Toxic Comment Classification

Mansi Nawani
Computer Science
The University of Texas at Dallas
Texas, USA
mxn180031@utdallas.edu

Satyam Rameshbhai Bhikadiya
Computer Science
The University of Texas at Dallas
Texas, USA
sxb180124@utdallas.edu

***Abstract*— Toxic Comment Classification is that the process of identifying abusive, insulting or hate-based comments from any online platform. The motivation of the matter comes from the multitude of online platforms where people actively make comments and cause deep personal attack to people. supported the analysis of the initial data we've used models like Logistic Regression model and Random Forest classifier. The results after using these models for the classification of toxic comments are compared, it is found that Logistic Regression works better for this data than Random Forest Classifier.**

Keywords—toxic, target, Logistic Regression, Random Forest.

I. INTRODUCTION

The remarkable achievements of engineering science and related fields has provided us one amongst the best developments of all times that's "the Internet" where one can connect with anyone on this planet with just two basic items : access to the web and a wise phone. The exchange of knowledge over the web has grown immensely with the introduction of social networking sites like Facebook, Twitter, Instagram, Snapchat, etc. thanks to rapid growing social networking platforms a very important task is development of algorithms to somehow classify the comments made on social networking platforms as "good" or bad". The contribution of this paper is to develop and illustrate some machine learning methods to investigate and classify disrespectful or hate comments. Toxic comment classification has developed over the amount of your time to become a vigorous research field with the aim of keeping the web conversations healthy and hate-free. it's the responsibility of the websites to supply an inclusive and healthy communication medium to the users. The aim is to coach a highly efficient model which classifies toxic comments and classifications and enables to spot these user handles which engage in hate speech and disrespectful comments. Many classification techniques and methods suffer from insufficient variance in training data and method which often lead to fail on the long end of globe data. Therefore for the aim of future research and discoveries, it's essential to develop models that are error prone and can understand current challenges. A dataset with 1.8 million of records from Kaggle Competition is picked by us to classify the comments as toxic and non-toxic by using big data technologies like Apache Spark and Machine Learning Models. There exist sub-categories in toxic comments which we are building, models identifies toxicity, toxicity type and toxicity topic. we'd also wish to evaluate the performance of those models supported accuracy, precision and F1 score to test for the foremost efficient model for classification.

It is comparatively easier for a person's to classify text or images except for a computer which identifies binary data it's difficult to spot images and text. Any variety of data has to be converted to a form which may be understood by the pc. The text classification algorithms use linguistic communication Processing (NLP), data processing and Machine Learning Techniques to classify disrespectful and toxic comments. to begin with we'd like to pre-process the information that's we've to research and vectorize the computer file and extract the features from the text.

II. BACKGROUND/RELATED WORK

Researchers are performing on Sentiment classification and analysis for a past few years where they need tried and used various machine learning systems to classify the comments and tackle the matter of toxicity in comments. The research for Comment Abuse Classification began with Yin et al's application of mixing TF-IDF with sentiment/contextual features. The performance of the model was compared supported F1 score of the classifier on chat style database. The TF-IDF model gave 6% increase within the F1score. Similarly, some more related works are done so far.

III. DATASET DESCIPTION

The training data comprises of 45 features with 1.8 million of records. The data will be further classified by us as toxic and non-toxic. The dataset accommodates comment id, comment text. The model should predict the target toxicity for test data as every comment within the dataset is own by a toxicity label (target). Labels represents fractional values of human haters and the adequate degree about the comment is described the term or label. Target with value greater than 0.5 are going to be classified as positive (toxic) and those with target value less than value 0.5 are classified as negative (non-toxic).There are subfields in toxicity that are: severe toxicity, obscene, threat, insult, identity attack, sexual explicit comment. The dataset also has identity attributes supported identities mentioned in comment which are : male, female, transgender, etc. The test for the individual comment can be found in comment_text column.

A spread of identity_attributes indicating the identities that are mentioned within the comments are labelled under subset of components and a example of that is shown below.

Comment: I'm a man in my early 20's and I love playing football.

- Toxicity Labels: All 0.0
- Identity Mention Labels: Male: 1.0, Asian: 0.0
- (all others 0.0)

IV. ALGORITHM AND TECHNIQUE

In our project we are building two forms of models supported two different labels. because the task is to work out whether the information belongs to zero, one or quite one categories out of the numerous categories listed above, the primary step before performing on the matter was to tell apart between multi-label and multi-class classification. In multi-class classification, we've got one self-evident truth that our data can belong to only 1 label out of all the labels we've. In multi-label classifications, data can belong to over one labels simultaneously. for instance in our project a comment could also be toxic, obscene and attacking the identity of the person at the identical time. It also can be an opening that the comment is non-toxic and hence doesn't belong to any of the six labels. Our project is multi-label classification. To classify the label as toxic or non-toxic we use binary classification where we are trying to predict the first label named target. We attempt to predict the type of toxicity which can fall into one of the following categories: Insult, sexual_explicit, threat, etc. this classification is often a multi-label classification task and for this we standardized across all models to use binary cross-entropy loss. In the subsequent step we have tried to classify the comments based on the subject i.e. Male, Female, transgender, etc. We've got to pre-process the input file and the extract the features from the text. The most primary step for the preprocessing is to remove the null values for the target column and comment_text column from the dataset and replace the null values within other columns as well. For the purpose of binary classification we have to classify the fractional values to 0 or 1. The subsequent pre-processing step involves converting multiple columns like: Severe_toxicity, Obscene, Threat, Insult, Identity_attack, sexual_explicit into one column (named as toxicity_type within the project) and provides it the best value column name among the columns. As an example given below:
Severe_toxicity: 0.5, Obscene: 0.2, Threat: 0.0, Insult: 0.8 , Identity_attack: 0.5, Sexual_explicit : 0.4
We set the column value of Toxic Topic as Insult because it has the largest value.
The next step within the pre-processing included Down Scaling. Down scaling helps to prevent over fitting in the model. As the dataset provided was unbalanced and the nontoxic comments present in the dataset were more as compared to the toxic comments, the model would predict majority of the test cases as non-toxic and thus would give a high accuracy which would lead it to overfitting. Hence, we attempted to balance the dataset and this down scaling wouldn't let the model overfit. We are going to be use Logistic Regression and Random Forest algorithms to create the model as it is a classification problem. Logistic regression model is a classification model which classifies binary target values and we were able to extend this model to classify multiclass or multiple targets. The target values produced by the regression values lies between 0 and 1 where the values between 0 to 0.5 are equivalent value of class 0 and the value greater than 0.5 are equivalent to class 1. To urge the simple gradient descent, we set the maximum iteration to 10. For the second classification algorithm we have used Random Forest. Random Forest classification algorithm uses an oversized number of decision trees with small heights usually a stump of 1 all acting as an ensemble. The essential idea behind random forest is that an oversized number of uncorrelated decision trees with minimum height all working together will outperform anyone classifier. This algorithm falls under ensemble methods. We set maximum number of trees in each model here also to be 10. After the pre-processing steps we had to make predictive models to classify the labels. For this we used our two aforementioned classifiers. We created a tokenizer which might split the words within the comment_text column and put these array of words during a new column. we've got attempted to filter the stopwords from the tokenizer output column. Stopwords are those words that are frequently employed in both written and verbal communication and thereby don't have either a positive/negative impact on our statement. For this we used StopWordRemover library from MLlib. After removing the StopWords we had to use the Hashing Term Frequency on the clean words and this could be our final features column. After this step we wanted the label to be encoded so we used StringIndexer as a label encoder. the subsequent step involved making the predictive models. during this step we created models of two different classifiers that's Random Forest and Logistic Regression and since we had to predict three different labels for every classifier there are a complete of six models. Since we've got to perform of these steps for training furthermore as testing and also the same step would be accustomed predict all the opposite labels also (Target, Toxicity Type and Toxicity Topic) we built a pipeline which might follow the sequential steps one after the opposite. We used the MultiClassEvaluator from the MLlib library to urge the Evaluation metrics which comprise of the Accuracy, Precision, Recall and F1 score. We obtained these metrics from the confusion matrix obtained for every iteration of hyper parameter tuning on the validation set. The confusion matrix summarizes the right and incorrect predictions through count values for each class that helps gain insight into the categories of

errors made by the model. Accuracy of the model is that the ratio of correctly predicted observations to the overall observations. Precision is that the ratio of correctly predicted positive observations to the whole predicted positive observations. Recall is that the ratio of correctly predicted positive observations to the all observations in actual positive class. F1 score is that the mean of precision and recall. it's a crucial lever to determine the performance of the classifier just in case of imbalanced cases. For the multi-dimensional classification task, additionally we will consider two versions of accuracy measurement, one which reports accuracy on correct classifications across all the classes on one comment (sentence accuracy) and therefore the other which reports label correctness across the complete dataset (label accuracy).

V. RESULT AND ANALYSIS

After testing the two models on the training dataset we evaluated our the results based on Accuracy, Precision and F1 Score. The maximum iteration for both the models was 10 and we also cross validated each of our model with the help grid parameters i.e. hashing term frequency num features. We found the highest accuracy for the “Target” and the “TopicType” was around 79% to 84% which is less than the “ToxicityType” predicted by the third model. Because of the imbalanced data we assume it led to lower accuracy for the “ToxicityTopic” in comparison to “Target” and “ToxicityType”. From the models that we have used for the classification of toxic comments in the pipeline, we conclude that Logistic Regression gives better result for this data in comparison to the Random Forest classifier, as it has low value for precision and F1 Score as compared to Logistic Regression Model.

VI. CONCLUSION AND FUTURE WORK

In this project we worked on various machine learning approaches for toxic comment classification. In future we'll include various other Machine Learning techniques and compare their performances. We plan on using linguistic communication Processing techniques for classifying unintended toxic comments. We also foresee to use Long Short Memory Networks (LSTM) to judge its performance on both binary (toxic vs non-toxic) and multi-label classification (classifying specific reasonably toxicity) tasks. We also seek to realize higher performance in terms of accuracy, precision and F1 score through applying richer word/character representations and using more complex deep learning models.

VII. REFERENCES

[1] https://spark.apache.org/docs/latest/ml-pipeline.html

[2] https://spark.apache.org/docs/latest/ml-classification-regression.html

[3] https://spark.apache.org/docs/latest/mllib-ensembles.html

[4] https://spark.apache.org/docs/2.2.0/api/java/org/apache/spark/ml/evaluation/MulticlassClassificationEvaluator.html