**CSCE 5290: Natural Language Processing**

**Increment 1**

**Fight Online Abuse**

**Toxic Comment Classification**

**Team Members**

1. Member

2. Member

3. Member

4. Member

( *Github :* [*github.com*](file:///E:\freelancer\rohith\github.com)*/insert\_your\_link\_here* )

**1. Goals and Objectives**

**1.1 Motivation**

Internet is a wide source of communication globally where people freely express their thoughts, feelings and opinions. While this situation contributes significantly, it involves great dangers since these written texts can be highly toxic causing offense, abuse, harassment and bullying behaviors. Online sites struggle to promote discussions effectively, leading many communities to limit or shut down user comments altogether. By analysing the text and restricting users to post toxic comments, we could promote safer discussions on various online portals.

**1.2 Significance**

Toxic comments passed by a user on the internet portals are offensive, rude and disrespectful. It may also take an extensive verge, where the users send threatening, degrading, or sexually explicit messages. As a result, detecting and reporting such toxic comments from online portals is crucial. Analyzing user comments over the internet is crucial for maintaining civility in online communities.

**1.3 Objectives**

We aim to create a machine learning model which detects the level of toxicity a piece of text contains. This model will be used for fighting online abuse making online communities grow better and healthier in user discussions and conversations openly. We hence intend to help online communities detect and restrict toxic comments which any user tries to post.

**1.4 Features**

The types of toxicity the text can contain are toxic, severe toxic, obscene, threat, insult and identity hate. We propose to develop a machine learning model which uses natural language processing techniques for detecting the type of toxicity a comment contains. This model provides, as a result the probability of each type of toxicity for a given comment.

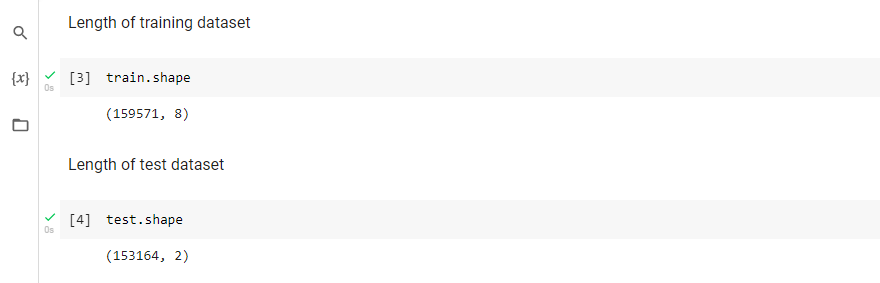
**2. Related Work (Background)**

In the age of Internet, people are free to express their thoughts in public portals. This is great in a way, but it can also take an extensive verge when people start posting toxic comments. Such comments can be rude, disrespectful, hurting and insulting. In order to remove such comments upon posting, several notable systems have been developed. This helps keep the community healthier in communication and maintaining civility in the same. We intend to create such a platform which analyses comments posted by the users and rate the nature of toxicity present in the comments. This shall help remove comments that are toxic and form a better community online.

**3. Dataset**

The dataset contains Wikipedia comments which have been rated by humans for toxic behaviour. The dataset contains two separate files, train.csv and test.csv. The train.csv file contains the training dataset which include comments with their corresponding binary labels. The test.csv file contains the test dataset also including comments whose toxicity probabilities are to be predicted.

The training dataset contains 159571 comments, whereas the test dataset contains 153164 comments.



**4. Detail design of Features**

The training dataset consists of the following features:

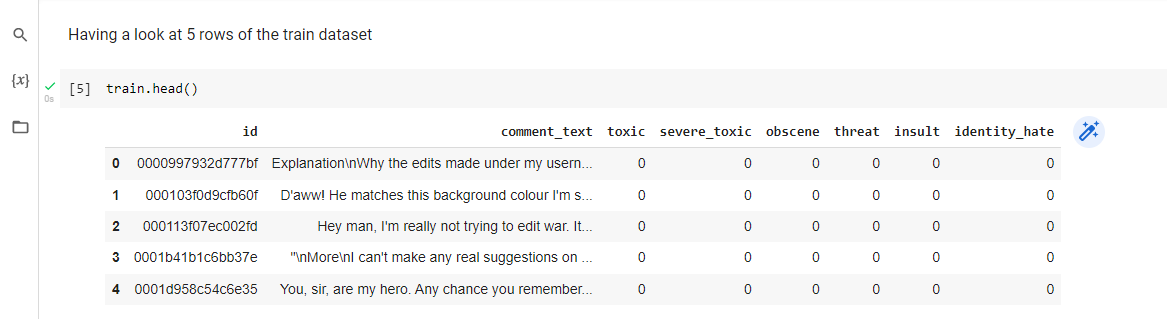
1. id
2. comment text
3. toxic
4. severe toxic
5. obscene
6. threat
7. insult
8. identity hate

The id is a unique set of characters for every comment. It is the primary key identifier of the row it belongs to. The comment text is a Wikipedia comment that contains strings of words and characters. The other six features are types of toxicity. These features are labelled in a binary format where ‘0’ means the comment is not toxic and ‘1’ implies that the comment is toxic.

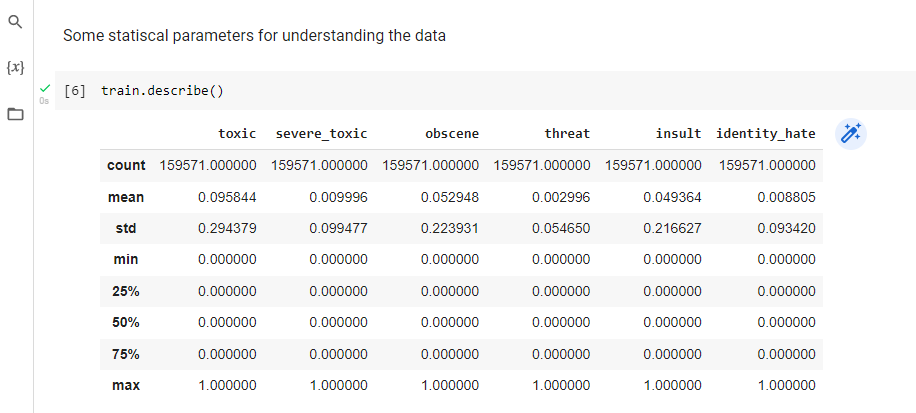
The test dataset consists of id and comment text where the probability of each type of toxicity is to be predicted.

**5. Analysis**

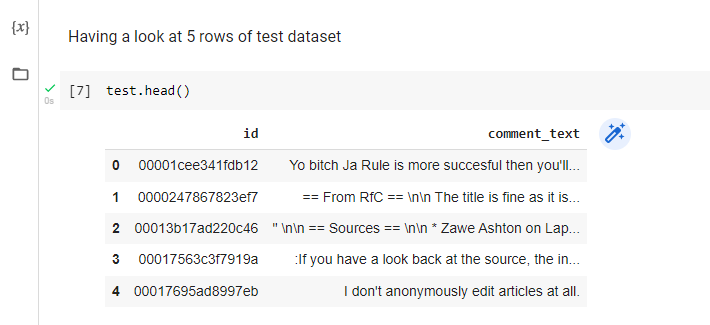
The first step of the analysis is to have a look at the data. The code output below shows the first five rows of the train dataset.



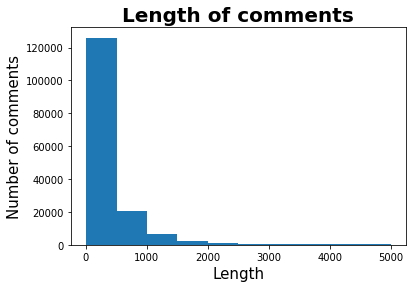
The features named toxic, severe toxic, obscene, threat, insult and identity hate are of numerical data-type. Some statistical measures such as mean, standard deviation, min and max of these features are described below.



For analysing the features of test dataset, the following code output shows first five rows of test dataset.

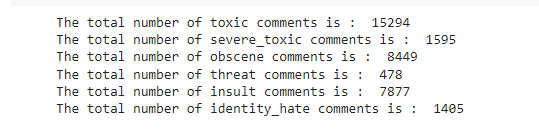


Since comments are of type strings, these can vary in length largely. The following plot shows the histogram of the length of comments.

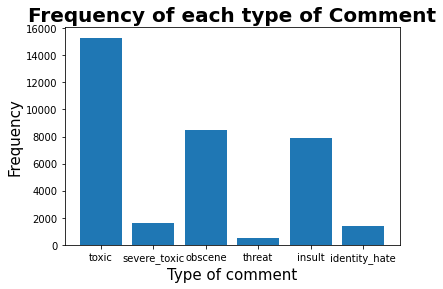


Many of the comments are no longer than 1000 characters and only a few of them have more than 2000 characters.

The number of comments of each type is given below.



Plots and diagrams always have a better readability over numbers in general. To have a better understanding and readability of the frequency of types of comments, the following bar plot shows the frequency of labels that appear in text comments.



It is observed that there are more toxic comments and very few threat comments.

An example for each type of comment is below.

*Toxic*:

Hey... what is it..

@ | talk .

What is it... an exclusive group of some WP TALIBANS...who are good at destroying, self-appointed purist who GANG UP any one who asks them questions abt their ANTI-SOCIAL and DESTRUCTIVE (non)-contribution at WP?

Ask Sityush to clean up his behavior than issue me nonsensical warnings...

*Severe Toxic*:

Stupid peace of shit stop deleting my stuff asshole go die and fall in a hole go to hell!

*Obscene*:

You are gay or antisemmitian?

Archangel WHite Tiger

Meow! Greetingshhh!

Uh, there are two ways, why you do erased my comment about WW2, that holocaust was brutally slaying of Jews and not gays/Gypsys/Slavs/anyone...

1 - If you are anti-semitian, than shave your head bald and go to the skinhead meetings!

2 - If you doubt words of the Bible, that homosexuality is a deadly sin, make a pentagram tatoo on your forehead go to the satanistic masses with your gay pals!

3 - First and last warning, you fucking gay - I won't appreciate if any more nazi shwain would write in my page! I don't wish to talk to you anymore!

Beware of the Dark Side!

*Threat*:

Hi! I am back again!

Last warning!

Stop undoing my edits or die!

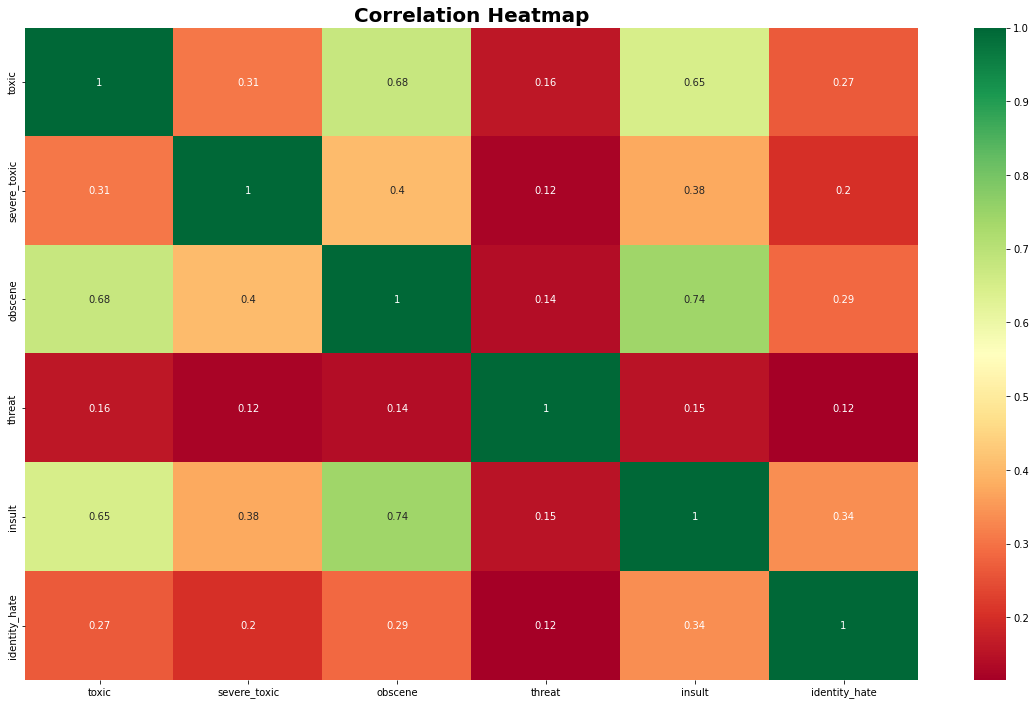
*Insult*:

COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK

*Identity hate*:

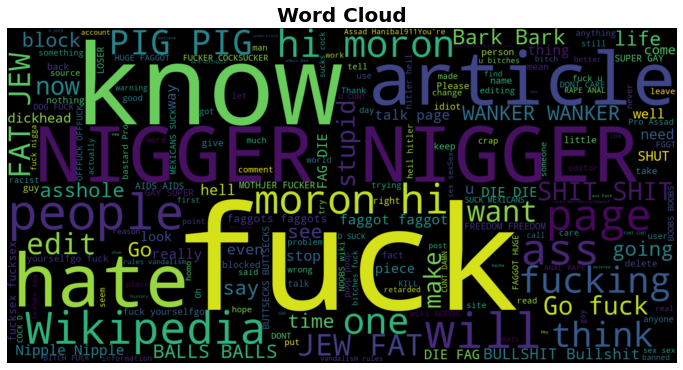
A pair of jew-hating weiner nazi schmucks.

The binary features of this dataset can have a correlation amongst them. To get the extent of correlation between the features, a correlation heat map is shown below which shows which labels are likely to appear together with a comment.



The highest correlation co-efficient in the matrix is 0.74 between the features obscene and insult which shows that there is a high probability of obscene comments to be insulting.

To have a look at the words that contribute the most in toxic comments, a word cloud of toxic comments is depicted below.



**6. Implementation**

The implementation includes data analysis necessary for further modelling of the data. The code is shown below.

Importing the libraries

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from wordcloud import WordCloud

Reading train and test csv files

train = pd.read\_csv('train.csv')

test = pd.read\_csv('test.csv')

Length of training dataset

train.shape

Length of test dataset

test.shape

Having a look at 5 rows of the train dataset

train.head()

Some statiscal parameters for understanding the data

train.describe()

Having a look at 5 rows of test dataset

test.head()

Length of comments

length\_of\_comments = train.comment\_text.str.len()

plt.figure()

plt.title('Length of comments', fontsize=20, fontweight='bold')

plt.xlabel('Length', fontsize=15)

plt.ylabel('Number of comments', fontsize=15)

plt.hist(length\_of\_comments)

plt.plot()

plt.show()

Frequency of each type of comment

print("The total number of toxic comments is : ", round(train['toxic'].sum()))

print("The total number of severe\_toxic comments is : ", round(train['severe\_toxic'].sum()))

print("The total number of obscene comments is : ", round(train['obscene'].sum()))

print("The total number of threat comments is : ", round(train['threat'].sum()))

print("The total number of insult comments is : ", round(train['insult'].sum()))

print("The total number of identity\_hate comments is : ", round(train['identity\_hate'].sum()))

Plot of frequency of each type of comment

labels\_name = ['toxic','severe\_toxic','obscene','threat','insult','identity\_hate']

labels = train[labels\_name]

label\_sum = labels.sum()

plt.figure()

plt.title('Frequency of each type of Comment', fontsize=20, fontweight='bold')

plt.xlabel('Type of comment', fontsize=15)

plt.ylabel('Frequency', fontsize=15)

plt.bar(labels\_name,label\_sum)

plt.plot()

plt.show()

Examples of each type of comments

print('Toxic:\n' + train[train.toxic == 1].iloc[1, 1])

print('Severe Toxic:\n' + train[train.severe\_toxic == 1].iloc[1, 1])

print('Obscene:\n' + train[train.obscene == 1].iloc[1, 1])

print('Threat:\n' + train[train.threat == 1].iloc[1, 1])

print('Insult:\n' + train[train.insult == 1].iloc[1, 1])

print('Identity hate:\n' + train[train.identity\_hate == 1].iloc[1, 1])

Heat Map to show the correlation between the types of comments

plt.figure(figsize=(20,12))

sns.heatmap(train.corr().abs(), annot=True, cmap='RdYlGn')

plt.title('Correlation Heatmap',fontsize=20, fontweight='bold')

plt.show()

Word Cloud

text = train[train.toxic == 1].comment\_text

wordcloud\_text = pd.Series(text).str.cat(sep=' ')

wordcloud = WordCloud(width=1600, height=800).generate(wordcloud\_text)

plt.figure(figsize=(12, 8))

plt.title('Word Cloud',fontsize=20, fontweight='bold')

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis("off")

plt.show()

**7. Preliminary Results**

The analysis of all the parameters has been carried out and the results throw some insights on the data.

* There are very few toxic (all types) comments, most of the text comments are non-toxic.
* Very few comments go beyond 2000 characters in length. Most of them fall in the range of 500 characters.
* Toxic comments are highest in number followed by obscene, insult, severe toxic, identity hate and threat.
* An obscene comment is highly likely to be an insulting comment.
* Some words appear frequently in toxic comments [shown in image].

**8. Project Management**

**8.1. Work completed**

**8.1.1. Description**

This increment contains analysing and interpreting the data, and its implementation. This includes studying the train and test data, data description, plotting graphs to analyse the data, finding the correlation between the features of the data. A deep analysis of the data is carried to better understand the data.

**8.1.2. Responsibility (Task, Person)**

Member 1

* Studying the complete dataset
* Analysing the text comment feature
* Understanding and Implementing Correlation Heat Map

Member 2:

* Studying the complete dataset
* Analysing the text comment feature
* Understanding and Implementing Correlation Heat Map

Member 3:

* Studying the complete dataset
* Analysing the types of toxic comments features
* Understanding and Implementing Word Cloud

Member 4:

* Studying the complete dataset
* Analysing the types of toxic comments features
* Understanding and Implementing Word Cloud

**8.1.3. Contributions (Members / Percentage)**

Member 1: 25%

Member 2: 25%

Member 3: 25%

Member 4: 25%

**8.2. Work to be completed**

**8.2.1. Description**

The following things are to be implemented. Firstly, the text comments are to be pre-processed which include, breaking down the text comments into words, removing special characters, etc. Next, we aim to create a few models and evaluate them using some evaluation metrics to see which model best fits our data. Then the test data is to be predicted by the best chosen model which predicts the probability of each type of toxic comments. Next, we tune the hyper-parameters of the model.

**8.2.2. Responsibility (Task, Person)**

Member 1: Pre-processing the data and creating a model.

Member 2: Creating models for the data.

Member 3: Evaluation of model.

Member 4: Hyper-parameter tuning.

**8.2.3. Issues / Concerns**

None

**References**

1. Navoneel Chakrabarty, “A Machine Learning approach to comment classification”, <https://arxiv.org/ftp/arxiv/papers/1903/1903.06765.pdf>
2. Iarjset Jornal, “Toxic comment classification using neural networks and machine learning”, <https://iarjset.com/papers/toxic-comment-classification-using-neural-networks-and-machine-learning/>
3. Darko Androcec, “Machine Learning methods for toxic comment classification: A systematic review”, <https://www.researchgate.net/publication/349929587_Machine_learning_methods_for_toxic_comment_classification_a_systematic_review>
4. Kaggle dataset, <https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge>
5. Matplotlib.pyplot, <https://matplotlib.org/stable/api/pyplot_summary.html>
6. Pandas, <https://pandas.pydata.org/docs/user_guide/index.html>
7. Seaborn, <https://seaborn.pydata.org/api.html>
8. Wordcloud, <https://pypi.org/project/wordcloud/>