Ensemble Learning solution for the Aspect-based sentimental analysis on IMDB reviews

**Kakavakam Jaswanth Sai**

CH.EN.U4CS20130

Department of Computer Science and Engineering

Amrita Vishwa Vidyapeetham

Chennai, India.

[ch.en.u4cse20130@ch.students.amrita.edu](mailto:ch.en.u4cse20130@ch.students.amrita.edu)

**Dr. Sangapu Sreenivasa Chakravarthi**

Associate Professor

Department of Computer Science and Engineering

Amrita Vishwa Vidyapeetham

Chennai, India.

[ss\_chakravarthi@ch.amrita.edu](mailto:ss_chakravarthi@ch.amrita.edu)

***Abstract :*** ***Nowadays, social media significantly influences how people form opinions about any type of business, politics, commerce, etc. based on user ratings. These reviews were examined using the field of sentiment analysis. This is a crucial component since well-designed and carried out sentiment assessments may lead to better and more accurate estimates in both business and politics. Sentiment analysis is skilled at overcoming a variety of difficulties, including issues with accuracy, problems with binary classification, problems with data sparsity, and polarity shift. There have been several approaches proposed and developed for this, but none of them have been effective in consistently extracting sentiment analysis. We reviewed the traditional lexicon-based method for sentiment analysis in this paper, and then we developed an ensemble model employing machine learning algorithms (Support vector Machine, Logistic regression, Naive Bayes, Random Forest) that outperformed the lexicon-based approach by 89 percent. Additionally, we have shown through a comparison study why the suggested model is the most effective.***

Keywords— Ensemble, Polarity, Sentiment, Lexicon, SVM, Logistic, Naïve bayes, Classification, Random Forest, NLTK, Opinion mining.

# **Introduction**

Sentimental analysis, also referred to as "opinion mining," is an important tool for assessing the viewpoint and general attitude of any sample population toward a specific good, service, event, or subject as expressed in text and shared on social media platforms through blog articles, comments, web reviews, etc. The difficulty of sifting through all posts and reviews and organizing them in a meaningful way may be significant from the standpoint of mining such data and opinionated content. Using a technique called sentiment analysis, text submitted by users to various digital platforms on the Internet is examined and opinions are divided into three groups: positive, neutral, and negative ones.

Sentimental Analysis is considered a three-layered approach (Classification levels). They are : **Document level:** At this level, a complete document is considered for SA. In this method of completing SA, the individual voicing an opinion is viewed as a lone source or an autonomous entity. **Sentence level** - When working at the document level, a fundamental problem is that not all utterances conveying opinions can be categorised as subjective sentences. As a result, the degree to which each sentence is divided and studied independently will influence how accurate the results are. **Aspect level** - Aspect level SA assumes that there are just four conceivable opinions: a positive, neutral, a negative, or an absolutely objective emotion. Based on this understanding we will discuss about lexicon and Machine learning approaches to do the sentimental analysis.

# **Literatur Survey**

There are several implementations of sentiment analysis that are now accessible. Let's look at some of the more well-known research works.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Year** | **Method** | **Domain** | **Accuracy** | **Author** |
| 2008 | Mixed ML algorithm | Product reviews | 83.30 | E. Boiy and M.-F. Moens [1] |
| 2011 | Lexicon Based | Movie reviews | 76.37 | J. Brooke, M. Tofiloski and M. Taboada [3] |
| 2013 | RNN (ML) | Movie reviews | 85.40 | R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts et al., [4] |
| 2015 | PRDM (DL) | Movie reviews | 86.50 | C. Li, B. Xu, G. Wu, S. He, G. Tian and Y. Zhou [6] |
| 2017 | SVM, RF, NN, Bagging | Weighted Fuzzy rule-based SA-Tweets | 73.2,  78.9,  65.9 | Syed Muzamil Basha, Yang Zhenning , Dharmendra Singh Rajput\*, Iyengar N.Ch.S.N and Ronnie D. Caytiles [2] |
| 2017 | SVM | Twitter Sentiment Analysis on Demonetization tweets | 72 | K.Arun 1 , A.Srinagesh 2 , M.Ramesh3 [5] |
| 2020 | Naïve bayes +SVM | Esports & education curriculum | 66.92 | K. Sentamilselvan, D. Aneri, A. C. Athithiya, P. Kani Kumar [7] |

***Table-1:*** *Literature Survey Table*

# **Implementation**

As we have discussed above there have been different approaches which have been designed for this sentimental analysis now we will be going through the famous lexicon-based approach on the IMDB review dataset.

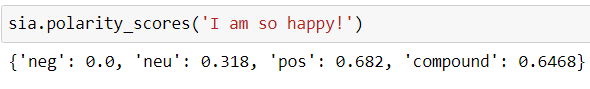
|  |
| --- |
| ***Dataset Description:*** *IMDB Review Dataset [8] is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. they provided a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. The only two attributes in this IMDB Dataset—which has been pre-processed in a way that a meets our goal—are Review and Sentiment.* |

## **LEXICON BASED APPROACH**

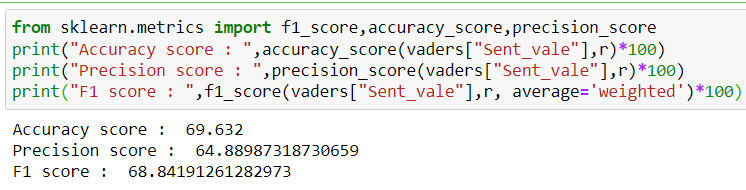
Lexicon-based approaches are one of the methods or procedures used in semantic analysis. This method uses lexical semantic orientation to determine the sentiment orientations of the entire text or group of sentences. Negative, neutral, or positive semantic orientation are all possible. Both manual and programmatically generated versions of the dictionary of lexicons are possible. In this lexicon based analysis again we will have two approaches dictionary based or corpus based approach. In this paper we are going to discuss the Dictionary based approach were there are different dictionary available such as Sent WordNet, AFINN Lexicon, Vader. In which we are going to use Vader sentimental dictionary.

The lexicon- and rule-based sentiment classification tool VADER (Valence Aware Dictionary and Sentiment Reasoner) is particularly geared to the feelings expressed in social media.It may be used to process unlabelled text data directly and is entirely open-sourced and included in the NLTK package. The polarity and strength of an emotion can be detected by VADER.

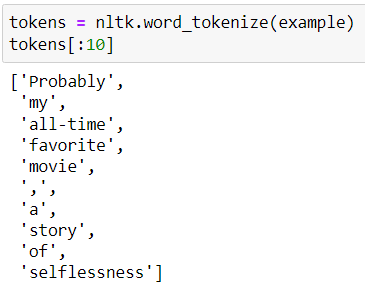
### **Preprocessing for lexicon approach**

Firstly, do the EDA and visualize the distribution of the data. Also, label Encode the sentiment in the given dataset as positive-1 & negitive-0.

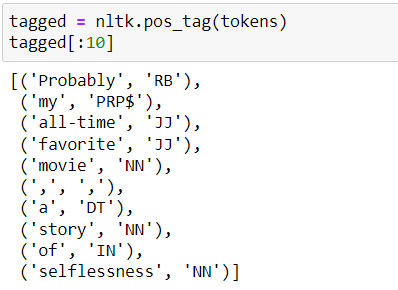


***Fig-1:*** *Data Distribution*

Now we have the text which is not preprocessed for sentimental analysis the first step is **tokenizing**. Tokenization is the process of dividing text into a set of meaningful pieces. These pieces are called tokens. We are going to use the famous NLTK package word tokenizing. For this process



***Fig-2:*** *Tokenization Example*

Each token's parts of speech tag will make it simple for us to distinguish between meanings. The pos\_tag () method from the same NLTK package is thus used.

***Fig-3:*** *Parts of Speech tagging and Chuck conversion*

After POS tagging, the data will already be in the dictionary; all that is left to do is break it up into chunks that can be combined to form words. The ne chunk() function is employed.

### **Analysing the Sentiment by polarity**

Using SentimentIntensityAnalyzer () function of nltk.sentiment package we will predict the polarity\_score of the processed text. polarity\_score() function provides us with the probability of negative, neutral, positive, and compound sentiment of the given text.

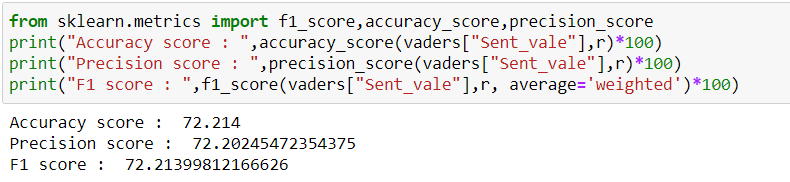
***Fig-4:*** *Polarity score of the example text.*

With a threshold value on compound probability, we can classify the text to positive and negative sentiment.

### **Threshold Selection**

Normally we can take the threshold as 0 where compound probability > 0 can be classified as Positive Text. compound probability <= 0 can be classified as negative Text.

***Fig-5:*** *Accuracy and precision and F1 score for threshold “0”*

If we see the accuracy is not that good (69.6) hence we need to examine it with different threshold values.

|  |  |  |  |
| --- | --- | --- | --- |
| **Threshold value** | **Accuracy** | **Precision** | **F1 score** |
| -0.2 | 69.054 | 64.083 | 68.059 |
| -0.1 | 69.368 | 64.525 | 68.492 |
| 0 | 69.632 | 64.889 | 68.841 |
| 0.1 | 69.886 | 65.243 | 69.171 |
| 0.2 | 70.17 | 65.694 | 69.551 |
| 0.3 | 70.509 | 66.254 | 69.995 |
| 0.4 | 70.852 | 66.871 | 70.440 |
| 0.5 | 71.2 | 67.651 | 70.906 |
| 0.6 | 71.676 | 68.668 | 71.490 |
| 0.7 | 71.99 | 69.9 | 71.912 |
| **0.8** | **72.258** | **71.767** | **72.254** |
| 0.9 | 71.492 | 74.986 | 71.351 |

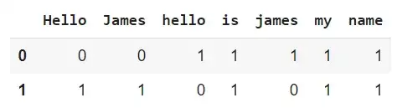
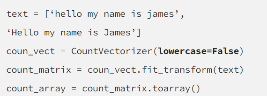
***Table-2 & Fig-6:*** *Optimal Threshold Selection and accuracy*

By calculating the accuracy with different threshold values, I have come up with an optimal threshold value of 0.8 Where the accuracy is **72.258** which is the highest compared to all other threshold values.

As we see the final accuracy from is just 72.258 which is not a best accuracy hence, we move to the some of famous binary classifiers and decide which to be considered or should ensemble it.

## **MACHINE LEANING APPROACH**

SVM, Logistic Regression, Naive Bayes, and Random Forest are the algorithm which we are going analyses SVM, Logistic Regression, Naive Bayes, and Random Forest are the algorithm I will be using. For training this model the preprocessing which we followed in lexicon based approach will not fit. For this we need to pre-processing of text in another way.

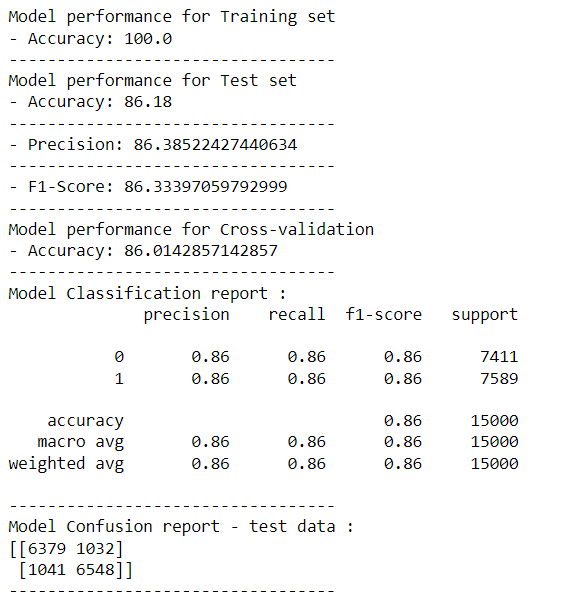
We have taken a whole dataset of 50000 data for my whole training and testing. 25000 are positive and 25000 and negatively classified sentiments. Also, I have checked for null values as there is no null value in the data. I have moved to Text preprocessing. Before moving to remove the noise in text data we will change the sentiment column from categorical to numerical with LableEncoder () where positive -> 1 and negative ->0. Firstly, Using the tweet-preprocessing package of python I have done the basic cleaning there after we have declared some regular expressions to clean the data. After cleaning the data, I split

***Fig-7:*** *Count Vectorizer Example*

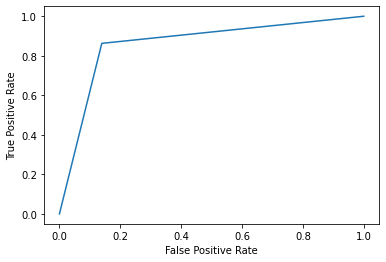
the data in to train and test using the train\_test\_split () function. The next step is Count Vectorizer which helps us to Convert a collection of text documents to a matrix of token counts. with parameters binary to true and stop word to English.

Now our data is ready to push in to the model and also, we can train and test the model. Before, we train the model we need to have a train and test on the data. Using train test split function, we have split the data into 70:30 ratio (i.e., 70% training data and 30% test data). Now we start training the model and examine the results.

### **Support Vector Machine**

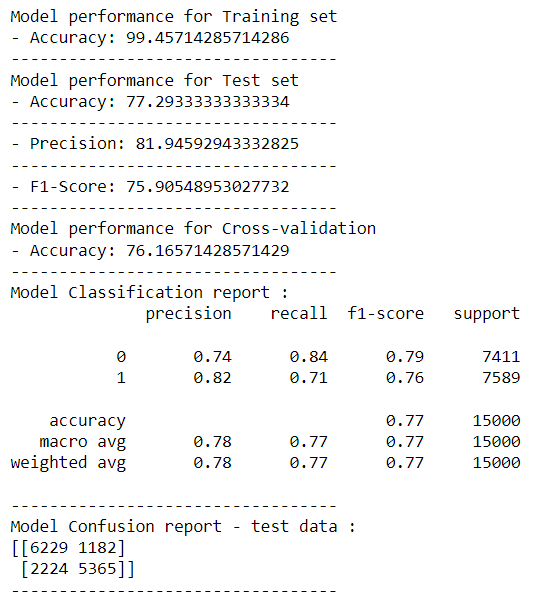
One of the most well-known supervised learning algorithms, Support Vector Machine, or SVM, is used to solve Classification and Regression problems. However, it is largely employed in Machine Learning Classification problems. We are using Linear SVC function scikit-learn package.

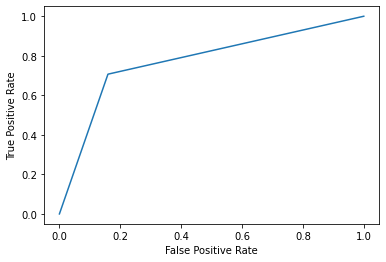
***Fig-8:*** *SVM Model performance Report*

As we see the model is trained with an accuracy of 100% but when it comes to the test data the accuracy is reduced to 86.16% with a precision of 86%. If you see the below ROC curve the graph is above the average line but even, we can improve the accuracy.

***Fig-9:*** *SVM Model ROC Curve*

### **Random Forest**

 The supervised learning approach includes the well-known machine learning algorithm Random Forest. It may be used for ML issues requiring both regression and classification. It is based on the concept of ensemble learning, which is a technique for combining several classifiers to handle challenging issues and enhance model performance. We are using RandomForestClassifier() function of ensemble module in scikit-learn package.

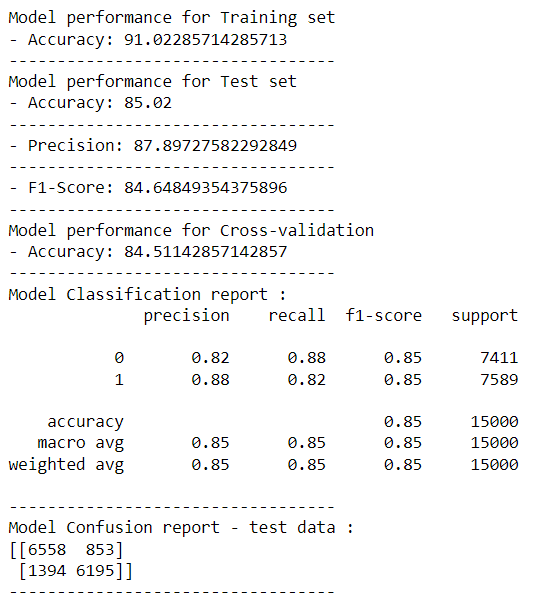
******

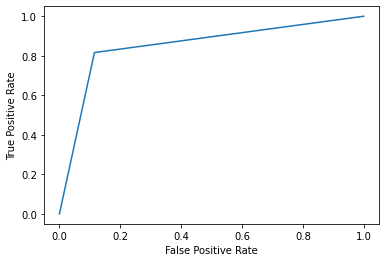
***Fig-10:*** *Random Forest Model Performance Report and ROC curve.*

Here, we can see that the random forest classifier is not a good fit for the data even though the training accuracy is 99.4% but the testing accuracy is quite low at 77.2%. Therefore, we must use a different algorithm than Random Forest.

### **Naïve Bayes – Bernoulli**

The Naive Bayes family includes Bernoulli Naive Bayes. It uses the Bernoulli Distribution as its basis and only accepts binary data, i.e., 0 or 1. If a dataset's features are binary, Bernoulli Naive Bayes is the method to employ, we can presume.

Here our data goal is also to find the binary classification. Hence, we can use this classifier for classifying our data.

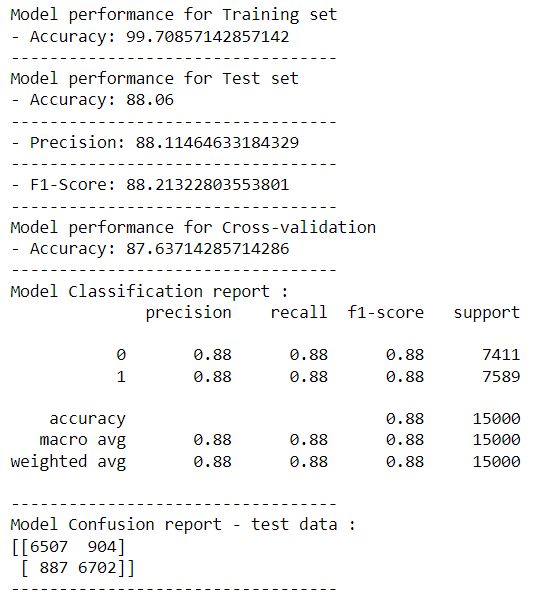


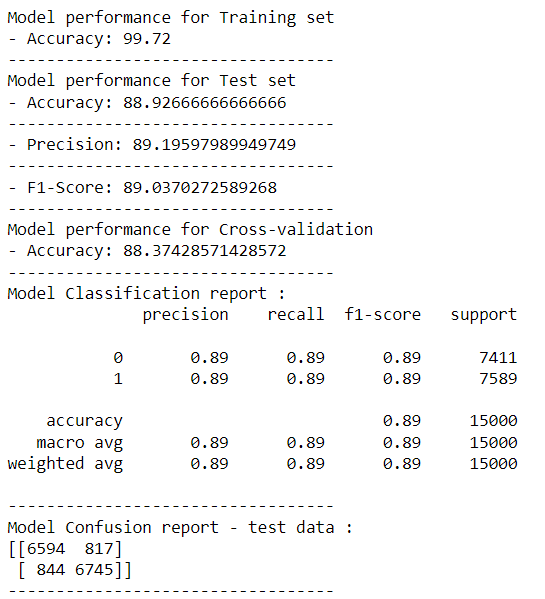
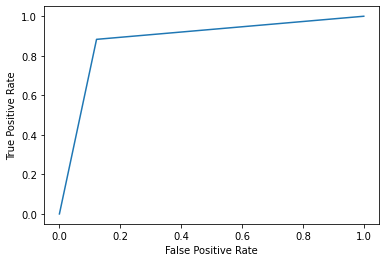
***Fig-11:*** *Naïve Bayes Model Performance Report and ROC curve.*

Since naive bayes is a probabilistic classifier, its output is superior to that of Random Forest, yet practically all of its predictions are identical to those of SVM. The testing accuracy is 85.2%. We may thus take this model into consideration, but as there is always room for improvement, we can use a classifier called logistic regression.

### **Logistic Regression**

Logistic regression is one of the most well-known machine learning algorithms in the supervised learning subcategory. The categorical dependent variable is predicted by this approach utilising a set of independent variables. Logistic regression is used to forecast an outcome of a dependent categorical variable. It can be either Yes or No, 0 or 1, true or false, etc., but instead of presenting the precise value like 0 or 1, it offers the probabilistic values that are in the range of 0 and 1. For this we are going to use Logistic Regression function from linear model module of scikit-learn package.



****

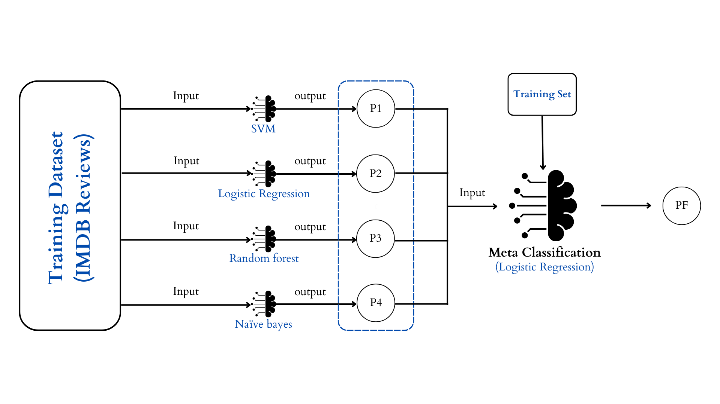
***Fig-12:*** *Logistic Regression Model Performance Report and ROC curve.*

Logistic regression provided us with an accuracy of 88% which much better compared to the all three algorithms which we have discussed previously and also precision is 88% which is best outcome for us. Hence, we can use this model as our primary model of training but if we compare the accuracy there is no much difference between SVM, RF and Logistic classifiers hence we can combine all these three and create a new model.

# **Proposed Solution**

Based on the results available in the four models which we have implemented. We have created an ensemble learning model using Stacking Classifier of scikit-learn package.

When we talk of ensemble learning there are three main methods they are Boosting, Bagging and Stacking methods. In which we are going to use Stacking learning method as our ensemble technique. Stacking frequently takes into account heterogeneous weak learners, learns them in parallel, and integrates together via training a meta-learner to output a prediction based on the various weak learners' predictions, unlike bagging and boosting employed homogeneous weak learners for ensemble. A meta learner attempts to learn the optimal way to combine the input predictions to produce a better output prediction by inputting the predictions as features and the goal as the ground truth values in the data.

 ***Fig-13:*** *Stacking Classifier Architecture. (Here P1, P2, P3, P4 are predictions from four different models and these predictions will be sent as input for our meta classifier and the Final prediction is made.)*

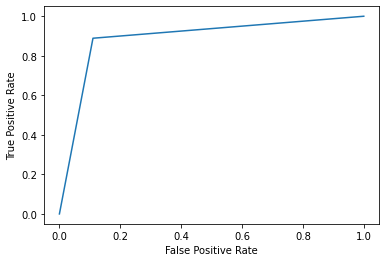
For our model learning we are considering all four algorithms SVM, RF, Logistic and Naïve bayes for our stacking classifier in ensemble learning and the predictions from these classifiers will be sent as input to the meta classifier (i.e., Logistic regression) and the final prediction is made.

# **Results**

For the proposed model the primary classifiers are SVM, Logistic regression, Random Forest and Naïve bayes and the meta classifier is Logistic regression based on this we trained and tested the Stacking Classifier and results are as follow:

***Fig-14:*** *Stacking Classifier Performance Report.*

For stacking classifier we can observe there is an increase in accuracy to 89% which good outcome from our ensemble and the error rate is also low compared to all other models. If we observer the ROC curve in fig-15 that lies in best fit region and makes this model a best compared to other models.



***Fig-15:*** *ROC curve of the Stacking Classifier.*

# **Comparitive Study**

Let’s us have a comparative study of all the models which we have implements in this research and discuss why stacking model is more efficient model compared to all other.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy train | Accuracy test | Precision test | F1\_score\_test |
| Lexicon approach | - | 72.21 | 72.20 | 72.21 |
| SVM linear | 100.0 | 86.18 | 86.38 | 86.33 |
| Random forest | 99.45 | 77.29 | 81.94 | 75.90 |
| Bernoulli Naive Bayes | 91.02 | 85.02 | 87.89 | 84.64 |
| Logistic Regression | 99.70 | 88.06 | 88.11 | 88.21 |
| Stacking Model | 99.72 | 88.92 | 89.19 | 89.03 |

***Table-3:*** *Comparative Study of all models implemented.*

As we can see in the **table-3** the stacking classifier performance stands out over all other models and the final accuracy we got is 89% which is best accuracy which haven’t achieved by Lexicon approach.

# **Future Scope & Conclusion**

In terms of the model's accuracy, there is always potential for improvement, as the ensemble learning research to date for this sentimental analysis is not extremely in-depth. As a result, we can broaden the scope of our research in this field. We can also ensemble the Lexicon and Stacking Classifier to produce a model that is superior to the one that was presented. By combining these two models, we can also improve accuracy. We conclude this paper by designating the stacking classifier as our final working model, with an accuracy of 89% and high precision. Where this model can anticipate a wide range of current enterprises.

# **References**

1. Erik Boiy, Marie-Francine Moens / “A machine learning approach to sentiment analysis in multilingual Web texts” Published: 26 September 2008
2. Syed Muzamil Basha, Yang Zhenning, Dharmendra Singh Rajput\*, Iyengar N.Ch.S.N and Ronnie D. Caytiles / “Weighted Fuzzy Rule Based Sentiment Prediction Analysis on Tweets” International Journal of Grid and Distributed Computing Vol. 10, No. 6 (2017), pp.41-54
3. J. Brooke, M. Tofiloski and M. Taboada/ “Cross-linguistic sentiment analysis: From English to Spanish” Published: 2011
4. R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts et al., / “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank”
5. K.Arun 1 , A.Srinagesh 2 , M.Ramesh3 / “Twitter Sentiment Analysis on Demonetization tweets in India Using R language” Impact Factor Value: 4.029 ISSN: 2349-7084 International Journal of Computer Engineering in Research Trends Volume 4, Issue 6, June-2017, pp. 252-258.
6. C. Li, B. Xu, G. Wu, S. He, G. Tian and Y. Zhou / “Parallel Recursive Deep Model for Sentiment Analysis” Conference: Pacific-Asia Conference on Knowledge Discovery and Data Mining.
7. K. Sentamilselvan, D. Aneri, A. C. Athithiya, P. Kani Kumar / “Twitter Sentiment Analysis using Machine Learning Techniques” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020
8. IMDB Dataset of 50K Movie Reviews / https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.