IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-8, Aug 2019 E-ISSN: 2347-2693

Twitter Sentiment Analysis using XGBoost and Logistic Regression: A Hybrid Approach

Ashwini M Joshi[1*], Sameer Prabhune[2]

[1,2]Department of CSE, SGBAU, Amaravati, India

Corresponding Author: ashwinimjoshi@pes.edu

DOI: https://doi.org/10.26438/ijcse/v7i8.356360 | Available online at: www.ijcseonline.org

Accepted: 28/Aug/2019, Published: 31/Aug/2019

***Abstract*—**World Wide Web is the largest source of information and huge information is available on the net. It is the growing tendency in users to express their opinions or thoughts using public opinion sites. Analysing all these opinions manually becomes challenging task so if we can develop the automated system to analyse what people want to say about any product, political party or any other thing it would be of great help. In this work we are trying to make readers life easier by providing the polarity of the reviews from user in automated way with better accuracy.
The hybrid model is built using XGBoost and Logistic Regression classifiers and the performance of the hybrid model is compared to both the static models. As per expectation the hybrid model is performing better.

***Keywords*—** XGBoost, Logistic Regression, Hybrid Model, Sentiment Analysis, Opinion Mining

I. INTRODUCTION

Sentiment analysis, as well known as opinion mining, is the area of research and study which analyses what people want to express from their reviews? What is their opinion, sentiment, attitude, emotion towards particular thing? [1]. This analysis can be done for people opinion in various areas like products, service, political party, individual or an organization, any particular event and many more. There are many names for opinion mining like opinion extraction, subjectivity analysis, polarity finding, review mining etc.

The applications in this domain are not only limited to products or hotels but also spread to healthcare and finance too. Sentiment analysis has been investigated mainly at three levels:

- Entire document at a time.
- Sentence by sentence.
- Entity by entity.

Sentiment analysis touches all the angles and aspects of NLP as it is a NLP problem. E.g. handling negations, disambiguating words with its senses, resolution of co-reference etc. It might be difficult in practical implementation as these problems are not yet solved completely. The scope of sentiment analysis problem is fixed. It need to understand only the polarity of the sentence or document as positive, negative or neutral and it need not fully understand the semantics of each word in the sentence. A Machine Learning approach is used by many researchers of sentiment identification. In this work, we have used XG Boost and Logistic Regression algorithms for sentence polarity identification and then developed the hybrid model by arranging these two in cascaded fashion and feeding the misclassified data of first model to second in order to improve the performance. Using XGBoost for sentiment identification is an uncommon idea as compared to other Machine Learning techniques. As XGBoost is designed for speed and performance it was selected as one of the method in our approach. As per our observation, this unique combination of XGBoost and Logistic Regression will help the users for better decision making and for developing better recommendation systems.

Some related work is reviewed in section II where the same methodologies used by various authors are analyzed and our innovative idea is proposed. Section III speaks about data preparation which includes preprocessing and feature extraction. The methodology and implementation of XG Boost and Logistic Regression algorithms is explained and proved with results and screenshots of execution in section IV. The idea of combining these two methodologies is suggested in section V whereas section VI gives results and comparison chart of individual methods with the hybrid model. Finally we provide conclusion of our work as well as the future scope which will tell the direction of our ongoing research.

The ideas suggested and the methodologies implemented are supported by the screenshots of code execution.

II. RELATED WORK

Sentiment analysis and opinion mining task is carried out using various ML techniques by many researchers, but no much work has been carried out using XGBoost and Logistic regression. As per the case study by Vasileios Anthanasiou and Manolis Maragoudak, a novel framework using gradient boosting for the languages is suggested where NLP resources are not available and if available, it is not sufficient [2]. This study is only limited to Greek language. This paper is the extended version of the paper published in 2016.

The sentiment analysis using Logistic Regression is done in [4]. In the proposed work, Logistic regression with unigram feature vector is used as classification method. The k fold cross validation technique is used for accuracy calculation. Tweet subjectivity is used for choosing specific training sample. The polarity of frequently used words, is calculated using word score heuristics. This can increase the speed of classification for sentiments with standard approaches of Machine Learning.

After studying various papers and books we found that even though XGBoost has better capabilities in terms of speed and performance it is not commonly used for sentiment analysis. Thus we decided to go with two uncommon classifiers, XGBoost and Logistic Regression and the combination of these two as hybrid model.

III. DATA PREPARATION

III.I Dataset

The twitter scraped data containing people reviews about US airlines is available for free download and is considered for this work. The data is initially labelled with positive, Negative and Neutral with the reason of negative review is written. Tweets are categorized into 'positive', 'negative' and 'neutral' which are the target values. The dataset consists of 14640 reviews where, 9178 are negative, 3099 are neutral and 2363 are positive tweets.

III.II Pre-processing

Only alphabets are considered from text column and rest of the characters are removed which doesn't contribute for feature extraction. Stop words are removed. As scikit learn handles only real numbers, categorical target is converted into numerical data using label encoder and this pre-processed data is used for further processing.

III.III Feature Extraction

The process of identifying features for a current learning or classification instance is feature extraction.

1. TF Score (Term Frequency)

By considering the collection of words in the document as vocabulary, how many times a particular word has occurred in that vocabulary is term frequency. For sentiment recognition, the word with highest frequency may not be useful as compared to word which is less frequent. Thus only alone term frequency may not be successful and sufficient feature for sentiment detection [11].

$$\text{TF (t)} = \frac{\text{(Number of times term t appears in a document)}}{\text{(Total number of terms in the document)}}$$

2. IDF Score (Inverse Document Frequency)

For weighing and ranking, we use frequency of the tern on the vocabulary. More meaningful information may get conveyed by the rare term than the frequent term. Inverse document frequency is the count of number of documents in which a particular word occurs. Thus the rare words are given more positive weights than the frequent words [11].

$$\text{IDF (t)} = log_e\left(\frac{\text{Total number of documents}}{\text{Number of documents with term t in it}}\right)$$

Combining these two terminologies (TF and IDF), we calculated these scores using log scale and refer it as tf-idf score. It is given by.

$$W_t,d = (1+\log (1+ tf_t, d)).\log_{10} Ndf_t$$

All values of n such that min_n <= n <= max_n will be used. The lower and upper boundary of the range of n-values for different n-grams to be extracted.

III.IV Word Embedding

In NLP we believe that similar words are having similar vector representations [12]. Thus the type of representation is nothing but word embedding. Standard word embedding do not support the embedding of non-English words. So we removed such words during pre-processing the text to train model, as this might add wrong meaning to the text.

IV.METHODOLOGY AND IMPLEMENTATION

IV.I XGBoost

XGBoost stands for eXtreme Gradient Boosting. The main goal of its development was improvement in model performance and speed of computation. It is an implementation of Gradient Boosting Machine which enhances the computing power for boosted trees algorithms. All hardware resources and every bit of memory is fully utilised here [13]. It also proposes many advanced features for computing environment, model tuning and enhancement of algorithm.

Fig.1 depicts the overall features and working characteristics of XGBoost algorithm. [13]. It majorly tells where the algorithm is used for.

Fig. 1. XGBoost

The three main forms of gradient boosting (Gradient Boosting (GB), Stochastic GB and Regularized GB) can be performed by XGBoost. Also it is capable of supporting fine tuning and addition of regularization parameters. The model is trained on tf-idf features. The terms are weighed by how less frequent they appear as uncommon words are more suggestive for the content. The step by step implementation of this model is as follows:

Step1: Pre-processing
Step2: Feature extraction using count vectorization, word level tf-idf and n-gram level tf-idf. Convert the text into vectors
Step3: Fit the training dataset, it learns the model
Step4: XGBoost classifier is used to predict the values
Once the XGBoost implementation is done, the accuracy calculation of this model is done with all the three vectorization techniques.

```
# Extreme Gradient Boosting on Counter Vectors
accuracy = train_model( xgboost. XGBClassifier (), xtrain_count.tocsc(), train_y, xvalid_count.tocsc())
print ("Xgb, Count Vectors: ", accuracy)

# Extreme Gradient Boosting on Word Level TF IDF Vectors
accuracy = train_model( xgboost. XGBClassifier (), xtrain_tfidf.tocsc(), train_y, xvalid_tfidf.tocsc())
print ("Xgb, WordLevel TF-IDF: ", accuracy)

# Extreme Gradient Boosting on Character Level TF IDF Vectors
accuracy = train_model( xgboost. XGBClassifier (), xtrain_tfidf_ngram_chars.tocsc(), train_y, xvalid_tfidf_tocsc())
print ( "Xgb, WordLevel TF-IDF: ", accuracy)

Xgb, Count Vectors: 0.7173934426229508
Xgb, WordLevel TF-IDF: 0.718306019289617
Xgb, CharLevel Vectors: 0.7426229508196721
```

Fig. 2. Accuracy calculation of XGBoost with three methods

IV.II Logistic Regression

For the prediction of the probability of a categorical dependent variable Logistic Regression, a Machine Learning classification algorithm is preferred. It is the most important and popularly used model in generalized linear models. Logistic Regression preserves the marginal probabilities of the training data [3]. The coefficients of the model also provide some hint of the relative importance of each input variable. When dependant variable is categorical, Logistic Regression is used. The main reason we selected Logistic Regression as our second algorithm is not only its efficiency but also its ability to work with less computational resources. The step by step implementation is as follows.

Step1: Pre-processing
Step2: Feature extraction using count vectorization, word level tf-idf and n-gram level tf-idf. Convert the text into vectors
Step3: Fit the training dataset, it learns the model
Step 4: To predict the probability that observation i has outcome k, a linear predictor function f (k,i) is used by multinomial logistic regression. The function is of the following form.
$f(k, i) = \beta 0 , i + \beta 1 , k x 1 , i + \beta 2 , k x 2 , i + \cdots + \beta M , k x M , i$,where $\beta m , k$ is a regression coefficient associated with the M^{th} explanatory variable and the k^{th} outcome.
Even though Logistic Regression has high reliance on proper representation data it worked very well on our dataset as the pre-processing is properly done. The accuracy calculation with all the three vectorization methods is shown in fig. 3.

```
# Liner Classifier on Counter Vectors
accuracy = train_model( linear_model. LogisticRegression(), xtrain_count, train_y, xvalid_count)
print ("LR, Count Vectors: ", accuracy)

# Liner Classifier on Word Level TF IDF Vectors
accuracy = train_model( linear_model. LogisticRegression(), xtrain_tfidf, train_y, xvalid_tfidf)
print ("LR, Word Level TF-IDF: ", accuracy)

# Liner Classifier on Ngram Level TF IDF Vectors
accuracy = train_model( linear_model. LogisticRegression(), xtrain_tfidf_ngram, train_y, xvalid_tfidf_ngram)
print ("LR, Count Vectors: ", accuracy)

# Liner Classifier on Character Level TF IDF Vectors
accuracy = train_model( linear_model. LogisticRegression(), xtrain_tfidf_ngram_chars, train_y, xvalid_tfidf_ngram_chars)
print ("LR, Count Vectors: ", accuracy)

/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbgfs' in 0.22
FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22
FutureWarning)

LR, Count Vectors: 0.7918032786885246
LR, WordLevel TF-IDF: 0.783879781420765
LR, N-Gram Vectors: 0.7008196721311475
LR, CharLevel Vectors: 0.7773224043715847
```

Fig. 3. Logistic Regression Model

V. HYBRID MODEL

Any method which is used by user for decision making should provide high accuracy and understanding of the decisions [10]. In this work we investigate two ML methods to solve the classification problem. The methods are combined and used to develop the hybrid model for improving the accuracy. The collaborative structure is developed in such a way that, the misclassified data of first model is fed to second for classification. The second model has higher performance than first when compared individually. Thus the instances which are wrongly classified by first will be again classified by second and thus the accuracy will be improved. With the correct classification the decision making will be better. The flow of actions in the Hybrid model is as shown in following fig. 4

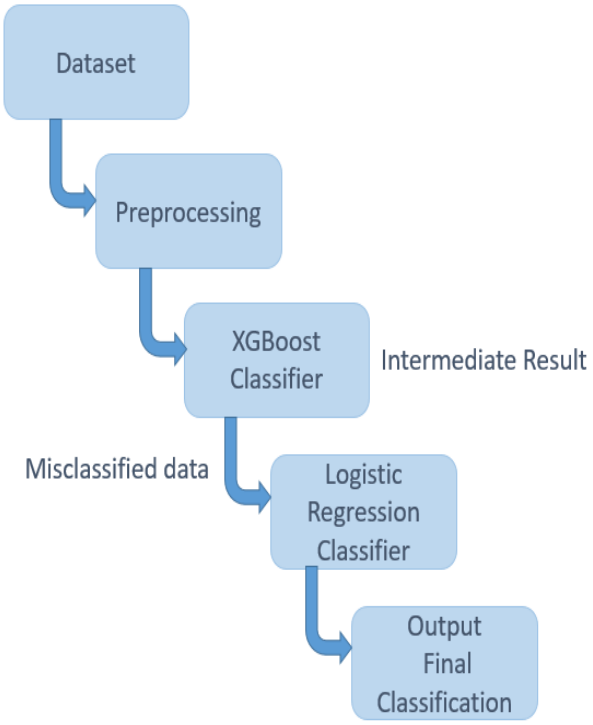

Fig. 4: Structure of a Hybrid model

In this work XGBoost and Logistic Regression methods are implemented and after comparing and analysing the performance of both the algorithms, we felt that if we can combine the capabilities of both the algorithms the performance of the resultant model will be better than the static models. As the logistic Regression algorithm is easy to regularize and outputs well calibrated predicted probabilities, we decided to combine it with XGBoost which itself is an ensemble algorithm. By keeping this idea in mind we fed the misclassified data from XGBoost to Logistic regression and found the improvement in accuracy. The step by step implementation of XGBoost-Logistic regression Hybrid model is given below.

Step1: Pre-processing
Step2: feature extraction using count vectorization, word level tf-idf and n-gram level tf-idf. Convert the text into vectors.
Step3: Fit the training dataset on XGBoost classifier and logistic regression: it learns the model
Step4: Predict the values on classifiers.
Step5: If the predicted values from XGBoost classifier is wrong then predicted values from logistic regression are considered.

The Hybrid model used only word level TF-IDF for both the algorithms. The accuracy is calculated for the model is shown in fig.5.

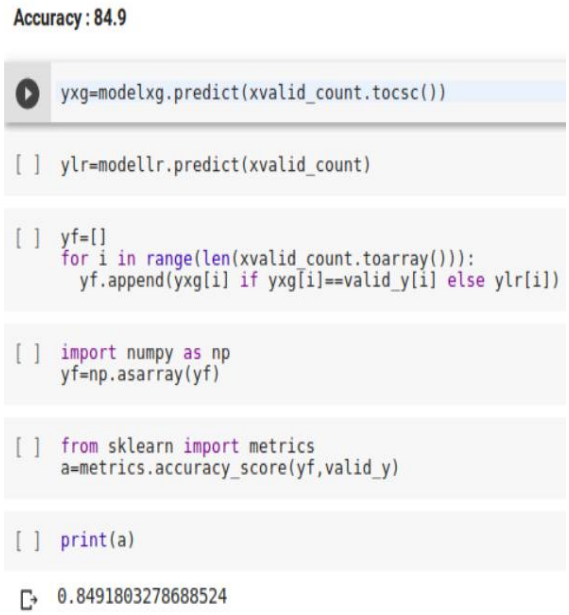

Fig. 5: Convolutional neural networks

VI.RESULTS AND CONCLUSION

The major role of Opinion mining and sentiment analysis is to guide people in making decisions by analysing people opinions. This opinion can be feedback, experience or just a suggestion. As per our work user decision can have support of our hybrid model. The following table gives the comparison of accuracies for the implementation of Static and Hybrid algorithms. The individual models are tested on all three vectorization methods and the hybrid for word level TF-IDF.

Table1: Comparison of accuracies with various implementation methods

Algorithm/ Model	**Implementation Details**	**Accuracy (%)**
XGBoost	Count Vectors	71.6 %
	Word Level Tf-Idf	71.8 %
	Character Level Vectors	74.2 %
Logistic Regression	Count Vectors	79.1 %
	Word Level Tf-Idf	78.3 %
	Character Level Vectors	77.7 %
Hybrid Model	Word Level Tf-Idf	84.9 %

As per the table above and with reference to fig5, we can conclude that combining the two techniques XGBoost and Logistic Regression is a feasible idea in terms of sentiment identification of public reviews from various public opinion sites. The hybrid approach suggested in the paper is certainly improving the accuracy with TF-IDF implementation. This model will help the decision makers to make good quality decision and recommendation.

FUTURE WORK

In future we would like to use our hybrid approach for various datasets and we can find the optimum data where our hybrid model performs much better. Also we would like to include real-time data for the analysis purpose so that various applications can be built for better decision making.

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to Amaravati University Research Cell and PES University for providing the research facilities and congenial environment for carrying out this work. We are also grateful to Dr. Shylaja S.S, Chairperson, Dept. of CSE, PES University, for motivating us and constantly supporting us.

REFERENCES

[1] "Sentiment Analysis and Opinion Mining", Bing Liu., Morgan & Claypool Publishers, May 2012
[2] "A Novel, Gradient Boosting Framework for Sentiment Analysis in Languages where NLP Resources Are Not Plentiful: A Case Study for Modern Greek", Vasileios Athanasiou and Manolis Maragoudakis , Artificial Intelligence Laboratory, University of the Aegean, 2017
[3] "Speech and Language Processing.", Daniel Jurafsky & James H. Martin. Draft of August 24, 2015
[4] "Sentiment Analysis using Logistic Regression and Effective Word Score Heuristic", Abhilasha Tyagi, Naresh Sharma, International Journal of Engineering and Technology, 2018
[5] Tiangi Chan, Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System", ACM digital Library, 2016.
[6] Nikolaos Malandrakis, Abe Kazemzadeh, Alexandros Potamianos, Shrikanth Narayanan, "SAIL: A hybrid approach to sentiment analysis", Second Joint Conference on Lexical and Computational Semantics, Volume 2, 2013.
[7] Ruchika Aggarwal, Latika Gupta, "A Hybrid Approach for Sentiment Analysis using Classification Algorithm", International Journal of Computer Science and Mobile Computing, Vol.6 Issue.6, June-2017
[8] Oscar Romero, Lombart, "Using Machine Learning Techniques for Sentiment Analysis", Final project on computer engineering, school of engineering, Universitat Autonoma de Barcelona 2017.
[9] Smitali Desai, Mayuri A. Mehta, "A Hybrid Classification Algorithm to classify Engineering Students' Problems and Perk", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.6, No.2, March 2016.
[10] "Machine Learning of Hybrid Classification Models for Decision Support", SINTEZA, the use of the internet and development perspectives, 2014.
[11] Dharmendra Sharma1, Suresh Jain, "Evaluation of Stemming and Stop Word Techniques on Text Classification Problem", International Journal of Scientific Research in Computer Science and Engineering, Volume-3, Issue-2 ISSN: 2320-7639, 2015.
[12] Okechukwu Cornelius, Aru Okereke Eze, "Development of an Optimized Intelligent Machine Learning Approach in Forex Trading Using Moving Average indicators." International Journal of Scientific Research, Vol.7, Issue.3, pp.15-21, E-ISSN: 2320, June 2019.
[13] "What is XGBoost Algorithm – Applied Machine Learning", DataFlair Team · Published February 1, 2018 · Updated November 16, 2018.

AUTHORS PROFILE

Ashwini M. Joshi is pursuing her Ph.D. in Computer Science and Eng. From SGBAU Amaravati University, Maharashtra. She holds a B.E Degree from Amaravati University and M.Tech from Bharti Vidyapeeth Pune. She is currently working as Asst. Professor in PES University in Bangalore. She has total 15+ years of experience in various academic institutions from Mumbai, Pune and Bangalore both in teaching and administration. Her research interest is in Natural Language processing in general and Sentiment Analysis and Opinion Mining in particular. Ashwini has attended INDIACOM-2018 at Delhi and presented a paper in IETE Conference at Mumbai in 2018. She holds total 4 publications on her name. She has attended various workshops and Faculty development Programs on Machine Learning, Python Programming and Networking.

Sameer S. Prabhune is currently working as the Principal in Govt. Polytechnic, Khamgaon and holds first rank in MPSC, Maharashtra. He holds B.E, M.E and Ph. D in Computer Science and Eng. He has total 23 years of experience in teaching. Formerly he was working as Head of the Information Science Department in SSGMCE, Shegaon, Maharashtra. He is the registered Ph. D guide in SGBAU, Amaravati in dept. of Computer Science and Eng. He is a life member of ISTE. He has presented papers in more than 15 conferences/Seminars, Journals and Proceedings. He has attended 20+ workshops on various technical topics all over India. Sameer has bagged Best Paper Award at IICT'06 in Database Track. His areas of Specialization are Data Mining, Database Management, Distributed DBMS, Big Data, Temporal Databases and Web Mining.

