TOXICITY OF WIKIPEDIA COMMENTS - MILESTONE REPORT

-Disclaimer: the dataset for this competition contains text that may be considered profane, vulgar, or offensive.

Introduction:

Wikipedia is a multilingual online encyclopedia based on open collaboration through a model of content edited by web-based applications like web browsers. It is the largest and most popular general reference work on the World Wide Web.

Open collaboration has its own difficulties since some authors insert contents that are toxic, which means they have improper language or are reflecting hatred towards a group of people etc. Moreover, the software that powers Wikipedia provides certain tools allowing anyone to review changes made by others, drawing criticism from opposing groups, which in some cases lead to vulgar language and vandalism.

Editors can enforce these rules by deleting or modifying non-compliant material. However, it is more practical to have computer programs to perform anti-toxic duties, blocking or deleting entries of such properties.

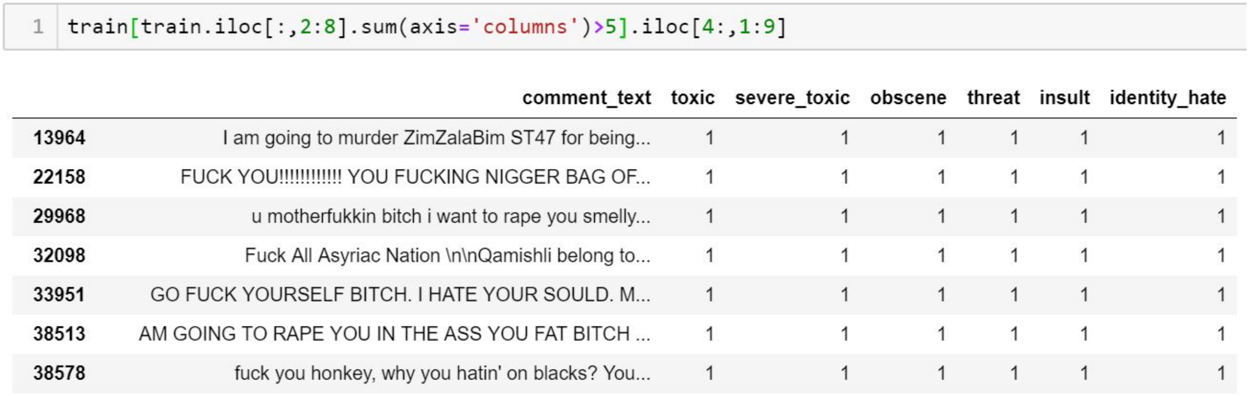
A model, utilizing neural networks (NN) , preferably Recursive NN, that’s capable of detecting different types of of toxicity like threats, obscenity, insults, and identity-based hate will be built and used to block improper comments.

Dataset:

Dataset consists of comments entered by authors word wide about various topics. In this project the aim is to build a multi-headed model that’s capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate. A successful model will hopefully help online discussion become more productive and respectful.

The number of observations in the dataset are more than 150 thousand, each including a document under column name ‘comment_text’, length of which ranges from just a couple of words to a few paragraph entry.

The nature of the label set, consisting of 6 columns brings challenge to this dataset because a comment may be both toxic and obscene, it may have elements of threat and identity hate - and even in some cases all six flags may be raised.



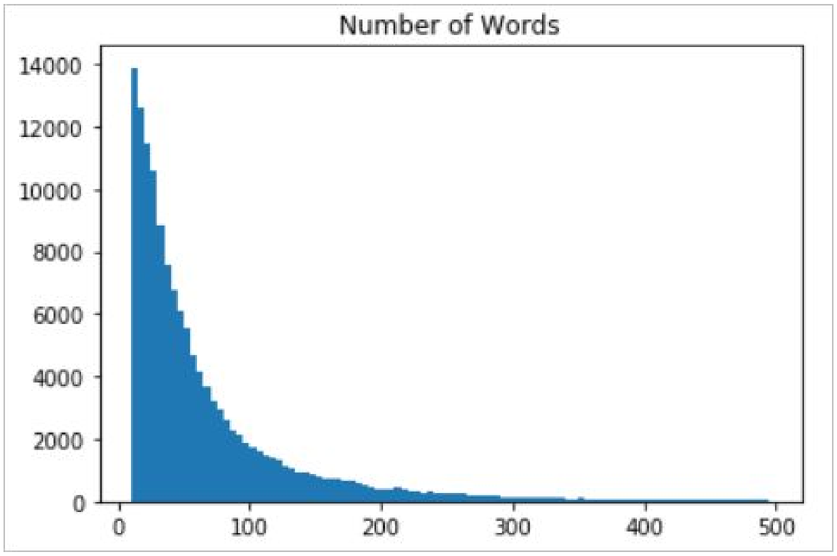
```
train[train.iloc[:,2:8].sum(axis='columns')>5].iloc[4:,1:9]
```

	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
13964	I am going to murder ZimZalaBim ST47 for being...	1	1	1	1	1	1
22158	FUCK YOU!!!!!!!!!!!! YOU FUCKING NIGGER BAG OF...	1	1	1	1	1	1
29968	u motherfukkin bitch i want to rape you smelly...	1	1	1	1	1	1
32098	Fuck All Asyriac Nation \n\nQamishli belong to...	1	1	1	1	1	1
33951	GO FUCK YOURSELF BITCH. I HATE YOUR SOULD. M...	1	1	1	1	1	1
38513	AM GOING TO RAPE YOU IN THE ASS YOU FAT BITCH ...	1	1	1	1	1	1
38578	fuck you honkey, why you hatin' on blacks? You...	1	1	1	1	1	1

This is a classification problem, but it comes with its slight variations since a comment can be labelled with more than one toxicity flag. Any specific comment may be represented with more than one label.

The aforementioned problem can be solved with different approaches, depending on the model used to solve the problem. For a baseline Multinomial Naive Bayes, model has to be run for each class label. However, on the other hand, with utilization of a neural network, output layer nodes can be adjusted to represent all class labels at once. Both methods were used for this study.

Data Exploration:

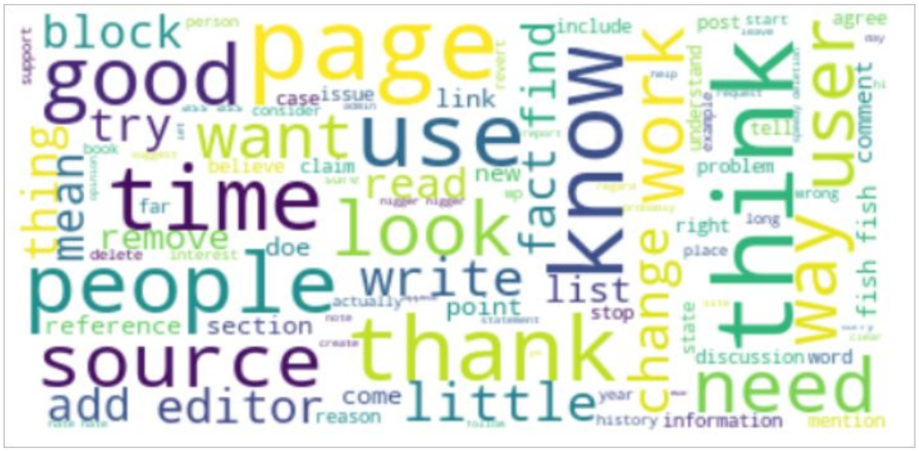


The 159’571 documents were joined into a corpus for analysis. The number of words in the corpus were 7’153’449 and number of unique words were 210’337.

Distribution of the number of words per document is represented by histogram on the left.

Most common 50 words, after text preprocessing were:
[('article', 8850), ('page', 7312), ('wikipedia', 5399), ('edit', 4893), ('talk', 4702), ('use', 4317), ('like', 3591), ('think', 3022), ('know', 2995), ('source', 2930), ('good', 2810), ('time', 2404), ('add', 2316), ('people', 2236), ('user', 2234), ('want', 1937), ('block', 1911), ('need', 1891), ('image', 1823), ('find', 1746), ('delete', 1746), ('link', 1728), ('look', 1715), ('work', 1714), ('remove', 1711), ('thank', 1627), ('information', 1602), ('write', 1582), ('fuck', 1541), ('change', 1512), ('way', 1504), ('little', 1489), ('comment', 1481), ('editor', 1457), ('thing', 1446), ('section', 1441), ('list', 1415), ('point', 1409), ('deletion', 1406), ('fact', 1384), ('try', 1378), ('help', 1377), ('thanks', 1375), ('read', 1360), ('doe', 1342), ('question', 1336), ('new', 1317), ('mean', 1316), ('wp', 1312), ('right', 1253)]

The unprocessed corpus, once input to a frequency based word cloud, generated the words represented by the graph on the right. Note that only the stop words were removed for the word cloud.



Data Cleaning / Text Pre-processing:

Text pre-processing is a very important stage of data cleaning and the methods chosen, their order and detail level has significant impact on the quality of the classifier. Since the corpus consists of more than 7 millions of words, pre-processing is time consuming and sometimes computationally expensive.

First of all, URL addresses were removed by Python built-in library Regex. Then accented characters were normalized into ASCII characters, which really comes into play when dealing with non-English alphabets. This also prevents some of the flagged words to go unnoticed, since sometimes collaborators try to cloak swear words by making minor alterations, such as changing an “e” to “é”.

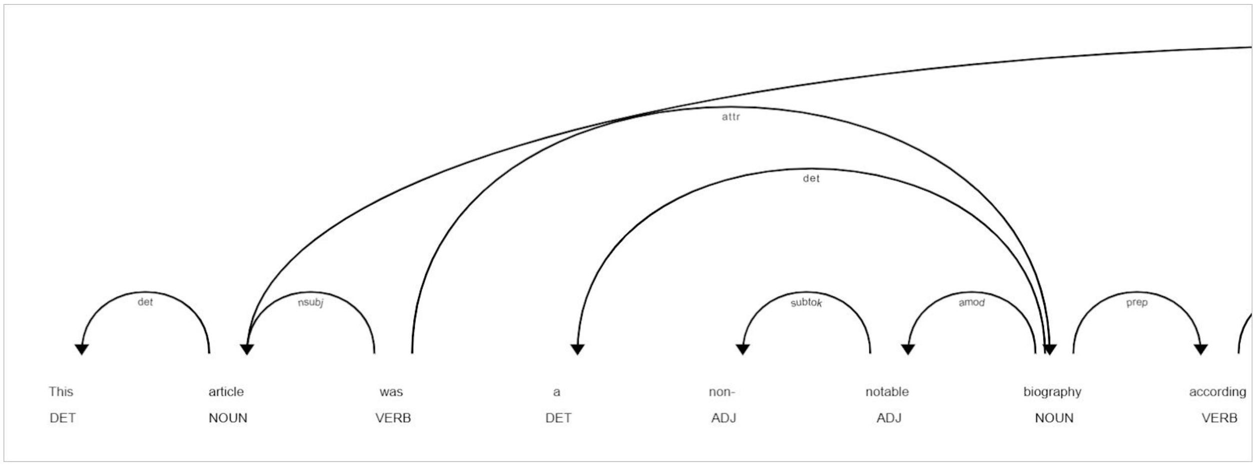
Next, contractions are expanded via a dictionary. Contractions are shortened versions of words or syllables. These shortened versions of existing words or phrases are created by removing specific letters and sounds. Examples would be, do not to don’t and I would to I’d. Converting each contraction to its expanded, original form often helps with text standardization.

Special characters including non-English alphabet and symbols were regarded as noise and removed with regular expressions code.

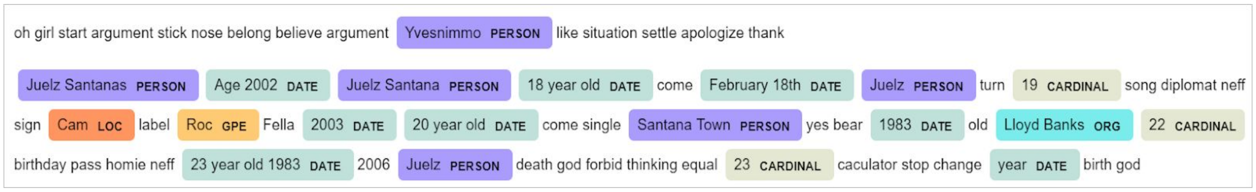
Next step in text preprocessing was lemmatization, where we remove word affixes to get to the base form of a word, which is always lexicographically correct. Lemmatization is computationally cumbersome compared to stemming, but it was preferred over stemming since it is more precise.

Later stop words were removed together with extra white spaces and text was converted to lowercase.

Spacy, a free open-source library for Natural Language Processing, was used in this project since it is one of the most sophisticated libraries and has great documentation. Spacy features NER, POS tagging, dependency parsing, word vectors.



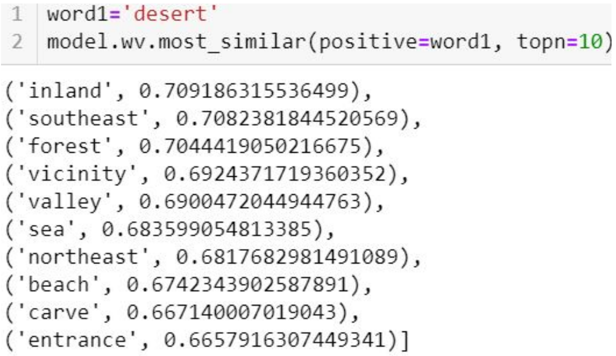
Two pre trained statistical English language models were used. The first model was ‘en_core_web_lg’, which supports NER and POS features, the other one was 'en_vectors_web_lg'. The latter model does not support NER or POS, but has 1070971 unique vectors (300 dimensions) trained with GloVe.


oh girl start argument stick nose belong believe argument Yvesnimmo PERSON like situation settle apologize thank

Juelz Santanas PERSON Age 2002 DATE Juelz Santana PERSON 18 year old DATE come February 18th DATE Juelz PERSON turn 19 CARDINAL song diplomat neff sign Cam LOC label Roc GPE Fella 2003 DATE 20 year old DATE come single Santana Town PERSON yes bear 1983 DATE old Lloyd Banks ORG 22 CARDINAL birthday pass homie neff 23 year old 1983 DATE 2006 Juelz PERSON death god forbid thinking equal 23 CARDINAL caculator stop change year DATE birth god

Non English comments:

This is an area of improvement since there are many comments that are not English. There are some libraries that can detect non-English words, however, this problem was not addressed at this time and left as provisional room for improvement. This also warrants a revisit to the filtering stage if the language uses a non-Latin alphabet, since such letters are filtered out.

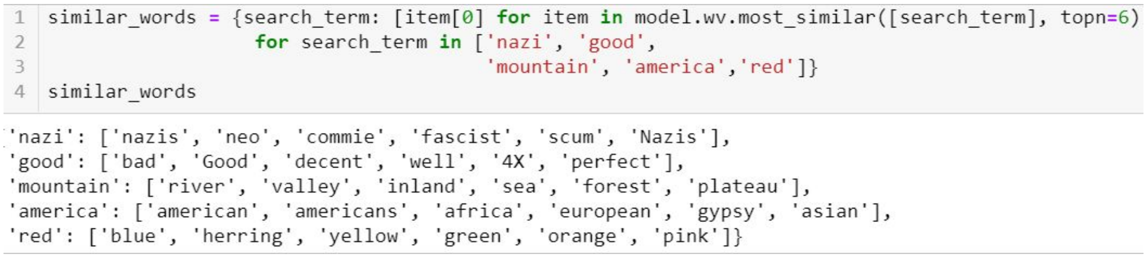


```
word1='desert'
model.wv.most_similar(positive=word1, topn=10)
```

```
('inland', 0.709186315536499),
('southeast', 0.7082381844520569),
('forest', 0.7044419050216675),
('vicinity', 0.6924371719360352),
('valley', 0.6900472044944763),
('sea', 0.683599054813385),
('northeast', 0.6817682981491089),
('beach', 0.6742343902587891),
('carve', 0.667140007019043),
('entrance', 0.6657916307449341)]
```

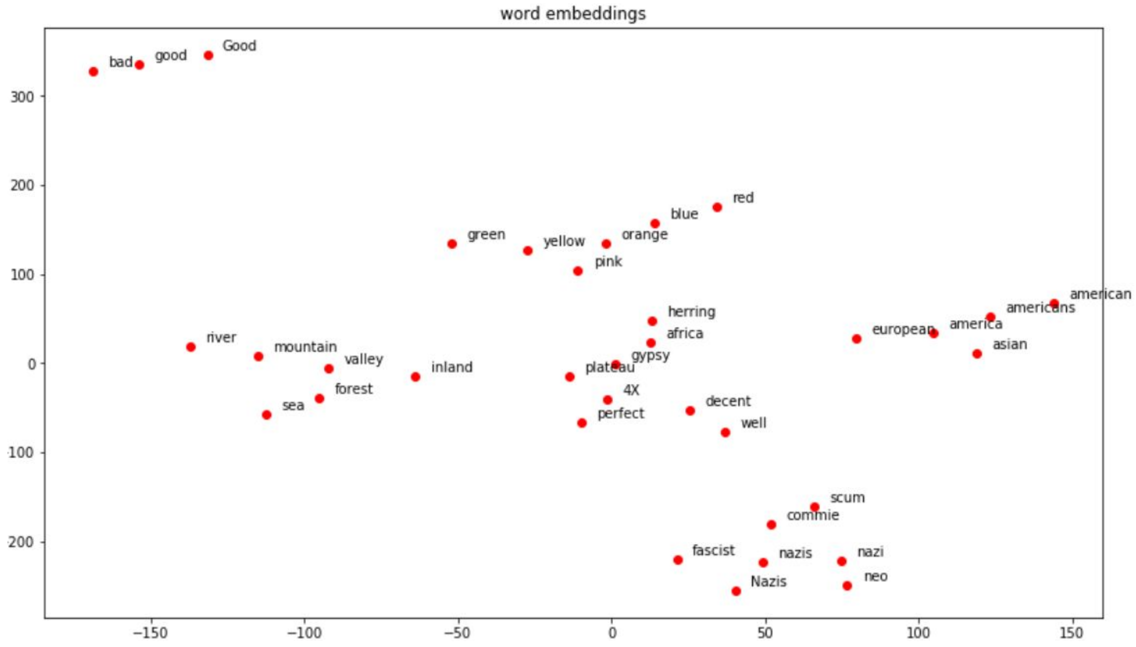
Similarities:

The pre-trained vectors of spacy did not work well for similarities since language of comment texts is not daily used language. After training own word2Vec vectors with Gensim, the similarity results were quite good. On the most similar words to selected word can be seen. For visualization of similarities, dimension reduction is required. T-distributed Stochastic Neighbor Embedding (t-SNE) algorithm for visualization was used for embedding high-dimensional data for visualization in a low-dimensional space of two dimensions. t-SNE models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability.



```
similar_words = {search_term: [item[0] for item in model.wv.most_similar([search_term], topn=6)
                    for search_term in ['nazi', 'good',
                                        'mountain', 'america','red']}
similar_words
```

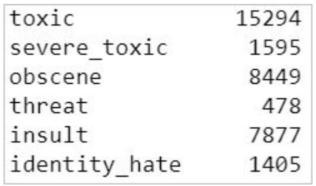
```
{'nazi': ['nazis', 'neo', 'commie', 'fascist', 'scum', 'Nazis'],
 'good': ['bad', 'Good', 'decent', 'well', '4X', 'perfect'],
 'mountain': ['river', 'valley', 'inland', 'sea', 'forest', 'plateau'],
 'america': ['american', 'americans', 'africa', 'european', 'gypsy', 'asian'],
 'red': ['blue', 'herring', 'yellow', 'green', 'orange', 'pink']}
```



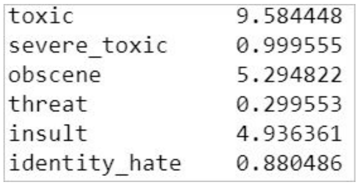
Baseline Models:

After text pre-processing; Multinomial Naive Bayes, Logistic regression and Support Vector Classification models were used to classify each comment for each of the six classes.

The biggest problem for approaching a dataset like this is class imbalance, which gives incorrect results. Accuracy terms higher than 0.995 were achieved, however, when we look at precision and/or recall we can see that actual results are not promising.



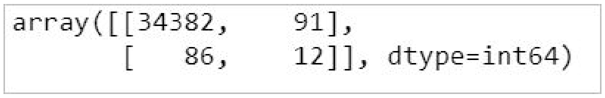
```
toxic            15294
severe_toxic      1595
obscene           8449
threat             478
insult            7877
identity_hate     1405
```



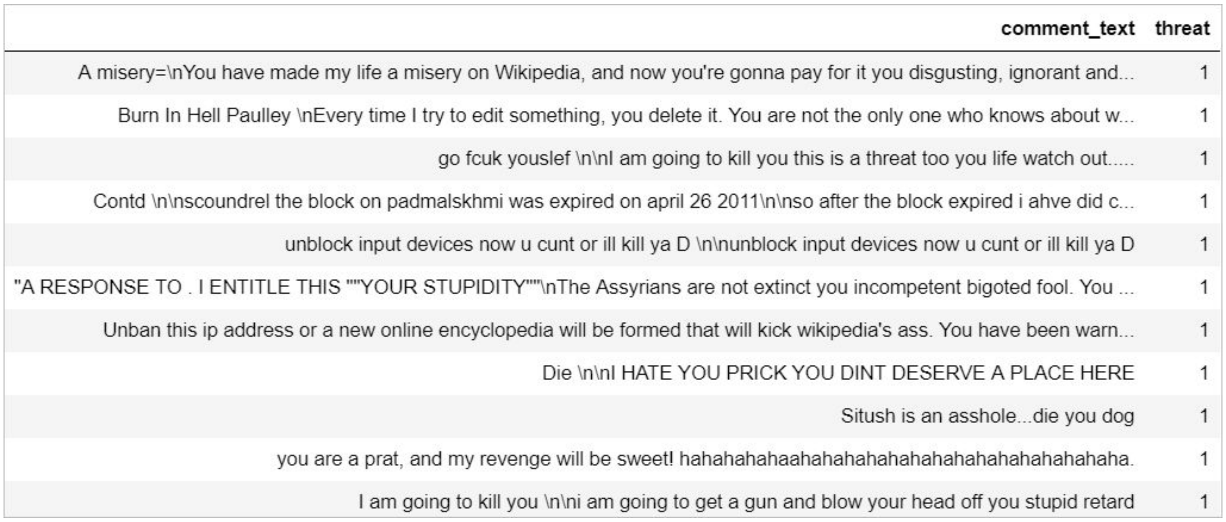
```
toxic            9.584448
severe_toxic     0.999555
obscene          5.294822
threat           0.299553
insult           4.936361
identity_hate    0.880486
```

Distribution of classes as depicted on charts on the left, are for the dataset size of 159 thousand. On the far left are number of “True” values for each feature, and to its right are the relating percentages of each variable. Weight imbalance was reduced by utilizing class weight and sample weight features of scikit learn.

Each model must be run 6 times, one for each class label. The validation accuracy score, 0.9948, looks interestingly high for “threat” label, but when we look at the confusion matrix we can see that the score is not what it appears like. The confusion matrix below clearly shows the “inaccuracy” of the accuracy calculation, and points out why other score functions such as precision, recall or roc auc should be used. Below are false negative results for threat.

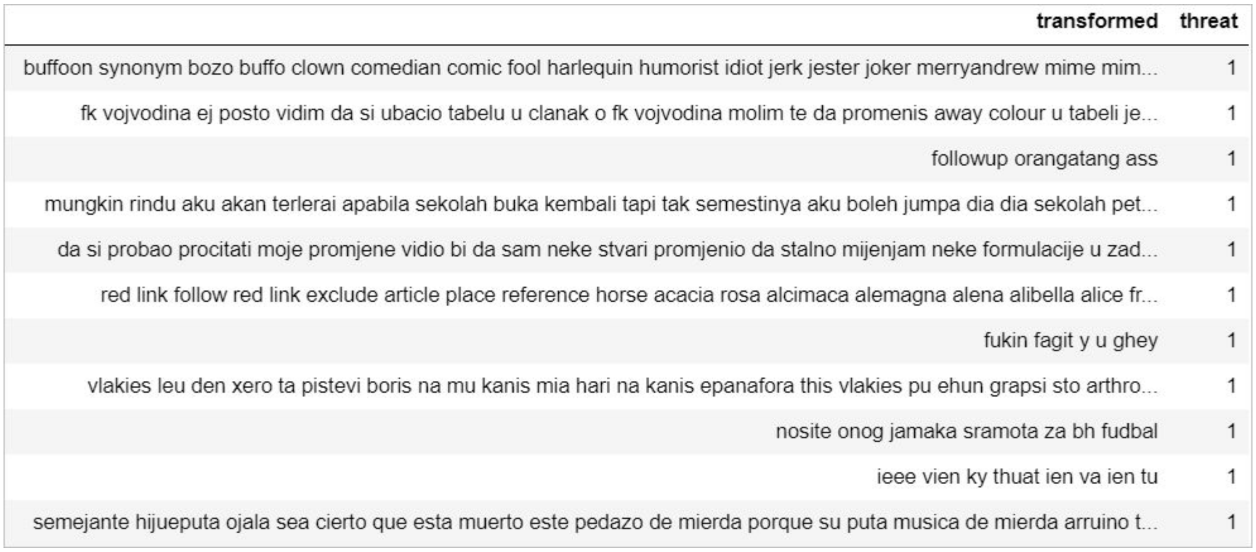


```
array([[34382,    91],
       [   86,    12]], dtype=int64)
```



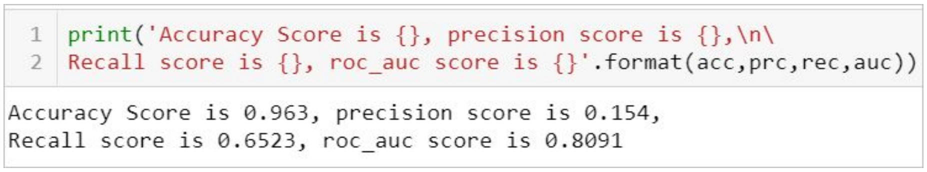
	comment_text	threat
	A misery=\nYou have made my life a misery on Wikipedia, and now you're gonna pay for it you disgusting, ignorant and...	1
	Burn In Hell Paulley \nEvery time I try to edit something, you delete it. You are not the only one who knows about w...	1
	go fcuk youslef \n\nI am going to kill you this is a threat too you life watch out.....	1
	Contd \n\nscoundrel the block on padmalskhmi was expired on april 26 2011\n\nso after the block expired i ahve did c...	1
	unblock input devices now u cunt or ill kill ya D \n\nunblock input devices now u cunt or ill kill ya D	1
	"A RESPONSE TO . I ENTITLE THIS ""YOUR STUPIDITY""\nThe Assyrians are not extinct you incompetent bigoted fool. You ...	1
	Unban this ip address or a new online encyclopedia will be formed that will kick wikipedia's ass. You have been warn...	1
	Die \n\nI HATE YOU PRICK YOU DINT DESERVE A PLACE HERE	1
	Sitush is an asshole...die you dog	1
	you are a prat, and my revenge will be sweet! hahahahahaahahahahahahahahahahahahahaha.	1
	I am going to kill you \n\ni am going to get a gun and blow your head off you stupid retard	1

Out of 98 threats of a 34571 validation comments, only 12 were picked out and labelled as threat.



transformed	threat
buffoon synonym bozo buffo clown comedian comic fool harlequin humorist idiot jerk jester joker merryandrew mime mim...	1
fk vojvodina ej posto vidim da si ubacio tabelu u clanak o fk vojvodina molim te da promenis away colour u tabeli je...	1
followup orangatang ass	1
mungkin rindu aku akan terlerai apabila sekolah buka kembali tapi tak semestinya aku boleh jumpa dia dia sekolah pet...	1
da si probao procitati moje promjene vidio bi da sam neke stvari promjenio da stalno mijenjam neke formulacije u zad...	1
red link follow red link exclude article place reference horse acacia rosa alcimaca alemagna alena alibella alice fr...	1
fukin fagit y u ghey	1
vlakies leu den xero ta pistevi boris na mu kanis mia hari na kanis epanafora this vlakies pu ehun grapsi sto arthro...	1
nosite onog jamaka sramota za bh fudbal	1
ieee vien ky thuat ien va ien tu	1
semejante hijueputa ojala sea cierto que esta muerto este pedazo de mierda porque su puta musica de mierda arruino t...	1

Above are the test dataset results. The text predicted as a member of threat class, but the contents are not really posing a threat. Please note that especially foreign languages have been merked as a threat, but almost all are false positives. The results of MNB classifier after weight balancing for "identity hate" feature is shown as below.

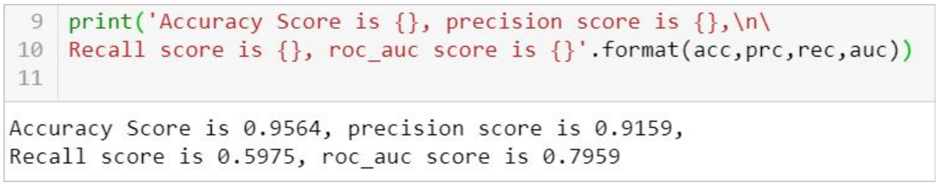


```
print('Accuracy Score is {}, precision score is {},\n\
Recall score is {}, roc_auc score is {}'.format(acc,prc,rec,auc))
```

```
Accuracy Score is 0.963, precision score is 0.154,
Recall score is 0.6523, roc_auc score is 0.8091
```

Logistic Regression:

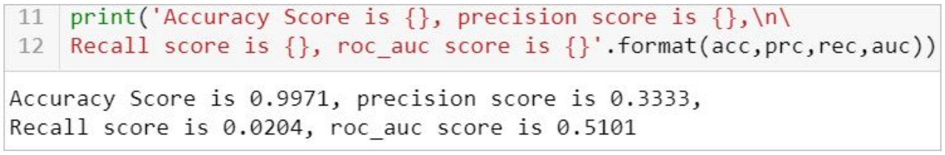
The results of Logistic Regression model for "toxic" and "threat" are shown below respectively. Please note that the model is good at explaining toxic feature, but not the threat.



```
print('Accuracy Score is {}, precision score is {},\n\
Recall score is {}, roc_auc score is {}'.format(acc,prc,rec,auc))

```

```
Accuracy Score is 0.9564, precision score is 0.9159,
Recall score is 0.5975, roc_auc score is 0.7959
```



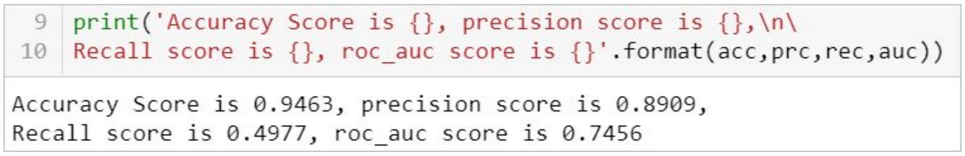
```
print('Accuracy Score is {}, precision score is {},\n\
Recall score is {}, roc_auc score is {}'.format(acc,prc,rec,auc))
```

```
Accuracy Score is 0.9971, precision score is 0.3333,
Recall score is 0.0204, roc_auc score is 0.5101
```

Support Vector Classification:

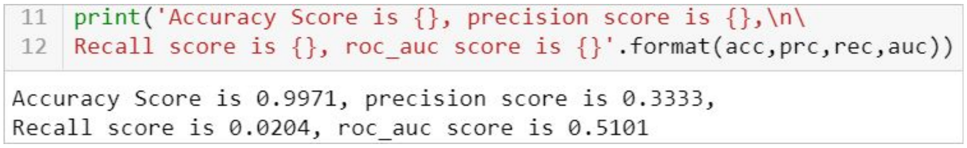
Final baseline model for classification used in this project is SVC. In order to reduce time required, number of features of TF-IDF were reduced using Singular Value Decomposition.

Again, the results of SVC model for “toxic” and “threat” are shown below respectively and the model acts similar to logistic regression in that it is good at explaining toxic feature, but not the threat.



```
print('Accuracy Score is {}, precision score is {},\n\
Recall score is {}, roc_auc score is {}'.format(acc,prc,rec,auc))
```

```
Accuracy Score is 0.9463, precision score is 0.8909,
Recall score is 0.4977, roc_auc score is 0.7456
```



```
print('Accuracy Score is {}, precision score is {},\n\
Recall score is {}, roc_auc score is {}'.format(acc,prc,rec,auc))
```

```
Accuracy Score is 0.9971, precision score is 0.3333,
Recall score is 0.0204, roc_auc score is 0.5101
```

Conclusion:

A 150 thousand long dataset comprising of comments entered about various topics has been preprocessed with a goal of building a multi-headed model that’s capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate. After intensive text pre-processing, and compensating for heavy class imbalance, various baseline models were employed in attempt to correctly label each feature. While models successfully explained “toxic, severe toxic, obscene”, they failed to generate promising results for variables such as “threat”.

In the following chapter, recurrent neuro-network models will be used to fill this gap and provide a better solution to the problem.