency (DF) method, (ii) Term occurrence frequency (TF) and the inverse document frequency (IDF) method, (iii) category frequency (CF) and document frequency (DF) method, (iv) method of principal component analysis (PCA). Out of these four methods, PCA proved to be the most reliable one with an reduction rate of 98.9%.

Damerau et al. [17], has presented us the outcomes for automated content classification the accumulation of Reuters-810000 news stories. The author has divided the data into groups of every month and given them an initial standard. For training data, they used two classifiers namely decision trees and linear classifier. The experiment result showed that the linear model is not much feasible for recency effect. Also, it requires less data for training purposes and it is more accurate than the decision trees. However if the categories are less frequent in a data then, decision trees proves to be more accurate.

ElAlami, in his research [18] has mentioned a new feature subset choosing algorithm which with the help of genetic algorithm filters the features which are required to obtain. This algorithm doesn't modify any training results and neither does it depend on any of the ANN networks to work properly, instead it is only dependent on input features of the training network. The role of genetic algorithm in is to optimize the features which relates to the output function of each class. Later after training this network, this algorithm is applied on Monk1's and Car Evaluation's database which shows that the dimensionality of those two databases is reduced by 50% and 33.33% respectively. The results of this experiment is checked against other data mining techniques for the same dataset and this new algorithm proves to be more stable.

Guo, et. al, in his research [19] have proposed a very unique method for weighting the terms in text categorization. The normal term weighting schemes provide only one weight to one term/word, even if it occurs under different labels. The author addresses it as an unreasonable thing to do. So the main idea used by the author is to provide multiple weights to a term so that each of the terms shows its importance in the context. Each weight assigned shows a different class for them. Thus each of these weighted word are then connected as a multi-channel picture which is contribution to a multi-channel CNN model to actualize characterization. The results of this experiment are then compared with other CNN based approaches under the same dataset and this new method proved to be better in most cases.

The research by Ferre [20] is just a comparative study on different method for choosing components for PCA. They made a report on these various methods of how they work, why they failed or why they give very low accuracy. After proper viewing they concluded that there was no perfect solution for the problem of dimensionality reduction so we can just use the most accurate one in these cases. The goal of this paper was to make a descriptive and predictive analysis which was achieved then.

Kępa, et. al [21], has obtained a new classifier approach for text documents. It is a two stage classifier which comprises of kth nearest neighbours(kNN) and SVM classifiers. In their new classifier they have made use of a new method called 'one-vs-near' which is an extension of 'one-vs-all'. In this method, the first classification stage is done by kNN classifier and the next or last stage is done for accurate classification of the trained data. The initial tests were conducted with small sets of data while finally large datasets were used while testing.

Fragoso et al. [22], proposed a new method for filtering the features while performing feature selection, thus to have an efficient vector size for feature selection automatically. The name of this method is Automatic Feature Subsets Analyzer (AFSA) which is an extension of Class-dependent Maximum Features per Document (cMFDR) method. This new method consumes less time in finding the best number of features than the old one. Also, AFSA doesn't need much training data to produce correct results than that of cMFDR. AFSA selects the similar number of features from the best output of the previous one (cMFDR). Experimental results with datasets namely WebKB, Reuters, 20 Newsgroup, TDT2 proved to provide better or similar results as of that in cMFDR method.

Jiang et al. [23], has proposed a new fuzzy approach for dimensionality reduction of features space. The method automatically generates clusters of words in a feature vector of a text data, based on a similarity test. Then, it extracts one feature from each of the clusters and use it further. As per the algorithm, the user is not required to tell the number of extracted features in advance, and the trial-and-error approach to find those extracted features can also be avoided. Experimental results say that this method be fast and has a good accuracy in obtaining the extracted features.

Salles et. al, in his work [24] has pointed out an important factor affecting the old and new algorithms developed for text data classification, which is 'temporal factor'. Many algorithms have been developed over time, but most of them are developed assuming that data doesn't change over time. So the author in one of their previous works has shown us some of the adverse effects due to three main temporal effects. In order to minimize these effects on data, a temporal weighting function (TWF) is incorporated with few Automatic Document Classification (ADC) algorithms which they call temporally-aware classifiers, namely Rocchio, KNN and Naive Bayes. These proposed method is evaluated with three real-world datasets that suffer with temporal effects, namely ACM DL, MEDLINE and AG-NEWS. The results tells that temporally aware classifiers significantly improve the results over the traditional ones, with gains up to 17%.

Uysal in his work [25] has proposed an improved global scheme version for feature selection. This scheme is same at all stages except the last stage of a common feature selection scheme is modified. The key aspect behind this method is to equally represent each class in the feature vector. Three distinct datasets were used namely, Reuters, WebKB, Classic3. This method has shown some improvement in performance in terms of Micro-F1 and Macro-F1 metrics.

GAPS IN LITERATURE REVIEW:

Sr. no.	Author/s and Year of Publication	Dataset	Problem/Objective	Method/s used	Accuracy	Limitations
1.	Aghdam, et. al, 2008	Reuters-21578 dataset	Features selection from texts	Ant Colony Optimization	Precision = 77.1343 Recall = 79.7546	The parameter values used for testing were determined based on their initial experiments. It is not certain that these vales are optimal. Thus, optimization of parameters can be done.
2.	Chang, et. al, 2008	Reuters-21578 ModeApte` Split Text Collection	Categorizing text with multiple labels	Mutual information term selection method, weighted indexing technique, categorysensitive refinement method	Microaveraged BEP = 81.2	
3.	Thomas, et. al, 2016	Reuters-21578	Classifying text in an efficient manner	Clustering, Similarity Measure for Text Processing (SMTP) for better accuracy	For similarity measure – SMTP, = 186	No proper dimensionality reduction techniques used. Thus, more execution time and less varieties of documents can be handled. It can only handle text data.
4.	Padmavathi.S, Dr. M. Chidambaram, 2018		Survey on different machine learning algorithms used for classifying text	Boost.SH algorithms, AFSA, inconsistency detection techniques, i-vector method, etc		
5.	Lin, 2015	20 newsgroups Dataset by Ken Lang, Reuters-21578, Reuters, Ltd. RCV1	Obtaining an energy efficient way for text classification	Naive Bayes, SVM and Perceptron as study objects		The only focus is energy efficiency, which is important but other points should also be kept in mind. Research done only for the three classifiers, few more classifiers like Decision trees could be helpful in the research. Computation could be moved to GPU for better energy usage.
6.	Sabbah, et. al, 2017	Reuters-21578, 20Newsgroups, and WebKB	Method for weighting terms in text classification	Four frequency-based term weighting schemes:- mTF, mTFIDF, TFmIDF, and mTFmIDF With SVM classifier	Microaverage F1 = 97	The method can be used to various feature extraction methods like PCA, and help in solving specific problems. Better performance could be obtained with the help of some deep learning algorithms and optimization based techniques.

7.	Joachims, 2005	"ModApte" split of the Reuters-21578 dataset, Ohsumed corpus by William Hersh	Text Categorization	Support Vector Machines (SVM)	Microaverage = 86	Use of SVM has its own disadvantage like:- Long training time is required for very large datasets. Difficulty in choice of kernel. Highly complex algorithm and extensive
8.	Mujtaba, et. al, 2018	Autopsy reports from University Malaya Medical Hospital, Kuala Lumpur, Malaysia	Predicting the cause of death	Various methods of text classification	Precision = 0.781 Recall = 0.783 F-measure = 0.782	The prediction accuracy of this model is not good enough for real time deployment. It can only classify upto eight different cause of deaths. It can only give cause of death level related to ICD-10. Thus, it cannot give the diagnosis upto a greater detail.
9.	Desmet, Hoste, 2018	Dutchlanguage forum and blog messages posted on Netlog	Suicide prevention by identifying related texts	Genetic algorithms		It only works for labelled data, so we have to be dependent on that. The data presented and used is specific to Dutch. No multi-lingual support. Reduction of linguistic noise could be done. Text variation could be reduced. Improvement in lexical recall is required.
10.	Shafiabady, et. al, 2016	20 Newsgroups, Reuters-21578(R8) and WebKB	Training of Support Vector Machine for text classification	Self-Organizing Maps (SOM), Correlation Coefficient (CorrCoef)	For method-3 in dataset:- Reuters = 96.98 Webkb = 94.73 20 Newsgroups = 99.72	
11.	Rehman, et. al, 2018	WebACE (WAP, K1a, K1b), Reuters (RE0, RE1), and 20 Newsgroups	Finding most relevant terms in a text data	Max-Min Ratio (MMR), multinomial naive Bayes (MNB) and support vector machines (SVM) classifiers	Macro F1 measure = 76.2 Micro F1 measure = 74.4	There is a need to develop effective local or class-wise feature selection algorithms. Work needs to be done on weighting scheme because it is an integral part of the process, no relevant works should be removed from the document.
12.	Vasa, 2016		Survey on statistical and machine learning approaches of text classification	k-nearest neighbours, support vector machine, naive Bayesian method, decision tree, rule based classification, neural network		
13.	Viegas, et. al, 2018	20 Newsgroups, Four Universities,	Classifying text in an effective and efficient manner	Lazy Super Parent Tree Augmented Naive Bayes (LSPTAN) method	MacF1 and MicF1 for:--	High computational costs such as time.

		Reuters, ACM Digital Library, RCV1, AGNews, Medline and Yahoo datasets			<p>20 Newsgroups = 90.58 ±1.16, 90.77 ±1.16</p> <p>Four Universities = 61.10 ±2.17, 71.60 ±1.84</p> <p>Reuters = 37.71 ±1.86, 66.74 ±1.10</p> <p>ACM Digital Library = 64.32 ±1.67, 75.93</p> <p>±0.71 RCV1 = 55.55 ±0.67, 73.25</p> <p>±0.28 AGNews = 61.56 ±0.03, 68.89 ±0.09</p> <p>Medline = 71.69 ±0.57, 84.61 ±0.03</p> <p>Yahoo dataset = 66.30 ±0.11, 66.62 ±0.11</p>	In large Automatic Document Classification (ADC) datasets, its performance is minimized.
14.	Faraz, 2015		Text mining with a mathematical approach	Mathematical notations and graphical modelling techniques		Mathematical models have limited scope in data mining
15.	Dogan, et. al, 2019	Reuters-21578, 20 Mini Newsgroups and 20 Newsgroups	An enhanced term weighting method for text classification	<p>Inverse gravity moment formula($\sqrt{\text{TF-IGMimp}}$ and TF-IGMimp);</p> <p>*imp means improved version</p>	For Reuters-21578, the Macro-F1 using SVM classifier is 96.57 (highest)	
16.	Lam, et. al, 1999	A subset of the Reuters-22173 test collection	Neural networks based feature reduction for categorizing text data	DF Method, TFxIDF Method, CF-DF Method, Principal Component Analysis	Reduction rate of 98.9%	PCA works on the basis of certain assumptions. It may give inaccurate results for a large variety of data. Also, it is a linear method.
17.	Damerau, et. al, 2004	Reuters-810000 collection	Text categorization	Decision tree classifier and linear classifier		Amount and recency of data used in training is dependent on the model
18.	ElAlami, 2009	Monk1's database and Car Evaluation database	Choosing a subset of features from a feature set	Genetic algorithm	For Monk1's database - 50%,	Testing is done with a limited data set type.

					Car Evaluation database – 33.33%	
19.	Guo, et. al, 2019	Movie review sentence polarity dataset v1.0, subjectivity data set, customer reviews of 14 products from Amazon, MPQA, TREC	Improved way for term weighting in text classification	A multi-channel Text-CNN model	Accuracy for, MR = 86.6, Subj = 95.6, CR = 87.5, MPQA = 93.0, TREC = 91.9	Size of the datasets used is not too large here. There are various word installing networks in our circumstance, it is thus hard to concentrate or posture how each inserting framework is identified with one another
20.	Ferre, 1995		Comparing methods for selecting components for PCA	Different methods of component selection in PCA		There is no proper or totally correct solution for solving the reducing the dimensions during PCA
21.	Kępa, et. al, 2015	Wikipedia	Classification of text documents in two stages	A new method named one-vs-near	Precision = 59.40% Recall = 33.34%	
22.	Fragoso, et. al, 2016	WebKB, Reuters, 20 Newsgroup, TDT2	Determination of vector size for feature selection in text categorization	Automatic Feature Subsets Analyzer (AFSA)	For TDT2 dataset, Micro-F1 value = 96.68	
23.	Jiang, et. al, 2011	20 Newsgroups Dataset, Reuters Corpus Volume 1 (RCV1) Data Set, Cade12 Data	Reduce dimensionality of feature space for text classification	Fuzzy self-constructing feature clustering (FFC) algorithm	For Cade12 dataset, Micro averaged accuracy = 93.55	
24.	Salles, et. al, 2017	ACM DL, MEDLINE and AG-NEWS	Efficient time-based text classification	A temporal weighting function (TWF) incorporated with classifiers namely Rocchio, KNN and Naive Bayes		This may affect the overall speed and performance.
25.	Uysal, 2015	Reuters, WebKB, Classic3 datasets	A global scheme for features selection in documents classification	Improved global feature selection scheme (IGFSS)	Micro-F1 scores (%) for Classic3 dataset = 98.973	

HARDWARE USED:

HP Laptop

RAM – 8gb

Storage – 1TB

Processor - Intel i5 8th Gen

SOFTWARE/TOOLS/LANGUAGE:

SOFTWARE: ANACONDA, SPYDER, JUPYTER NOTEBOOK, VS CODE

LANGUAGE: PYTHON,PHP

Module Description:

NLTK

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.” NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more.

NLP Approach

This approach utilizes the publicly available library of SentiWordNet, which provides a sentiment polarity values for every term occurring in the document. In this lexical resource each term t occurring in WordNet is associated to three numerical scores $\text{obj}(t)$, $\text{pos}(t)$ and $\text{neg}(t)$, describing the objective, positive and negative polarities of the term, respectively. These three 26 scores are computed by combining the results produced by eight ternary classifiers. WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. WordNet's structure makes it a useful tool for computational linguistics and natural language processing. It groups words together based on their meanings. Synet is nothing but a set of one or more Synonyms. This approach uses Semantics to understand the language.

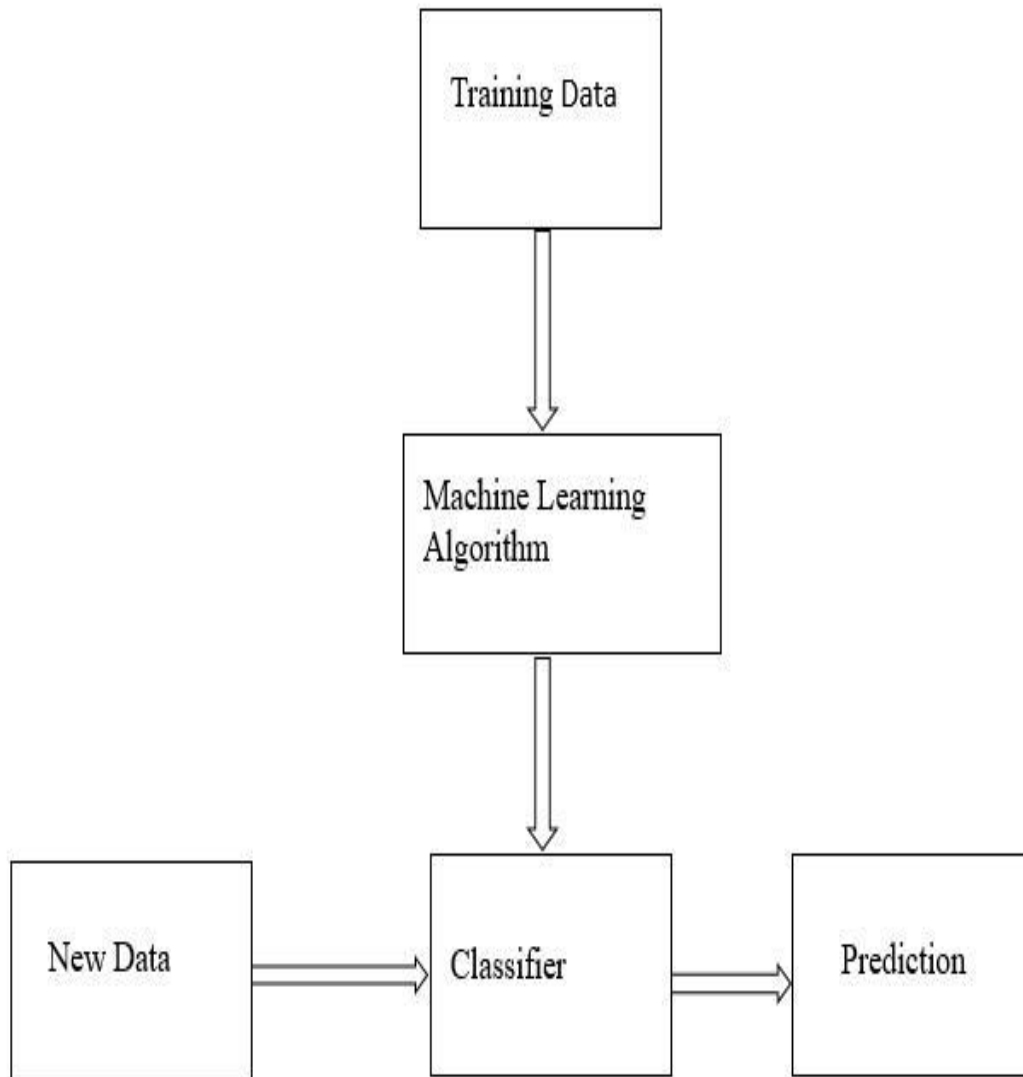
Major tasks in NLP that helps in extracting sentiment from a sentence:

- Extracting part of the sentence that reflects the sentiment
- Understanding the structure of the sentence
- Different tools which help process the textual data

Basically, Positive and Negative scores got from SentiWordNet according to its part-of-speech tag and then by counting the total positive and negative scores we determine the sentiment polarity based on which class (i.e. either positive or negative) has received the highest score.

TWEEPY: An easy-to-use Python library for accessing the Twitter API

System Flow Diagram:



Algorithm Presentation / Procedure Description

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.

Feature Extraction:

We extract two types of features from our dataset, namely Term Frequency-Inverse Document Frequency and Count Vectors.

• ***Term Frequency-Inverse Document Frequency (TF-IDF)***: This parameter gives the relative importance of a term in the data and is a measure of how frequently and rarely it appears in the text.

• ***Count Vectors***: This is another feature we consider and as the name suggests we transform our tweet into an array having the count of appearances of each word in it. The intuition here is that the text that conveys similar emotions may have the same words repeated over and over again. This is more of a direct approach.

Training the models:

Naive Bayes

Naive Bayes is a simple model which can be used for text classification. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.

Random Forest

Random Forest is an ensemble learning algorithm for classification and regression. Random Forest generates a multitude of decision trees classifies based on the aggregated decision of those trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

SVM

SVM, also known as support vector machines, is a non-probabilistic binary linear classifier. For a training set of points 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

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

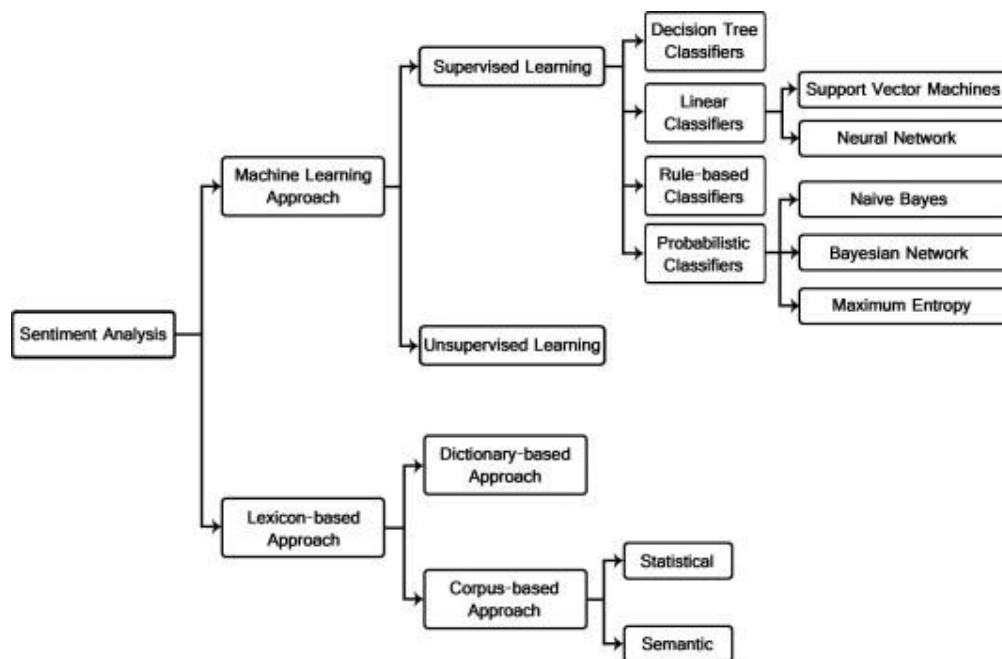
We want to maximize the margin, denoted by γ , as follows

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

in order to separate the points well.

Logistic regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value).



Step 1: Import the basic libraries such as pandas, numpy, textblob, nltk, tweepy

Step 2: Read the dataset.

Step 3: Filter the dataset. Choose four basic emotion labels – Positive, Negative, Opinion and Factual as output labels.

Step 4: make all the words lowercase

Step 5: remove symbols used for punctuation

Step 6: stop words such as prepositions

Step 7: convert all words to their root form- convert all plural words, or words in past tense etc, to the root form (lemmatisation)

Step 8: assuming that hardly any word in the dataset would have letters repeating more than twice consecutively, revert repetition of letters (removing words that are unnecessary)

Step 9: find rare words and remove them

Step 10: Classify the data into the four output labels. Split the dataset into training and testing data in the ratio 90:10

Step 11: extract Term Frequency-Inverse Document Frequency (TF-IDF) parameters.

Step 12: extract Count Vectors parameters

Step 13: build Multinomial Naive Bayes Classifier, Linear SVM, logistic regression, Random Forest Classifier models using TF-IDF features.

Step 14: build Multinomial Naive Bayes Classifier, Linear SVM, logistic regression, Random Forest Classifier models using Count Vectors features.

Step 15: Test the models. By accessing the tweets by topics, assess if the tweets are classified properly under the output labels.

Sample Code:

```
###

import pandas as pd

import numpy as np

import nltk

import tweepy

nltk.download('stopwords')

auth=tweepy.OAuthHandler(Consumer_key, consumer_secret)

auth.set_access_token(access_token, access_token_secret)

#Making all letters lowercase

data['content'] = data['content'].apply(lambda x: " ".join(x.lower() for x in x.split()))

#Removing Punctuation, Symbols

data['content'] = data['content'].str.replace('[^\w\s]',' ')

#Removing Stop Words using NLTK

from nltk.corpus import stopwords

stop = stopwords.words('english')

data['content'] = data['content'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))

#Lemmatisation

from textblob import Word

data['content'] = data['content'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))

#Correcting Letter Repetitions

import re

def de_repeat(text):

    pattern = re.compile(r"(\1{2,})")

    return pattern.sub(r"\1\1", text)
```

```

###

data['content'] = data['content'].apply(lambda x: " ".join(de_repeat(x) for x in x.split()))

# Code to find the top 10,000 rarest words appearing in the data

freq = pd.Series(' '.join(data['content']).split()).value_counts()[-10000:]

# Removing all those rarely appearing words from the data

freq = list(freq.index)

data['content'] = data['content'].apply(lambda x: " ".join(x for x in x.split() if x not in freq))

#Encoding output labels 'sadness' as '1' & 'happiness' as '0'

from sklearn import preprocessing

lbl_enc = preprocessing.LabelEncoder()

y = lbl_enc.fit_transform(data.sentiment.values)

# Splitting into training and testing data in 90:10 ratio

from sklearn.model_selection import train_test_split

X_train, X_val, y_train, y_val = train_test_split(data.content.values, y, stratify=y, random_state=42,
test_size=0.1, shuffle=True)

# Extracting TF-IDF parameters

from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(max_features=1000, analyzer='word',ngram_range=(1,3))

X_train_tfidf = tfidf.fit_transform(X_train)

X_val_tfidf = tfidf.fit_transform(X_val)

# Extracting Count Vectors Parameters

from sklearn.feature_extraction.text import CountVectorizer

count_vect = CountVectorizer(analyzer='word')

count_vect.fit(data['content'])

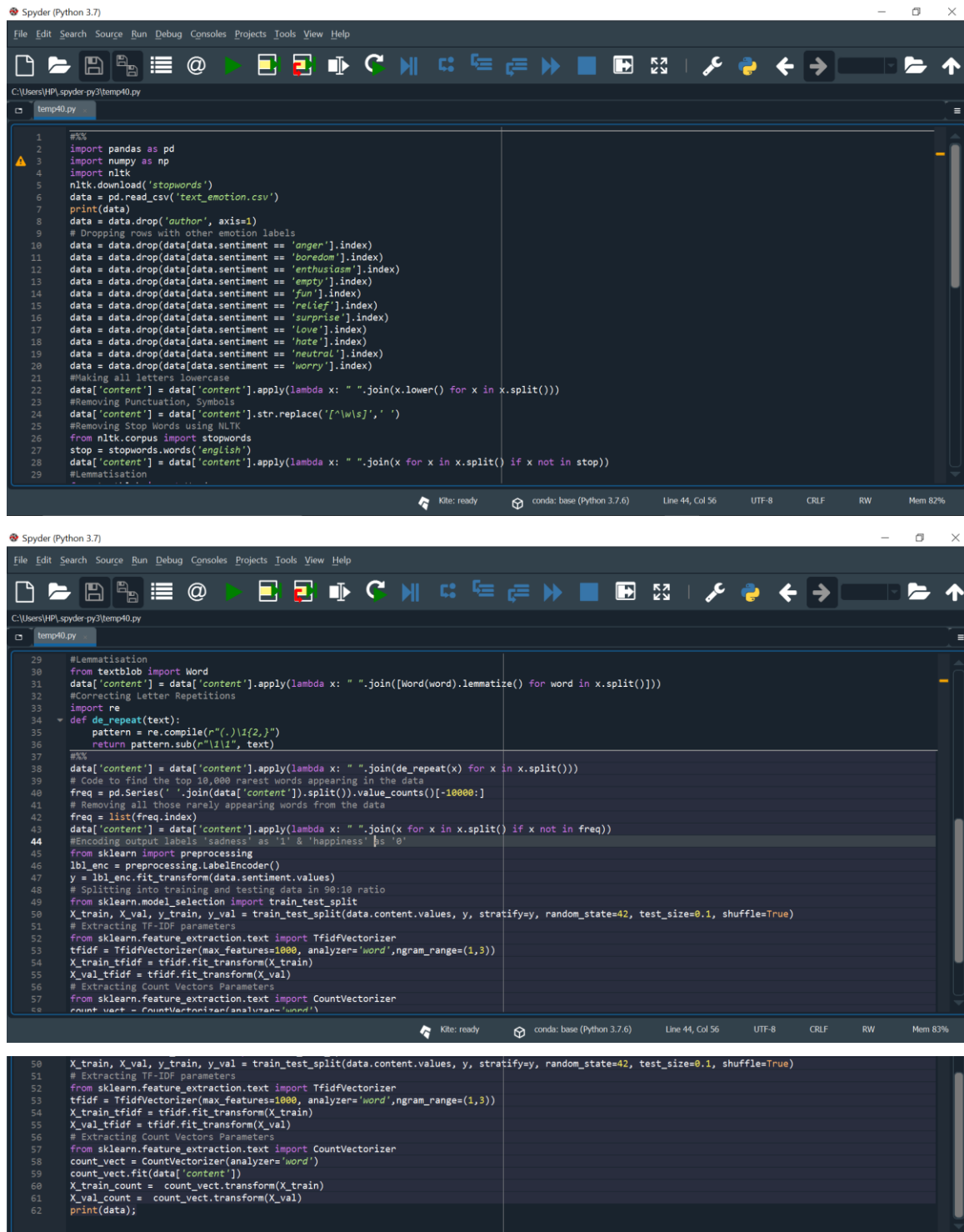
X_train_count = count_vect.transform(X_train)

X_val_count = count_vect.transform(X_val)

print(data);

```

Snapshots:



```
1  #####
2  import pandas as pd
3  import numpy as np
4  import nltk
5  nltk.download('stopwords')
6  data = pd.read_csv('text_emotion.csv')
7  print(data)
8  data = data.drop('author', axis=1)
9  # Dropping rows with other emotion labels
10 data = data.drop(data[data.sentiment == 'anger'].index)
11 data = data.drop(data[data.sentiment == 'boredom'].index)
12 data = data.drop(data[data.sentiment == 'enthusiasm'].index)
13 data = data.drop(data[data.sentiment == 'empty'].index)
14 data = data.drop(data[data.sentiment == 'fun'].index)
15 data = data.drop(data[data.sentiment == 'relief'].index)
16 data = data.drop(data[data.sentiment == 'surprise'].index)
17 data = data.drop(data[data.sentiment == 'Love'].index)
18 data = data.drop(data[data.sentiment == 'hate'].index)
19 data = data.drop(data[data.sentiment == 'neutral'].index)
20 data = data.drop(data[data.sentiment == 'worry'].index)
21 #Making all letters lowercase
22 data['content'] = data['content'].apply(lambda x: " ".join(x.lower() for x in x.split()))
23 #Removing Punctuation, Symbols
24 data['content'] = data['content'].str.replace('[^\w\s]',' ')
25 #Removing Stop Words using NLTK
26 from nltk.corpus import stopwords
27 stop = stopwords.words('english')
28 data['content'] = data['content'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
29 #Lemmatisation
30 from textblob import Word
31 data['content'] = data['content'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))
32 #Correcting Letter Repetitions
33 import re
34 def de_repeat(text):
35     pattern = re.compile(r'(\w)\1{2,}')
36     return pattern.sub(r'\1', text)
37
38 data['content'] = data['content'].apply(lambda x: " ".join(de_repeat(x) for x in x.split()))
39 # Code to find the top 10,000 rarest words appearing in the data
40 freq = pd.Series(' '.join(data['content']).split()).value_counts()[-10000:]
41 # Removing all those rarely appearing words from the data
42 freq = list(freq.index)
43 data['content'] = data['content'].apply(lambda x: " ".join(x for x in x.split() if x not in freq))
44 #Encoding output labels 'sadness' as '1' & 'happiness' as '0'
45 from sklearn import preprocessing
46 lbl_enc = preprocessing.LabelEncoder()
47 y = lbl_enc.fit_transform(data.sentiment.values)
48 # Splitting into training and testing data in 90:10 ratio
49 from sklearn.model_selection import train_test_split
50 X_train, X_val, y_train, y_val = train_test_split(data.content.values, y, stratify=y, random_state=42, test_size=0.1, shuffle=True)
51 # Extracting TF-IDF parameters
52 from sklearn.feature_extraction.text import TfidfVectorizer
53 tfidf = TfidfVectorizer(max_features=1000, analyzer='word', ngram_range=(1,3))
54 X_train_tfidf = tfidf.fit_transform(X_train)
55 X_val_tfidf = tfidf.fit_transform(X_val)
56 # Extracting Count Vectors Parameters
57 from sklearn.feature_extraction.text import CountVectorizer
58 count_vect = CountVectorizer(analyzer='word')
59 count_vect.fit(data['content'])
60 X_train_count = count_vect.transform(X_train)
61 X_val_count = count_vect.transform(X_val)
62 print(data);
```

The above three snapshots show the code typed in Spyder IDE. Packages such as pandas, numpy, textblob, nltk module are used to carry out the steps of the algorithm, that is, to read the data, filter it , pre-process it, classify it and do the feature extraction. This is part of the code used in the Spider interface. Using php interface we accessed the online Tweets using the tweepy module to classify the tweets under four different labels of classification.

Output snapshot:

Tweets Analysis

Search for your favorite topic in twitter, using Sentiment filter!

What tweets you want to see!

Use the sentiment and factual filters below

music

Search Tweets

Reset Filters

Positive Tweets

Negative Tweets

Opinion Tweet

Factual Tweet

1. RT @btsanalytics: @BTS_twt "Dynamite" on Korean music shows: • Most music show wins of 2020 • 2nd most music show wins for BTS after "Boy...
2. RT @yunowantsmilk: from home charting at no.10 on the music core chart 🍀! <https://t.co/wtF7qwkBxx>
3. RT @TheRealHoarse: Taylor Swift has 87.3 million followers on Twitter - the exact same number as Trump. This is the first campaign ad she...
4. RT @tinkswonu: jeonghan talked about how wonwoo and dino didnt look at the room in the mv where mingyu and jeonghan were hiding and wonwoo...
5. 'I know you're upset so we just gonna go get you food' music to my fcking ears 🍀
6. RT @Ride_MXtruck: 🍀 Mock streaming Monbebe Let's start to do mock streaming! Our pick song for this time is Newton!!!! No matter what musi...
7. RT @chol_bts2: They gave a spoiler 'Life Goes On' at UN, but actually it was not a spoiler, they just always talk about their real stories...
8. @86th_Bassman @FonzTramontano @Scorpx11 @tonymixclash @peterharich @ciroyelle @KarNeVor_Music @sleuthmusic1 @Seanthenumber1 @Stickupmusic @TrioMinutes @DJCrazyAnt yeah!! 🍀🍀🍀
9. RT @arashi5official: ニューヨークのタイムスクエアに嵐が出現 🍀🍀🍀 @spotifyJP のNew Music Fridayで今すぐ #PartyStarters も聴いてね 🍀 #ARASHIonSpotify <https://t.co/vM0Qem9...>

Tweets Analysis

Search for your favorite topic in twitter, using Sentiment filter!

What tweets you want to see!

Use the sentiment and factual filters below

music

Search Tweets

Reset Filters

Positive Tweets

Negative Tweets

Opinion Tweet

Factual Tweet

1. RT @baraju_SuperHit: Gorgeous and Talented @MalavikaM_ is on board for the @dhanushkraj starrer #D43 ! Directed by @karthicknaren_M & Musi...
2. Which is top tier music?
3. RT @otakuwwo: いのち/AZKi 文字PV Music 瀬名航 様 Vocal AZKi 様 Movie OTK #うごめろ #うごめろ3D #文字PV <https://t.co/6IX1ZrOT4E>
4. RT @sanbenito: #D4KITI 🍀🍀🍀🍀 <https://t.co/nFi0aU4xwv> <https://t.co/Dfozmxsnpn>
5. RT @BB_Votaciones: INFO 🍀🍀 @BTS_twt Felicidades a BTS que obtuvo el 1er lugar en Music Core por 8va semana consecutiva. 'Dynamite' extendi...
6. RT @WhoPaved_TheWay: OFICIALMENTE DYNAMITE DE @BTS_twt HA OBTENIDO SU WIN #20 Y ADEMÁS SE POSICIONA EN EL PUESTO #1 POR OCTAVA SEMANA CONSE...
7. RT @LLS2703: Lisa คัดอันดับ 15 ผู้ทรงอิทธิพลในอินสตาแกรม และอันดับ 6 ผู้ทรงอิทธิพลด้านดนตรีในอินสตาแกรม ในวันที่ 30 ต.ค. จัดอันดับโดย Hypea...
8. RT @BWBestPartners: the background music 🍀 sepanx - still2gether #SiamParagonxBrightWin #winmetawin #bbrightvc #โบรห์วิน <https://t.co/...>
9. @HornyFacts <https://t.co/lnqcl7YPDpy> thank me later! 🍀
10. bcp trop fort et sous cote ce mec <https://t.co/C5DFU1UcM1>

Tweets Analysis

Search for your favorite topic in twitter, using Sentiment filter!

What tweets you want to see!

Use the sentiment and factual filters below



music

Positive Tweets

Factual Tweet

1. RT @Univers_Bangtan: Parabéns ao BTS por ganhar o primeiro lugar pela 8ª semana consecutiva no Music Core por "Dynamite", sua 20ª vitória...

Tweets Analysis

Search for your favorite topic in twitter, using Sentiment filter!

What tweets you want to see!

Use the sentiment and factual filters below



music

Factual Tweet

1. RT @2cool4skull: Rt this so everyone finally learns about this, YOU CAN DOWNLOAD APPLE MUSIC AND STREAM ON YOUR ANDROID AS WELL!!! THERE IS...
2. OOR...バンドのレベルが馬鹿みたいに上がってる。本当に日本のバンドかよ。再結成後の Luna Sea と One Ok Rock はマジでヤバイ グループが海外のそこらのバンドと全然レベルが違う。OORに関しては関係者含めてだけど。<https://t.co/37ye7VkcZd>
3. RT @BTS_History613: [👉] Congratulations to @BTS_twt 'Dynamite' for winning 1st place on Music Core today. 🎊🏆🥇🥈🥉🏅🏆🥇🥈🥉🏅🏆🥇🥈🥉🏅🏆🥇🥈🥉🏅
🏆🏆🏆🏆🏆🏆🏆 Overall...
4. starting tomorrow I will strictly only be listening to christmas music.

Tweets Analysis

Search for your favorite topic in twitter, using Sentiment filter!

What tweets you want to see!

Use the sentiment and factual filters below

music

Search Tweets

Reset Filters

Positive Tweets

Opinion Tweet

Negative Tweets

Factual Tweet

1. RT @timetojinhyuk2: 191031 LEE JIN HYUK <I Like That> MUSIC THUMBNAIL #이진혁 #진혁 <https://t.co/aikyoyLvXZa>

2. @music_of_Kane 嬉し😊

3. my cat , music , rain <https://t.co/jMvkfjsG8R>

4. RT @btsyoutubedata: Most viewed @BTS_twt music videos in the past 24h <https://t.co/ExFd7wW41n>

5. RT @nhk_songs: 【#朝ドラエール 裏話&名曲】 今夜23:00 ~ #SONGS は #森山直太朗 #山崎育三郎 が名場面を振り返る! ◆御手洗ティファとの7D'リア合戦!? ◆藤堂先生 #故郷 歌唱♪で大切にしたこと ◆久志 #栄冠は君に輝く...

6. RT @NeoGlobalTeam: 🇺🇸 Reminder to vote for #NCT127 for the American Music Awards: Vote here: <https://t.co/S4ePWklsal> I'm voting for NCT 12...

7. Water Music Suite #2 in D HWV 349 by George Frideric Handel performed by Academy for Ancient Music, Berlin

8. This morning's soundtrack by @sigurros. Music that is unfailingly helpful for raising spirits--- and consciousness.

9. RT @nctshotarotops: #SHOTARO in Korea Drive-In Music Festival ! <https://t.co/LRyzu4puEf>

10. 腰痛い バッセンで遊びすぎた

11. RT @itskeyon: I can't trust people who don't like R&B music.

The above snapshots show the output displayed in the php server. The page shows the user with the space to enter their choice of topic to arrow down the tweets to apply the text emotion and sentiment analysis on it. It shows the tweets that fall under the topic the user entered and the user has options to filter the tweets to the four output labels as shown in the screenshots. The tweets are then filtered, and according to the filter label selected by the user, the corresponding tweets are displayed. The user can also reset the search, so that they can enter a fresh topic to classify the tweets into the labels.

Result analysis:

Data is retrieved online from Twitter by using the tweepy module. The data is then filtered and pre-processed and trained. Classifiers are built and is training with the data to classify them into the output labels. The algorithm works well and the interface created makes it easier for the user to use and understand. Using this people can classify data from various topics and who knows, this can even help to create big changes for day-to-day problems or solutions.

Conclusions:

While working with any dataset, filtering the dataset and making it clean, precise and error-free is mandatory for going on to the other stages of the program or project. Otherwise, unwanted data, or noise may hinder the progress or give error or wrong outputs. After that, feature extraction is the crucial step wherein we extract some parameters which we can represent numerically to carry on with the other stages. The implementation that is shown read the dataset and filtered it. It also removed unnecessary data from the dataset. Feature extraction is done by extracting Term Frequency-Inverse Document Frequency (TF-IDF) and Count Vectors parameters and classifying them into their respective labels. It built the Multinomial Naive Bayes Classifier, Linear SVM, logistic regression, Random Forest Classifier models with both the above features. Even though analysing text to find the emotions may sound and seem quite challenging, with learning the various options and ways, it can be done.

Scope of the project: This project will be helpful to the companies, political parties as well as to the common people. It will be helpful to political party for reviewing about the program that they are going to do or the program that they have performed. Similarly companies also can get review about their new product on newly released hardwares or softwares. Also the movie maker can take review on the currently running movie. By analyzing the tweets analyzer can get result on how positive or negative or neutral are peoples about it.

Future work will include enhancing the feature extraction, specifically in the area of removing the unwanted words, changing its form. Other works which can make this classifying algorithm more powerful is classifying using the emojis entered with the text.

REFERENCES:

- [1] Mehdi Hosseinzadeh Aghdam, Nasser Ghasem-Aghaee, Mohammad Ehsan Basiri, “Text feature selection using ant colony optimization”
- [2] Yu-Chuan Chang, Shyi-Ming Chen, Churn-Jung Liao, “Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method”
- [3] Anisha Mariam Thomas, Resmipriya M G, “An Efficient Text Classification Scheme Using Clustering”
- [4] Padmavathi.S, Dr. M. Chidambaram, “A Brief Survey on Text Classification Using Various Machine Learning Techniques”
- [5] Hao Lin, “Research on Energy-Efficient Text Classification”
- [6] Thabit Sabbah, Ali Selamat, Md Hafiz Selamat, Fawaz S. Al-Anzi, Enrique Herrera Viedma, Ondrej Krejcar and Hamido Fujita, “Modified Frequency-Based Term Weighting Schemes for Text Classification”
- [7] Thorsten Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”
- [8] Ghulam Mujtaba, Liyana Shuib, Ram Gopal Raj, Retnagowri Rajandram, Khairunisa Shaikh, “Prediction of Cause of Death from Forensic Autopsy Reports using Text Classification Techniques: A Comparative Study”
- [9] Bart Desmet, Véronique Hoste, “Online suicide prevention through optimised text classification”
- [10] Niusha Shafiabady, L.H. Lee, R. Rajkumar, V.P. Kallimani, Nik Ahmad Akram, Dino Isa, “Using unsupervised clustering approach to train the Support Vector Machine for text classification”
- [11] Abdur Rehman, Kashif Javed, Haroon A. Babri, Nabeel Asim, “Selection of the Most Relevant Terms Based on a Max-Min Ratio metric for Text Classification”
- [12] Krina Vasa, “Text Classification through Statistical and Machine Learning Methods: A Survey”

- [13] Felipe Viegas, Leonardo Rocha, Elaine Resende, Thiago Salles, Wellington Martins, Mateus Ferreira e Freitas, Marcos André Gonçalves, “Exploiting efficient and effective lazy Semi-Bayesian strategies for text classification”
- [14] Ahmed Faraz, “An elaboration of text categorization and automatic text classification through mathematical and graphical modelling”
- [15] Turgut Dogan and Alper Kursat Uysal, “Improved inverse gravity moment term weighting for text classification”
- [16] Savio L. Y. Lam & Dik Lun Lee, “Feature Reduction for Neural Network Based Text Categorization”
- [17] Fred J. Damerau, Tong Zhang, Sholom M. Weiss, Nitin Indurkha, “Text categorization for a comprehensive time-dependent benchmark”
- [18] M.E. ElAlami, “A filter model for feature subset selection based on genetic algorithm”
- [19] Bao Guo, Chunxia Zhang, Junmin Liu, Xiaoyi Ma, “Improving text classification with weighted word embeddings via a multi-channel TextCNN model”
- [20] Louis Ferre, “Selection of components in principal component analysis: A comparison of methods”
- [21] Marcin Kępa and Julian Szymański, “Two Stage SVM and kNN Text Documents Classifier”
- [22] Rogério C. P. Fragoso, Roberto H. W. Pinheiro, George D. C. Cavalcanti, “A method for automatic determination of the feature vector size for text categorization”
- [23] Jung-Yi Jiang, Ren-Jia Liou, and Shie-Jue Lee, “A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification”
- [24] Thiago Salles, Leonardo Rocha, Fernando Mourão, Marcos Gonçalves, Felipe Viegas, Wagner Meira Jr., “A Two-Stage Machine Learning Approach for Temporally-Robust Text Classification”
- [25] Alper Kursat Uysal, “An improved global feature selection scheme for text classification”