International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

A Comparative Study of different Classifiers on Political Data for Classification Of Hinglish Text

Ruchira Nehete[1], Amisha Paralkar[2], Osheen Pandita[3], Prof. Amrapali Mhaisgawali[4]

[1,2,3] *Dept. of Computer Science and Technology, Usha Mittal Institute of Technology, Maharashtra, India*
[4]*Professor, Dept. of Computer Science and Technology, Usha Mittal Institute of Technology, Maharashtra, India*

Abstract - *The information mining process has a wide scope of uses. This exploration uncovered key regions like information planning, highlight designing, model preparation, and assessment procedures inside the setting of CRISP-DM, an information mining project. Opinion investigation/assessment digging is a strategy for sorting and characterizing computationally the sentiments or sentiments communicated in a piece of message to decide if individuals' mentalities toward a theme, premium, or item are positive, negative, or impartial. Profound learning is one of the most well-known approaches in numerous normal language handling undertakings. On normal language handling undertakings, profound learning calculations beat conventional AI calculations. It is normal practice these days for individuals to utilize online entertainment to impart their considerations and insights on an assortment of subjects, with legislative issues being quite possibly the most famous themes among numerous social medium posts or recordings. Consistently, an enormous number of political recordings are transferred to YouTube, and most of individuals express their perspectives on ideological groups in the remarks part of these recordings. The review analyzes how well profound gaining strategies remove the political inclinations of YouTubers from the Hinglish dataset. This study will act as an establishment for giving helpful data to ideological groups during the political decision crusading by understanding and following up on electors' feelings.*

Key Words: *Social Media, Sentiment Analysis, Supervised Machine Learning Models.*

1. INTRODUCTION

INDIA is a political entity comprised of partially self-governing provinces, states, and other regions united under a central federal government governed by parliamentary systems governed by the Indian constitution[1] (A. a. N. S. Kaushik). The President of India serves as the supreme commander-in-chief of all Indian armed forces and as the country's ceremonial head [2] (P. a. R. S. Chakravartty).However, the Prime Minister of India, who exercises the majority of executive authority over matters requiring country-wide authority under a federal system, is the leader of the party or political alliance that won a majority in the country's Lok Sabha elections [3] (S. a. J. K. a. C. J. Ahmed).We are attempting to analyze one of India's most historic elections in this research [4](S. a. J. K. a. C. J. Ahmed)

2. OBJECTIVE

The aim is to assess the efficacy of deep learning approaches and determine whether deep learning models applied to political datasets in the Hinglish language provide useful insights that can aid political parties in their campaigning.

The specific goal of this research is to identify a promising classifier capable of effectively classifying the sentiments of Hinglish comments made on YouTube. It may provide some useful statistical data for improving the political campaign.

3. LITERATURE SURVEY

Along with the success of deep learning in many other application domains, deep learning is also popularly used in sentiment analysis in recent years [5](L. a. L. Y. Deng), [6] (A. a. G. D. a. S. S. a. S. P. P. Garg)implemented CNN of 4 convolutional layers on MNIST data set (having a huge number of handwritten text data) MNIST data set with 98.45% accuracy to show how deep learning can achieve high performance in digit or character recognition[7](H. Li)Mi- crosoft Researcher, proposed a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection achieved high accuracy of 66% to 68% compared to previous work on difficult objects. [8](L. a. M. G. a. M. K.-R. a. S. W. Arras)applied the extended LRP version to a bi-directional LSTM model for the sentiment predic- tion of sentences, demonstrating that the resulting word relevance's trustworthy reveal words supporting the classifier's decision for or against a specific class, and perform better than those obtained by a gradient-based decomposition [9](Y. a. L. Y. a. O. A. a. L. S. a. W. S. a. L. M. S. Guo) implemented regional CNN-LSTM model to predict the VA ratings of texts. By capturing both local (regional) information within sentences and long- distance dependency across sentences, the proposed method outperformed regression- and conventional NN-based methods presented in previous studies. These researches show that there is scope for exploring these models on the political domain dataset.

4. PROPOSED METHODOLOGY

The working of the system is divided into the following parts as shown in Fig 1 [10](A. a. W. A. a. M. A. Botzenhardt)

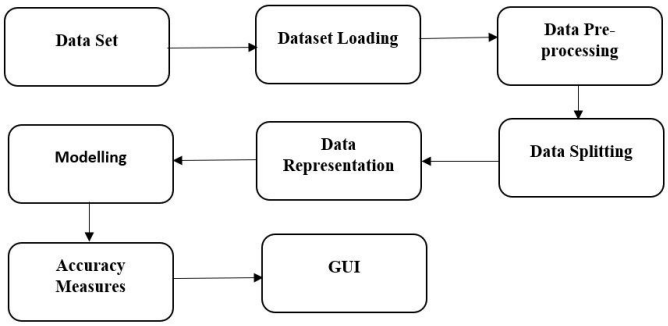


Fig. 1. Flow Diagram

4.1. Business Understanding

The business issue is to evaluate from a dataset of YouTube remarks, which partners the perspectives of the balloter on the political circumstances during the 2014 Indian general political decision. The objective is to utilize profound learning techniques to investigate the opinion of the citizens, whether they are leaning toward a party or against them, and to observe how well these elector's feelings decide

The results of the race. The general business objective of this venture is to observe the best prescient profound learning model among Convolution brain organization and Multilayer Perceptron to order the citizen's opinions.

4.2. Data Understanding

The data interpretation phase begins with data collection and gaining first insights into the data to get acquainted with the data. In this research data is available from gathered from political comments on a political video published on YouTube.The datasets are taken from Various YouTube videos of the top two Indian political parties named Indian National Congress (INC) and Bhartiya Janata Party (BJP). Both the datasets are divided into two categories: -

Label 1- Positive

Label 0- Negative

Label 2- Neutral

All the labeling has been done manually by us In the dataset, each remark is appropriately named with paired numbers 1(one) and 0(zero). One method in favor and 0 methods remark is against the political gathering. Dataset is separated into two CSV documents, for every one of the ideological groups referenced previously. After getting the information, for the displaying reason both the dataset has been converted to a solitary dataset.

4.3. Data Preparation (Pre-Processing)

In pre-processing on both the datasets, which included removal of stop words, null values, numbers, special characters, and punctuation, converting the entire text into lower case, tokenization, and stemming. In the pre-processing phase, some steps d to be taken to access the data better. All the comments without subjects are modified by attaching political party names so that the model can learn easily. For example, In the BJP comments dataset, ”nice” a comment in favor is modified like this:” BJP is nice”. There are some comments which include only political leader names, to identify those comments better, concatenated with political party names BJP or Congress. For instance, “Modi is better than Rahul” will be converted to ”BJP Modi is better than Congress Rahul.” If the comment is a single word or without noun comments like ”Nice” and in favor of BJP, then ”BJP” is added at the beginning of that comment. similarly, if ”Nice” is in favor of congress” Congress” is added to at the beginning of the comment. After removing such ambiguities, the comments can be easily understandable by the model. Afterward, different word vectorization techniques: TF-IDF and Keras token- izer prepare data into deep learning require a format. After pre-processing, various techniques of deep learning and their performance comparison are done with a different configuration of layers and nodes.

1)Removing Stop Words*:* Stop words are the words that add no importance to the sentences. For instance, "is", "am", "we", and so forth. It likewise incorporates accentuation. Such stop words ought to be eliminated from the text corpora before preparing a model. In any case, it is prudent to initially prepare models without eliminating stop words to get the believability of text information. If the model neglects to give preferred exactness over just, we ought to attempt to eliminate stop words and take a look at execution. One explanation for this is that we can't comprehend the main impression that which words are significant and which words are of less significance for building a model. In the investigation, subafteriminating the stop words model gave terrible showing contrasted with the model prepared on information with stop words. Thus, not all stop words however just accentuations were taken out from the crude text information.

2) Stemming*:* Stem is also referred to as a root form of a word. So, a stemming sentence converts each word of a sentence to its original form after removing suffixes and prefixes. For Example: changing to change, played to play. The experiment stemming reduced model performance because of the Hinglish nature of the data. Nltk or Keras does not pose the functions to work with

Hinglish language data. Hence, stemmed words change the meaning of Hinglish words. That reduces the performance of the model.

3)Lemmatization: Then again, morphological examination of the word is a consideration while changing the world over to its root structure. The stem probably won't be a genuine word though, the lemma is a real language word. To do such, nitty-gritty word references are expected for lemmatization calculations to glance through and connect the word structure back to its lemma.

4)Tokenizing: Involves breaking each sentence into the word or character level tokens in N-grams after filtering and removing words that do not add meaning to the sentences Vectorizing techniques like TF-IDF or Keras text to sequence converts this raw text data into vectors representing each word as an integer value [1]((A. a. N. S. Kaushik). By limiting maximum words, the vectorized data trains deep neural networks and are classified with deep learning techniques. [11](A. a. A. M. R. a. M. N. a. A. A. Hassan), [12] (Y. a. W. B. Zhang)In the end, they categorized either in favor of the BJP or against the BJP in favor of Congress.

4.4. Data Splitting

After the data is prepared, it is divided into training and test data with 75% and 25% respectively, using the Hold-Out technique to evaluate the models on unseen/different data than it was trained on, because if we use the same data that we used to create the model, the model will simply remember the entire training set and will always predict the correct label, resulting in overfitting.

4.5 Data Representation

N-gram is a feature extraction method that is carried out by merging N-words into a single set and using each set as a function in the dataset. Like this, the bi-gram approach looks at every pair of terms, and the tri-gram approach looks at every triplet, etc. In this research, we are trying to compare the model performance for different N-grams and how it affects the accuracy of both models.

1)TF-IDF*:* The aim behind TF-IDF is that not all words occurring most frequently in a document are important or related to documents. Instead of just counting the occurrence of words, this algorithm allocates weights or importance to each word and creates a tabular form of words and documents [13](S. Robertson). Inverse Document Frequency defines a measure of the importance of a word. The terms like ”is”, “of”, and” that” which occur in many documents should be less important than one like ”election” which occur in a few politics-related documents. [13] (S. Robertson). TF (Term Frequency) measures the frequency of the term in the document itself. The importance of the word is now the multiplication of these two values IDF and TF. It is called a TF-IDF weight.

IDF(t) = log e(Total number of records/Number of reports with term t in it).

TF(t) = (Number of times term t shows up in a record)/(Total number of terms in the report).

TF-IDF weight = TF(t) * IDF(t)We have around 5324 features maximum, we restricted vector size to 5300 most frequent words generated by TF-IDF trigrams.

2)Keras text to Sequence*:* Change each text in the summary into an entire number progression. Simply the primary ten most often used words will be thought of. Simply words that the tokenizer sees will be considered. We've kept the best language size at 5300 experiencing the same thing as well.

4.6. Modelling

4.6.1. Multilayer Perceptron: Every neuron in the input layer, an output layer, and hidden layers are connected to each neuron in the next and previous layer neurons in MLP architectures. It is the most common type of deep neural network framework and is indeed a nonlinear classifier model to fit high order dimensions for binary classification of political comments. The complexity of the model increases with three hidden layers, although they are hopefully high-performance. For each input word or dimension, it learns weights for each connection to achieve the lowest error. During the training process, the loss function is minimized by updating weights in each iteration with back-propagation techniques like Adams, adamax. Lack of neurons means a lack of the connections that can cause the problem of underfitting and less accuracy. While more layers can lead to the problem of overfitting. With proper hyperparameters, it learns the weights such that the loss function is minimum to achieve the best accuracy. in this research, we examine how the different number of layers and the different number of neurons affect the accuracy of the binary classifier network. As shown in figure 2, we optimize the number of

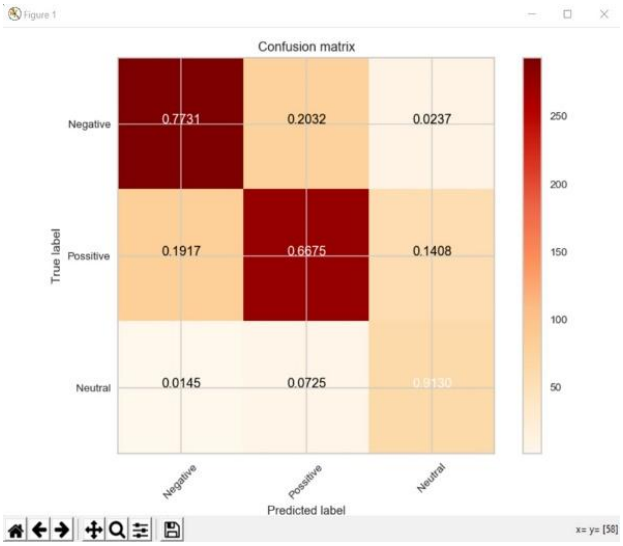
Layer (type)	Output Shape	Param #
dense_36 (Dense)	(None, 500)	2650500
dropout_29 (Dropout)	(None, 500)	0
dense_37 (Dense)	(None, 550)	275550
dropout_30 (Dropout)	(None, 550)	0
dense_38 (Dense)	(None, 100)	55100
dropout_31 (Dropout)	(None, 100)	0
dense_39 (Dense)	(None, 10)	1010
dropout_32 (Dropout)	(None, 10)	0
dense_40 (Dense)	(None, 1)	11

Total params: 2,982,171
Trainable params: 2,982,171
Non-trainable params: 0

None

Fig. 2. Model Summary of Multilayer

hidden layers and the number of neurons in each hidden layer so that the developed model can achieve the best performance. MLP used in the experiment was developed and tested for different number layers containing different neurons. The best performance was achieved on Multilayer perceptron as shown in figure 2.I/P layers of 500 neurons. Each hidden layer holds neurons with Relu (Rectified Linear Unit) activation function with maximum features of 5300. The first hidden layer contained 550 neurons. The second hidden layer occupied 150 neurons and the third hidden layer contained 10 hidden neurons.



Graph. 1. Confusion matrix of Multilayer perceptron

4.6.2. Convolutional Neural Network with Long Short Term Memory: CNN is a form of an artificial neural network. But it differs from standard Multilayer Perceptron (MLP). CNN has a hidden layer called the convolutional layer. It can also have other non-convolutional layers. But the basis of the CNN is the convolutional layer.

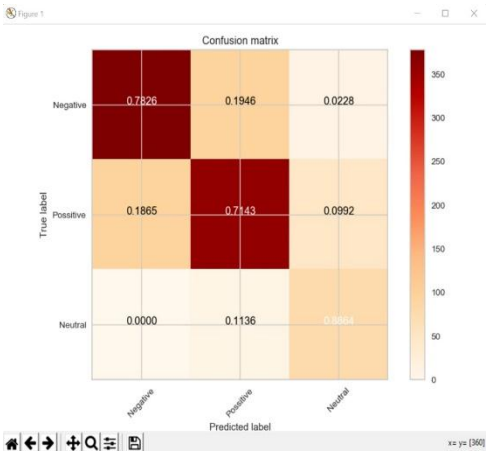
i)Convolutional Layer and filters: The function of the convolutional layer is Just like any other layer, it receives input, performs on input, and transfers it to the next layer neurons in its network. This transformation is a convolutional operation within the convolutional layer. Convolutional layers can detect patterns and images. To be precise, the convolutional layer must be specified with the number of filters. These filters are what detects the patterns. The precise meaning of patterns could be edged in mages, words in sentences, objects like eyes, a nose of faces, etc. The deeper the network is the more sophisticated is the network. And deeper networks can detect even full objects like a dog, cats, or birds [14](R. Girshick).Let's consider a network detecting digits 0 to 9 in the MNIST dataset. Our network is classifying them into images of 1, 2, 3 [6](A. a. G. D. a. S. S. a. S. P. P. Garg). so let's assume the first layer is a convolutional; layer. As mentioned before, we must specify the number of filters the convolutional layers have. The filters can think of as a relatively small matrix with several rows and columns. The values within the matrix are initialized with a random number. When the layer receives an input, the filter considers 3by3 size will slide over each 3*3 set pixel from the input until it slides overall 3by3 block pixels in the entire image. This sliding is referred to as convolving. We can say that filter is going to convolve over each input image. When the filter starts convolving, the dot product of each input 3by3 pixel set and filter matrix values is generated and stored in the convolution layer. The resultant matrix or output of the dot product is passed to the next convolutional layer as an input. And same dot product is repeated for the input matrix and filter matrix. The generated output values are converted into colorsours to visualize predicted numbers.

ii)Pooling Layer: Pooling is added after a convolutional layer. The convoluted matrix would contain a lot of pixel values. If the matrix or previous layer result is very large (ex. Large pixel matrix), the network will result in a slow learner. Typically, it is easy for any network to learn the features if the convoluted matrix size is necessarily reduced. The pooling layer replaces certain locations of input representation with values computed from nearby values. This operation is processed on every slice of the input representation or convoluted matrix. Finally, reducing or

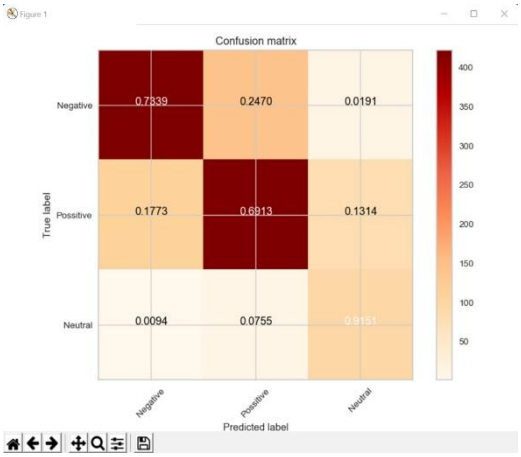
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 5300, 150)         795000
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 5296, 64)          48064
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 1324, 64)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 1323, 64)          8256
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 330, 64)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                33024
_________________________________________________________________
dense_41 (Dense)             (None, 1)                 65
=================================================================
Total params: 884,409
Trainable params: 884,409
Non-trainable params: 0
_________________________________________________________________
None
```

Fig. 3. Model Summary of Convolutional Neural Network with Long Short Term Memory

transformed representation reduces the required computation. There are several functions for pooling. For example, representation by selecting the maximum values of the input matrix. It can also downsample based on averaging on the input matrix. But most of the applications use max pooling operations. And the pooled matrix is passed to the fully connected layers. Flatten layer used after pooling layer to pass the data to fully connected layers. In this experiment, CNN-LSTM was developed and tested for different number layers containing different neurons.



Graph. 2. Confusion matrix of CNN with LSTM



Graph. 3. Confusion matrix of Long Short Term Memory

4.6.3. Long Short Term Memory*:* The LSTM comprises units or memory blocks in the repetitive secret layer, which contains memory cells with self-association putting away the fleeting condition of the organization. Notwithstanding this, the organization has exceptional multiplicative units called entryways to control the progression of data in the organization. Here we make a record planning word reference so that your often happening words are allowed lower files. The primary layer is the Embedded layer that

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 40, 300)	1583400
lstm_1 (LSTM)	(None, 40, 100)	160400
dense_1 (Dense)	(None, 40, 50)	5050
flatten_1 (Flatten)	(None, 2000)	0
dense_2 (Dense)	(None, 460)	920460
dense_3 (Dense)	(None, 180)	82980
dropout_1 (Dropout)	(None, 180)	0
dense_4 (Dense)	(None, 50)	9050
dense_5 (Dense)	(None, 20)	1020
dense_6 (Dense)	(None, 1)	21

Fig. 4. Model summary of Long Short Term Memory

utilizes 300 length vectors to address each word. The following layer is the LSTM layer with 100 memory units (savvy neurons). At long last, since this is an arrangement issue we utilize a Dense result layer with a solitary neuron and a sigmoid initiation capacity to make expectations for the three classes in the issue. Since it is a parallel grouping issue, misfortune is utilized as the misfortune work (binary_crossentropy in Keras). The proficient ADAM enhancement calculation is utilized. Utilizing model designated spot and callbacks, we are saving the model loads when approval exactness is greatest. A number of epoch we utilized are 10, cause we notice that the approval exactness rapidly begins falling, showing that the model is overfitted.

4.6.4. Convolutional Neural Network:

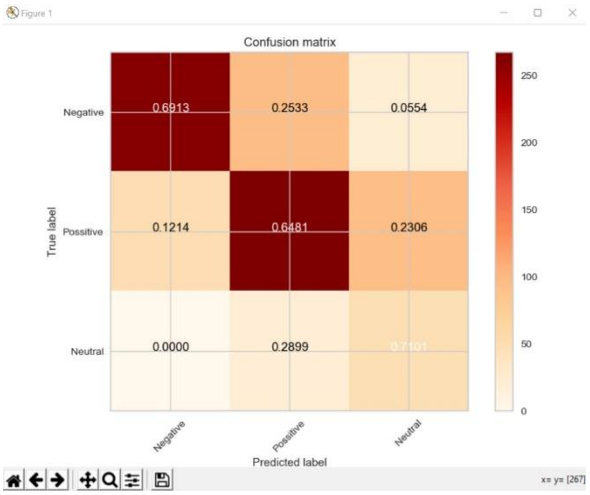
1)Word Embedding: The information from the message feed is inserted into a mathematical structure for the contribution of the CNN. Each word is addressed by a genuine worth vector. The circulated portrayal of words is learned by the procedure of move learning.

2)Convolution Neural Network

Vectors produced by word inserting are the contribution of the brain network layers. The convolution layer is the primary layer of CNN result of this layer is given as far as possible pooling layer and a completely associated network is created. Softmax work is utilized after the completely associated layer to create the result.

Layer (type)	Output Shape	Param #
dense_7 (Dense)	(None, 300)	90300
dense_8 (Dense)	(None, 160)	48160
dense_9 (Dense)	(None, 80)	12880
dense_10 (Dense)	(None, 50)	4050
dense_11 (Dense)	(None, 20)	1020
dense_12 (Dense)	(None, 1)	21

Fig. 5. Model Summary of CNN for this research



Graph. 4. Confusion matrix of CNN

4.6.5. Stacking Classifier: Stacking is a group learning method to consolidate numerous models through a meta-classifier. Dissimilar to helping, stacking isn't ordinarily used to consolidate models of a similar sort for instance a bunch of choice trees. Rather it is applied to models worked by various learning calculations. Assume you have a choice tree inducer, a Naïve Bayes student, and an occurrence-based learning plan and you need to frame a classifier for a given dataset. The standard method is to gauge the normal mistake of every calculation by cross-approval and to pick the best one to shape a model for expectation of future information. In any case, rather than doing this would we be able to simply join every one of them for expectation and consolidate the result. This should be possible by stacking, stacking presents the idea of a meta-student, which replaces the past system. Stacking attempts to realize which classifiers are the solid ones, utilizing another learning calculation the meta-student to find how best to join the result of the base students.

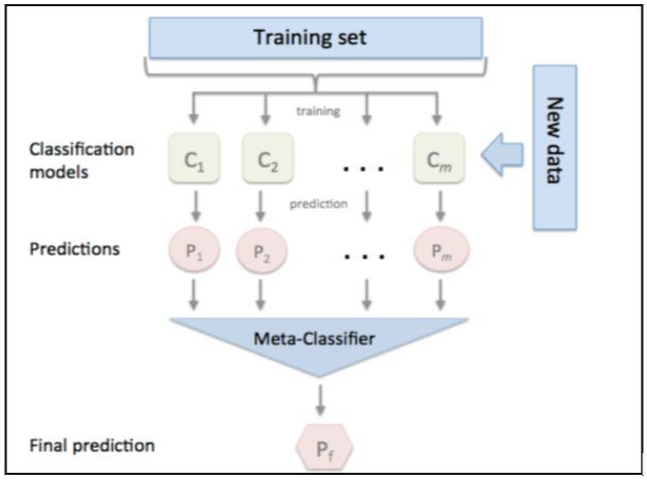
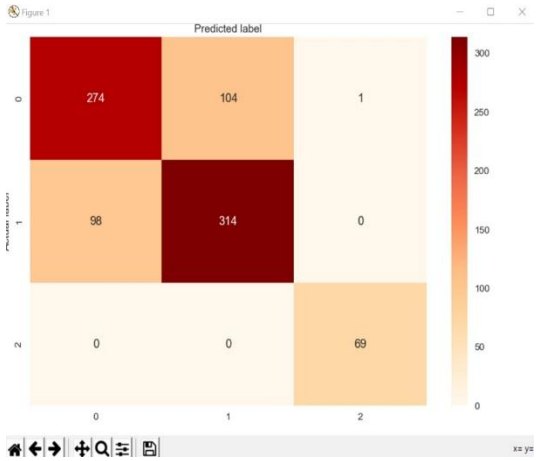


Fig. 6. The architecture of Stacking Classifier



Graph. 5. Confusion matrix of stacking Model

4.7. Accuracy Measures

The accuracy measure is calculated as the total no of correct predictions divided by the total number of records

$$\text{Accuracy} = \frac{TP+TN+TNe}{P+N+Ne}$$

Where, TP =true positive P=positive

TN=true Negative N=negative

TNe=true Neutral Ne=neutral

Sr No	Models	Accuracy
1	Multilayer Perceptron	83%
2	Convolution Neural Network	75%
3	Long Short Term Memory	86%
4	Convolution Neural Network with Long Short Term Memory	87%
5	Stacking Hybrid Model	79%

Fig. 7. Accuracy Table of all Models

Here, the accuracies of all five models were obtained and analyzed which model has the highest accuracy. As mentioned in fig. 7, Convolution Neural Network with Long Short Term Memory has the highest accuracy.

4.8. Graphical User Interface

For Graphical User Interface, Convolution Neural Network with Long Short Term Memory has to be used as it has the best accuracy but, after performing it was noticed that it does not give correct output as it seemed to be over-fitted, so Stacking Classifier was used and correct output was obtained.

5. RESULT

Fig. 8. The classifier has correctly predicted Negative

Fig. 9. Classifier has correctly predicted Neutral

Fig 10. The classifier has correctly predicted Positive

6. CONCLUSION

Sentiment analysis goes under the class of text and assessment mining. It centers around dissecting the opinions of the tweets and taking care of the information to a machine learning model to prepare it and afterward actually take a look at its exactness so that we can involve this model for later use as indicated by the outcomes. It includes steps like data collection, text preprocessing, sentiment detection, sentiment classification, training, and testing of the model. However, it comes up short on the aspect of variety in the information. Additionally, it's as yet not tried how exact the model will be for points other than the one in thought. Thus sentiment analysis has a very splendid extent of advancement in future.

References

[1] A. a. N. S. Kaushik, "A study on sentiment analysis: methods and tools," *Int. J. Sci. Res.(IJSR),* pp. 2319--7064, 2015.

[2] P. a. R. S. Chakravartty, "Mr. Modi goes to Delhi: Mediated populism and the 2014 Indian elections," *Television \& New Media,* vol. 4, pp. 311--322, 2015.

[3] S. a. J. K. a. C. J. Ahmed, "The 2014 Indian elections on Twitter: A comparison of campaign strategies of political parties," *Telematics and Informatics,* pp. 1071--1087, 2016.

[4] A. Kalra, "Twitter to take India election innovations global. Reuters," 2014.

[5] L. a. L. Y. Deng, "Deep learning in natural language processing," 2018.

[6] A. a. G. D. a. S. S. a. S. P. P. Garg, "Validation of random dataset using an efficient CNN model trained on MNIST handwritten dataset," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2019, pp. 602--606.

[7] H. Li, "Proceedings of the Ninth IEEE International Conference on Computer Vision," 2003.

[8] L. a. M. G. a. M. K.-R. a. S. W. Arras, "Explaining recurrent neural network predictions in sentiment analysis," *arXiv preprint arXiv:1706.07206,* 2017.

[9] Y. a. L. Y. a. O. A. a. L. S. a. W. S. a. L. M. S. Guo, "Deep learning for visual understanding: A review," *Neurocomputing,* vol. 187, pp. 27--48, 2016.

[10] A. a. W. A. a. M. A. Botzenhardt, "A Text Mining Application for Exploring the Voice of the Customer," 2011.

[11] A. a. A. M. R. a. M. N. a. A. A. Hassan, "Sentiment analysis on bangla and romanized bangla text (BRBT) using deep recurrent models," *arXiv preprint arXiv:1610.00369,* 2016.

[12] Y. a. W. B. Zhang, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820,* 2015.

[13] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," *Journal of documentation,* 2004.

[14] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015.