Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 2, May : 2023

Social Media Sentiment Analysis using NLP and AI Concepts

Shravani Joshi, G. J. Laxmi Priya UG Student,
B. Durga Bhavani Assistant Professor,
Department of Information Technology, Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Secunderabad, Telangana, India

ABSTRACT
Opinions are generally expressed for anything. For Example, a service, a product, a person, a topic, or an organization. The entity under observation has different components and may also have sub-components. Thus, the entity is called an object for sentiment analysis. Feature-based sentiment analysis uses the hierarchical model because objects are hierarchical in nature. The object may have sub-components and attributes. Therefore, it is difficult for general people to understand these technical terms (attribute or components). So, a simple word "Feature" is used for featured-based opining mining. Opinion or sentiment can be expressed in one sentence or in multiple sentences as a paragraph. Opinion word orientation determines the orientation of opinion. One single sentence can one or more opinion words. With the advancement of web technology and its growth, there is a huge volume of data present in the web for internet users and a lot of data is generated too. The Internet has become a platform for online learning, exchanging ideas and sharing opinions. Social networking sites like Twitter, Facebook, Google+ are rapidly gaining popularity as they allow people to share and express their views about topics, have discussions with different communities, or post messages across the world. There has been lot of work in the field of sentiment analysis of twitter data. This project focuses mainly on sentiment analysis of twitter data which is helpful to analyze the information in the tweets where opinions are highly unstructured, heterogeneous and are either positive or negative, or neutral in some cases.

Keywords: social media, content analysis, sentiment analysis, natural language processing, machine learning.

1. INTRODUCTION

1.1 Overview

Twitter is a social media platform for computer-mediated online communication, which shapes an emerging social structure. This communication platform has 1.3 billion accounts and 336 million active users posting 500 million tweets per day [1]. Twitter users can post comments known as ''tweets,'' each restricted to 140 characters prior to October 2018 and currently, 280 characters. Unless tweets are made private, they are publicly available and Twitter users can show their reaction to and engagement with a tweet by sharing it on their profile (retweet), clicking the like button, tagging someone's user name, or responding to the author of the tweet [2]. Twitter has also provided Application Programming Interfaces (APIs) to facilitate data collection. To access the API, a user can apply for a developer account.1 Following the application approval, the user has access to four keys: consumer key, consumer secret, access token, and access secret. These keys authenticate the user to access Twitter data such as tweets and profile information. Twitter's own API is the most potent available tool for collecting data generated through the interaction of Twitter users. Representing different demographic categories, Twitter data is a diverse and salient data source for researchers and policymakers. This global data source has earned the focus of several studies to address a wide range of research questions in different applications such as health and politics. While most studies used Twitter APIs for data collection such as, other studies manually collected data like, acquired Twitter data from commercial companies such as, or utilized previously collected data from other studies like.

The 21st century has witnessed a torrential flow of data. The data has sprung massively in various fields over the last two decades, which has led to the birth of big data [3]. Moreover, the influx of technology in the digital world has opened the doors for the development of big data. Citizens of the world are now becoming technology savvy with devices ranges from digital sensors, communication tools including social media applications, and actuators and data processors. For instance, organizations capture the mushrooming volume of transactional data, through which trillions of bytes of information are generated regarding aspects from suppliers to customers. The physical world has millions of network sensors embedded in devices like smart phones, smart energy meters, automobiles, and industrial machines. Such advances in digital sensors and communication technologies have led to the development of the Internet of Things (IoT). With such a development, social networking sites and communication devices like smart phones, laptops, and PCs allow individuals to interact with one another to create massive amounts of big data. For instance, Twitter's wide network of 467 million users generates 175 million tweets on a daily basis [5]. Similarly, the amount of space needed to store one second of a high-definition video is 2000 times more than the space needed to store a page of plain text. Furthermore, according to the International Data Corporation report in 2011, the world is already generated about 1 zettabyte (ZB) of data, and the rate at which this amount is growing has been exploding; the amount of data grew to 7ZB by the end of 2014. Moreover, by 2020, the amount of data generated is expected to reach 44ZB, with at least half of them being textual data [4] that is generated through social media technologies like Facebook, Twitter, and mobile instant messaging apps such as WhatsApp and Telegram. It has been determined that 500 million tweets are sent each day, while 40 million of those are shared daily. Meanwhile, it is estimated that 4.3 billion messages on Facebook are posted with 5.75 billion likes on a daily basis. Moreover, it is expected that the amount of data will continuously grow because of the influx of digital technologies that have already sprung up in the digital era.

1.2 Objecti1ve of the Project

With the advancement of web technology and its growth, there is a huge volume of data present on the web for internet users and a lot of data is generated too. The Internet has become a platform for online learning, exchanging ideas and sharing opinions. Social networking sites like Twitter, Facebook, Google+ are rapidly gaining popularity as they allow people to share and express their views about topics, have discussions with different communities, or post messages across the world. There has been lot of work in the field of sentiment analysis of twitter data. This project focuses mainly on sentiment analysis of twitter data which is helpful to analyze the information in the tweets where opinions are highly unstructured, heterogeneous and are either positive or negative, or neutral in some cases.

2. LITERATURE SURVEY

Jianqiang et. al [6] introduced a word embeddings method obtained by unsupervised learning based on large twitter corpora, this method using latent contextual semantic relationships and co-occurrence statistical characteristics between words in tweets. These word embeddings are combined with n-grams features and word sentiment polarity score features to form a sentiment feature set of tweets. The feature set is integrated into a deep convolution neural network for training and predicting sentiment classification labels. They experimentally compare the performance of our model with the baseline model that is a word n-grams model on five Twitter data sets, the results indicate that our model performs better on the accuracy and F1-measure for twitter sentiment classification.

Pai et. al [7] presented a framework that consists of both time series forecasting models and multivariate regression technique to predict monthly total vehicle sales. Deseasonalizing procedures were employed to deal with different types of data. The numerical results indicate that forecasting vehicle sales by hybrid multivariate regression data with desonalizing procedures can obtain more accurate forecasting results than other forecasting models. The superior forecasting performance

could be concluded as follows. First, the use of hybrid data containing sentiment analysis of social media and stock market values can improve the forecasting accuracy. Secondly the deceasonalizing procedures both in condition variables and decision variables do help to increase the prediction performance. For future study, since the determination of keywords for Twitter significantly affects the search results of tweets and have influences on forecasting accuracy, a more systematized technique for selecting proper keywords from Twitter could be a direction for future study. Another possible direction for future study is to employ other social media data, such as Facebook and YouTube to forecast vehicle sales. Finally, the geographical information collection of Twitter possibly could be an essential issue for future study to improve tweets analysis.

Shayaa et. al [8] presents a comprehensive systematic literature review, aims to discuss both technical aspect of OMSA (techniques and types) and non-technical aspect in the form of application areas are discussed. Furthermore, this paper also highlighted both technical aspects of OMSA in the form of challenges in the development of its technique and non-technical challenges mainly based on its application. These challenges are presented as a future direction for research.

Naseem et. al [9] aims to inform policy that can be applied to social media platforms; for example, determining what degree of moderation is necessary to curtail misinformation on social media. This study also analyzes views concerning COVID-19 by focusing on people who interact and share social media on Twitter. As a platform for our experiments, we present a new large-scale sentiment data set COVIDSENTI, which consists of 90 000 COVID-19-related tweets collected in the early stages of the pandemic, from February to March 2020. The tweets have been labeled into positive, negative, and neutral sentiment classes. We analyzed the collected tweets for sentiment classification using different sets of features and classifiers. Negative opinion played an important role in conditioning public sentiment, for instance, we observed that people favored lockdown earlier in the pandemic; however, as expected, sentiment shifted by mid-March. Our study supports the view that there is a need to develop a proactive and agile public health presence to combat the spread of negative sentiment on social media following a pandemic.

Usman Naseem et. al [10] present, a transformer-based method for sentiment analysis that encodes representation from a transformer and applies deep intelligent contextual embedding to enhance the quality of tweets by removing noise while taking word sentiments, polysemy, syntax, and semantic knowledge into account. We also use the bidirectional long- and short-term memory network to determine the sentiment of a tweet. To validate the performance of the proposed framework, we perform extensive experiments on three benchmark datasets, and results show that considerably outperforms the state of the art in sentiment classification.

Naseem et. al [11] proposed a Deep Intelligent Contextual Embedding (DICE), which enhances the tweet quality by handling noises within contexts, and then integrates four embeddings to involve polysemy in context, semantics, syntax, and sentiment knowledge of words in a tweet. DICE is then fed to a Bi-directional Long Short-Term Memory (BiLSTM) network with attention to determine the sentiment of a tweet. The experimental results show that our model outperforms several baselines of both classic classifiers and combinations of various word embedding models in the sentiment analysis of airline-related tweets.

Aloufi et. al [12] focused on analyzing sentiment expressed by football fans through Twitter. These tweets reflect the changes in the fans' sentiment as they watch the game and react to the events of the game, e.g., goal scoring, penalties, and so on. Collecting and examining the sentiment conveyed through these tweets will help to draw a complete picture which expresses fan interaction during a specific football event. The objective of this work is to propose a domain-specific approach for understanding sentiments expressed in football fans' conversations. To achieve our goal, they start by developing a football-specific sentiment dataset which they label manually. They then utilize our dataset to automatically create a football-specific sentiment lexicon. Finally, they develop a sentiment classifier which is capable of recognizing sentiments expressed in football conversation. They conduct extensive experiments on our dataset to compare the performance of different learning

algorithms in identifying the sentiment expressed in football related tweets. These results show that our approach is effective in recognizing the fans' sentiment during football events.
Rahman et. al [13] presented a model that can perform sentiment analysis of real data collected from Twitter. Data in Twitter is highly unstructured which makes it difficult to analyze. However, our proposed model is different from prior work in this field because it combined the use of supervised and unsupervised machine learning algorithms. The process of performing sentiment analysis as follows: Tweet extracted directly from Twitter API, then cleaning and discovery of data performed. After that the data were fed into several models for the purpose of training. Each tweet extracted classified based on its sentiment whether it is a positive, negative or neutral. Data were collected on two subjects McDonalds and KFC to show which restaurant has more popularity. Different machine learning algorithms were used. The result from these models were tested using various testing metrics like cross validation and f-score. Moreover, our model demonstrates strong performance on mining texts extracted directly from Twitter.
Arora et. al [14] proposes a text normalization with deep convolutional character level embedding (Conv-char-Emb) neural network model for SA of unstructured data. This model can tackle the problems: (1) processing the noisy sentence for sentiment detection (2) handling small memory space in word level embedded learning (3) accurate sentiment analysis of the unstructured data. The initial preprocessing stage for performing text normalization includes the following steps: tokenization, out of vocabulary (OOV) detection and its replacement, lemmatization and stemming. A character-based embedding in convolutional neural network (CNN) is an effective and efficient technique for SA that uses less learnable parameters in feature representation. Thus, the proposed method performs both the normalization and classification of sentiments for unstructured sentences. The experimental results are evaluated in the Twitter dataset by a different point polarity (positive, negative and neutral). As a result, our model performs well in normalization and sentiment analysis of the raw Twitter data enriched with hidden information.
Wang et. al [15] focused on how to fuse textual information of Twitter messages and sentiment diffusion patterns to obtain better performance of sentiment analysis on Twitter data. To this end, we first analyze sentiment diffusion by investigating a phenomenon called sentiment reversal, and find some interesting properties of sentiment reversals. Then, we consider the inter-relationships between textual information of Twitter messages and sentiment diffusion patterns, and propose an iterative algorithm called SentiDiff to predict sentiment polarities expressed in Twitter messages. To the best of our knowledge, this work is the first to utilize sentiment diffusion patterns to help improve Twitter sentiment analysis. Extensive experiments on real-world dataset demonstrate that compared with state-of-the-art textual information-based sentiment analysis algorithms, our proposed algorithm yields PR-AUC improvements between 5.09 and 8.38 percent on Twitter sentiment classification tasks.

3. EXISTING SYSTEM

3.1 Multinominal Naive Bayes Model

Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.
Naive Bayes classifier is a collection of many algorithms where all the algorithms share one common principle, and that is each feature being classified is not related to any other feature. The presence or absence of a feature does not affect the presence or absence of the other feature.

How Multinomial Naive Bayes works?

Naive Bayes is a powerful algorithm that is used for text data analysis and with problems with multiple classes. To understand Naive Bayes theorem's working, it is important to understand the Bayes theorem concept first as it is based on the latter.

Bayes theorem, formulated by Thomas Bayes, calculates the probability of an event occurring based on the prior knowledge of conditions related to an event. It is based on the following formula:

$$P(A|B) = P(A) * P(B|A)/P(B)$$

Where we are calculating the probability of class A when predictor B is already provided.
P(B) = prior probability of B
P(A) = prior probability of class A
P(B|A) = occurrence of predictor B given class A probability

3.2 Disadvantages of existing system

The Naive Bayes algorithm has the following disadvantages:

- The prediction accuracy of this algorithm is lower than the other probability algorithms.
- It is not suitable for regression. Naive Bayes algorithm is only used for textual data classification and cannot be used to predict numeric values.

4. PROPOSED SYSTEM

4.1 Data Preprocessing in Machine learning

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.
When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

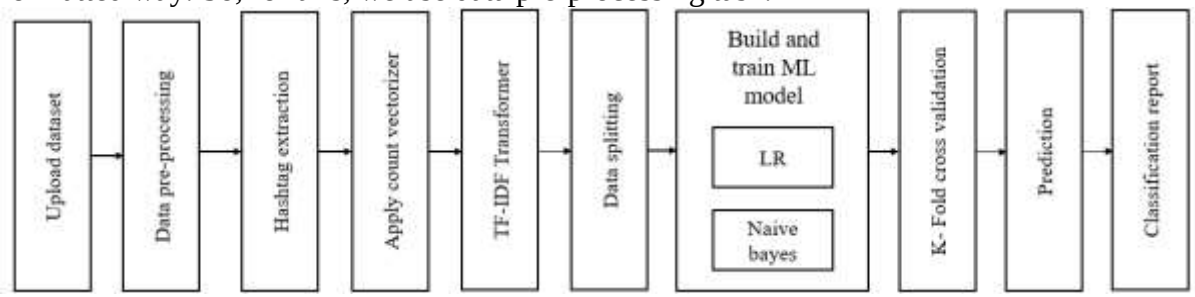


Fig. 4.1: Block diagram of proposed system.

4.2 TF-IDF Feature extraction

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. Let's take an example, we have a string or Bag of Words (BOW) and we have to extract information from it, then we can use this approach. The tf-idf value increases in proportion to the number of times a word appears in the document but is often offset by the frequency of the word in the corpus, which helps to adjust with respect to the fact that some words appear more frequently in general. TF-IDF use two statistical methods, first is Term Frequency and the other is Inverse Document Frequency. Term frequency refers to the total number of times a given term t appears in the document doc against (per) the total number of all words in the document and The inverse document frequency measure of how much information the word provides. It measures the weight of a given word in the entire document. IDF show how common or rare a given word is across all documents. TF-IDF can be computed as tf * idf

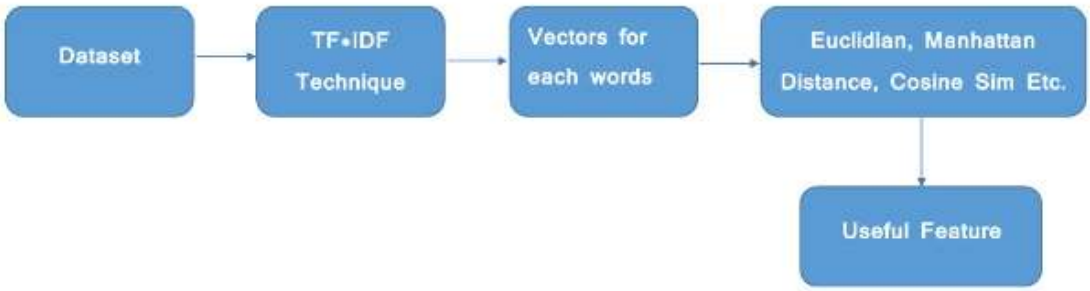


Fig. 4.3: TF-IDF block diagram.

TF-IDF do not convert directly raw data into useful features. Firstly, it converts raw strings or dataset into vectors and each word has its own vector. Then we'll use a particular technique for retrieving the feature like Cosine Similarity which works on vectors, etc.

Terminology

t — term (word)

d — document (set of words)

N — count of corpus

corpus — the total document set

Step 1: Term Frequency (TF): Suppose we have a set of English text documents and wish to rank which document is most relevant to the query, "Data Science is awesome!" A simple way to start out is by eliminating documents that do not contain all three words "Data" is", "Science", and "awesome", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its term frequency. The weight of a term that occurs in a document is simply proportional to the term frequency.

$$tf(t,d) = count\ of\ t\ in\ d\ /\ number\ of\ words\ in\ d$$

Step 2: Document Frequency: This measures the importance of document in whole set of corpora, this is very similar to TF. The only difference is that TF is frequency counter for a term t in document d, whereas DF is the count of occurrences of term t in the document set N. In other words, DF is the number of documents in which the word is present. We consider one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.

$$df(t) = occurrence\ of\ t\ in\ documents$$

Step 3: Inverse Document Frequency (IDF): While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing IDF, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. The IDF is the inverse of the document frequency which measures the informativeness of term t. When we calculate IDF, it will be very low for the most occurring words such as stop words (because stop words such as "is" is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.

$$idf(t) = N/df$$

Now there are few other problems with the IDF, in case of a large corpus, say 100,000,000, the IDF value explodes , to avoid the effect we take the log of idf . During the query time, when a word which is not in vocab occurs, the df will be 0. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.

$$idf(t) = log(N/(df + 1))$$

The TF-IDF now is at the right measure to evaluate how important a word is to a document in a collection or corpus. Here are many different variations of TF-IDF but for now let us concentrate on this basic version.

$$tf - idf(t,d) = tf(t,d) * log(N/(df + 1))$$

Step 4: Implementing TF-IDF: To make TF-IDF from scratch in python, let's imagine those two sentences from different document:
first sentence: "Data Science is the sexiest job of the 21st century".
second sentence: "machine learning is the key for data science".

Natural Language Toolkit (NLTK)
NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenization, lower case conversion, Stop Words removal, stemming, and lemmatization.

Tokenization
The breaking down of text into smaller units is called tokens. tokens are a small part of that text. If we have a sentence, the idea is to separate each word and build a vocabulary such that we can represent all words uniquely in a list. Numbers, words, etc. all fall under tokens.

Lower case conversion
We want our model to not get confused by seeing the same word with different cases like one starting with capital and one without and interpret both differently. So we convert all words into the lower case to avoid redundancy in the token list.

Stop Words removal
When we use the features from a text to model, we will encounter a lot of noise. These are the stop words like the, he, her, etc… which don't help us and just be removed before processing for cleaner processing inside the model. With NLTK we can see all the stop words available in the English language.

Stemming
In our text we may find many words like playing, played, playfully, etc… which have a root word, play all of these convey the same meaning. So we can just extract the root word and remove the rest. Here the root word formed is called 'stem' and it is not necessarily that stem needs to exist and have a meaning. Just by committing the suffix and prefix, we generate the stems.

Lemmatization
We want to extract the base form of the word here. The word extracted here is called Lemma and it is available in the dictionary. We have the WordNet corpus and the lemma generated will be available in this corpus. NLTK provides us with the WordNet Lemmatizer that makes use of the WordNet Database to lookup lemmas of words.

4.3 Logistic Regression
Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical). A linear regression is not appropriate for predicting the value of a binary variable for two reasons:

- A linear regression will predict values outside the acceptable range (e.g. predicting probabilities
- outside the range 0 to 1)
- Since the dichotomous experiments can only have one of two possible values for each experiment, the residuals will not be normally distributed about the predicted line.

On the other hand, a logistic regression produces a logistic curve, which is limited to values between 0 and 1. Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the "odds" of the target variable, rather than the probability. Moreover, the predictors do not have to be normally distributed or have equal variance in each group.

In the logistic regression the constant (b0) moves the curve left and right and the slope (b1) defines the steepness of the curve. By simple transformation, the logistic regression equation can be written in terms of an odds ratio.

$$\frac{p}{1-p} = \exp(b_0 + b_1 x)$$

Finally, taking the natural log of both sides, we can write the equation in terms of log-odds (logit) which is a linear function of the predictors. The coefficient (b_1) is the amount the logit (log-odds) changes with a one unit change in x.

$$ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x$$

As mentioned before, logistic regression can handle any number of numerical and/or categorical variables.

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}}$$

There are several analogies between linear regression and logistic regression. Just as ordinary least square regression is the method used to estimate coefficients for the best fit line in linear regression, logistic regression uses maximum likelihood estimation (MLE) to obtain the model coefficients that relate predictors to the target. After this initial function is estimated, the process is repeated until LL (Log Likelihood) does not change significantly.

$$\boldsymbol{\beta^1 = \beta^0 + [X^T W X]^{-1} . X^T (y - \mu)}$$

$\boldsymbol{\beta}$ is a vector of the logistic regression coefficients.

$\boldsymbol{W}$ is a square matrix of order N with elements $n_i \pi_i (1 - \pi_i)$ on the diagonal and zeros everywhere else.

$\boldsymbol{\mu}$ is a vector of length N with elements $\mu_i = n_i \pi_i$.

A pseudo R^2 value is also available to indicate the adequacy of the regression model. Likelihood ratio test is a test of the significance of the difference between the likelihood ratio for the baseline model minus the likelihood ratio for a reduced model. This difference is called "model chi-square". Wald test is used to test the statistical significance of each coefficient (b) in the model (i.e., predictors contribution).

Pseudo R2

There are several measures intended to mimic the R2 analysis to evaluate the goodness-of-fit of logistic models, but they cannot be interpreted as one would interpret an R2 and different pseudo R2 can arrive at very different values. Here we discuss three pseudo R2measures.

Likelihood Ratio Test

The likelihood ratio test provides the means for comparing the likelihood of the data under one model (e.g., full model) against the likelihood of the data under another, more restricted model (e.g., intercept model).

$$LL = \sum_{i=1}^{n} y_i ln(p_i) + (1 - y_i) ln(1 - p_i)$$

where *'p'* is the logistic model predicted probability. The next step is to calculate the difference between these two log-likelihoods.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 2, May : 2023

$$2(LL_1 - LL_2)$$

The difference between two likelihoods is multiplied by a factor of 2 in order to be assessed for statistical significance using standard significance levels (Chi^2 test). The degrees of freedom for the test will equal the difference in the number of parameters being estimated under the models (e.g., full and intercept).

4.4 Advantages of proposed system

- Logistic regression is easier to implement, interpret, and very efficient to train.
- It can easily extend to multiple classes (multinomial regression) and a natural probabilistic view of class predictions.
- Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.
- Logistic regression is less inclined to over-fitting but it can overfit in high dimensional datasets. One may consider Regularization (L1 and L2) techniques to avoid over-fifty-ninth ese scenarios.

5. RESULTS AND DISCUSSION

Sample training data

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation

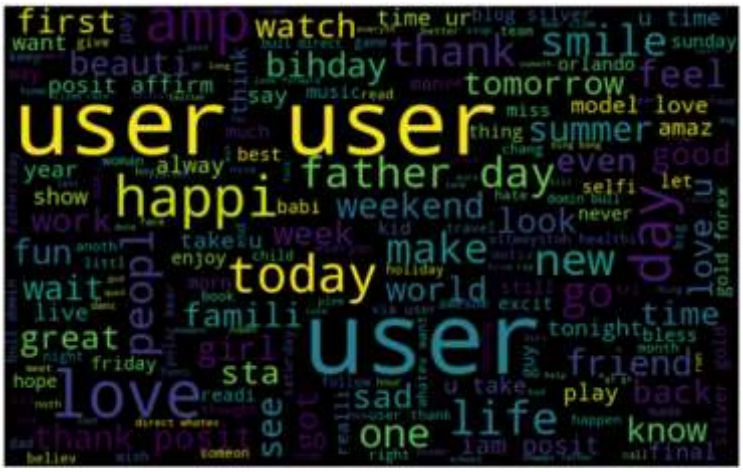
Data description

	id	label
count	31962.000000	31962.000000
mean	15981.500000	0.070146
std	9226.778988	0.255397
min	1.000000	0.000000
25%	7991.250000	0.000000
50%	15981.500000	0.000000
75%	23971.750000	0.000000
max	31962.000000	1.000000

Word cloud of training dataset

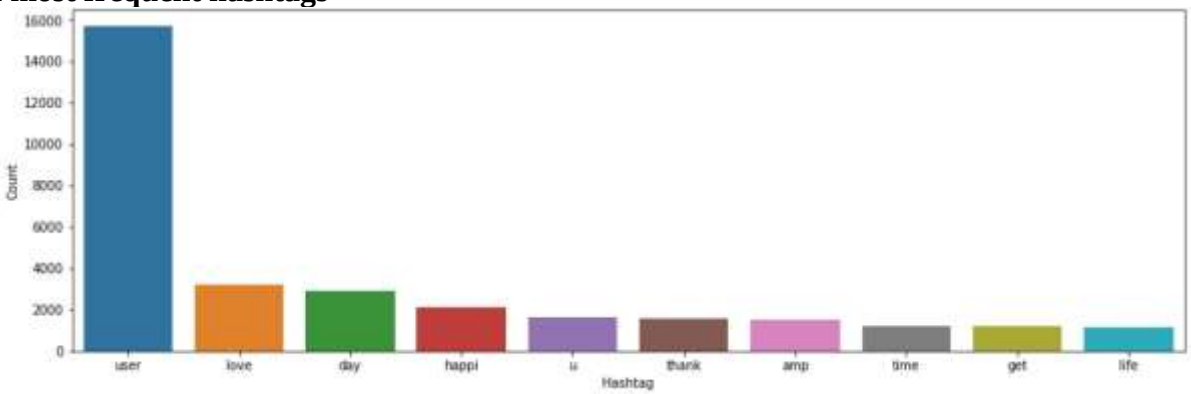
Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 2, May : 2023

Word cloud of normal words



Word cloud of negative words

Ten most frequent hashtags

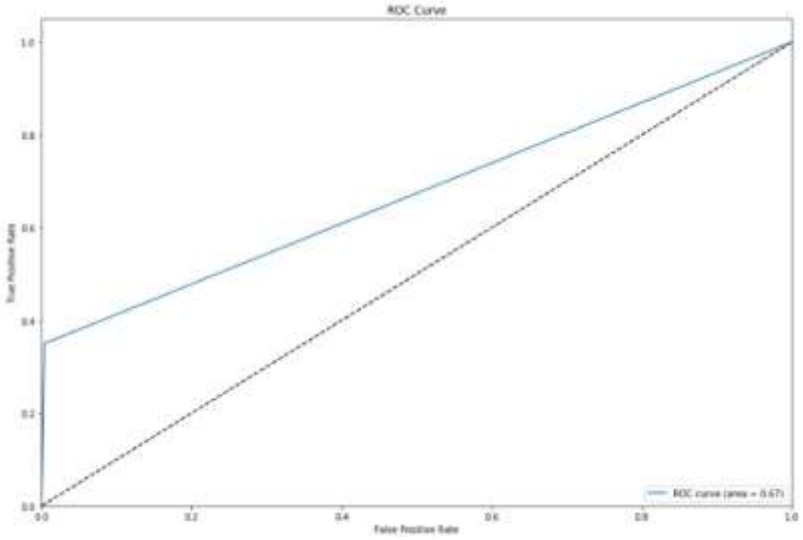


Classification report of LR

```
print(classification_report(text_y, y_pred))
```

	precision	recall	f1-score	support
0	0.96	1.00	0.98	7460
1	0.87	0.35	0.50	531
accuracy			0.95	7991
macro avg	0.91	0.67	0.74	7991
weighted avg	0.95	0.95	0.94	7991

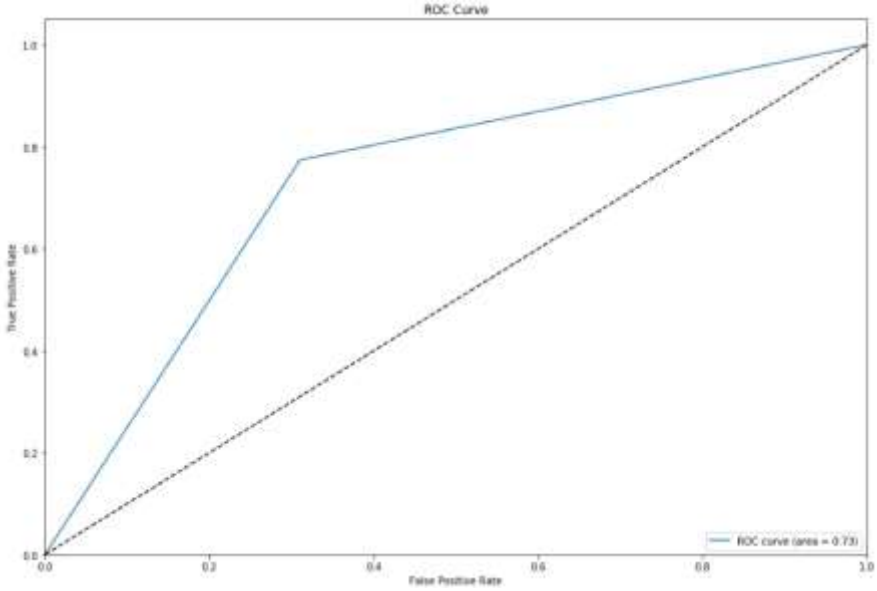
Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 2, May : 2023



Classification report of GNB

```
print(classification_report(text_y, y_predict))
```

	precision	recall	f1-score	support
0	0.98	0.69	0.81	7460
1	0.15	0.77	0.25	531
accuracy			0.70	7991
macro avg	0.56	0.73	0.53	7991
weighted avg	0.92	0.70	0.77	7991



6. CONCLUSION

With the advancement of web technology and its growth, there is a huge volume of data present on the web for internet users and a lot of data is generated too. The Internet has become a platform for online learning, exchanging ideas and sharing opinions. Social networking sites like Twitter, Facebook, Google+ are rapidly gaining popularity as they allow people to share and express their views about topics, have discussions with different communities, or post messages across the world. Therefore, this project implemented the sentiment analysis of twitter dataset for opinion mining using NLP, AI, and lexicon-based approaches, together with evaluation metrics. Using various machine learning algorithms like Naive Bayes, and logistic regression, this work provided research

on twitter data streams. In addition, this project has also discussed general challenges and applications of Sentiment Analysis on Twitter.

REFERENCES

[1] M. Ahlgren. (2020). 40+ Twitter Statistics & Facts For 2020. Accessed: Mar. 22, 2020. [Online]. Available: https://www.websitehostingrating.com/twitter-statistics/

[2] D. Arigo, S. Pagoto, L. Carter-Harris, S. E. Lillie, and C. Nebeker, ‘‘Using social media for health research: Methodological and ethical considerations for recruitment and intervention delivery,’’ Digit. Health, vol. 4, Jan. 2018, Art. no. 205520761877175

[3] R. Addo-tenkorang and P. T. Helo, ‘‘Big data applications in operations/supply-chain management: A literature review,’’ Comput. Ind. Eng., vol. 101, pp. 528–543, Nov. 2016.

[4] M. Khoso. (2016). How Much Data is Produced Every Day? [Online]. Available: http://www.northeastern.edu/levelblog/2016/05/13/how-muchdata-produced-every

[5] A. Yasin, Y. Ben-Asner, and A. Menaeison, ‘‘Deep-dive analysis or the data analytics workload in cloudsuite,’’ in Proc. IEEE Int. Symp. Workload Characterization (IISWC), Oct. 2014, pp. 202–211.

[6] Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," in IEEE Access, vol. 6, pp. 23253-23260, 2018, doi: 10.1109/ACCESS.2017.2776930.

[7] P. -F. Pai and C. -H. Liu, "Predicting Vehicle Sales by Sentiment Analysis of Twitter Data and Stock Market Values," in IEEE Access, vol. 6, pp. 57655-57662, 2018, doi: 10.1109/ACCESS.2018.2873730.

[8] S. Shayaa, "Sentiment Analysis of Big Data: Methods, Applications, and Open Challenges," in IEEE Access, vol. 6, pp. 37807-37827, 2018, doi: 10.1109/ACCESS.2018.2851311.

[9] U. Naseem, I. Razzak, M. Khushi, P. W. Eklund and J. Kim, "COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis," in IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 1003-1015, Aug. 2021, doi: 10.1109/TCSS.2021.3051189.

[10] Usman Naseem, Imran Razzak, Katarzyna Musial, Muhammad Imran, Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis, Future Generation Computer Systems, Volume 113, 2020, Pages 58-69, ISSN 0167-739X, https://doi.org/10.1016/j.future.2020.06.050.

[11] U. Naseem and K. Musial, "DICE: Deep Intelligent Contextual Embedding for Twitter Sentiment Analysis," 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 953-958, doi: 10.1109/ICDAR.2019.00157.

[12] S. Aloufi and A. E. Saddik, "Sentiment Identification in Football-Specific Tweets," in IEEE Access, vol. 6, pp. 78609-78621, 2018, doi: 10.1109/ACCESS.2018.2885117.

[13] S. A. El Rahman, F. A. AlOtaibi and W. A. AlShehri, "Sentiment Analysis of Twitter Data," 2019 International Conference on Computer and Information Sciences (ICCIS), 2019, pp. 1-4, doi: 10.1109/ICCISci.2019.8716464.

[14] Arora, M., Kansal, V. Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis. Soc. Netw. Anal. Min. 9, 12 (2019). https://doi.org/10.1007/s13278-019-0557-y

[15] L. Wang, J. Niu and S. Yu, "SentiDiff: Combining Textual Information and Sentiment Diffusion Patterns for Twitter Sentiment Analysis," in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 10, pp. 2026-2039, 1 Oct. 2020, doi: 10.1109/TKDE.2019.2913641.