iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** VII **Month of publication:** July 2019

DOI: http://doi.org/10.22214/ijraset.2019.7041

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VII, July 2019- Available at www.ijraset.com

Sentiment Analysis of College Reviews using Machine Learning & Data Mining

Mr. Ghanshyam Gupta[1], Mrs. Mayura Nagar[2], Mr. Ashish Mishra[3]
[1, 3]*Student,* [2]*Professor, MCA Department Sardar Patel Institute of Technology, Mumbai, India.*

Abstract: ***Now-a-days there is no shortage of review forums, blogs and other social networking sites in this digital world. Many studies have been done in analyzing the sentiment of those reviews but the area of "college reviews" has not got much attention in the field of sentiment analysis. So, we have worked on analyzing college reviews and created a system called as College Review Analysis System which will be an online web based platform for automatically analyzing and predicting student review sentiment with the help of machine learning. This system analyses the sentiment of the review of every college in its database. Based on the sentiment, system will classify top college as per their review sentiment. This classification problem is tackled with the help of an algorithm called: "Random Forest". We compared the accuracy of Random Forest with other classifiers as well including "Naïve Bayes", "Decision Trees" and "Support Vector Machine" it turned out that Random Forest performed better in comparison with those classifiers.***

Keywords: Sentiment Analysis, College Review Analysis, Text Classification, Machine Learning, Data Mining, Random Forest.

I. INTRODUCTION

Sentiment Analysis is a challenging field of the current generation in computer science. The challenging part of this domain is the vast variety and unpredictability of sentence formation. Although Natural Language Processing and other Machine Learning approaches have made huge improvement in handling this task, it will be safe to say that we can't apply a particular approach for a problem statement to another problem. In this internet age, people are now more freely voicing their opinion about any subject in the form of text. Thus it becomes more necessary to accurately analyze these text data. This will help to greatly improve the customer experience and many product quality as a whole. This analysis will help us to assess the user sentiment about a particular topic. Here, the topic of our research is "College Reviews". Not many work has been done in this domain even though this is one of the important areas of Sentiment Analysis. Analyzing sentiment of student reviews about their college will help the college to work on their areas of improvement and also improve the quality of education which will help produce better students. This can be one of the novel applications of sentiment analysis. A very trivial approach to solve this type of problem is the "Bag of Words" model. In this model, one or many sentence strings is represented as a bag (sets) of its words, without considering its grammar and even the serial in which the words occur but keeping number of occurrences of a word [1]. We, unlike the Bag of Words model have implemented a more sophisticated approach i.e. Machine learning method in document classification. The classification algorithm we have used is the "Random Forest" for our dataset which has performed reasonably well as compared to the popular "Naïve Bayes" algorithm which has a similar approach as the Bag of Words model. We have used two classes for classification namely "positive" and "negative". This will help the system to determine whether the student is happy about the college or not. Using all these techniques, we created the system which is capable of performing text sentiment analysis as a web-based platform. In ideal scenario, a review should be termed as positive if the opinion or the feeling of the reviewer is inclined positively towards the subject (in this case College reviews). On the contrary, a review should be termed as "negative" if the reviewer is unhappy about the subject or expresses disagreement regarding a subject. We have developed a system as a web platform based on "Java Server Pages" and "Servlet". Along with that we have used a freely available data mining tool named "Weka" which has a java API (Application Programming Interface) from which we have performed the task of predicting the sentiment of a given review.

II. LITERATURE REVIEW

Ross Quinlan (1993) developed C4.5 classifier algorithm which is an extension of ID3 algorithm earlier developed by him [15]. These decision tree classifiers are one of the most popular algorithms in data mining and machine learning due to their simplicity and effectiveness [17]. Both the algorithm develops decision tree from the training dataset provided to it. At every node of the tree, the classifier chooses one attribute from the dataset which most effectively splits the node.

Decision trees that grow deep are subjected to overfitting the data by learning from very irregular patterns. This issue makes the classifier unreliable for predicting future data. Random Forests overcome this difficulty of decision trees by averaging multiple decision trees which are trained on same training data but on different parts. This change greatly increases the performance of decision trees [14].

Thorsten Joachims (2001) introduced the idea of text classification from a simple decision tree classifier to relatively complex Support Vector Machines (SVM) and also demonstrated that SVM is well suited to text classification with many documents [2].

Zhang, Harry (2004) showed that "Naïve Bayes" classifier performs very good even if the attributes are strongly depended upon each other. He also showed that these dependencies cancel out each other, hence the algorithm performs well in most cases. Because of this accuracy, Naïve Bayes outperforms many start-of-the-art algorithms [16].

M. Ikonomakis, S. Kotsiantis and V. Tampakas (2005) observes that the performances of the various classifiers even for a particular classification method is majorly dependent on the training data (corpus) which is provided to the classifier. Author even observes that good training data may produce good quality classifiers. Also the authors look for a common feature selection method across different classifiers and questions the performance benefits of using good feature selection methods [3].

Zi-Qiang Wang, Xia Sun, De-Xian Zhang, Xin Li (2006) on the other hand focused on the performances of the classifiers itself and developed a variant of Support Vector Machine (SVM) algorithm via multiple optimal strategies and demonstrated that this variant of SVM performed much better that other conventional algorithms for text classification [4].

Similarly, Pingpeng Yuan, Yuqin Chen, Hai Jin and Li Huang (2008) proposed a different hybrid classifier by combining SVM and k-Nearest Neighbor (kNN). They first identified cons in the two classifiers and then proposed to combine the features of the two classifiers to make a better one. They also showed that this hybrid approach performed better than SVM and kNN in most cases [5].

Considering that [3] and [4] focused on the classifier performance and accuracy, Mita K. Dalal et al (2011) in her paper "Automatic Text Classification: A Technical Review" discussed on the importance of data pre-processing as a generic step in text classification due to their importance in the accuracy and efficiency of the machine leaning model. Along with this, author identifies lack of classifiers for regional languages which the author considers would be helpful for government organizations [6].

Shifting the focus from traditional machine learning classifiers to highly advanced Neural Networks, Ghiassi, M and Skinner, J and Zimbra, David (2013) successfully implemented "Artificial Neural Networks" into text classification (twitter) with improved accuracy over classical machine learning algorithms [7].

On the other hand, Arundeep Kaur, AP Nidhi (2013) implemented Neural Networks to predict the sentiment of movie reviews [8].

III. OBJECTIVES

Human language is a prominent barrier in better communication between humans and technologies and integration of technologies in human lives. Since interaction between humans and technology happens mostly via an interface, it would be much helpful and comforting if this communication can happen orally (i.e. Computers understanding human language). Thus it becomes substantial to better understand human language for the betterment of human lives.

Keeping the above motivation in mind following are the main objectives of text classification:

A. To design and develop efficient and accurate techniques in performing text classification.
B. An accurate and efficient text classification technique will help producer to improve their product.
C. To use the emerging techniques to create technological products that can understand humans and their linguistics in a better way.
D. As a result of all these benefits, sentiment analysis will help us build a better planet wherein humans and technology will go hand-in-hand.

IV. METHODOLOGY

In order to create a system for sentiment analysis of text using machine learning and data mining approach, we first require data (i.e. Text data in the form of college reviews).

Although, currently, we have many platforms wherein we can get our required datasets (i.e. Kaggle, UCI Machine Learning Repository and many more) but unfortunately it seemed there were no sources from where we could get our required dataset of college reviews. Even though there are web forums, websites where students post their review about their colleges, these websites do not provide a standard API for accessing those data.

But there is a way. We can create a web crawler which will scrape a given website and gather all the required data. A web crawler, sometimes called a spider, is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web indexing (web spidering) or text mining.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VII, July 2019- Available at www.ijraset.com

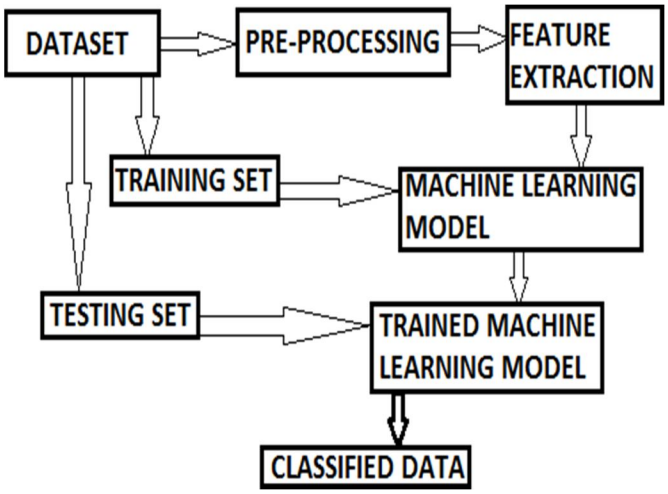


Figure 1 – General Methodology of sentiment analysis [9]

Figure 1 shows the generic methodology in creating a sentiment mining technique. This methodology typically needs to be performed in the following order:

1) Collecting Dataset
2) Pre-processing the collected dataset
3) Dividing the dataset into training and testing set
4) Feature extraction from the dataset
5) Passing the dataset as input to a Machine Learning model in order to train the model
6) Courtesy of above, the machine learning model will learn(train) from the dataset
7) After our model is trained, we can pass the testing dataset (segregated in step3) or a new instance of data to test the accuracy of the model.
8) Measure the accuracy and thrive to improve upon it

A. Collecting the Dataset

As mentioned, since we did not find our required dataset, we went on to create a web crawler in python programming language. This web crawler got us all the student reviews from a particular website along with the college details as well.

For the sake of simplicity, we only chose to get the data of top engineering colleges under Mumbai University. However, this dataset can be expanded to state level and further.

B. Pre-Processing the Dataset

Data preprocessing is a very important step in building any kinds of machine learning or data mining. The primary reason being that this step determines the accuracy and efficiency of our model. Another reason is that the data which we collect or extract is filled with unnecessary clutter or noise.

Example: Review with slang language, unnecessary stop words (special characters like .,:"!), more than one language, spelling mistakes etc.

In order to filter our data from these unnecessary noises, we must perform pre-processing or cleaning of data.

1) *Formatting:* Formatting is done in order to convert the data in the format in which we want to use it. Since we are using "Weka" which is a free data mining tool to perform sentiment analysis, we need to convert the text data "Attribute-Relation File Format" which is the default file format Weka understands. This conversion can be performed via Weka inbuilt conversion tool namely "TextDirectoryLoader". Because we are dealing with classification problem, we first labelled the dataset we collected as either positive or negative manually for every review.
2) *Transformation:* After converting the data into Weka, we need to apply some filtering process to the reviews. This filter is called as "StringToWordVector" which transforms the reviews into a vector (1D array) of words. Mainly it converts String attributes into a vector of attributes which displays word occurrence information from the text present in the strings.

C. Feature Extraction

Feature extraction is a data retrieval process just like a filter wherein it filters outs unnecessary or redundant data attributes from the dataset and only extracts useful features from the provided dataset. Basically, feature extraction acts like a function in which we pass our dataset as input and the function generates output features from it. The generated output may contain useful information and thus may improve the accuracy and efficiency [10].

There are few machine learning algorithms like J48 which have inbuilt feature extraction method but not in Support Vector Machines. So we need to perform this step from our side.

Again, we would use Weka's "StringToWordVector" (STWV) to perform feature extraction. STWV has few extraction tools namely:

1) *TF Transform & IDF Transform (TF-IDF):* Term Frequency–Inverse Document Frequency (TF-IDF) is a feature extraction tool which highlights the importance of a word in the document [11]. In Weka, we have a different version of TF-IDF. Below are the definitions of the same [12]:

a) *TF Transform:* Sets whether if the word frequencies should be transformed into: log(1+fij) where fij is the frequency of word i in document (instance) j.

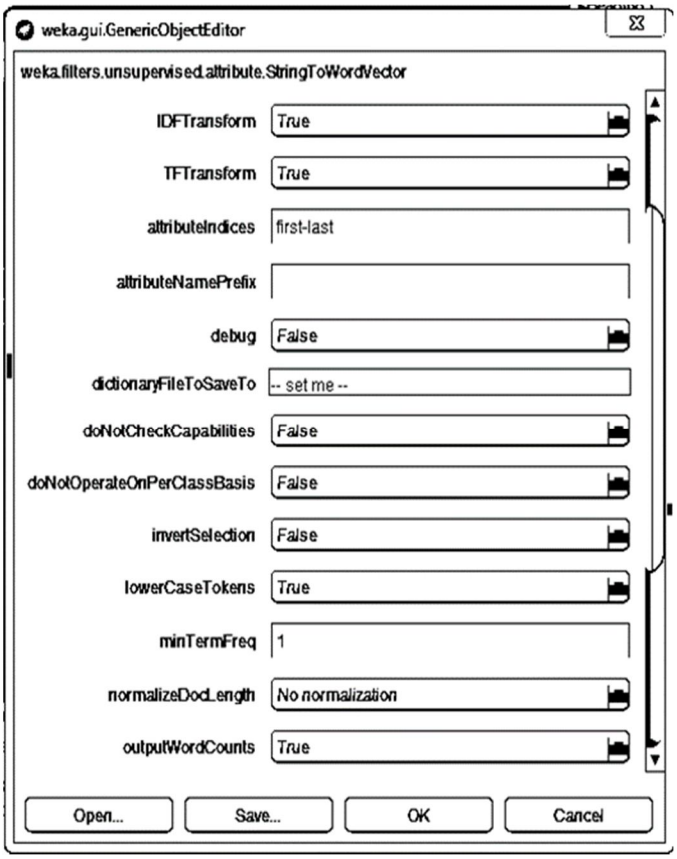


Figure 2 – Illustrating Weka's StringToWordVector filter

b) *IDF Transform: Sets whether if the word frequencies in a document should be transformed into: fij*log(num of Docs/num of Docs with word i) where fij is the frequency of word i in document (instance) j[13].*

2) *Output Word Count:* If we set this filter to true, then actual word counts are used instead of just "1" for word occurrences. This value is the basis for the TF-IDF transformation. Since we set TF-IDF, we set this value to true.

3) *Lower Case Token:* This filter converts all the characters to lower case. Since case of a word does not matter substantially. As shown in the above figure 4.1, there are many other filters like stemmers, stopwords handler and so on which can be applied to the dataset. The key here is what works best for a particular use case.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VII, July 2019- Available at www.ijraset.com

D. Build a Machine Learning Model

Before building a machine learning model, we must choose which machine learning algorithm to choose. Since Weka has many great machine learning algorithms inbuilt, we don't need to look anywhere else.

Since our problem is classification problem we tried with many classification algorithms. Below are the results of them:

1) *J48:* J48 is a Java variant of C4.5 algorithm in the Weka data mining tool. It is a type of decision tree that splits the data into two or more classes or target classes based on the most important attribute in a data. It further splits the node according to the similar rule.

Below are the results of a J48 Classifier

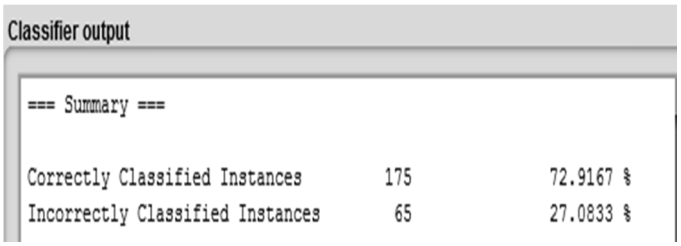


Figure 3 – Result of J48 Classifier

We built the model by training from 467 instances and tested on 240 instances and got the accuracy of 72.91%.

2) *Naïve Bayes:* Naïve Bayes is a probabilistic type of classifier wherein the probability of an instance belonging to a particular class is determined by its probability. Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector

x = (x1, x2, ..., xn) representing some n features (independent variables), it assigns to this instance probabilities:

P(Ck | x1,…, xn) for each of k possible outcomes or classes Ck [13].

Below are the results of a Naïve Bayes Classifier:

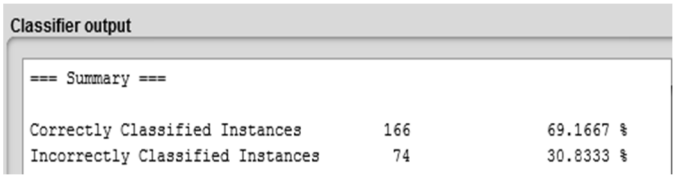


Figure 4 – Result of Naïve Bayes Classifier

We built the model by training from 467 instances and tested on 240 instances and got the accuracy of 69.16%.

3) *Support Vector Machine (SVM):* "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. This algorithm works by plotting the values of every attribute of an instance in a n dimensional vector space. N is the number of features in the dataset. After plotting the values, the algorithm creates a hyper-plane which divides the number of classes in the most appropriate way. Weka uses "Sequential Minimal Optimization" algorithm for training an SVM classifier.

Classifier

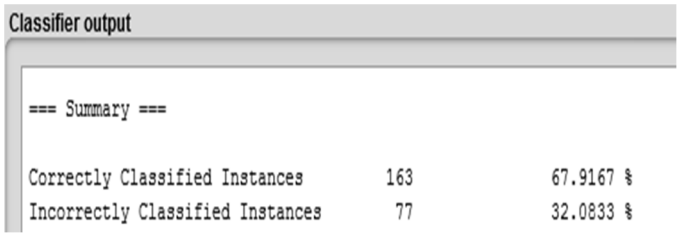


Figure 5 – Result of Support Vector Machine

We built the model by training from 467 instances and tested on 240 instances and got the accuracy of 67.91%.

4) *Random Forest:* Random Forest classification algorithm are mainly used for classification and other tasks. They work by creating multiple decision trees during training period and selecting the class that is the mode of the classes of the individual trees. Random forests are good option in terms of avoiding overfitting from training data [14].

Below are the results of a Random Forest Classifier

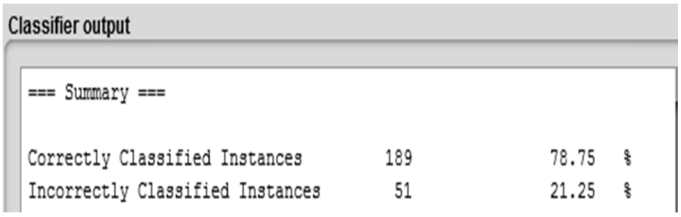


Figure 6 – Result of Random Forest Classifier

We built the model by training from 467 instances and tested on 240 instances and got the accuracy of 78.75%.
It is because of the highest accuracy; we chose Random Forest as our classifier for analyzing sentiment.

E. Evaluating a Classifier

Since we worked on many different classifiers i.e. J48 Classifier, Naïve Bayes and Support Vector Machine, we found that Random Forest had the highest accuracy of 75.83%.
Although, Random Forest has the highest level of accuracy, its comparatively inefficient with respect to time complexity. One of the reason for this is this algorithm creates many decision trees unlike J48 which creates only a single tree.

Classifier	Total Instances (240)		*Prediction Accuracy*
	Correct	*Incorrect*	
J48	162	78	67.5
Naïve Bayes	148	92	61.67
SVM	166	74	69.17
Random Forest	182	58	75.83

Table 1 – Comparison of classification algorithms with their respective prediction accuracy

Below are some of the screen shots of our web application

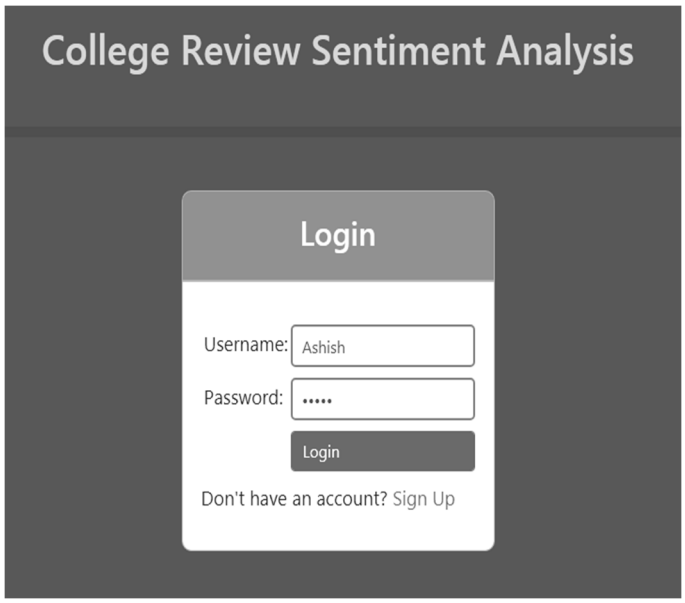


Figure 7 – Login Page

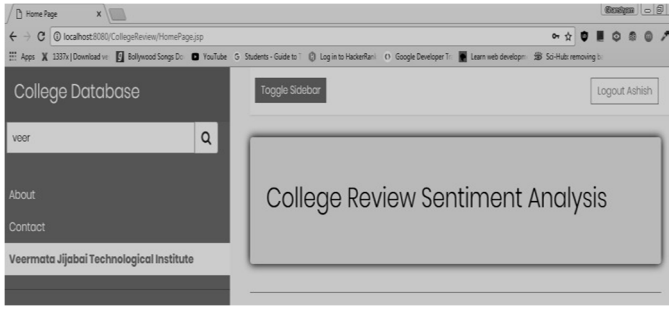


Figure 8 – Home Page where user can search colleges by their names

In the above case, we are searching "Veermata Jijabai Technological Institute".

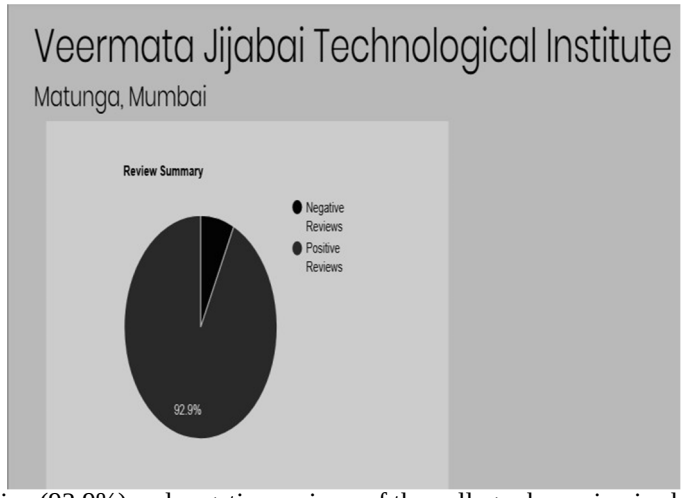


Figure 9 – Total count of positive (92.9%) and negative reviews of the college shown in pie chart with gray and black color respectively

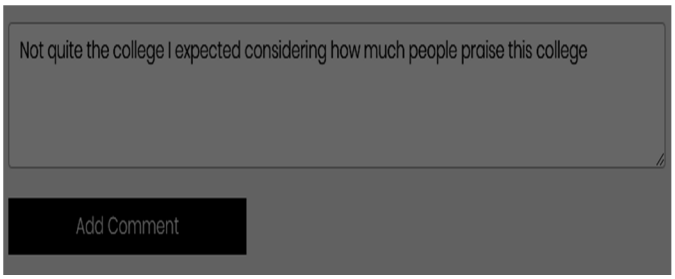


Figure 10 – Sample review saying "Not quite the college I expected considering how much people praise this college"

Figure 11 – Predicted Sentiment "Negative" which is correct in this context

V. CONCLUSION

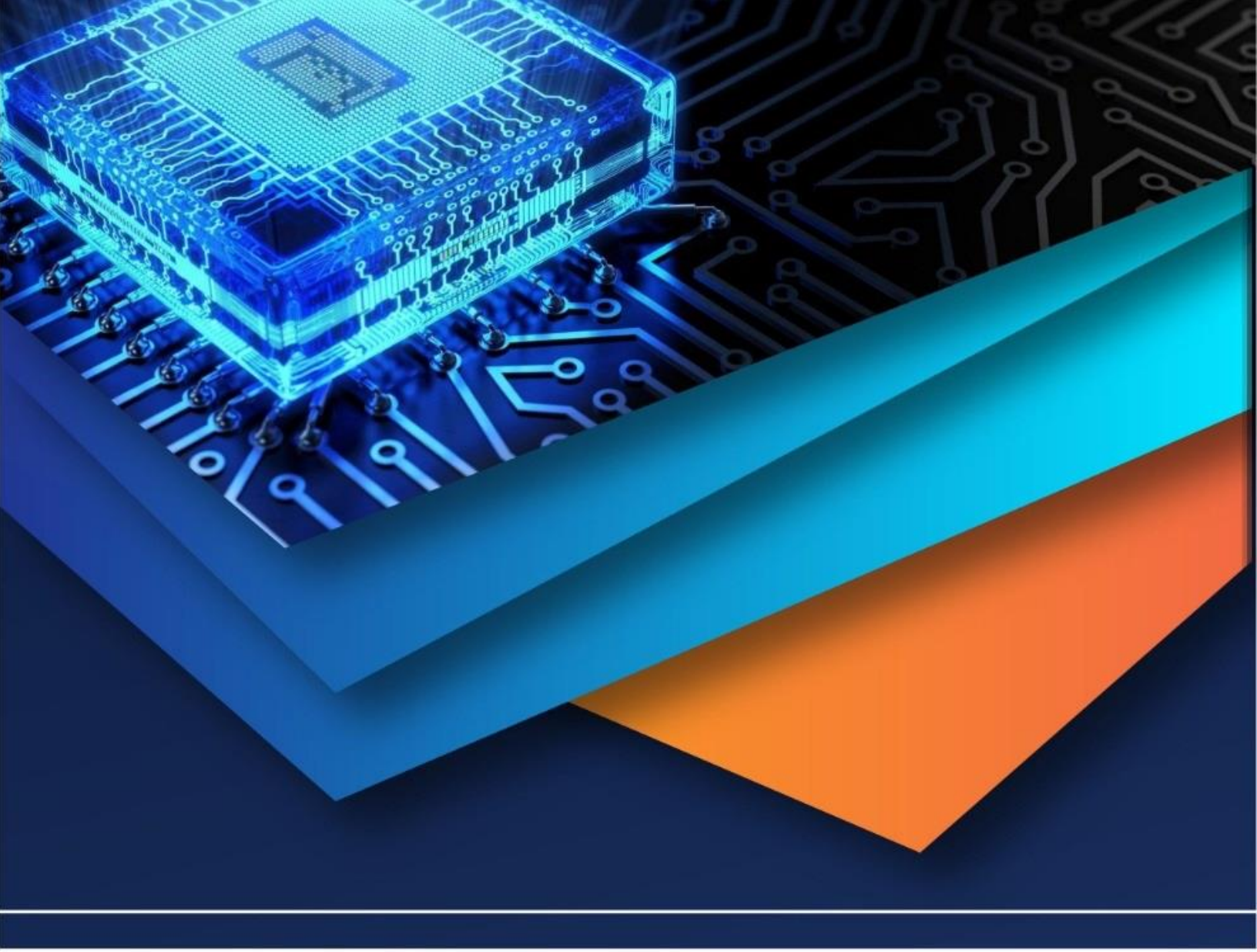
In conclusion, we built a web-based text sentiment analysis platform for reviewing college on the basis of student reviews.
We performed comparative analysis of four different types of machine learning classifiers using Weka Data mining tool on College Reviews dataset which we gathered ourselves.
We compared various classifiers against each other with respect to their advantages and their disadvantages in text classification:

1) J48 Classifier gave the accuracy of 67.5%.
2) Although it is a simple and easy to understand classifier, the classifier is prone to "overfitting" thus resulting accuracy is not the best which we can achieve.
3) Naïve Bayes which is a probabilistic type of classifier is another simple and effective classifier but could only produce prediction accuracy of only about 62.67%. The algorithm treats every word independently in a sentence thus it lacks identifies sentiment of complex sentences.
4) Next, we tried Support Vector Machines, which is comparatively complex than the above algorithm. It gave the accuracy of 69.17 which is good but not the best which we can do considering that it runs slowly.
5) Finally, we used Random Forest which eliminates the issue of over-fitting in J48 but is comparatively slower. But it gives the best accuracy amongst the above algorithms which is 75.83%.

The above accuracies may change with respect to the quality of preprocessing done. Hence, data preprocessing plays an important role in the accuracy of these classifiers.

REFERENCES

[1] Sivic, Josef (April 2009). "Efficient visual search of videos cast as text retrieval". IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 4. IEEE. pp. 591–605.
[2] T. Joachims, "A statistical learning model of text classification with support vector machines". Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp.128-136, 2001.
[3] Ikonomakis, M., Kotsiantis, S. & Tampakas, V. Text classification using machine learning techniques. WSEAS Transactions on Computers, 4, 966—974, 2005.
[4] W. Zi-Qiang, S. Xia, Z. De-Xian, L. Xin, "An Optimal SVM-Based Text Classification Algorithm," Fifth International Conference on Machine Learning and Cybernetics, Dalian, 2006.
[5] Pingpeng Yuan , Yuqin Chen , Hai Jin , Li Huang, MSVM-kNN. Combining SVM and k-NN for Multi-class Text Classification, Proceedings of the IEEE International Workshop on Semantic Computing and Systems, p.133-140, July 14-15, 2008.
[6] Mita K Dalal and Mukesh A. Zaveri. Article: TODV: Automatic Text Classification: A Technical Review. International Journal of Computer Applications 28(2):37-40, August 2011.
[7] Ghiassi, M, Skinner, J & Zimbra, David. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. Expert Systems with Applications. 40. 6266–6282. 10.1016/j.eswa.2013.05.057, 2013.
[8] Arundeep Kaur, AP Nidhi. Predicting Movie Success Using Neural Network. ISSN: 2319-7064, Volume 2 Issue 9, September 2013.
[9] Bhumika Gupta, Monika Negi, Kanika Vishwakarma, Goldi Rawat and Priyanka Badhani. Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python. International Journal of Computer Applications 165(9):29-34, May 2017.
[10] Alpaydin, Ethem (2010). Introduction to Machine Learning. London: The MIT Press. p. 110. ISBN 978-0-262-01243-0. Retrieved 4 February 2017.
[11] Rajaraman, A.; Ullman, J. D. "Data Mining". Mining of Massive Datasets. pp. 1–17. ISBN 978-1-139-05845-2. doi:10.1017/CBO9781139058452.002, 2005.
[12] Ian H. Witten and Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. ISBN 1-.55860-552- 5, 2000.
[13] Narasimha Murty, M.; Susheela Devi, V. Pattern Recognition: An Algorithmic Approach. ISBN 0857294946, 2011.
[14] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284- 5.
[15] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
[16] Zhang, Harry. The Optimality of Naive Bayes. FLAIRS2004 conference.
[17] Ian H. Witten; Eibe Frank; Mark A. Hall (2011). "Data Mining: Practical machine learning tools and techniques, 3rd Edition". Morgan Kaufmann, San Francisco. p. 191.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY