Twitter Sentiment Analysis using XG Boost and Random Forest Classification Algorithms: A Hybrid Approach

Ashwini M. Joshi, Research Student, SGBA University, Amravati, India.
ashwinimjoshi@rediffmail.com
Sameer Prabhune, Principal, Govt. Polytechnic, Khamgaon, India. ssprabhune@gmail.com
Nesara B R, Student, PES University, Bengaluru, India. nesararamaswamy@gmail.com

Abstract: **Availability of social media to express our thoughts is creating huge data bank which can be referred by users in analysis and decision making. For any individual it a tedious task to analyze all these opinions, reviews manually if they want to make use of these reviews in decision making or any kind of recommendation. Thus if we can develop an automated system to understand public views, it can lead to better decision making. In this paper we are proposing the Hybrid algorithm which is developed from two classifiers XGBoost and Random Forest. Also the individual performances of both these classifiers are compared with the performance of Hybrid model and finally the better model is suggested for sentiment classification. This work majorly aims at performance comparison of both the algorithms used and then building the Hybrid model for improving the accuracy. This improved model will definitely help the users to understand what people want to express through reviews.**

Keywords — Accuracy, Data Mining, Hybrid Model, Opinion Mining, Random Forest, Sentiment Analysis, XG Boost

I. INTRODUCTION

Sentiment analysis, also called opinion mining, is the research area in which the public opinions are considered and analysis of these opinions is done to understand what people think about a particular thing [1]. As the tendency of people to express thoughts, reviews, experiences, feedback is growing, the public opinion sites are getting overwhelmed data which can be properly analysed so that it will be useful for the recommendation. As huge data is available it is not possible to take care of it manually and thus we need to take help of some technique to automate it. There are various Machine Learning techniques and classifiers which can be used to do this. This job comes under Sentiment analysis where we need to find out the polarity of the review and can be classified as positive, negative and neutral.

As the input is from public opinion sites it is in Natural Language and a lot of pre-processing is required on it. The pre-processed data is then fed to classifiers to get the result. In this work we have used XGBoost and Random Forest Classifiers and then the Hybrid model is developed by combining the capabilities of these two. The only aim in building this Hybrid model is performance improvement. The reason behind selecting these two algorithms is its uniqueness in work. It is not as common in sentiment analysis as other Machine Learning techniques.

Initially people used to put their reviews for a particular product or hotel but now it is spread to all the possible domains including healthcare and finance. The sentiment of people can be investigated majorly at three levels

- Document
- Sentence
- Word

As the input is from people it has many challenges like slangs, abbreviations, mixed languages and many more. In this work only valid English reviews are considered and the public opinion platform is only limited to twitter.

Many Machine Learning algorithms are used in sentiment classification. In this work we used XGBoost as it is mainly designed for accuracy and performance. Another algorithm is Random forest as it is very robust, deals with noise and outliers very efficiently and because of its high accuracy it is preferred in our work. Every algorithm has its own capabilities as well as limitations. Thus if we can combine the capabilities it will definitely be a stronger model than the individual classifiers. By keeping this idea in mind we have developed the hybrid model where the misclassified data of XGBoost is again provided as an input to Random Forest classifier. The sequence in the cascading is purely based on individual performance.

DOI : 10.35291/2454-9150.2019.0433

The related work is analyzed in next section of this paper followed by data preparation and implementation details. Finally results are compared.

We are sure that this contribution of ours will help the readers and users to understand people inclination in more accurate way and can be further used in good selection.

II. Related Work

Sentiment analysis is the research area from early 2000 and many researchers are still working on it. Using Machine Learning algorithms for sentiment classification is very common but based on our literature review we observed that using XGBoost algorithm for sentiment analysis is not as common as other ML algorithms. As per the case study by Vasileios Anthanasiou and Manolis Maragoudak, a novel framework using gradient boosting for the languages 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.

XGBoost's main purpose is to push the extreme of the computation limits of machine to provide a scalable, portable and distributed tree boosting [7]. Authors says, in order to improve their Neural Networks models, they decided to use XGBoost to combine several models they constructed so as to generate a more accurate prediction.

Palak, Apoorva and Neelam has analysed movie reviews using Naïve Bayes, K-Nearest Neighbours and Random Forest algorithms [4]. In this paper authors elaborately stated the applications of sentiment analysis which includes Quality improvement of products or services, Purchasing, decision making, recommender system, marketing, research, flame detection etc. This paper uses 2 datasets of IMDb Movie Reviews and WEKA tool is used for further processing. For validation, 10 fold cross validation is used. Finally the accuracies of all the three techniques are compared.

An improved version of random forest classifier is given in [5]. This algorithm is designed especially for multiple classes and very high dimensional data. A new feature weighing method and tree selection method are adopted which improves classification performance with less error bound. This paper uses out-of-bag, estimate for error bound, test accuracy and F1-measure as performance evaluation methods.

As stated in [6] Naïve Bayes approach gives poor performance in context of twitter messages. The authors says that their classifier gives better performance on SMS data. Here 2 annotated datasets were used and the baseline for future research is created.

Based on our observation we have decided to use XGBoost and Random forest for our study and we are trying to build the unique model by combining the capabilities of these two algorithms which will lead to proper decision making.

III. Data Preparation

A. Dataset

The dataset containing people reviews which are scraped from twitter is considered here for this work. The dataset has initial labels positive, negative and neutral. Also it contains the reason or the experience of people for negative reviews. These tags are the target values for the test set. The dataset consists of 14640 reviews where, 9178 are negative, 3099 are neutral and 2363 are positive tweets.

B. Pre-processing

Pre-processing is the very important step before application of any method on the dataset. If the dataset is pre-processed as per the requirement of methods to be implemented. In this work we are only using text as input and the other characters which are not contributing for feature extraction are discarded. Stop word removal is done to minimize the processing load and time. 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.

C. Feature Extraction

Feature Extraction is a very important processing step in NLP and Machine Learning. The extraction of features which are informative with respect to sentiment analysis task needs to be extracted. Sentiment is a view, feeling, opinion or assessment of a person for some product, event or service. Sentiment analysis is a challenging text mining and NLP problem for automatic extraction, classification and summarization of sentiments. In this work we have used Word Level TF-IDF, Count Vectors and word vectors to find how important the word is for the document or corpus [10].

Count Vectorization involves counting the number of occurrences each word appears in a document. This sets how many features or words we want countvectorizer to count.

Countvectorizer just counts word frequencies with TF-IDF vectorizer. The value increases proportionally to count but is offset by the frequency of the word in the corpus [8, 9]

Word Vectors are simply vectors of numbers that represent the meaning of a word. Here words or phrases from the vocabulary are in NLP mapped to vectors at real numbers.

Word Level TF-IDF

1. TF Score (Term Frequency)

If we consider each document is the glossary of words, the count of every word in that glossary is term frequency. For sentiment detection, the word with highest frequency may not be useful as compared to word which is less frequent.

DOI : 10.35291/2454-9150.2019.0433

Thus alone term frequency cannot be successful feature for sentiment detection.

2. IDF Score (Inverse Document Frequency)

We make use of frequency of the term in collection for weighting and ranking. 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 frequent words should be given lower positive weights and rare ones with higher.

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

$$\mathbf{W_{t},d= (1+log\ (1+ tf_{t},\ d)).log_{10}\ Ndf_{t}}$$

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.

IV. Methodology And Implementation

A. XGBoost

XGBoost stands for eXtreme Gradient Boosting. In order to improve model performance and speed of computation, XGBoost was developed [2]. It improves the resource utilization of hardware resources and maximizes memory utilization too. It is an implementation of Gradient Boosting Machine which enhances the computing power for boosted trees algorithms. It also proposes many advanced features for computing environment, model tuning and algorithm enhancement. Three main forms of Gradient Boosting (GB), Stochastic GB and Regularized GB are getting performed and it is capable of supporting fine tuning and addition of regularization parameters [11].

This model is trained on tf-idf features. The terms are weighed by how less frequent they are as uncommon words are more suggestive for the content.

Table 1 Accuracy calculation of XGBoost with three methods

Algorithm/ Model	Implementation Details	Accuracy (%)
XGBoost	Count Vectors Word Level Tf-Idf Character Level Vectors	71.6 % 71.8 % 74.2 %

The step by step implementation of this model is as follows:

1. Data Preparation.
2. Text to vector conversion by extracting features using count vectorization, word level tf-idf and n-gram level tf-idf.
3. Train the model by fitting the training dataset.
4. Get the class prediction as a result using XGBoost Classifier.

Fig. 1 mentions the accuracies obtained on test data with all the three techniques.

B. Random Forest

A set of decision trees gets created by randomly selected training data by the random forest classifier. The final class of the test object is decided by aggregating the votes from different decision trees. As single decision tree can be more noise prone, this model works with better accuracy as many decision trees are aggregated. It reduces the noise and gives more accurate results. It is considered as one of the robust methods among all machine learning algorithms. The f1-score of this method is better as it has lower classification error. The random subset creation prevents overfitting and thus this method does not suffer from overfitting problem. The main reason is that it takes the average of all the predictions, which cancels out the biases.

Even though Random Forest classifier is complex and requires more training period because of its robustness and ability to deal with missing values and outliers, this method is used in our work. The step by step implementation is as follows.

1. Data Preparation.
2. Text to vector conversion by extracting features using count vectorization, word level tf-idf and n-gram level tf-idf.
3. Train the model by fitting the training dataset.
4. It aggregates the prediction based on different decision tree which is created on subsets of dataset.

The implementation and accuracy calculation is shown in following table.

Fig. 2 Accuracy calculation of Random Forest Classifier

Algorithm/ Model	Implementation Details	Accuracy (%)
Random Forest	Count Vectors Word Level Tf-Idf Character Level Vectors	73.03 % 73.16 % 68.00%

C. Hybrid Model

XGBoost and Random Forest methods were 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. By keeping this idea in mind we fed the misclassified data from XGBoost to Random Forest and found the improvement in accuracy. The individual models accuracies are used to decide the sequence of systems in cascaded fashion. The accuracy obtained from this model is mentioned in Fig. 3. Also the screenshot of the accuracy calculation is given if Fig. 4. The step by step implementation is as follows:

1. Data Preparation.

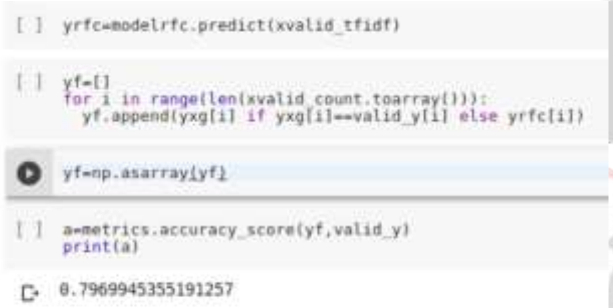
DOI : 10.35291/2454-9150.2019.0433

2. Text to vector conversion by extracting features using count vectorization, word level tf-idf and n-gram level tf-idf.
3. Train the model by fitting the training dataset using XGBoost and Random Forest Classifiers.
4. Get the class predictions.
5. If the predicted values from XGBoost classifier are wrong then predicted values from random forest are considered.

Table 3: Accuracy calculation of Hybrid Model.

Algorithm/ Model	Implementation Details	Accuracy (%)
Hybrid Model	Word Level Tf-Idf	79.6 %

The Word Level TF-IDF implementation of the hybrid model and the respective accuracy is given in the following fig.1



```
[ ]  yrfc=modelrfc.predict(xvalid_tfidf)

[ ]  yf=[]
     for i in range(len(xvalid_count.toarray())):
       yf.append(yxg[i] if yxg[i]==valid_y[i] else yrfc[i])

     yf=np.asarray(yf)

[ ]  a=metrics.accuracy_score(yf,valid_y)
     print(a)

     0.7969945355191257
```

Fig. 1. Hybrid model accuracy calculation with word level tf-idf.

V. Results and Conclusion

In this research XGBoost and Random Forest classifiers were implemented and the accuracies are compared which are mentioned in tables 1 1nd 2 respectively. The basic idea of this work is use both the algorithms in cascaded fashion and compare the accuracy with the individual performances. The accuracy calculation of this hybrid algorithm is shown in fig 1. and depicted in table 3. Thus we can conclude that this Hybrid implementation is a feasible idea in terms of sentiment identification of public reviews from various public opinion sites. If readers of this paper uses the hybrid approach suggested in this paper or this model, it will help them for good quality sentiment analysis and gradually better decision making or better recommendation.

VI. Future Work

In extension to current work, we would like to use combination of more such algorithms for further performance improvement. apply this hybrid algorithm on various datasets and we'll try to find the optimum result for concluding the type of data suitable to this algorithm. Also instead of using only two algorithms

VII. 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.

References

[1] Bing Liu, "Sentiment Analysis and Opinion Mining," Morgan & Claypool Publishers, May 2012.

[2] Vasileios Athanasiou and Manolis Maragoudakis, "A Novel, Gradient Boosting Framework for Sentiment Analysis in Languages where NLP Resources Are Not Plentiful: A Case Study for Modern Greek," Artificial Intelligence Laboratory, University of the Aegean, 2017.

[3] Daniel Jurafsky & James H. Martin, "Speech and Language Processing," Draft of August 24, 2015.

[4] Palak Baid, Apoorva Gupta, Neelam Chaplot "Sentiment Analysis of Movie Reviews using Machine Learning Techniques," International Journal of Computer Applications (0975 –8887)Volume 179 – No.7, December2017.

[5] Baoxun Xu Shenzhen, Xiufeng Guo, Yunming Ye, "An Improved Random Forest Classifier for Text Categorization," Journal of computers, vol. 7, no. 12, December 2012.

[6] Silvio Moreiral, Jo˜ao Filgueiras, Bruno Martins, Francisco Couto, M´ario J. Silva, "REACTION: A naive machine learning approach for sentiment classification", Second Joint Conference on Lexical and Computational Semantics, Volume 2: Seventh International Workshop on Semantic Evaluation (Sem Eval 2013), pages 490–494, Atlanta, Georgia, June 14-15, 2013.c©2013 Association for Computational Linguistics.

[7] Sung-Lin Chan, Xiang zhe Meng, S¨uha Kagan K¨ose, "EPFLMachineLearningCS-433-Project2 Twitter Sentiment Analysis", 2018.

[8] https://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

[9] https://en.wikipedia.org/wiki/Tf-idf.

[10] Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov, "Enriching Word Vectors with Subword Information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

[11] Ashwini M Joshi, Sameer Prabhune "Twitter Sentiment Analysis using XGBoost and Logistic Regression: A Hybrid Approach," International Journal of Computer Sciences and Engineering,Vol.-7, Issue-8, Aug 2019.

DOI : 10.35291/2454-9150.2019.0433