INTERNATIONAL JOURNAL OF RESEARCH AND DEVELOPMENT IN ENGINEERING SCIENCES
e-ISSN: 2582 - 4201 Volume: 3 | Issue: 5 | Year: 2021

Machine Learning Methods for Online Toxic Comment Classification

Kondala Geethanjali
M.Tech-CST(Cyber Security & Data Analytics)
Department of IT & CA, AUCE(A), Andhra University, Visakhapatnam
e-Mail: k.geethanjali8@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 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

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 the flow of data over the internet change 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 the antisocial behavior of people. 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 to the given data set and calculate and compare their accuracy, log loss, and hamming loss. The online communications between people generate a huge amount of data, giving life to what the researchers defined Big Data: a very huge dataset of information from which it is difficult to extract useful information in a reasonable time if we do not use specific tools and strategies (e.g., Machine Learning, Natural Language Processing, etc.). Some examples of such sources of data are the communications related to social networks, blogs, and so on, an ever-increasing number of information that many companies exploit to offer free or paid services to their users (e.g., targeted recommendation of products and services).

In this scenario, the downside is represented by the risks associated with toxic comments, since the advantages related to the aforementioned kind of information (social networks comments, reviews, political opinions, etc.) are dramatically reduced by those who intervene with verbal attacks, verbal bullying, threats to the person, harassment and, more generally, behaviors that lead some participants to abandon the discussion. Considering that an automatic approach designed to classify the toxicity related to a text must face a multi-class multi-label problem because a text can be classified in more than a class, to define an effective, flexible, and scalably approachable to tackle this problem, this paper proposes a novel method that exploits the following three components:

(a). The first component is the Apache Spark, which is used as a big data framework with its Machine Learning library (MLlib);
(b). The second component is the semantics of words obtained by using the word embeddings representations, a modeling approach that can be profitably applied in different domains (Boratto et al., 2016b; Boratto et al., 2016a);
(c). A last component is a huge number of reviews taken from Udemy1, which represents

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 contain the 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 having dissenting opinions [3]. So, it is the need of the hour to detect and restrict antisocial behavior over the online discussion

forums [4]. Although, there were efforts in the past to increase 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 the 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 to given data set and set of labels by applying the data on a function that will assign a value to each data value of the data set [8].

In this context, Wulczyn et al.'s [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 online toxicity, threats, and offensive content with the help of machine learning algorithms. In another approach, Convolutional Neural Networks (CNN) were used in text classification over online content [11], without any knowledge of syntactic or semantic language [12].

In this 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 [15] online toxic comments with better accuracy.

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 multiclass 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.

(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 to it, so we will use KNN classification for the 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.

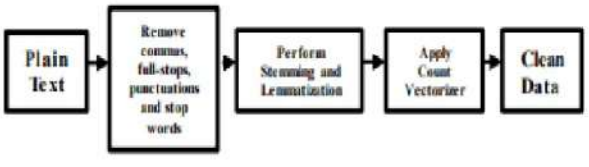

Fig 1: Pre-processing steps for data cleaning.

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 a better picture of our cleaned data we will go for exploratory visualization.

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.

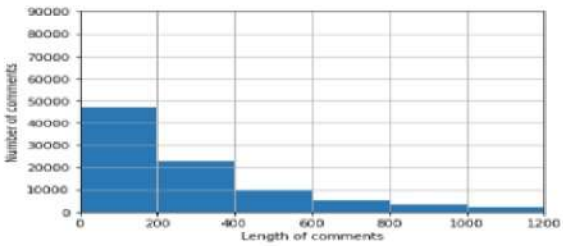

Fig 2: First Visualization of cleaned Data

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.

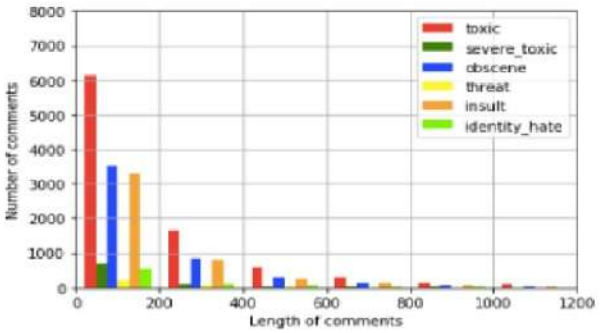

Fig 3: Second Visualization of cleaned data

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 fewer than 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 to 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.

(e). **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 like

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

Here, is exclusive-or, NL is the number of labels, is the predicted value, and is the actual value for the ith comment on lth label value.

$$\text{Log-Loss}=-\frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{M} y_{il}\, log p_{il}$$

Here, N is the number of samples; M is the number of labels, is a binary indicator of the correct classification, and is model probability.

G. 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 AND DISCUSSION

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 Hammingloss, 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.

Model	Hamming loss	Accuracy	Log loss
Logistic Regression	2.432451957809565	89.46684005201561	2.1435876 92292937
Naive Bayes	3.764629388 816645	86.592977 893368	2.3524402 426558524
Decision Tree	3.028464094 783991	86.684005 20156047	2.2582552 11101094
Random Forest	5.436353128 16067	85.652362 37537928	0.5844382 317269664
KNN Classificatio n	3.839040601 0692093	87.121803 20762896	1.6592639 816189594
SVM classifier	2.764651043 1735623	88.697077 82814157	2.2914469 953676764

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

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.

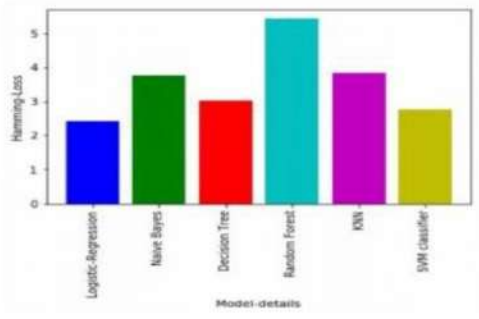

Fig 4: Graphical comparison of Hamming loss for Machine learning Models

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.

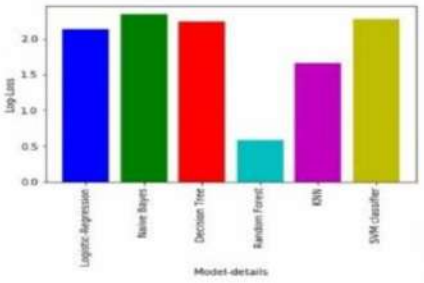

Fig 5: Graphical comparison of Log loss for Machine learning Models

After analyzing figure 5, we can conclude that the best model would be Random Forest Regression since It had a logloss 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.

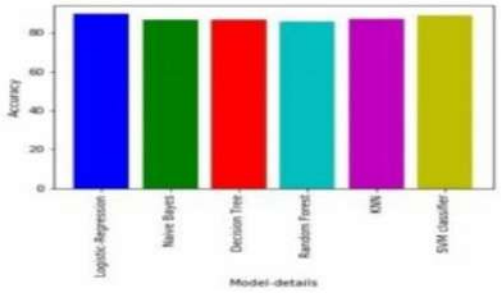

Fig 6: Graphical comparison of Accuracy for Machine learning Models

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.

REFERENCES

[1]. H. M. Saleem, K. P . Dillon, S. Benesch, and D. Rut hs, "A Web of Hat e: Tackling Hateful Speech in Online Social

Spaces," 2017, [Online].Available: ht tp://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. Abbot t, and J. King, " Acorpus for research on deliberation and debate," Proc. 8t h Int. Conf. Lang. Resour. Eval. Lr. 2012, pp. 812–817, 12.

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

[5]. B. Mat hew 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. Mehrdad, and Y. Chang," Abusive language detection in online user content," 25t h Int. WorldWide Web Conf. WWW 2016, pp. 145–153, 2016, doi:10.1145/2872427.2883062.

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

[8]. M. R. Murt y, J. V. .Murthy, and P . Reddy P . V.G.D, " Text DocumentClassificat ion based on 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, " Exmachina: Personal at t acts 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. P Ravindran, " DeceivingGoogle's Perspective API Built for Detecting Toxic Comment s," 2017,[Online]. Available: ht hp://arxiv.org/abs/1702.08138.

[11]. Y. Kim, " Convolutional neural networks for sent enceclassificat ion," EMNLP 2014 - 2014 Conf. Empir. Methods Nat . Lang. Process. Proc.Conf., pp. 1746–1751, 2014, DOI: 0.3115/v1/d14-181

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