Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020)
IEEE Xplore Part Number:CFP20K74-ART; ISBN: 978-1-7281-4876-2

Classification of Online Toxic Comments Using Machine Learning Algorithms

Rahul
Department of Computer Science & Engineering
Delhi Technological University
Delhi, India
rahul@dtu.ac.in

Harsh Kajla
Department of Computer Science & Engineering
Delhi Technological University
Delhi, India
harshcns@gmail.com

Jatin Hooda
Department of Computer Science & Engineering
Delhi Technological University
Delhi, India
jatindce210@gmail.com

Gajanand Saini
Department of Computer Science & Engineering
Delhi Technological University
Delhi, India
saini14gaja@gmail.com

***Abstract*— Toxic comments are disrespectful, abusive, or unreasonable online comments that usually make other users leave a discussion. The danger of online bullying and harassment affects the free flow of thoughts by restricting the dissenting opinions of people. Sites struggle to promote discussions effectively, leading many communities to limit or close down user comments altogether. This paper will systematically examine the extent of online harassment and classify the content into labels to examine the toxicity as correctly as possible. Here, we will use six machine learning algorithms and apply them to our data to solve the problem of text classification and to identify the best machine learning algorithm based on our evaluation metrics for toxic comments classification. We will aim at examining the toxicity with high accuracy to limit down its adverse effects which will be an incentive for organizations to take the necessary steps.**

Keywords—Machine Learning, Toxic Comments Classification, Text Classification, Accuracy

I. INTRODUCTION

The exponential development of computer science and technology provides us with one of the greatest innovations of the "Internet" of the 21st century, where one person can communicate to another worldwide with the help of a mere smartphone and internet.

In the initial days of the internet, people used to communicate with each other through Email only and it was filled with spam emails. In those days, it was a big task to classify the emails as positive or negative i.e. spam or not-spam. As time flows, communication, and flow of data over the internet got changed drastically, especially after the appearance of social media sites. With the advancement of social media, it becomes highly important to classify the content into positive and negative terms, to prevent any form of harm to society and to control antisocial behavior of people.

In recent times there have many instances where authorities arrest people due to their harmful and toxic social media contents[1]. For example, one 28-year-old man was arrested in Bengal for posting an abusive comment against Mamata Banerjee on Facebook and one man from Indonesia was arrested for insulting the police of Indonesia on Facebook. Thus, there is an alarming situation and it is the need of the hour to detect such content before they got published because these negative contents are creating the internet an unsafe place and affecting people adversely.

Suppose there is a comment on social media "Nonsense? Kiss off, geek. What I said is true", it can be easily identified that the words like Nonsense and Kiss off are negative and thus this comment is toxic. But to mine the toxicity technically this comment needs to go through a particular procedure and then classification technique will be applied on it to verify the precision of the obtained result.

Different machine learning algorithms will be used in the classification of toxic comments on the Data set of Kaggle.com. This paper includes six machine learning techniques i.e. logistic regression, random forest, SVM classifier, naive bayes, decision tree, and KNN classification to solve the problem of text classification. So, we will apply all the six machine learning algorithms on the given data set and calculate and compare their accuracy, log loss, and hamming loss.

The rest of the paper is arranged as follows: Section II includes related work, Section III deals with the proposed methodology, and section IV and section V contains result and conclusion respectively.

II. RELATED WORK

A huge amount of data is released daily through social media sites. This huge amount of data is affecting the quality of human life significantly, but unfortunately due to the presence of toxicity that is there on the internet, it is negatively affecting the lives of humans [2]. Due to this negativity, there is a lack of healthy discussion on social media sites since toxic comments are restricting people to express themselves and to have dissenting opinions [3]. So, it is the need of the hour to detect and restrict the antisocial behavior over the online discussion forums [4]. Although, there were efforts in the past to increase the online safety by site moderation through crowd-sourcing

schemes and comment denouncing, in most cases these techniques fail to detect the toxicity [5]. So, we have to find a potential technique that can detect the online toxicity of user content effectively [6].

As Computer works on binary data and in real-world we have data in various other forms i.e. images or text. Therefore, we have to convert the data of the real world into binary form for proper processing through the computer. In this paper, We will use this converted data and apply Machine learning techniques to classify online comments [7]. Text classification can be easily applied on given data set and set of labels by applying the data on a function, that will assign a value to each data value of data set [8].

In this context, Wulczyn et al. [9] research introduced a technique that incorporates crowdsourcing and machine learning to evaluate on-scale personal attacks. Recently, a project called perspective [10] was introduced by Google and Jigsaw, to detect the online toxicity, threats, and offensive content with the help of machine learning algorithms. In another approach, Convolutional Neural Networks (CNN) was used in text classification over online content [11], without any knowledge of syntactic or semantic language [12]. In the approach used by Y. Chen et al. [13], introduced a combination of a parser and lexical feature to detect the toxic language in YouTube comments to protect adolescents. In the approach used by Sulke et al. [14], Online comments are classified with the help of machine learning algorithms. So, lots of work has already been done to detect and classify online toxic comments. In our research paper, we will learn from the already published work and use machine learning algorithms to detect and classify online toxic comments with better accuracy [15].

III. Proposed Methodology

A. Type of classification

In this paper, we have to classify the data into six categories i.e. threat, insult, toxic, severe toxic, obscene, or identity hate and we can put one data value into zero, one or more than one category. Before the start of any processing on our data, our first task will be to identify whether our classification is multi-class or multi-label in nature.

In multi-label classification, one data value can belong to more than one category, E.g. a given sketch of a garden may contain a tree, monument, walking path, or a combination of these and thus sketch can belong to zero, one or more than one categories.

While in multi-class classification, one data value can belong to only one category, E.g. a given car can belong to Honda, Hyundai, Tata Motors, or none of the above companies and thus belongs to either 1 category or of none of them.

In our data set, since our data value can belong to zero, one or more than one category, we have a Multi-Label Classification problem to solve.

B. Machine learning Methodologies

For classifying the online toxic comments we will use six machine learning methodologies i.e. logistic regression, random forest, SVM classifier, naive Bayes, decision tree, and KNN classification.

Since either the comment belongs to the toxic group or will not belong to that, we will use Logistic regression because it will be used to calculate the probability of a comment being toxic or not. Since we can classify the comments into broad categories of toxic and non-toxic and further into 6 labels in case of toxic comment, we will make use of SVM classifier since it distinctively classifies the data values and can also use decision tree and random forest methodology, since in both the methodologies we will use the concept of decision tree and then the final classification of online toxic comments will be done based on the best solution through voting in the decision tree. Since in our data, comments are independent of each other and two distinct comments have no relation in between, we will use Naïve Bayes classification on our data. As we have labeled input data and we can easily apply a supervised machine learning algorithm on it, so we will use KNN classification for classification of online toxic comments.

C. Data Cleaning and Exploratory Visualization of Cleaned Data

The next step in our methodology is to clean the data and extract important features from it. We took our data set directly from the Kaggle website and it is there in the form of CSV files. Firstly we clean it using proper procedure and then we will go for exploratory visualization from it to extract important features.

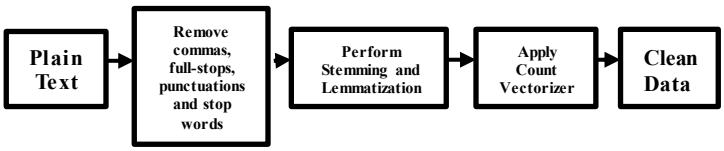

Fig 1: Pre-processing steps for data cleaning.

The process followed in the cleaning of data is shown in fig 1. We will take raw data from the Kaggle website in the form of plain text and apply our techniques to clean the data. Initially, we will remove commas, full-stops, and punctuations. After this, we will remove the stop words. After this, we will perform stemming and lemmatization to get the root word and in the end, we will apply the count vectorizer to get the clean data.

After extracting and analyzing the cleaned data, We got to know that we have a total of 95981 samples of comments and labeled data, which can be loaded from the train.csv file.

To get the better picture of our cleaned data we will go for exploratory visualization.

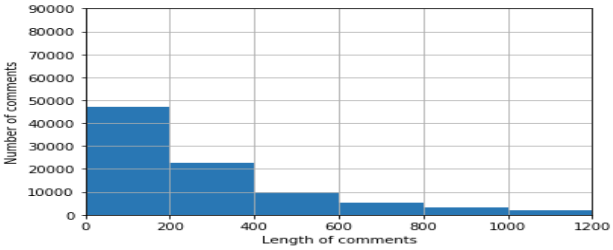

Fig 2: First Visualization of cleaned Data

From Fig 2, we can conclude that in our data set comments are there of varying lengths from within 200 up to 1200 and the number of comments decreases as the length of the comments increase. We can also observe that maximum comments are there of 0 to 200 lengths. Since, as we move forward towards greater length comments, the total number of comments increases manifold, so we have to put a threshold limit on length to get the best result.

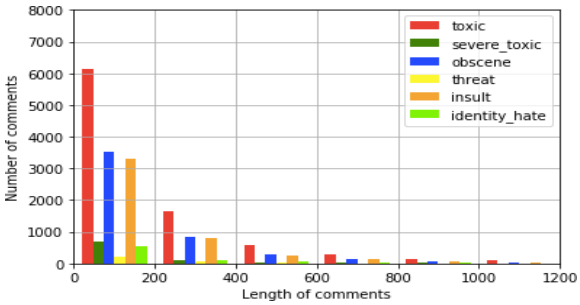

Fig 3: Second Visualization of cleaned data

From Fig 3, we can observe that it is the updated version of fig 2. Here, we are showing all the comments under a definite length range with the number of comments falling under different labels i.e. obscene, toxic, threat, etc. From here we can conclude that the maximum number of comments is of under 200 lengths and as the length of comments increases, the number of comments decreases.

After going through exploratory visualization, we can conclude that we will put the threshold on 400 lengths and select the comments of 4 to 400 lengths.

D. *Finalizing Evaluation Metrics*

Evaluation metrics are used to calculate the quality of machine learning algorithms. Therefore, before applying any machine learning algorithms on our processed data, we have to select the suitable evaluation metrics for our data set to calculate and compare all the techniques. For Multi-label classification there are two major types of metrics:

- Example-Based Metrics: Here we will calculate the value for each data value and then average the result across the data set. Example Hamming Loss, Accuracy, etc.
- Label-Based Metric: Here we will calculate the value for each label of our classification and then we will average out all the values without taking any relation between labels into count. Example average precision, one-error, etc.

We are taking data from the Kaggle website and most of that data is non-toxic. So accuracy as a metric will not give us the true result as 90 % of our data is non-toxic and if we select a simple algorithm that predicts non-toxic nature to every data, it will also result in 90% accuracy. So, it will be a better choice to select the metric that will calculate the loss. So, for our machine learning algorithms, we will select Log-Loss and Hamming Loss as metrics to compare the results of different models.

Equations for calculating Hamming loss and log loss for our data are shown in Equation 1 and Equation 2 respectively

$$\text{Hamming-Loss}=\frac{1}{NL}\sum_{l=1}^{L}\sum_{i=1}^{N} Y_{i,l} \oplus X_{i,l} \quad (1)$$

Here, $\oplus$ is exclusive-or, NL is the number of labels, $Y_{i,l}$ is the predicted value and $X_{i,l}$ is the actual value for the ith comment on lth label value.

$$\text{Log-Loss}=-\frac{1}{N}\sum_{l=1}^{N}\sum_{i=1}^{M} y_{li}\, log p_{li} \quad (2)$$

Here, N is the number of samples, M is the number of labels, y_{li} is a binary indicator of the correct classification and p_{li} is model probability.

E. *Applying algorithms*

Now, since we are ready with clean data and suitable evaluation metrics, we have to select a machine learning model that will give the most optimal result. So, we will apply our machine learning algorithms to our already processed data and calculate and compare their results. We will use the sklearn. metrics and sklearn.linear_model to extract important features from the available comments data.

IV. Result & Analysis

After applying all the 6 machine learning techniques over the cleaned data set of Kaggle, we will get the required result of each machine learning technique in the form of Hamming-loss, Accuracy, and Log-loss. As we have to select the best machine learning model, we have to properly analyze and compare these results.

Hamming-loss, accuracy, and log-loss for each machine learning algorithm are presented in table 1.

Table 1: Hamming loss, accuracy and log loss for machine learning models

Model	Hamming loss	Accuracy	Log loss
Logistic Regression	2.432451957809565	89.46684005201561	2.143587692292937
Naive Bayes	3.764629388816645	86.592977893368	2.3524402426558524
Decision Tree	3.028464094783991	86.68400520156047	2.258255211101094
Random Forest	5.43635312816067	85.65236237537928	0.5844382317269664
KNN Classification	3.8390406010692093	87.12180320762896	1.6592639816189594
SVM classifier	2.7646510431735623	88.69707782814157	2.2914469953676764

The following figures will compare the losses produced by each machine learning algorithm. Since less loss is desirable, the best model will produce the minimum loss.

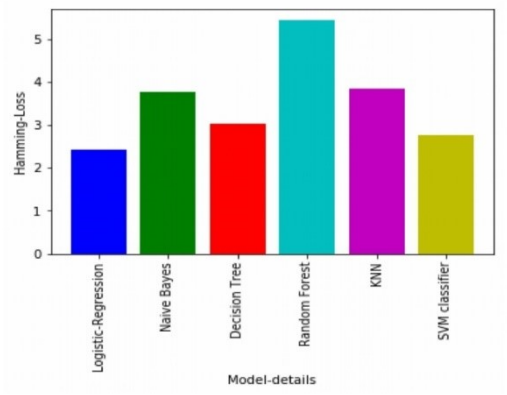

Fig 4: Graphical comparison of Hamming loss for Machine learning Models

After analyzing figure 4, we can conclude that the best model would be logistic regression since It had a hamming-loss of 2.43 % only.

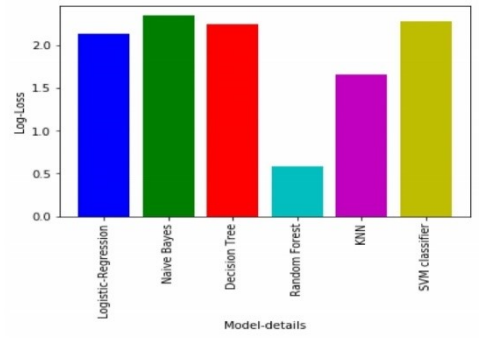

Fig 5: Graphical comparison of Log loss for Machine learning Models

After analyzing figure 5, we can conclude that the best model would be Random Forest Regression since It had a log-loss of 0.58 % only.

The following figure will compare the accuracy produced by each machine learning algorithm. Since high accuracy is desirable, the best model will produce maximum accuracy.

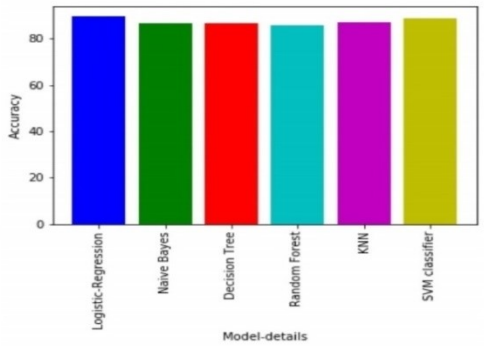

Fig 6: Graphical comparison of Accuracy for Machine learning Models

After analyzing figure 6, we can conclude that the best model would be Logistic Regression since it had an accuracy of 89.46 %.

V. CONCLUSION

We have discussed six Machine learning techniques i.e. logistic regression, Naive Bayes, decision tree, random forest, KNN classification, and SVM classifier, and compared their hamming loss, accuracy, and log loss in this paper. Now after proper analysis, we can say that in terms of hamming loss, logistic regression performs best because in that case, our hamming loss is least, while in terms of accuracy, logistic regression performs best because accuracy is best in that model in comparison to other ones and terms of log loss, random forest works best due to least possible log loss in that model.

So, our final model selection will be based on the combination of hamming loss and accuracy. Since we got the maximum accuracy i.e. 89.46 % and least possible hamming loss i.e. 2.43 % in case of the logistic regression model. We will select the logistic regression model as our final machine learning technique since it works best for our data.

VI. FUTURE WORK

In further research, other machine learning models can be used to calculate accuracy, hamming loss, and log loss for better results. We can also explore some deep learning algorithms such as LSTM (long short-term memory recurrent neural network), multi-layer perceptron, and GRU. So, we can explore many other techniques which will help us to improve the obtained result.

REFERENCES

[1] H. M. Saleem, K. P. Dillon, S. Benesch, and D. Ruths, "A Web of Hate: Tackling Hateful Speech in Online Social Spaces," 2017, [Online]. Available: http://arxiv.org/abs/1709.10159.

[2] M. Duggan, "Online harassment 2017," Pew Res., pp. 1–85, 2017, doi: 202.419.4372.

[3] M. A. Walker, P. Anand, J. E. F. Tree, R. Abbott, and J. King, "A corpus for research on deliberation and debate," Proc. 8th Int. Conf. Lang. Resour. Eval. Lr. 2012, pp. 812–817, 2012.

[4] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Antisocial behavior in online discussion communities," Proc. 9th Int. Conf. Web Soc. Media, ICWSM 2015, pp. 61–70, 2015.

[5] B. Mathew et al., "Thou shalt not hate: Countering online hate speech," Proc. 13th Int. Conf. Web Soc. Media, ICWSM 2019, no. August, pp. 369–380, 2019.

[6] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," 25th Int. World Wide Web Conf. WWW 2016, pp. 145–153, 2016, doi: 10.1145/2872427.2883062.

[7] E. K. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text Classification Using Machine Learning Techniques," no. August, 2005.

[8] M. R. Murty, J. V. . Murthy, and P. Reddy P.V.G.D, "Text Document Classification basedon Least Square Support Vector Machines with Singular Value Decomposition," Int. J. Comput. Appl., vol. 27, no. 7, pp. 21–26, 2011, doi: 10.5120/3312-4540.

[9] E. Wulczyn, N. Thain, and L. Dixon, "Ex machina: Personal attacks seen at scale," 26th Int. World Wide Web Conf. WWW 2017, pp. 1391–1399, 2017, doi: 10.1145/3038912.3052591.

[10] H. Hosseini, S. Kannan, B. Zhang, and R. Poovendran, "Deceiving Google's Perspective API Built for Detecting Toxic Comments," 2017, [Online]. Available: http://arxiv.org/abs/1702.08138.

[11] Y. Kim, "Convolutional neural networks for sentence classification," EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf., pp. 1746–1751, 2014, doi: 10.3115/v1/d14-1181.

[12] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," NAACL HLT 2015

- 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf., no. 2011, pp. 103–112, 2015, doi: 10.3115/v1/n15-1011.

[13] Y. Chen and S. Zhu, "Detecting Offensive Language in Social Media to Protect Adolescents," [Online]. Available: http://www.cse.psu.edu/~sxz16/papers/SocialCom2012.pdf.

[14] A. L. Sulke and A. S. Varude, "Classification of Online Pernicious Comments using Machine Learning," no. October, 2019.

[15] N. Chakrabarty, "A Machine Learning Approach to Comment Toxicity Classification," Adv. Intell. Syst. Comput., vol. 999, pp. 183–193, 2020, doi: 10.1007/978-981-13-9042-5_16.