IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Monitoring people's emotion using Sentiment Analysis and Deep Learning on COVID-19 Related Tweets

Dr.P. RAMYA, Ph.D[1],GOKUL.M[2], GURUMOORTHI.P[3],GOKUL.C[4],MEIBHARATHI.R[5]
1 ASSOCIATE PROFESSOR, 2,3,4,5 UG STUDENTS
Computer science and engineering
Mahendra Engineering College, Tamilnadu, India

ABSTRACT

Between January and October of 2020, the severe acute respiratory syndrome coronavirus 2(SARS-CoV-2) virus has infected more than 34 million persons in a worldwide pandemic leading to over one million deaths worldwide (data from the Johns Hopkins University). Since the virus begun tospread, emergency departments were busy with COVID-19 patients for whom a quick decision regarding in- or outpatient care was required. The virus can cause characteristic abnormalities in chest radiographs(CXR), but, due to the low sensitivity of CXR, additional variables and criteria are needed to accurately predict risk. Here, we describe a computerized system primarily aimed at extracting the most relevant radiological, clinical, and laboratory variables for improving patient risk prediction, and secondarily at presenting an explainable machine learning system, which may provide simple decision criteria to be used by clinicians as a support for assessing patient risk. To achieve robust and reliable variable selection, Boruta and Random Forest (RF) are combined in a 10-fold cross-validation scheme to produce a variable importance estimate not biased by the presence of surrogates. The most important variables are then selected to train a RF classifier, whose rules may be extracted, simplified, and pruned to finally build an associative tree, particularly appealing for its simplicity. Results show that the radiological score automatically computed through a neural network is highly correlated with the score computed by radiologists, and that laboratory variables, together with the number of comorbidities, aid risk prediction. The prediction performance of our approach was compared to that that of generalized linear models and shown to be effective and robust. The proposed machine learning-based computational system can be easily deployed and used in emergency departments for rapid and accurate risk prediction in COVID-19 patients.

KEYWORDS: Accurate risk prediction, emergency, coronavirus, machine learning system

INTRODUCTION

The age of Internet has changed the way people express their views. It is now done through blog posts, online discussion forums, product review websites etc. People depend upon this user generated content to a great extent. When someone wants to buy a product, they will look up its reviews online before taking a decision. The amount of user generated content is too large for a normal user to analyze. So to automate this, various sentiment analysis techniques are used. Symbolic techniques or Knowledge base approach and Machine learning techniques are the two main techniques used in sentiment analysis. Knowledge base approach requires a large database of predefined emotions and an efficient knowledge representation for identifying sentiments. Machine learning approach makes use of a training set to develop a sentiment classifier that classifies sentiments. Since a predefined database of entire emotions is not required for machine learning

approach, it is rather simpler than Knowledge base approach. In this paper, we use different machine learning techniques for classifying tweets. Sentiment analysis is usually conducted at different levels varying from coarse level to fine level. Coarse level sentiment analysis deals with determining the sentiment of an entire document and Fine level deals with attribute level sentiment analysis. Sentence level sentiment analysis comes in between these two. There are many researches on the area of sentiment analysis of user reviews. Previous researches show that the performances of sentiment classifiers are dependent on topics. Because of that we cannot say that one classifier is the best for all topics since one classifier does not consistently outperforms the other. Sentiment Analysis in twitter is quite difficult due to its short length. Presence of emoticons, slang words and misspellings in tweets forced to have a preprocessing step before feature extraction.

LITERATURE SURVEY

AN EMPIRICAL STUDY OF SMOOTHING TECHNIQUES FOR LANGUAGE MODELING

AUTHORS: STANLEY F. CHEN, JOSHUA GOODMAN.

Language models are a staple in many domains including speech recognition, optical character recognition, handwriting recognition, machine translation, and spelling correction. The dominant technology in language modeling is n-gram models, which are straightforward to construct except for the issue of smoothing, a technique used to better estimate probabilities when there is tweet data to estimate probabilities accurately. An enormous number of techniques have been proposed for smoothing n-gram models, many more than we could possibly describe here; however, there has been a conspicuous absence of studies that systematically compare the relative performance of more than just a few of these algorithms on multiple data sets. As a result, from the literature it is impossible to gauge the relative performance of existing algorithms in all but a handful of situations. In this work, we attempt to dispel some of the mystery surrounding smoothing by determining which algorithms work well in which situations, and why. We begin by giving a tutorial introduction to n-gram models and smoothing, and survey the most widely-used smoothing techniques.

SMOOTHING:

Smoothing is used to address this problem. The term smoothing describes techniques for adjusting the maximum likelihood estimate of probabilities to produce more accurate probabilities. The name smoothing comes from the fact that these techniques tend to make distributions more uniform, by adjusting low probabilities such as zero probabilities upward, and high probabilities downward. Not only do smoothing methods generally prevent zero probabilities, but they also attempt to improve the accuracy of the model as a whole. Whenever a probability is estimated from few counts, smoothing has the potential to significantly improve estimation. While smoothing is a central issue in language modeling, the literature lacks a denitive comparison between the many existing techniques. Previous studies (Nadas, 1984; Katz, 1987; Church and Gale, 1991; MacKay and Peto, 1995; Kneser and Ney, 1995) only compare a small number of methods (typically two) on one or two corpora and using a single training set size. As a result, it is currently diult for a researcher to intelligently choose among smoothing schemes.

EMOTICON SMOOTHED LANGUAGE MODELS FOR TWITTER SENTIMENT ANALYSIS

AUTHORS: KUN-LIN LIU, WU-JUN LI, MINYI GUO.

In this paper, we present a novel model, called emoticon smoothed language model (ESLAM), to handle this challenge. The basic idea is to train a language model based on the manually labeled data, and then use the noisy emoticon data for smoothing. Experiments on real data sets demonstrate that ESLAM can effectively integrate both kinds of data to outperform those methods using only one of them. Sentiment analysis (SA) (Pang and Lee 2007) (also known as opinion mining) is mainly about discovering “what others think” from data such as product reviews and news articles. On one hand, consumers can seek advices about a product to make informed decisions in the consuming process. On the other hand, vendors are paying more and more attention to online opinions about their products and services. Hence, SA has attracted increasing attention from many research communities such as machine learning, data mining, and natural language processing. The sentiment of a document or sentence can be positive, negative or neutral. Hence, SA is actually a three-way classification problem. In practice, most methods adopt a two-step strategy for SA (Pang

and Lee 2007). In the subjectivity classification step, the target is classified to be subjective or neutral (objective), and in the polarity classification step, the subjective targets are further classified as positive or negative. Hence, two classifiers are trained for the whole SA process, one is called subjectivity classifier, and the other is called polarity classifier. Since formulated SA as a machine learning based text classification problem, more and more machine learning methods have been proposed for SA (Pang and Lee 2007).

One way to perform Twitter sentiment analysis (TSA) is to directly exploit traditional SA methods (Pang and Lee 2007). However, tweets are quite different from other text forms like product reviews and news articles. Firstly, tweets are often short and ambiguous because of the limitation of characters. Secondly, there're more misspelled words, slang, modal particles and acronyms on Twitter because of its casual form. Thirdly, a huge amount of unlabeled or noisy labeled data can be easily downloaded through Twitter API. Therefore, many novel SA methods have been specially developed for TSA. These methods can be mainly divided into two categories: fully supervised methods and distantly supervised methods2.The fully supervised methods try to learn the classifiers from manually labeled data. (Jansen et al. 2009) uses the multinomial Bayes model to perform automatic TSA. Compares support vector machine (SVM) and multinomial naive Bayes (MNB) for both blog and microblog SA, and finds that SVM outperforms MNB on blogs with long text but MNB outperforms SVM on microblogs with short text. One problem with the fully supervised methods is that it is very labor-intensive and time-consuming to manually label the data and hence the training data sets for most methods are often too small to guarantee a good performance.

Considering the shortcomings of the fully supervised and distantly supervised methods, we argue that the best strategy is to utilize both manually labeled data and noisy labeled data for training. However, how to seamlessly integrate these two different kinds of data into the same learning framework is still a challenge. In this paper, we propose a novel model, called emoticon smoothed language model (ESLAM), to handle this challenge. The main contributions of ESLAM are outlined as follows:

- ESLAM uses the noisy emoticon data to smooth the language model trained from manually labeled data. Hence, ESLAM seamlessly integrate both manually
- labeled data and noisy labeled data into a probabilistic framework. The large amount of noisy emoticon data gives ESLAM have the power to deal with misspelled words, slang, modal particles, acronyms, and the unforseen test words, which cannot be easily handled by fully supervised methods.
- Besides the polarity classification, ESLAM can also be used for subjectivity classification which cannot be handled by most existing distantly supervised methods.
- Rather than crawling a large amount of noisy data to local disks which is a typical choice by existing distantly supervised methods, we propose an efficient and convenient way to directly estimate the word probabilities from Twitter API without downloading any tweet. This is very promising because it is very expensive in terms of time and storage to download and process large amount of tweets.

Language models (LM) can be either probabilistic or no probabilistic. In this paper, we refer to probabilistic language models which are widely used in information retrieval and natural language processing (Ponte and Croft 1998; Zhai and Lafferty 2004; Manning, Raghavan, and Schutze 2009). A LM assign a probability to a sequence of words. In information retrieval, first we estimate a LM for each document, then we can compute a likelihood measuring how likely a query is generated by each document LM and rank the documents with respect to the likelihoods.

TSA is actually a classification problem. To adapt LM for TSA, we concatenate all the tweets from the same class to form one synthetic document. Hence, for the polarity classification problem, one document is constructed from positive training tweets, and the other document is constructed from negative training tweets. Then we learn two LMs, one for positive class and the other for negative class. The LM learning procedure for subjectivity classification is similar. During the test phase, we treat each test tweet as a query, and then we can use the likelihoods to rank the classes. The class with the highest likelihood will be chosen as the label of the test tweet.

CLASSIFICATION EXPERIMENTS:

To evaluate T-CLASS's ability to classify entity mentions in context, we annotated the 2,400 tweets with 10 types which are both popular on Twitter, and have good coverage in Freebase: PERSON, GEO-LOCATION, COMPANY, PRODUCT, FACILITY, TV-SHOW, MOVIE, SPORTSTEAM, BAND, and OTHER. Note that these type annotations are only used for evaluation purposes, and not used during training T-CLASS, which relies only on distant supervision. In some cases, we combine multiple Freebase types to create a dictionary of entities
Representing a single type (for example the COMPANY dictionary contains Freebase types /business/ consumer company and /business/brand). Because our approach does not rely on any manually labeled examples, it is straightforward to extend it for a different sets of types based on the needs of downstream applications.

In contrast to previous work, we have demonstrated the utility of features based on Twitter specific POS taggers and Shallow Parsers in segmenting Named Entities. In addition we take a distantly supervised approach to Named Entity Classification which exploits large dictionaries of entities gathered from Freebase, requires no manually annotated data, and as a result is able to handle a larger number of types than previous work. Although we found manually annotated data to be very beneficial for named entity segmentation, we were motivated
to explore approaches that don't rely on manual labels for classification due to Twitter's wide range of named entity types. Additionally, unlike previous work on NER in informal text, our approach allows the sharing of information across an entity's mentions which is quite beneficial due to Twitter's terse nature.

We identified named entity classification as a particularly challenging task on Twitter. Due to their terse nature, tweets often lack enough contexts to identify the types of the entities they contain. In addition, a plethora of distinctive named entity types are present, necessitating large amounts of training data. To address both these issues we have presented and evaluated a distantly supervised approach based on LabeledLDA, which obtains a 25% increase in F1 score over the co-training approach to Named Entity Classification suggested by Collins and Singer (1999) when applied to Twitter.

SYSTEM ANALYSIS

EXISTING SYSTEM:

Twitter is categorized as a microblogging service. Microblogging is a form of blogging that enables users to send brief text updates or micro media such as photographs or audio clips. Microblogging services other than Twitter include Tumblr, Plurk, Jaiku, identi.ca, and others. Users can know how other users are doing and often what they are thinking about now, users repeatedly return to the site and check to see what other people are doing

PROPOSED SYSTEM:

This paper presents an investigation of the real-time nature of Twitter that is designed to ascertain whether we can extract valid information from it. We propose an event notification system that monitors tweets and delivers notification promptly using knowledge from the investigation. In this research, we take three steps: first, we crawl numerous tweets related to target events; second, we propose probabilistic models to extract events from those tweets and estimate locations of events; finally, we developed an earthquake reporting system that extracts earthquakes from Twitter and sends a message to registered users.

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the

proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SOFTWARE ENVIRONMENT

TENSORFLOW:

Introduction to TensorFlow

Introduction to TensorFlow TensorFlow is a multipurpose open source so2ware library for numerical computation using data flow graphs. It has been designed with deep learning in mind but it is applicable to a much wider range of problems. In this tutorial I will cover the very basics of TensorFlow not going much into deep learning at all. TensorFlow can be used from many programming languages. I will give simple examples, such as linear regression, showing the python API as well as the recent interface to R.

ANACONDA:

Anaconda is the installation program used by Fedora, Red Hat Enterprise Linux and some other distributions. During installation, a target computer's hardware is identified and configured and the appropriate file systems for the system's architecture are created. Finally, anaconda allows the user to install the operating system software on the target computer. Anaconda can also upgrade existing installations of earlier versions of the same distribution. After the installation is complete, you can reboot into your installed system and continue doing customization using the initial setup program. Anaconda is a fairly sophisticated installer. It supports installation from local and remote sources such as CDs and DVDs, images stored on a hard drive, NFS, HTTP, and FTP. Installation can be scripted with kickstart to provide a fully unattended installation that can be duplicated on scores of machines. It can also be run over VNC on headless machines. A variety of advanced storage devices including LVM, RAID, iSCSI, and multipath are supported from the

partitioning program. Anaconda provides advanced debugging features such as remote logging, access to the python interactive debugger, and remote saving of exception dumps.

SYSTEM IMPLEMENTATION

AUTOMATED COVID-19 DIAGNOSIS FROM LUNG IMAGES

Since the beginning of 2020, several deep neural models have proven their effectiveness in the diagnosis of COVID-19 infection from either lung CT or CXR images. Although the proposed deep neural networks were developed upon completely different architectures, and exploit different training losses and optimization algorithms, their common trait is the ``Active, Incremental Learning'' approach used for learning which is especially needed when the available datasets are limited in size and only small numbers of new cases can be acquired incrementally. Thanks to the existence of large open datasets containing either lung CT or CXR images from patients with various diseases other than COVID-19 (e.g., lung cancer, pneumonia, pleural effusion, and others), the problem of COVID-19 diagnosis is commonly addressed by training well-known existing deep neural networks such as ResNet Inception-Net or VGG on the existing, large datasets. In this way, the network is first trained on a similar task, such as lung cancer or pneumonia diagnosis. Next, the knowledge of the pre trained network is ``incremented'' by applying a training phase where an augmented COVID-dataset is used.

RISK PREDICTION MODELS FOR COVID-19 PATIENTS

A recent exhaustive survey of the literature on multivariate models and scoring systems for predicting COVID-19 related outcomes revealed 107 studies describing 145 prediction models. Of these, four models aim to identify people at risk in the general population; 60 exploit medical imaging for diagnosing COVID-19 in patients with suspected infection; nine models diagnose disease severity; and 50 propose prognostic models for predicting mortality risk, progression to severe disease, intensive care unit admission, ventilation, intubation, or length of hospital stay. Besides being a precise report of all the available state-ofthe- art works (up to May 5th, 2020) for COVID-19-related predictions based on patient data, the method proposed in is

CHEST X-RAY ANALYSIS AND AUTOMATED PROCESSING

The Fleischner Society presented three different scenarios and an algorithm for recommending chest imaging that includes CT and/or CXR to direct patient management during the COVID-19 pandemic. Ultimately, the choice of imaging modality is left to the judgment of clinical teams at the point of care, accounting for the differing attributes of CXR and CT, local resources, and expertise [2]. Though CXR shows clear patterns, distinguishable from those of pneumonia [108], when COVID-19 infection becomes serious, it is insensitive in mild or early infection stages [108]. In contrast, lung CT has greater sensitivity for early pneumonic changes, but this advantage is partially diminished by the huge burden placed on radiology departments in terms of staff commitment, CT room workow, and disinfection procedures. Therefore, many Italian hospitals decided to employ CXR as a rst-line triage tool.

MISSING DATA IMPUTATION

At the state of the art, several imputation models for MCAR methods have been presented that can deal with ``complex'' data . Among such methods, we experimented both Multiple Imputations by Chained Equations (MICE using either predictive mean matching (micePMM) or Random Forest classifiers (miceRF) as the base imputation model, and missForest, which also exploits RFs. More precisely, MI techniques are an effective strategy that exploits randomness for producing unbiased estimates, with a reduced dependency on the normality assumption MIs are mainly used for estimating the linear or logistic regression coefcients that link predictor variables to a response variable. In this case, given a dataset (with MCAR or MAR values) and an imputation model containing some randomness, m imputed datasets are drawn, and subsequently processed separately but identically by the chosen estimator.

TYPES OF TESTS:

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box . you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

SYSTEM DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

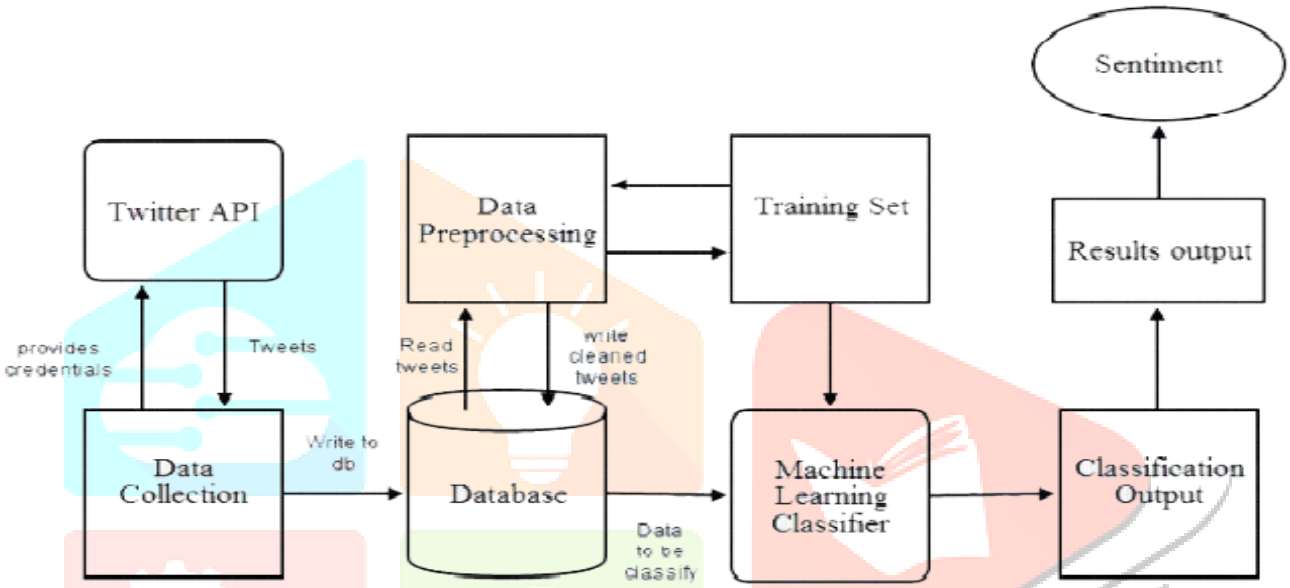
2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

DATA FLOW DIAGRAM



CONCLUSION

There are different Symbolic and Machine Learning techniques to identify sentiments from text. Machine Learning techniques are simpler and efficient than Symbolic techniques. These techniques can be applied for twitter sentiment analysis. There are certain issues while dealing with identifying emotional keyword from tweets having multiple keywords. It is also difficult to handle misspellings and slang words. To deal with these issues, an efficient feature vector is created by doing feature extraction in two steps after proper preprocessing. In the first step, twitter specific features are extracted and added to the feature vector. After that, these features are removed from tweets and again feature extraction is done as if it is done on normal text. These features are also added to the feature vector. Classification accuracy of the feature vector is tested using different classifiers like Nave Bayes, SVM, Maximum Entropy and Ensemble classifiers. All these classifiers have almost similar accuracy for the new feature vector. This feature vector performs well for electronic products domain.

SCREENSHOT

```
%matplotlib inline
```

```
Requirement already satisfied: wordcloud in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (1.6.0)
Requirement already satisfied: matplotlib in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (from wordcloud)
(2.2.2)
Requirement already satisfied: numpy>=1.6.1 in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (from wordcloud)
(1.16.5)
Requirement already satisfied: pillow in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (from wordcloud) (7.0.
0)
Requirement already satisfied: cycler>=0.10 in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (from matplotlib
->wordcloud) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\immatech\anaconda3\envs\tensorflow\lib\si
te-packages (from matplotlib->wordcloud) (2.4.2)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (from ma
tplotlib->wordcloud) (2.8.0)
Requirement already satisfied: pytz in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (from matplotlib->wordcl
oud) (2019.3)
Requirement already satisfied: six>=1.10 in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (from matplotlib->w
ordcloud) (1.12.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\immatech\anaconda3\envs\tensorflow\lib\site-packages (from matpl
otlib->wordcloud) (1.1.0)
```

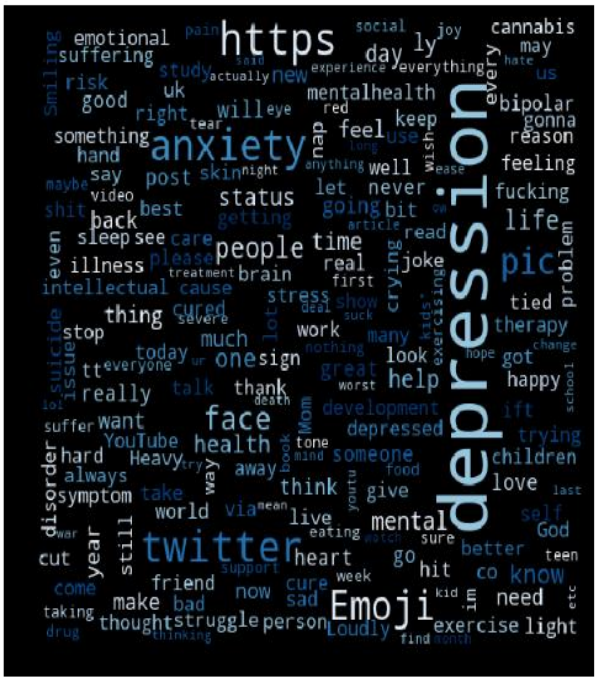
Out[2]:

	Unnamed: 0	message	label
0	106	just had a real good moment. i missssssssss hi...	0
1	217	is reading manga http://plurk.com/p/mzp1e	0
2	220	@comeagainjen http://twitpic.com/2y2lx - http:...	0
3	288	@lapcat Need to send 'em to my accountant tomo...	0
4	540	ADD ME ON MYSPACE!!! myspace.com/LookThunder	0
5	624	so sleepy. good times tonight though	0
6	701	@SilkCharm re: #nbn as someone already said, d...	0
7	808	23 or 24ï¿½C possible today. Nice	0
8	1193	nite twitterville workout in the am -ciao	0
9	1324	@daNanner Night, darlin'! Sweet dreams to you	0
10	1332	Good morning everybody!	0
11	1368	Finally! I just created my WordPress Blog. The...	0
12	1578	kisha they cnt get over u til they get out frm...	0
13	1595	@nicolerichie Yes i remember that band, It was...	0
14	1861	I really love reflections and shadows	0
15	1889	@blueaero ooo it's fantasy? i like fantasy no...	0
16	1899	@rokchic28 no probs, I sell nothing other than...	0
17	1919	@shipovalov "NOKLA connecting people&quot...	0
18	1992	Once again stayed up to late and have to start...	0
19	2097	@Kal_Penn I just read about your new job, CONG...	0

```
In [4]: tweets['label'].value_counts()

Out[4]: 0    8000
        1    2314
        Name: label, dtype: int64

In [5]: tweets.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 10314 entries, 0 to 10313
        Data columns (total 2 columns):
        message    10314 non-null object
        label      10314 non-null int64
        dtypes: int64(1), object(1)
        memory usage: 161.3+ KB
```

Splitting the Data in Training and Testing Sets

```
plt.show()
```



```
Precision:  0.9545454545454546
Recall:  0.40384615384615385
F-score:  0.5675675675675677
Accuracy:  0.8525345622119815
```

```
In [18]: sc_bow = TweetClassifier(trainData, 'bow')
         sc_bow.train()
         preds_bow = sc_bow.predict(testData['message'])
         metrics(testData['label'], preds_bow)
```

```
Precision:  0.9230769230769231
Recall:  0.23076923076923078
F-score:  0.36923076923076925
Accuracy:  0.8110599078341014
```

Predictions with TF-IDF

Depressive Tweets

```
In [19]: pm = process_message('Lately I have been feeling unsure of myself as a person & an artist')
         sc_tf_idf.classify(pm)
Out[19]: False

In [20]: pm = process_message('Extreme sadness, lack of energy, hopelessness')
         sc_tf_idf.classify(pm)
Out[20]: True

In [21]: pm = process_message('Hi hello depression and anxiety are the worst')
         sc_tf_idf.classify(pm)
Out[21]: True

In [22]: pm = process_message('I am officially done with @kanyewest')
         sc_tf_idf.classify(pm)
Out[22]: True
```

Predictions with Bag-of-Words (BOW)

Depressive tweets

```
In [29]: pm = process_message('Hi hello depression and anxiety are the worst')
         sc_bow.classify(pm)
Out[29]: True

In [30]: pm = process_message('My depression will not let me work out')
         sc_bow.classify(pm)
Out[30]: True

In [31]: pm = process_message('Good...')
         sc_bow.classify(pm)
Out[31]: False
```

Positive Tweets

```
In [32]: pm = process_message('Loving how me and my lovely partner is talking about what we want.')
         sc_bow.classify(pm)

Out[32]: False

In [33]: pm = process_message('Very rewarding when a patient hugs you and tells you they feel great after changing the diet and daily hab
         sc_bow.classify(pm)

Out[33]: False

In [34]: pm = process_message('Happy Thursday everyone. Thought today was Wednesday so super happy tomorrow is Friday yayyyyy')
         sc_bow.classify(pm)

Out[34]: False
```

REFERENCES

[1] C. Huang et al., ``Clinical features of patients infected with 2019 novel coronavirus inWuhan, China," Lancet, vol. 395, no. 10223, pp. 497506, 2020.

[2] Z. Huang, W. Dong, L. Ji, and H. Duan, ``Outcome prediction in clinical treatment rocesses," J. Med. Syst., vol. 40, no. 1, p. 8, Jan. 2016, doi: 10.1007/s10916-015-0380-6.

[3] J. Gliozzo, P. Perlasca, M. Mesiti, E. Casiraghi, V. Vallacchi, E. Vergani, M. Frasca, G. Grossi, A. Petrini, M. Re, A. Paccanaro, and G. Valentini, ``Network modeling of patients' biomolecular proles for clinical phenotype/ outcome prediction," Sci. Rep., vol. 10, no. 1, Dec. 2020, Art. no. 3612, doi: 10.1038/s41598-020-60235-8.

[4] B. R. Barricelli, E. Casiraghi, J. Gliozzo, A. Petrini, and S. Valtolina, ``Human digital twin for tness management," IEEE Access, vol. 8, pp. 266326664, 2020.

[5] P. Ramya, S. G. Balakrishnan, M. Kannan, "Recommendation system to improve students' performance using machine learning" in IOP: Materials Science and Engineering, ISSN: 872 (2020) 012038, DOI:10.1088/1757-899X/872/1/012038, during 09 - 10, April 2020.

[6] Q. Pham, D. C. Nguyen, T. Huynh-The, W. Hwang, and P. N. Pathirana, ``Articial intelligence (AI) and big data for coronavirus (COVID-19) pandemic: A survey on the state-of-the-arts," IEEE Access, vol. 8, pp. 130830820, 2020.

[7] S. Fong, G. Li, N. Dey, R. G. Crespo, and E. Herrera-Viedma, ``Composite Monte Carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction," Appl. Soft Comput., vol. 93, pp. 121, Apr. 2020.

[8] A. Joshi, N. Dey, and K. Santosh, Intelligent Systems and Methods to Combat COVID-19 (Springer Briefs in Computational Intelligence). Singapore: Springer, 2020.

[9] R. Vaishya, M. Javaid, I. H. Khan, and A. Haleem, ``Articial intelligence (AI) applications for COVID-19 pandemic," Diabetes Metabolic Syndrome, Clin. Res. Rev., vol. 14, no. 4, pp. 33339, Jul. 2020, doi: 10.1016/j.dsx.2020.04.012.

[10] E. Neri, V. Miele, F. Coppola, and R. Grassi, ``Use of CT and artificial intelligence in suspected or COVID-19 positive patients: Statement of the Italian society of medical and interventional radiology," La Radiol. Med., vol. 125, no. 5, pp. 505508, May 2020.

[11] A. Wong, Z. Q. Lin, L. Wang, A. G. Chung, B. Shen, A. Abbasi, M. Hoshmand-Kochi, and T. Q. Duong, ``COVIDNet-S: Towards computer-aided severity assessment via training and validation of deep neural networks for geographic extent and opacity extent scoring of chest X-rays for SARS-CoV-2 lung disease severity," 2020, arXiv:2005.12855. [Online]. Available: http://arxiv.org/abs/2005.12855

[12] S. Ahuja, B. Panigrahi, N. Dey, T. Gandhi, and V. Rajinikanth, ``Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices," Techrxiv, 2020, doi: 10.36227/techrxiv.12334265.v2.

[13] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang,Q. Song, K. Cao, D. Liu, G.Wang, Q. Xu, X. Fang, S. Zhang, J. Xia, and J. Xia, ``Articial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT," Radiology, vol. 296, Mar. 2020,

Art. no. 200905.

[14] L. Wang and A. Wong, ``COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," 2020, arXiv:2003.09871. [Online]. Available: http://arxiv.org/abs/2003.09871

[15] L. Wynants et al., ``Prediction models for diagnosis and prognosis of COVID-19: Systematic review and critical appraisal," BMJ, vol. 369, no. 369, 2020. [Online]. Available: https://www.bmj.com/content/369/bmj.m1328