Random Forest: A Hybrid Implementation for Sarcasm Detection in Public Opinion Mining

Ashwini M Joshi, Sameer Prabhune

Abstract: ***Modelling the sentiment with context is one of the most important part in Sentiment analysis. There are various classifiers which helps in detecting and classifying it. Detection of sentiment with consideration of sarcasm would make it more accurate. But detection of sarcasm in people review is a challenging task and it may lead to wrong decision making or classification if not detected. This paper uses Decision Tree and Random forest classifiers and compares the performance of both. Here we consider the random forest as hybrid decision tree classifier. We propose that performance of random forest classifier is better than any other normal decision tree classifier with appropriate reasoning.***

Keywords—Random forest, Decision Tree, Pruning, Diverse, Hybrid.

I. INTRODUCTION

With the advent of luxurious mobile phone and 4G internet speed the number of mobile phone have grown drastically. Social media platforms like Instagram, Twitter and Facebook are so well established that anybody can express their feelings in anyway about anything in the world [11]. In the recent surveys it is observed that Twitter has around 500 million users among which 332 million users are monthly active and 126 million users are active everyday. There are about 500 million tweets every day and 1.6 billion searches [12]. Hence Twitter data is used as one of the major dataset for the experiments related to this work. This huge amount of customer data has become a boon to CRM based industries to understand the feelings of their customers. This is why Sentiment analysis with sarcasm detection has become a major source of attention to NLP experts.

Sarcasm is form of speech through which people may taunt each other. It's a way people express their negative feeling with the help of positive expressions. Sarcasm is a highest form of intelligence which increases the creativity of both who expresses and who receives and understands it [13]. Sentiment analysis being one of the most important natural language processing problem fails to better accuracies because of its misclassification due to sarcastic sentences .In this work we have used Decision Tree and Random Forest classifiers for detecting the sarcasm. Decision tree classifiers are best known for their interpretation of feature splits for a particular data set. They don't consider the randomness of data and fit into only one kind of data set. On the other hand Random forest classifier randomly selects observations and specific features to build multiple Decision Trees and averages/votes for the result. This paper proposes that the use of random forest classifiers are proved to be working well than decision tree classifiers. The implementation is considered with various parameters like unigrams, bigrams and trigrams. The application of this will be helping users for correct classification and better decision making.

The Literature Review with respect to the work done by various researchers in this area is mentioned and analyses in next section. The methodologies and implementation of random forest and decision tree are described in section III. Section IV of this paper proposes a method for sentiment analysis including sarcasm detection. The results are mentioned and discussed in section V and the last section of the paper ends with conclusion and scope of future work.

II. LITERATURE SURVEY

D.V.Nagarjana Devi, Dr.T.V.Rajanikanth and Dr.V.V.S.S.S. Balaram [2] has used various Machine Learning classification methods for the detection of Sarcasm in plain text. The actual or hidden meaning from people opinion can be discovered by identifying the sarcasm in people opinion. As part of pre-processing all Hash tags #sarcasm and #sarcastic were filtered so that it won't influence the classification of their model due to its presence. After feature engineering the classification is done using Naïve Bayes, SVM, Logistic Regression and Decision Trees. After doing the results analysis, authors feels that they are getting best results with Logistic Regression and Decision Tree classifications techniques. Though the only few best working classification methods are analysed in this paper for the detection of sarcasm in the plain text with highest accuracy, some hybrid classification technique/s can be built in future.

Anukarsh G Prasad; Sanjana S, Skanda M Bhat, B S Harish, compares various classification algorithms such as Random Forest, Gradient Boosting, Decision Tree, Adaptive Boosting, Logistic Regression and Gaussian Naïve Bayes to detect sarcasm in tweets from the twitter streaming API [3]. This paper uses a novel idea of introducing emojis and slang dictionary mapping. Here the emoji dictionary is manually created. The paper suggests various improvements in existing classification algorithm for improving the accuracy in detecting whether the tweet is sarcastic or non-sarcastic. This model is tested in real time as well.

Revised Manuscript Received on October 05, 2019.

Ashwini M Joshi, Department of CSE, SGBAU, Amaravati, ashwinimjoshi@redffmial.com

Sameer Prabhune, Department of CSE, SGBAU, Amaravati, ssprabhune@gmail.com

In [4] the authors of this paper used reditt dataset as they wanted to remove the limit of number of words in their analysis. The sarcasm detection of people review which has happened so far has usually a limit of words up to 140. Here due to reditt dataset the public post can be a word, a sentence or even a paragraph. To detect the sarcasm in posts, the sentence patterns are identified with syntactic, semantic and sentiment based features. The features which are considered here are related to punctuation, sentiment, syntactic, semantic and pattern. The sub-patterns of length 4 to 10 are considered in the model which helps the model to classify more accurately. Current method is telling whether the post is sarcastic or not which can be further extended to find the sentiment behind it as whether it is positive or negative.

III. METHODOLOGIES AND IMPLEMENTATION

Any work in NLP starts with the data preparation and then implementation of algorithms on the preprocessed dataset. This procedure is explained in following sections of this paper.

A. Dataset

The dataset used for this work is a standard Kaggle data with the labels sarcastic in it. This dataset contains 1.3 million sarcastic comments from the Internet commentary website Reddit. The dataset was generated by scraping comments from Reddit (not by us) containing the \s (sarcasm) tag. This tag is often used by Redditors to indicate that their comment is in jest and not meant to be taken seriously, and is generally a reliable indicator of sarcastic comment content.

Sarcastic comments: 505405
Non-sarcastic comments: 505368

B. Decision Trees in Sarcasm Detection

Decision trees are one of the most powerful models which are easy to understand, visualise and interpret [9]. Computation of impurity and further splitting based on the most impure feature happens in decision tree classification. They need less pre-processing compared to other classifiers. Decision tree classification is very robust to overfitting.

In case of text classification once a text is fed in, it measures the informativeness of the word and its entropy at each split and classifies into a particular sentiment.

The accuracy and the way decision trees get trained varies for every dataset. In the experiments performed, same dataset was reshuffled for about 6 times and the accuracies were tested on a validation set and it was observed that the results were random. Hence the order in which the text was fed and trained was also quite important. It gives model convoluted decision boundaries with piece wise approximations. [17]

Overfitting being one of the major problem with decision trees, Decision trees fails to both with huge data and less data due to its piece wise approximation logic. In the Experiments performed it was observed that unigram and bigram training is less efficient compared to n gram level of decision tree training.[17]

C. Random Forests in Sarcasm Detection

Random Forest consists of a large number of individual decision trees that operates as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction [15].

As per data science speak, Random forest is, ***"****A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models."* And this is the reason that the random forest model works so well.

Random Forests are the ensemble of randomly sampled n decision tress .It is possible that some trees might go wrong and other trees might perform really well but on an average random forest classifier perform better than decision tree classifiers.

As Random forests are the ensembles of k decision trees and m bootstrap samples. After necessary pre-processing like noise removal, stemming, lemmatization and cleaning, the following steps are followed in order to use it for sarcasm detection in text.

Step 1:

At any given node in the decision tree randomly select p features, such that p<x, where x is total number of features.

Step 2:

Compute the best split point based on a specific metric, say Information gain

$$Information\ gain = entropy(parent) - [weights\ average] * entopy(children)$$

Split the node into 2 child nodes and reduce the total number of features X for the next split.

Step 3:

Repeat the steps 1 to 2 until you reach a maximum depth say 'l' or till X=0

Step 4:

Repeat the steps 1 to 3 for all k trees in the forest. Choosing optimal value of k is also important and can be done only via trial and error method.

Step 5:

At the end you can either choose the output in either of the 2 ways.

1. Vote and find out the best tree and output the same
2. Aggregate and find the average output of all k decision trees.

In this work, the first way is opted.

D. Why Random Forests are better?

After comparing Decision Tree and Random Forest, here are few reasons why Random Forest is better.

1.Decision trees are usually prone to sampling unlike random forests :

Decision trees are robust to outliers. If the training data has slightly different data distribution they fail to produce good results.

They have high tendency to overfit and suffers from sampling errors. In case of random forests k number of such trees are trained with m number of random samples and hence less prone to overfitting.

2.Decision trees follows greedy approach unlike random forests:

In case of decision trees the splitting happens in a local way. Tree splits inorder to reduce the impurity to the maximum possible (greedy approach) that might not always be the best way. It only concentrates to reduce the local impurity level and not the global impurity. It is observed `that reducing the local impurity does not lead to the reduction in global impurity. Hence fails to give better results.

This is not the case with Random forest classifiers. They lead to the ensemble of k uncorrelated trees and votes out the best one hence they reduce the global impurity and not just the local impurity.[15]

IV. SENTIMENT ANALYSIS WITH SARCASM DETECTION: PROPOSED METHOD

When the given text is sarcastic in nature, the sentiment the classifier conveys is exactly opposite to the hidden sentiment. E.g. If the classifier classifies it as positive the hidden sentiment behind that text is negative if it is sarcastic. Thus when sentiment analysis has to be done including sarcasm detection, the sentiment of text should be considered exactly opposite of the classifier result if it is sarcastic.

One naïve way of doing it is by setting a threshold and comparing the probabilities of a sentence being sarcastic (by sarcasm detection) and the sentence being positive or negative (by sentiment analysis). The step by step procedure for this proposed approach is as follows:

Step1:

Pre-process a common data set using processing techniques like noise removal, stemming and lemmatization.

Step2:

Train 2 classifier models. One model is detecting whether the text is sarcastic or not and the other detecting whether the text is positive or negative.

Step3:

Extract the probabilities of text being positive or negative and sarcastic or not.

If the probability of a text being sarcastic is greater than 0.6 (Or the other threshold value) then the sentence can be considered as sarcastic so output the opposite result of the sentiment analysis model.

V. RESULTS & CONCLUSION

Decision Tree and Random Forest classifiers are implemented on the dataset by considering Unigram, Bigram and Trigram Parameters. Also the stability is analysed by sampling and shuffling techniques. Fig 5 shows the results of both the classifiers with three parameters. The table 1 shows the comparison of results for both the classifiers.

Sampling: Refers to training the models with different samples of data with same data set (6 such experiments were conducted on each model to understand its stability of each model.)

Shuffling: Refers to training the models by rearranging the same data. (For understanding the stability, 6 such experiments were conducted on each model.)

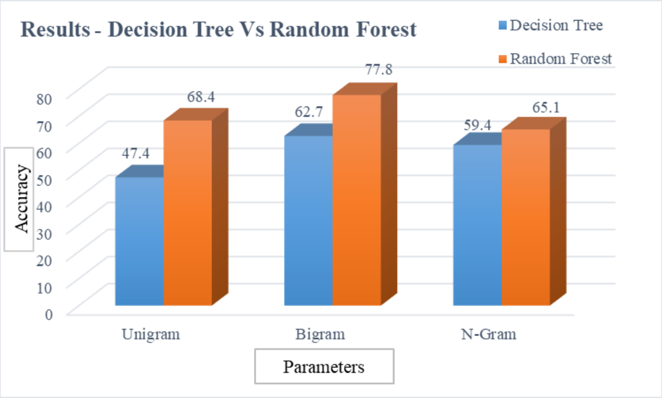

Fig. 1.Results of random forests and decision trees with consideration of parameters like unigram,bigram and trigrams.

After analysing the table I, we can conclude that the random forest can be considered as an ensemble model of decision tree and the performance of random forest is high in terms of both accuracy and stability as compared to individual decision tree. This paper also proposes the method for sentiment analysis including sarcasm detection.

Table- I: Comparative results of Decision Tree Vz Random Forest

Classifier	Accuracy w.r.t Parameter		Stability	
	Parameter	***Accuracy***	***On Sampling***	***On Shuffling***
Decision Tree	Unigram	47.4	Low	Low
	Bigram	62.7	Low	Moderate
	N-Gram	59.4	Moderate	High
Random Forest	Unigram	K=6, m=4 68.4	High	High
	Bigram	K=13, m=4 77.8	High	High
	N-Gram	K=, m=4 65.1	High	High

FUTURE SCOPE

In continuation to this work we would like to implement sentiment analysis method which is proposed in this paper including sarcasm detection. Implementation of the same will make Sentiment analysis job more accurate as well as full proof.

ACKNOWLEDGMENT

Authors would like to thank Research Cell of SGBAU, Amaravati for providing the environment for carrying out this work.

REFERENCES

1. A Review on Sarcasm Detection from Machine-Learning Perspective, Setra Genyang Wicana, Taha Yasin İbisoglu.
2. “Sarcasm Detection in Plain Text Using Machine Learning “, D.V.Nagarjana Devi,]Dr.T.V.Rajanikanth, [3] Dr.V.V.S.S.S. Balaram, International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 4, April 2018
3. “Sentiment Analysis for Sarcasm Detection on Streaming Short Text Data”, Anukarsh G Prasad; Sanjana S, Skanda M Bhat, B S Harish, 2nd International Conference on Knowledge Engineering and Applications, 2017
4. “Sentiment Analysis for Sarcasm Detection on Streaming Short Text Data”, Anukarsh G Prasad; Sanjana S, Skanda M Bhat, B S Harish, 2nd International Conference on Knowledge Engineering and Applications, 2017.
5. “Sarcasm Detection in Reddit”, B.Lakshmanan1, A.Anjana, International Journal of Recent Engineering Research and Development (IJRERD), February 2018
6. The Importance of Multimodality in Sarcasm Detection for Sentiment Analysis , Md Saifullah Razali, Alfian Abdul Halin 2017
7. Detecting Sarcasm in Multimodal Social Platforms, Rossano Schifanella, Paloma de Juan, Joel Tetreault,2016
8. A hybrid approach for Sarcasm Detection of Social Media Data N.Vijayalaksmi* , Dr. A.Senthilrajan**
9. A deep learning approach for identifying sarcasm in text Bachelor’s thesis in Computer Science and Engineering OSCAR BARK ANDREAS GRIGORIADIS ,2017.
10. https://en.wikipedia.org/wiki/Decision_tree
11. www.google.com
12. “Understanding the Twitter Usage of Humanities and SocialSciences Academic Journals”, Aravind Sesagiri Raamkumar, Edie Rasmussen, Mojisola Erdt, Yin-LengTheng, HarshaVijayakumar, Master Journal List of Clarivate Analytics http://ip-science.thomsonreuters.com/mjl, 2018
13. Twitter by the Numbers: Stats, Demographics & Fun Facts, https://www.omnicoreagency.com/twitter-statistics/, 2019
14. “The highest form of intelligence: Sarcasm increases creativity for bothexpressers and recipients”, Organizational Behavior and Human Decision Processes 131 , 162–177, 2015
15. https://www.datascience.com/blog/random-forests-decision-trees-ensemble-methods
16. https://towardsdatascience.com/understanding-random-forest-58381e0602d2
17. https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5

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.