Does unlabeled data improve comment toxicity classification?

Anonymous

1. INTRODUCTION

In the age of social media, it is difficult for developers to monitor every single remark, blog. People utilize this platform to share and learn information at times. However, online users can make toxic remarks that are insulting, vulgar, and can cause people to abandon the debate. As a result, individuals can stop expressing themselves and stop searching out alternative viewpoints. Because platforms fail to properly facilitate dialogues, communities can limit or totally disable user comments.

Machine learning techniques were applied to the jigsaw text data in this notebook, and their accuracy performance is compared on various subsets of dataset to find out the most predictive features behind well performing models. The models under consideration include Random Forest, naive-Bayes, and MLP Classifier, and their performance evaluated using an accuracy metric. While most algorithms in this notebook use a feature matrix based on weights of certain phrases in the comment-text (TF-IDF). The question "Does unlabeled data improve comment toxicity classification?" is investigated and to address this question, the unsupervised algorithms are implemented and being trained on unlabeled data to check whether they enhance classification accuracy or not.

2. LITERATURE REVIEW

Textual aggression is a complicated topic that is being studied and addressed by several knowledge disciplines. This review of related work focuses on a computer science viewpoint on aggressiveness detection, a newly developing field. The scientific research of automated recognition of violent writing using information technology approaches is currently expanding. Several pieces of linked literature are employed in this study to express various sorts of aggressiveness. Hate (Tarasova et al. [1]), cyberbullying (Adamic [2]), abusive language (Nobata et al. [3]), toxicity (Hanson [4]), flaming (Waseem et al. [5]), extremism (Kumar et al. [6]), radicalization (Aggarwal & Zhai [7]), and hate speech are some examples (Georgeakopoulos et al. [8]). Regardless of the variations between such notions, earlier research might provide insight into approaches to the challenge of recognizing aggressive interactions. The automated identification of hate speech is focused on. Georgeakopoulos et al. [8], for example, presents a concise, thorough, systematic, and critical assessment of the topic of automatic hate speech identification in natural language processing.

Many Machine and Deep Learning Approaches have been attempted for detecting types of toxicity in comments.
– Georgeakopoulos et al. proposed a Deep Learning Approach involving Convolutional Neural Networks (CNNs) for text analytics in toxicity classification, obtaining a Mean Accuracy of 91.2% [8].
– Khieu et al. applied various Deep Learning approaches involving Long-Short Term Memory Networks (LSTMs) for the task of classifying toxicity in online comments, obtaining a Label Accuracy of 92.7% [9].
– Chu et al. implemented a Convolutional Neural Network (CNN) with character level embedding for detecting types of toxicity in online comments, obtaining
a Mean Accuracy of 94% [10].
– Kohli et al. proposed a Deep Learning Approach involving Recurrent Neural Network (RNN) Long-Short Term Memory with Custom Embeddings for comment toxicity classification, obtaining a Mean Accuracy of 97.78% [11].

3. METHOD

This section examines the features of the data acquired for this investigation. This section also discusses various machine learning models used on different subsets of dataset to compare the prediction results with respect to the identities provided in the comments dataset. Confusion matrix, and Accuracy are used as metrics to evaluate the models.

3.1. Data Collection

The University of Melbourne provided the data to undertake this research and experiment.

3.2. Description of Proposed Machine Learning Techniques

This section provides an overview of the Term Frequency-Inverse Document Frequency (TF-IDF) approach, the MLP Classifier algorithm, Random Forest, Naive Bayes, DBSCAN, GMM, K-Means, Accuracy, and sklearn. The strategies given are then applied to the problem of toxicity classification and prediction under various subsets of data to analyze the bias. This section provides an explanation of several machine learning approaches.

1) Term Frequency Text Processing Method Technique of Inverse Document Frequency (TF-IDF): The goal of feature extraction is to minimize dimensionality and eliminate unnecessary information to enhance the efficiency and effectiveness of classification algorithms.

2) MLP Classifier: MLP Classifier stands for Multi-layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLP Classifier relies on an underlying Neural Network to perform the task of classification.

3) Random Forest: The random forest method is a classification system made up of numerous decision trees. When creating each individual tree, it employs bagging and feature randomization to produce an uncorrelated forest of trees whose forecast by committee is more accurate than that of any individual tree.

4) DBSCAN: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is the foundation algorithm for density-based clustering. It can find clusters of various forms and sizes in a huge quantity of data that include noise and outliers.

5) GMM: A Gaussian mixture model is a probabilistic model in which all data points are assumed to be created by a mixing of a finite number of Gaussian distributions with unknown parameters.

6) K-Means: K-means clustering is a basic and widely used unsupervised machine learning technique. Unsupervised algorithms often make inferences from datasets using just input vectors and without reference to known, or labelled, outcomes.

7) Given the class, naive Bayes is a basic learning technique that employs the Bayes rule together with the strong assumption that the attributes are conditionally independent. Even though this independence requirement is frequently broken in practice, naive Bayes typically produces competitive classification accuracy.

8) Accuracy: The accuracy of a machine learning classification algorithm is one technique to determine how frequently the system properly identifies a data point. The amount of accurately anticipated data points out of all data points is referred to as accuracy.

9) The machine learning models used for the predictive analysis of dataset are taken from the sklearn library which contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction. Sklearn is used because it is an open-source library which uses the BSD license, and it is widely used in industry as well as in academia.

3.3. Research Question

The experiment will investigate whether unlabeled data will increase toxicity classification or not. This is addressed with using unsupervised learning techniques like DBSCAN Clustering, Gaussian Mixture Model and K-Means Clustering on provided unlabeled dataset to observe the results of the trained models and to compare the results with supervised learning algorithms to answer the question.

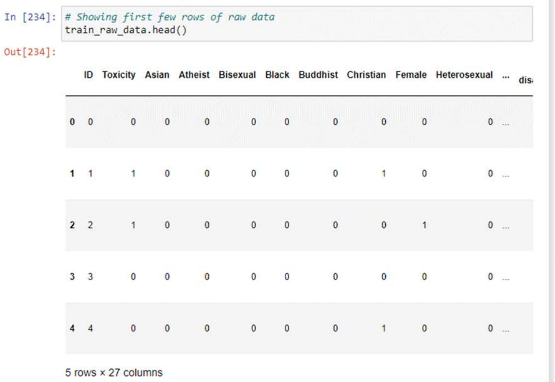
4. RESULTS

Will describe the results of the various models on different subsets of the data. It will then analyse the steps taken and give a comparison between the models used.

4.1. Implementation

This study's methods were implemented in a Jupyter notebook on ANACONDA IDE written in

Python programming language on a personal computer with Windows 10 operating system and an i5 CPU. This section discusses the outcomes of the various machine learning models trained with different subsets of dataset to investigate which categories are more predictive compared to others. The distribution of the categories is shown in Figure 4.1: toxicity, Asian, atheist, black, Christian, and female. The principal labels from the distribution are poisonous, severe toxic, and obscene. Figure 4.1.2 is used to check for missing values and count the overall amount of data presented.

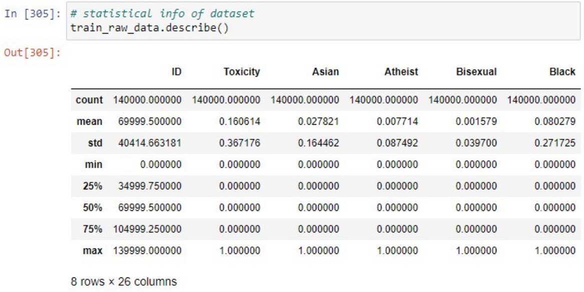


```
In [234]: # Showing first few rows of raw data
          train_raw_data.head()
Out[234]:
```

	ID	Toxicity	Asian	Atheist	Bisexual	Black	Buddhist	Christian	Female	Heterosexual	...	dis
0	0	0	0	0	0	0	0	0	0	0	...	
1	1	1	0	0	0	0	0	1	0	0	...	
2	2	1	0	0	0	0	0	0	1	0	...	
3	3	0	0	0	0	0	0	0	0	0	...	
4	4	0	0	0	0	0	0	1	0	0	...	

5 rows × 27 columns

Fig. 4.1.1 Initial rows of Dataset



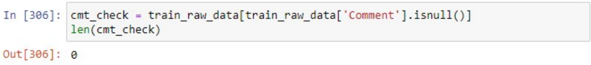
```
In [305]: # statistical info of dataset
          train_raw_data.describe()
Out[305]:
```

	ID	Toxicity	Asian	Atheist	Bisexual	Black
count	140000.000000	140000.000000	140000.000000	140000.000000	140000.000000	140000.000000
mean	69999.500000	0.160614	0.027821	0.007714	0.001579	0.080279
std	40414.663181	0.367176	0.164462	0.087492	0.039700	0.271725
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	34999.750000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	69999.500000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	104999.250000	0.000000	0.000000	0.000000	0.000000	0.000000
max	139999.000000	1.000000	1.000000	1.000000	1.000000	1.000000

8 rows × 26 columns

Fig. 4.1.2 Statistics of Dataset

Figure 4.1.3 shows the output to check for comments that do not fall into one of the categories (toxic or non-toxic). This results in zero (0) output, indicating that there is no output that is not toxic or non-toxic. If it falls into any of the categories, it will be 0 all the way through, making it a non-toxic comment.



```
In [306]: cmt_check = train_raw_data[train_raw_data['Comment'].isnull()]
          len(cmt_check)

Out[306]: 0
```

Fig. 4.1.3 Number of null comments

Figure 4.1.4 Shows the various toxicity and non-toxicity ratio towards different identities. The most affected identities are Female, Black, Homosexual gay, White, Muslim, Male and Christian.

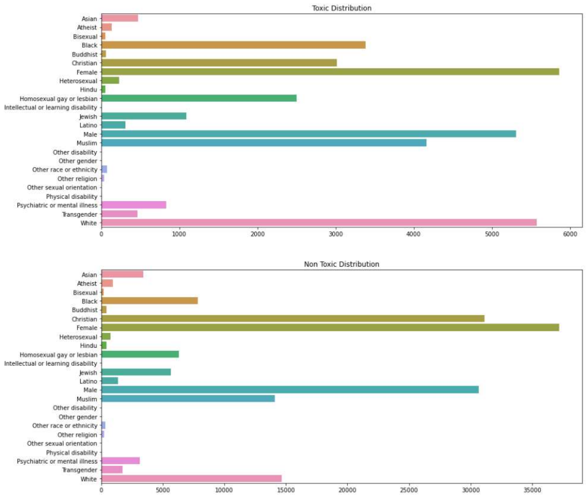
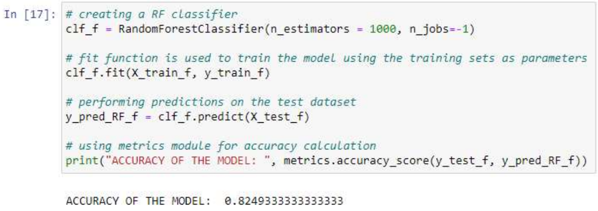


Fig 4.1.4. Toxic and Non-Toxic ratio identity-wise

4.2. Data Training

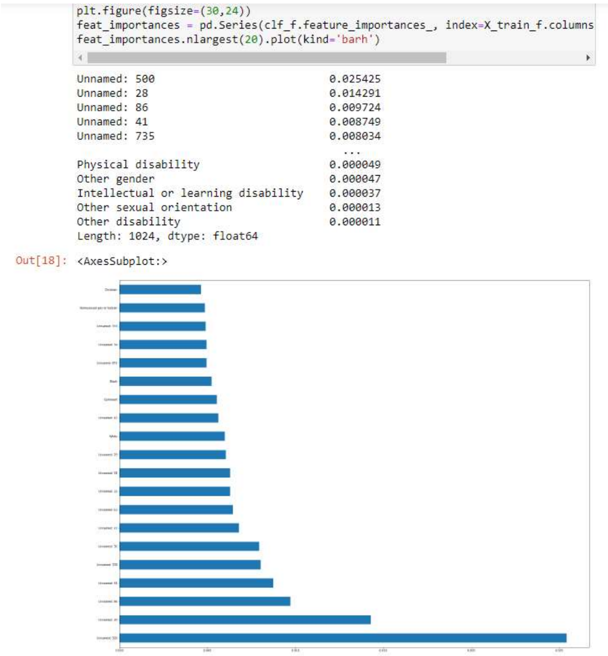
The processed dataset is trained by treating the problem as a binary label classification problem. The pre-processed complete dataset is trained by Random Forest Classifier, with the training shown in Fig. 4.2.1 and reported in Table 4.2.1.



```
In [17]: # creating a RF classifier
         clf_f = RandomForestClassifier(n_estimators = 1000, n_jobs=-1)

         # fit function is used to train the model using the training sets as parameters
         clf_f.fit(X_train_f, y_train_f)

         # performing predictions on the test dataset
         y_pred_RF_f = clf_f.predict(X_test_f)

         # using metrics module for accuracy calculation
         print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test_f, y_pred_RF_f))

         ACCURACY OF THE MODEL:  0.8249333333333333
```

Fig. 4.2.1 Training Random Forest

Figure 4.2.2 depicts the predictive features from the dataset for Random Forest Classifier. The predictive features are: Unnamed: 500, Unnamed: 28, Unnamed: 41, Unnamed: 86, Unnamed: 63, Unnamed: 735, White, Comment, Black, Christian, and homosexual gay or lesbian.



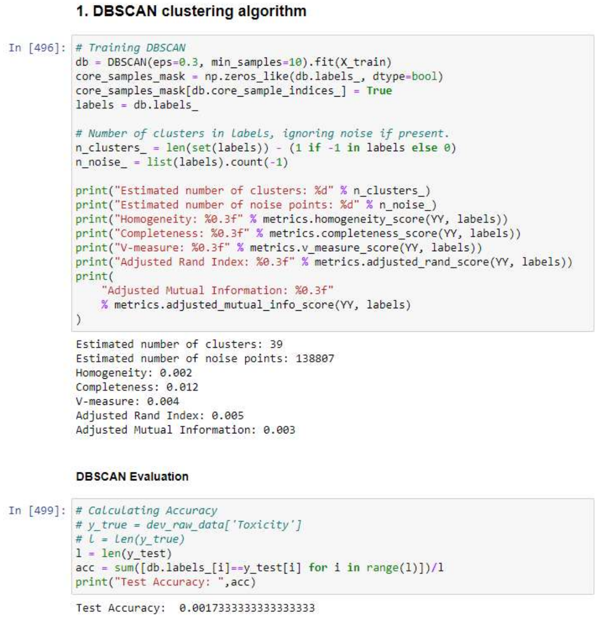
```
plt.figure(figsize=(30,24))
feat_importances = pd.Series(clf_f.feature_importances_, index=X_train_f.columns
feat_importances.nlargest(20).plot(kind='barh')
```

```
Unnamed: 500                                 0.025425
Unnamed: 28                                  0.014291
Unnamed: 86                                  0.009724
Unnamed: 41                                  0.008749
Unnamed: 735                                 0.008034
                                               ...
Physical disability                          0.000049
Other gender                                 0.000047
Intellectual or learning disability          0.000037
Other sexual orientation                     0.000013
Other disability                             0.000011
Length: 1024, dtype: float64
```

```
Out[18]: <AxesSubplot:>
```

Fig 4.2.2 Feature Importance

Figure 4.2.3 shows the training of the unsupervised learning models, which are being trained on unlabeled dataset to find pattern in the data and then classify based on that pattern in various categories.



1. DBSCAN clustering algorithm

```
In [496]: # Training DBSCAN
          db = DBSCAN(eps=0.3, min_samples=10).fit(X_train)
          core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
          core_samples_mask[db.core_sample_indices_] = True
          labels = db.labels_

          # Number of clusters in labels, ignoring noise if present.
          n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
          n_noise_ = list(labels).count(-1)

          print("Estimated number of clusters: %d" % n_clusters_)
          print("Estimated number of noise points: %d" % n_noise_)
          print("Homogeneity: %0.3f" % metrics.homogeneity_score(YY, labels))
          print("Completeness: %0.3f" % metrics.completeness_score(YY, labels))
          print("V-measure: %0.3f" % metrics.v_measure_score(YY, labels))
          print("Adjusted Rand Index: %0.3f" % metrics.adjusted_rand_score(YY, labels))
          print(
              "Adjusted Mutual Information: %0.3f"
              % metrics.adjusted_mutual_info_score(YY, labels)
          )

          Estimated number of clusters: 39
          Estimated number of noise points: 138807
          Homogeneity: 0.002
          Completeness: 0.012
          V-measure: 0.004
          Adjusted Rand Index: 0.005
          Adjusted Mutual Information: 0.003
```

DBSCAN Evaluation

```
In [499]: # Calculating Accuracy
          # y_true = dev_raw_data['Toxicity']
          # l = len(y_true)
          l = len(y_test)
          acc = sum([db.labels_[i]==y_test[i] for i in range(l)])/l
          print("Test Accuracy: ",acc)

          Test Accuracy:  0.0017333333333333333
```

Fig. 4.2.3 Unsupervised DBSCAN Training

Table 4.2.1 Accuracies of various Unsupervised models

Model	**Accuracy**
K-Means	73.76%
GMM	36.04%
DBSCAN	0.17%

The testing accuracies obtained from the unsupervised techniques are very low because the dataset contains high number of features and huge sparse data, that's why unsupervised learning algorithms didn't succeed to find the exact clusters/patterns for the binary classification and hence unlabeled data which is used to train the unsupervised learning algorithms does not improve the classification accuracy of toxicity classification.

Table 4.2.2 Testing Accuracies of various models on different subsets of dataset.

Model	**Accuracy**
RF – Baseline (TFIDF values)	82.43%
RF – Baseline (Whole Dataset)	82.49%
RF – Baseline (TFIDF + Religion)	82.37%
RF – Baseline (TFIDF + Gender)	82.46%
MLP Classifier (TFIDF values)	78.88%
Multinomial Naive Bayes (TFIDF values)	81.32%

The testing accuracy for the various models acquired from training on various subsets of comments dataset is good. Let us analyze various model's performance on different subsets of dataset.

Random Forest with using hyperparameter n_estimator equal to 1000 as higher number of trees gives better performance on large dataset keeping other parameters default because of the optimal results, provided maximum testing accuracy of 82.49% for the whole dataset because it is trained with all categories and features and as RF can generalize over the large data in a better way due to its decorrelation of trees with the introduction of splitting on a random subset of features. Random Forest as baseline also trained with a subset of data including TFIDF features and obtained a testing accuracy of 82.43%, the small decrease in accuracy because of the less feature i.e., identity categories that were not used in training. Random Forest trained with the subset of TFIDF features and one category named religion obtained 82.37% which shows that this category is not important to algorithm for predictive features, it may be due to biased data provided. Lastly RF trained with a subset of TFIDF features and one category named gender obtained 82.46% due to the gender category's predictive importance.

MLP Classifier with keeping hyperparameters default due to their optimal outcome where hidden_layer_sizes=100, activation='relu', solver='adam' and learning_rate_init=0.001, is trained with a subset of dataset including only TFIDF features obtained 78.88% testing accuracy, the reason behind this low accuracy is because the MLP includes too many parameters because it is fully connected. Each node is connected to another in a very dense web – resulting in redundancy and inefficiency.

Naïve bayes with keeping hyperparameters default as there was no difference in performance, obtained 81.32% testing accuracy on the subset of dataset including only TFIDF values which is a good accuracy because of its time and space complexity scales well with the sparse datasets, which is why it is known as popular classification algorithm for sparse datasets.

To summarize the various subsets of data and their accuracy results, the dataset is biased due to which ML models are predicting some categories more than others, and the dataset has considerable number of zeros which is why ML algorithms like MLP doesn't work well with sparse datasets.

5. CONCLUSION

Communication is one of the most essential needs in everyone's life. People must communicate and engage with one another to express themselves. Because of the increased usage of the internet, media and social networking have grown dramatically over the years. Daily, a flood of knowledge emerges through online interaction, as individuals can communicate, express themselves, and voice their opinions. While this situation has the potential to be incredibly productive and improve human life quality, it also has the potential to be destructive and extremely hazardous. The management of social media is responsible for controlling and monitoring these remarks.

Using Random Forest, Multi-layer Perceptron (MLP), and Naive Bayes, this study aims to create a model that can automatically identify a comment as toxic or non-toxic and analyze its results with various subsets of dataset. As a result, the goal of this research is to create a binary classification model to identify the toxicity of a comment and analyze its performance and feature importance on various subsets of dataset. Developing a binary classification model will detect the toxic comments using Random Forest, MLP, and Naive Bayes to train the dataset, and evaluate the model using metrics such as confusion matrix and accuracy by collecting and preprocessing toxicity classified comments for training and testing using term frequency -inverse document frequency (TF-IDF) algorithm.

The Random Forest provided testing accuracy of 82.49% on complete dataset using all identities which is greater than the accuracies of same baseline model on other subsets of dataset because of the availability of all features which are critical to the predictivity of the model. The most key features are named: 500, Unnamed: 28, Unnamed: 41, Unnamed: 86, Unnamed: 63, Unnamed: 735, White, Comment, Black, Christian, and homosexual gay or lesbian, other gender. While this also proves that the dataset is biased, that is why few categories are more predictive than others. To close the performance gap between models on various subsets of data we need to collect relevant data for training and deploy pipelines that will feed data to the model when it is in production. In addition, the subset of dataset including only TFIDF values is also trained on DBSCAN, GMM, and K-Means and the accuracies from these unsupervised learning algorithms are poor mentioned in Table 4.2.1., because of features with more sparse values. which shows that unlabeled data does not increase classification accuracy.

Ethical concerns that may arise are whether data is biased, or not enough data was used and as a result there is prejudicial distinction in the treatment of distinct categories of users. To mitigate such events, the algorithm must be trained on unbiased and complete data which represents every aspect of the online community.

REFERENCES

[1] Tarasova, Z., Khlinovskaya-Rockhill, E. Tuprina, O., Skryabin,V., (2017). Urbanization and the Shifting of Boundaries, Contemporary Transformations in Kinship and Child Circulation among the Sakha. Europe-Asia Studies vol.67, no.7, pp. 1106-1125.Sara Zaheri, Jeff Leath, and David Stroud. Toxic comment classification. SMU Data Science Review, 3(1):13, 2020.

[2] Adamic. L., (2016). The small world web, Research and Advanced Technology for Digital Libraries, pp. 852–852.

[3] Nobata, C., Tetreault, J., Thomas, A. Mehdad, Y., Chang, Y., (2016). Abusive Language Detection in Online User Content. International Conference on World Wide Web, pp. 145-153.

[4] Hanson, R., (2014). Foul play in information markets. George Mason University, vol. 18, no. 2, pp, pp. 107-126.

[5] Waseem, Z., Thorne, J., Bingel, J., (2018). Bridgingthegaps: Multitask Learningfor Domain Transfer of Hate Speech Detection. Online Harassment, pp 29–55.

[6] Kumar, R., Ojha, A., Malmasi, S., Zampieri, M., (2018). Benchmarking Aggression Identification in social media, Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 1-11.

[7] Aggarwal, C., Zhai, C., (2012). A Survey of Text Classification Algorithms. In Mining Text Data. Springer, pp. 163-222.

[8] Georgakopoulos, V., Tasoulis, S., Vrahatis, A., Plagianakos, P., (2018). Convolutional Neural Networks for Toxic Comment Classification, ACM Proceedings of the 10th Hellenic Conference on Artificial Intelligence, pp. 35.

[9] 2 Kevin Khieu and Neha Narwal:” Detecting and Classifying Toxic Comments”, https://web.stanford.edu/class/cs224n/reports/6837517.pdf

[10] 1 Theodora Chu, Kylie Jue and Max Wang:” Comment Abuse Classification with Deep Learning”, https://web.stanford.edu/class/cs224n/reports/2762092.pdf

[11] 9 Manav Kohli, Emily Kuehler and John Palowitch:” Paying attention to toxic comments online”, https://web.stanford.edu/class/cs224n/reports/6856482.pdf