Journal of Engineering Sciences
Vol 13, Issue 02, FEB /2022
ISSN NO:0377-9254

# TOXIC AND NON TOXIC WORD DETECTION

Email:jeevanaaembari@gmail.com
Name:Aembari Jeevana
M.tech-IT
JNTUH college of engineering,nachupally,jagtial
Guide:Asha Jyothi
Email: asha4prathap@gmail.com
Name:Asha Jyothi
Professor of IT department,
JNTUH college of engineering,nachupally, jagtial
Name: Suresh Kumar
HOD of IT department,
JNTUH college of engineering,nachupally, jagtial

## ABSTRACT

The introduction of social media brought about a revolution in the world of digitalization and communication. These platforms were initially developed with a purpose of connecting people across the global boundaries while allowing them to express their views and opinions and learn from others' ideas. With the incoming of the pandemic, the usage of these sites has risen significantly be it by the businesses, educational institutions, students or general public. The increasing ubiquity of social media platforms like Twitter and Facebook has been an issue of major concern since a long time. Along with providing a way for enhanced communication, these platforms also allow internet users to voice their opinions which get circulated among the masses within seconds. Moreover, given the different backgrounds, believes, ethnicity and cultures that the users on these platforms come from, many of them tend to use mean, aggressive and hateful content during their discussions with people not hailing from a background similar to theirs. The amount of hate speech and offensive content has been increasing exponentially. Terms like "profane", "hate", and "offensive" are used interchangeably, and hence these have been classified under a broader category of "Toxic" content. A major part of our dataset focuses on conversations prevailing among the youth. After the preprocessing of this dataset using NLP and bunch of Machine Learning (SVM,RF, KNN, LR & Voting Classifier).

## I. INTRODUCTION

In today’s era of online connections, with the growing prevalence of social media sites like Facebook, Twitter, Instagram, Youtube and Snapchat, more than half of the population of the world seeks to connect and converse through these platforms. This ability of being able to connect with a mass audience by generating and sharing content to interact over large distances, has changed the way these users are involved in public affairs, politics and also with each other. All this has led to provoking violence and amplified the propagation of hateful content. Most of these social media platforms are motivated to draw attention as a part of their business model. Since this offensive content attracts the attention of the masses, it becomes more audible on such platforms. Many a times, it is the inability of people to understand and acknowledge opinions, ideas and views of people hailing from different gender, socio-economic backgrounds and cultures that acts as the driving cause of the spread of this toxicity and hence the hate content targets a specific gender, religion, ethnic group or racial community. Of the population being targeted, adolescents form a major group that are vulnerable to such hatred. Being extremely involved on these platforms, the youth community is a major contributor and receiver of this content. Therefore a major part of our dataset contains tweets circulating among the youth.

As observed through the analysis of our dataset, the prevailing use of various slangs and abbreviations, like “wtf”, “asap”, “tbh”, “idc” etc., emojis to make the content more interactive and expressive, extensive repetitions as a form of emotional emphasis in a statement, like “Hate him sooomuchhh” is used as an indication of extent of despise the sender holds towards the addressee, and the extensive use of punctuations, like “Shut Up You !!!!!”, all of this makes itdifficult for the researchers and the government to study the content and check it for the presence of any toxicity.

## EXISTING SYSTEM:

Manual filtering can be very time consuming with very low accuracy. Automatic systems are thus required to carry out this process in a much more efficient way while saving a lot of time and effort. Recently, there has been a significant development in the fields of ML which provide us a means to analyse the text semantically and make predictions while understanding the content.

## DISADVANTAGES OF EXISTING SYSTEM:

- ❖ It is a manual Process and time Consuming.
- ❖ The propagation of such content is leading to increased violence in matters such as communal riots and lynching globally.

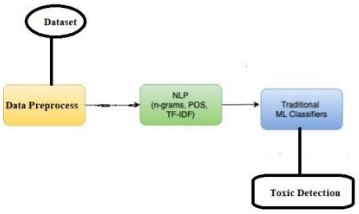
## PROPOSED SYSTEM:

- ❖ In our research work for the purpose of classifying text into either of the three categories: toxic, non-toxic or unclear. As an initial step, we have combined two different datasets to formulate our dataset for this research. Further analysis involving the application of ML classifiers. As a part of the first step, the data is pre-processed before being fed into machine learning algorithms of LR, SVM, KNN,RF and Voting.

## ADVANTAGES OF PROPOSED SYSTEM:

- A major part of our dataset focuses on conversations prevailing among the youth.
- It help to detect toxic using machine learning algorithms.

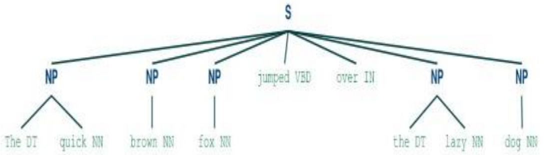
## SYSTEM ARCHITECTURE:



## NLTK

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania.[4] NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook.

NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning.[7] NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems. There are 32 universities in the US and 25 countries using NLTK in their courses. NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.



## Modules:

Preprocessing: Natural Language Processing is basically the interaction between humans and machines where machines detect and analyse large amount of data in the form of text, similar to the way humans do. NLP is used to extract useful information from the text being studied. The NLTK library has been used to work with the WordNet corpus. Further steps of tokenization with an ngram range of (1,3), stopwords removal, stemming, lemmatization, Part-Of-Speech tagging for understanding of grammar have been performed during the data pre-processing. To obtain more accurate results, we have used TF-IDF approach over the traditional BOW approach for creating the count vectorizer.

Machine Learning

Logistic Regression (LR): This is one of the most common algorithms to be used when the dependent variable is categorical, especially in case of binary classification. The core method or the middle of logistic regression is the logistic function, which uses sigmoid function as main entity.

Support Vector Machine (SVM): This algorithm is extensively used in high dimensionality space for classification, regression or outlier detection. It classifies data points by developing a hyperplane in an N-dimensional space (represented by N features).

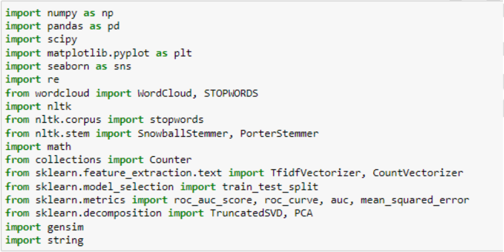
Random Forest (RF): This is a supervised algorithm that can be used for both classification and regression. This algorithm creates a forest of trees with higher number of trees signifying higher accuracy results and reduced overfitting. To calculate the importance of each feature, random forest uses either MDI (mean decrease in impurity) or Gini Importance. In this work, the count vectorizer generated from TF-IDF approach is fed into each of these algorithms and the results are then compared.

**Algorithms:**

i) SVM
ii) Random Forest Classifier
iii) Decision Tree Classifier
iv) KNN classifier
v) Logistic Regression
vi) Voting Classifier( adding Logistic regression, SVM, Decision Tree Classifier)

**SCREENSHOTS**

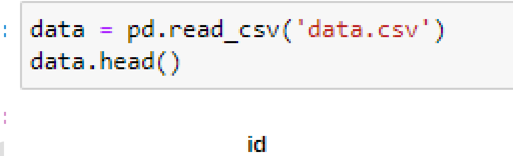
1) Importing Libraries



```
import numpy as np
import pandas as pd
import scipy
import matplotlib.pyplot as plt
import seaborn as sns
import re
from wordcloud import WordCloud, STOPWORDS
import nltk
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer, PorterStemmer
import math
from collections import Counter
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score, roc_curve, auc, mean_squared_error
from sklearn.decomposition import TruncatedSVD, PCA
import gensim
import string
```

( **numpy, pandas** for data pre-processing, **scipy** - SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering, **matplotlib and seaborn** for data visualization part, **re -** A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing), **wordcloud** - visual representations of words that give greater prominence to words that appear more frequently, **nltk** - is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language, **sklearn**– for model building)

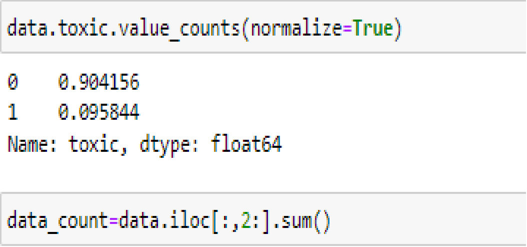
**2) Importing Dataset**



```
data = pd.read_csv('data.csv')
data.head()
```

id

**3) Data Pre-processing**



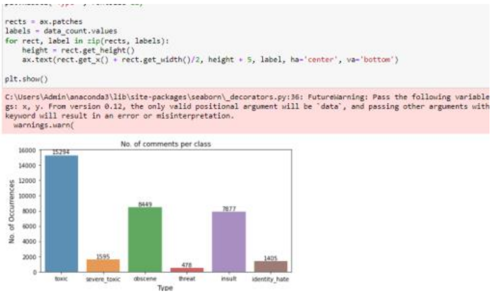
```
data.toxic.value_counts(normalize=True)
```

```
0    0.904156
1    0.095844
Name: toxic, dtype: float64
```

```
data_count=data.iloc[:,2:].sum()
```

(Calculating the toxic values)

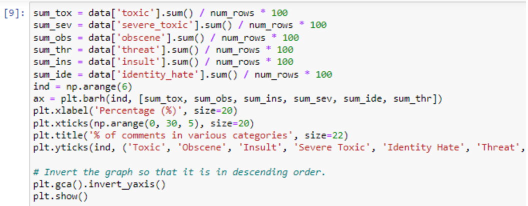
**4) Data Visualization**



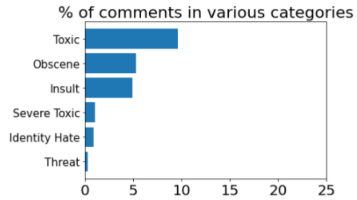
```
rects = ax.patches
labels = data_count.values
for rect, label in zip(rects, labels):
    height = rect.get_height()
    ax.text(rect.get_x() + rect.get_width()/2, height + 5, label, ha='center', va='bottom')

plt.show()
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments with
keyword will result in an error or misinterpretation.
  warnings.warn(
```

(Representing the different comments per class).

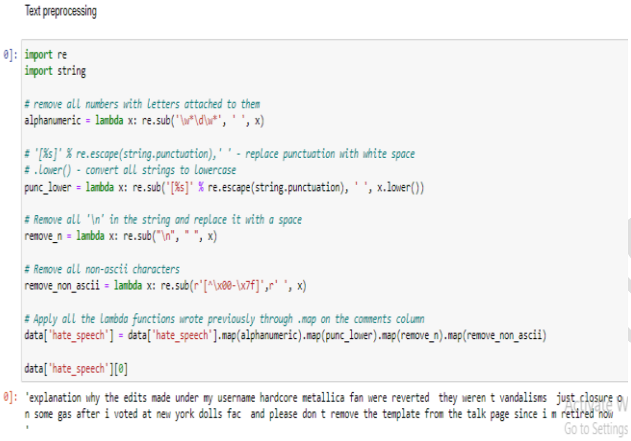


```
[9]: sum_tox = data['toxic'].sum() / num_rows * 100
     sum_sev = data['severe_toxic'].sum() / num_rows * 100
     sum_obs = data['obscene'].sum() / num_rows * 100
     sum_thr = data['threat'].sum() / num_rows * 100
     sum_ins = data['insult'].sum() / num_rows * 100
     sum_ide = data['identity_hate'].sum() / num_rows * 100
     ind = np.arange(6)
     ax = plt.barh(ind, [sum_tox, sum_obs, sum_ins, sum_sev, sum_ide, sum_thr])
     plt.xlabel('Percentage (%)', size=20)
     plt.xticks(np.arange(0, 30, 5), size=20)
     plt.title('% of comments in various categories', size=22)
     plt.yticks(ind, ('Toxic', 'Obscene', 'Insult', 'Severe Toxic', 'Identity Hate', 'Threat',

     # Invert the graph so that it is in descending order.
     plt.gca().invert_yaxis()
     plt.show()
```



(Representing the comments in various categories).

## 5) Text Pre-processing

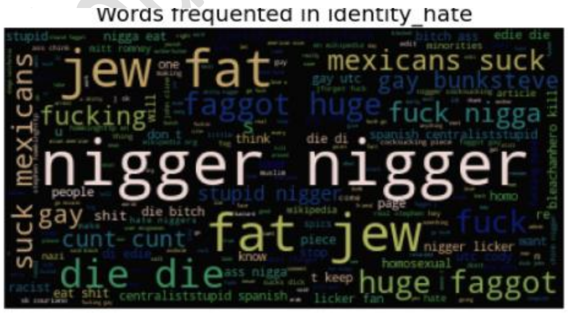


Text preprocessing

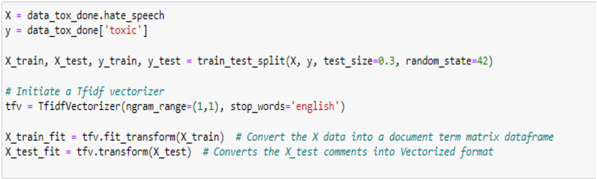
```
0]: import re
    import string

    # remove all numbers with letters attached to them
    alphanumeric = lambda x: re.sub('\w*\d\w*', ' ', x)

    # '[%s]' % re.escape(string.punctuation),' ' - replace punctuation with white space
    # .lower() - convert all strings to lowercase
    punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ', x.lower())

    # Remove all '\n' in the string and replace it with a space
    remove_n = lambda x: re.sub("\n", " ", x)

    # Remove all non-ascii characters
    remove_non_ascii = lambda x: re.sub(r'[^\x00-\x7f]',r' ', x)

    # Apply all the lambda functions wrote previously through .map on the comments column
    data['hate_speech'] = data['hate_speech'].map(alphanumeric).map(punc_lower).map(remove_n).map(remove_non_ascii)

    data['hate_speech'][0]

0]: 'explanation why the edits made under my username hardcore metallica fan were reverted  they weren t vandalisms  just closure o
    n some gas after i voted at new york dolls fac  and please don t remove the template from the talk page since i m retired now
    '
```

(Remove all the numbers, convert the strings to lower case, remove all '\n', and remove all the non-ascii characters)

## 6) Representing the most hate words



## 7) Assigning values to x and y, splitting the dataset into test and train



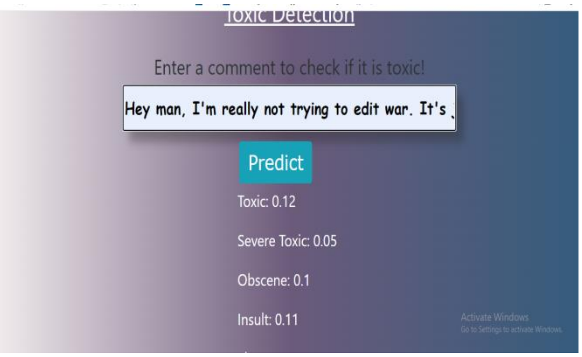
```
X = data_tox_done.hate_speech
y = data_tox_done['toxic']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initiate a Tfidf vectorizer
tfv = TfidfVectorizer(ngram_range=(1,1), stop_words='english')

X_train_fit = tfv.fit_transform(X_train)  # Convert the X data into a document term matrix dataframe
X_test_fit = tfv.transform(X_test)  # Converts the X_test comments into Vectorized format
```

Assigning hate_speech to x and toxic to y
Training – 70% and testing – 30%.

## 8) Applying the machine learning algorithms

vii) SVM
viii) Random Forest Classifier
ix) Decision Tree Classifier
x) Knn classifier
xi) Logistic Regression
xii) Voting Classifier( adding Logistic regression, SVM, Decision Tree Classifier)

## 9) Model Comparison



Toxic Detection

Enter a comment to check if it is toxic!

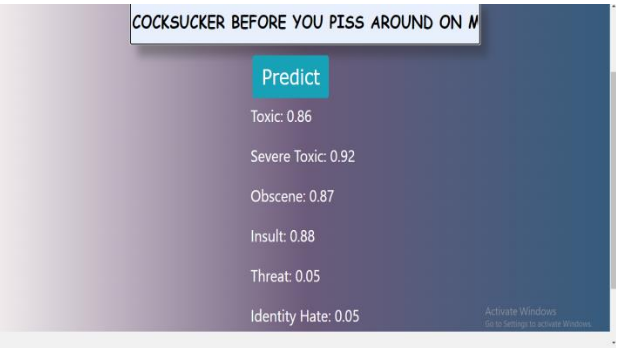
Hey man, I'm really not trying to edit war. It's

Toxic: 0.12

Severe Toxic: 0.05

Obscene: 0.1

Insult: 0.11



COCKSUCKER BEFORE YOU PISS AROUND ON M

Toxic: 0.86

Severe Toxic: 0.92

Obscene: 0.87

Insult: 0.88

Threat: 0.05

Identity Hate: 0.05

## CONCLUSION

Alone is a dataset entirely based on youths' toxic conversations on Twitter. We have performed data pre-processing followed by various machine learning and ensemble algorithms using TF-IDF, POS tagging and trigrams approach in which LR and SVM performed the best for us.

## FUTURE ENHANCEMENT

For future work, there are many models and combination of deep learning paradigms that we would like to test and explore such as combining convolutional and recurrent neural networks, as well as state-of-the-art SVM models and feature extraction mechanisms used for toxic speech detection. We think that the cascading model shows promise. While we only implemented a simple version, it would be interesting to further optimize the criteria to determine the confidence of the baseline and to improve both the efficiency and accuracy of the model.

## REFERENCES

[1] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” arXiv, no. Icwsm, pp. 512–515, 2017.

[2] F. Del Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Tesconi, “Hate me, hate me not: Hate speech detection on Facebook,” CEUR Workshop Proc., vol. 1816, pp. 86–95, 2017.

[3] P. L. Teh, C. Bin Cheng, and W. M. Chee, “Identifying and categorising profane words in hate speech,” ACM Int. Conf. Proceeding Ser., pp. 65–69, 2018.

[4] A. Mittos, S. Zannettou, J. Blackburn, and E. De Cristofaro, “‘And we will fight for our race!’ a measurement study of genetic testing conversations on reddit and 4chan,” Proc. 14th Int. AAAI Conf. Web Soc. Media, ICWSM 2020, no. Icwsm, pp. 452–463, 2020.

[5] A. Olteanu, C. Castillo, J. Boy, and K. R. Varshney, “The effect of extremist violence on hateful speech online,” arXiv, no. Icwsm, pp. 221–230, 2018. [6] P. Fortuna and S. Nunes, “A survey on automaticdetection of hate speech in text,” ACM Comput. Surv., vol. 51, no. 4, 2018.

[7] Z. Waseem, J. Thorne, and J. Bingel, “Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection,” Springer International Publishing, 2018.

[8] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, “Detecting hate speech and offensive language on twitter using machine learning: An N-gram and TFIDF based approach,” arXiv, 2018.

[9] Z. Zhang, D. Robinson, and J. Tepper, “Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network,” vol. 10843 LNCS. Springer International Publishing, 2018.

[10] T. Ranasinghe, M. Zampieri, and H. Hettiarachchi, “BRUMS at HASOC 2019: Deep learning models for multilingual hate speech and offensive language identification,” CEUR Workshop Proc., vol. 2517, pp. 199–207, 2019.