MALIGNANT COMMENTS CLASSIFICATION PROJECT

Submitted by:

Snehal Kumawat

INTRODUCTION

• Business Problem Framing

The proliferation of social media enables people to express their opinions widely online. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users. Although researchers have found that hate is a problem across multiple platforms, there is a lack of models for online hate detection. Our goal is to build a prototype of online hate and abuse comment classifier which can used to classify hate and offensive comments so that it can be controlled and restricted from spreading hatred and cyberbullying.

• Conceptual Background of the Domain Problem

The background for the problem originates from the multitude of online forums, where-in people participate actively and make comments. As the comments some times may be abusive, insulting or even hate-based, it becomes the responsibility of the hosting organizations to ensure that these conversations are not of negative type. The task was thus to build a model which could make prediction to classify the comments into various categories.

• Review of Literature

Online hate, described as abusive language, aggression, cyberbullying, hatefulness and many others has been identified as a major threat on online social media platforms. Social media platforms are the most prominent grounds for such toxic behaviour.

There has been a remarkable increase in the cases of cyberbullying and trolls on various social media platforms. Many celebrities and influences are facing backlashes from people and have to come across hateful and offensive comments. This can take a toll on anyone and affect them mentally leading to depression, mental illness, self-hatred and suicidal thoughts.

• Motivation for the Problem Undertaken

Internet comments are bastions of hatred and vitriol. While online anonymity has provided a new outlet for aggression and hate speech, machine learning can be used to fight it. The problem we sought to solve was the tagging of internet comments that are aggressive towards other users. This means that insults to third parties such as celebrities will be tagged as unoffensive, but “u are an idiot” is clearly offensive.

• Data Sources and their formats

The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains nearly 1,53,000 samples. All the data samples contain 8 fields which includes ‘Id’, ‘Comments’, ‘Malignant’, ‘Highly malignant’, ‘Rude’, ‘Threat’, ‘Abuse’ and ‘Loathe’.

The label can be either 0 or 1, where 0 denotes a NO while 1 denotes a YES. There are various comments which have multiple labels. The first attribute is a unique ID associated with each comment.

The data set includes:

- **Malignant:** It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.
- **Highly Malignant:** It denotes comments that are highly malignant and hurtful.
- **Rude:** It denotes comments that are very rude and offensive.
- **Threat:** It contains indication of the comments that are giving any threat to someone.

- **Abuse:** It is for comments that are abusive in nature.
- **Loathe:** It describes the comments which are hateful and loathing in nature.
- **ID:** It includes unique Ids associated with each comment text given.
- **Comment text:** This column contains the comments extracted from various social media platforms.

1. Data Preparation and Cleaning

Fig. 1 captures the research framework for the malignant comment classification problem. This section focuses on different algorithms and the various stages that are involved for the proposed Toxic comment classification system such as 'logistic regression', 'Decision Tree Classifier' and 'Random Forest Classifier' which helps us in classifying the comments and results in a decisive outcome. It includes five major blocks, namely data collection, data preparation, feature processing, mode training, classification and model evaluation. These blocks of the diagram are explained in detail in the next subsections.

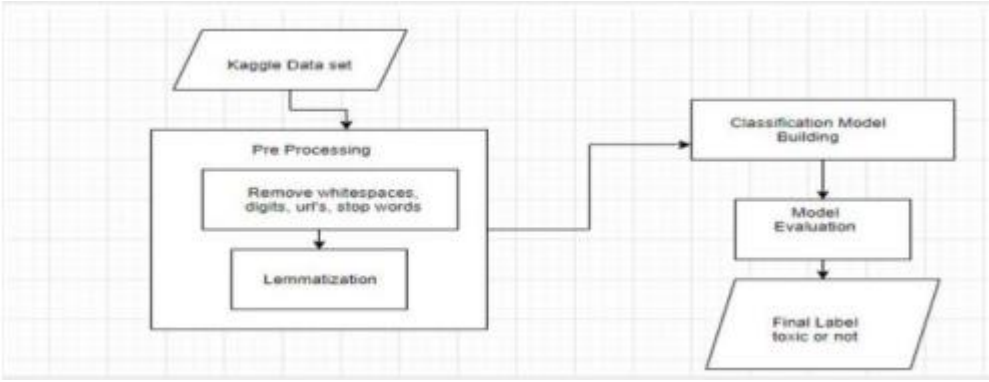


fig 1. Research framework

About Project

The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains nearly 1,53,000 samples. All the data samples contain 8 fields which includes 'Id', 'Comments', 'Malignant', 'Highly malignant', 'Rude', 'Threat', 'Abuse' and 'Loathe'. The project is related multilabel classification.

Packages used: Pandas, Numpy, Seaborn, Matplotlib, Scikit and Stats models, NLTK, Lemmeaizer.

Multilabel vs Multiclass classification

As the task was to figure out whether the data belongs to zero, one, or more than one categories out of the six listed above, the first step before working on the problem was to distinguish between multi-label and multi-class classification.

In multi-class classification, we have one basic assumption that our data can belong to only one label out of all the labels we have. For example, a given picture of a fruit may be an apple, orange or guava only and not a combination of these.

In multi-label classification, data can belong to more than one label simultaneously. For example, in our case a comment may be toxic, obscene and insulting at the same time. It may also happen that the comment is non-toxic and hence does not belong to any of the six labels.

Step 2: Studying data & identifying hidden patterns

For the next visualisation, I had length of comments on the independent axis. But instead of co unting number of comments, I counted comments belonging to each of the different categories.

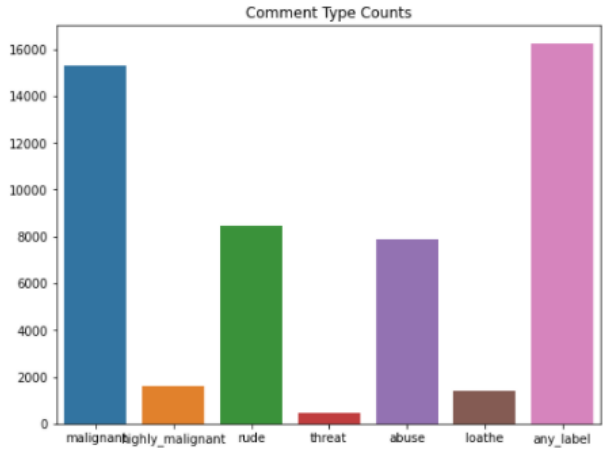


fig2.comments belonging to different categories

2. **Data Preprocessing**

1. **Preparation for removal of punctuation marks:** I imported the string library comprisi ng all punctuation characters and appended the numeric digits to it, as those were requir ed to be removed too.

```
import string
print(string.punctuation)
punctuation_edit = string.punctuation.replace('\'','') +"0123456789"
print (punctuation_edit)
outtab = "                                         "
trantab = str.maketrans(punctuation_edit, outtab)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
!"#$%&()*+,-./:;<=>?@[\]^_`{|}~0123456789
```

2. Updating the list of stop words : Stop words are those words that are frequently used in both written and verbal communication and thereby do not have either a positive/negative impact on our statement. E.g. is, this, us, etc.

Python has a built-in dictionary of stop words. I used the same and also appended the single letters like 'b', 'c' …. to it, which might be pre-existing or have generated during data preprocessing.

```
from stop_words import get_stop_words
stop_words = get_stop_words('english')
stop_words.append('')

for x in range(ord('b'), ord('z')+1):
    stop_words.append(chr(x))
```

3. Stemming and Lemmatising : Stemming is the process of converting inflected/derived words to their word stem or the root form. This helps in achieving the training process with a better accuracy.

Lemmatising is the process of grouping together the inflected forms of a word so they can be analyzed as a single item. I used the **word-net library** in nltk for this purpose. Stemmer and Lemmatizer were imported from nltk

```
import nltk
from nltk.stem import PorterStemmer, WordNetLemmatizer
```

```
#create objects for stemmer and lemmatizer
lemmatiser = WordNetLemmatizer()
stemmer = PorterStemmer()
#download words from wordnet library
nltk.download('wordnet')
```

```
for i in range(len(comments)):
    comments[i] = comments[i].lower().translate(trantab)
    l = []
    for word in comments[i].split():
        l.append(stemmer.stem(lemmatiser.lemmatize(word,pos="v")))
    comments[i] = " ".join(l)
```

4. Applying Count Vectorizer : Count Vectorizer is used for converting a string of wo rds into a matrix of words. Column headers have the words themselves and the cell val ues signify the frequency of occurrence of the word.

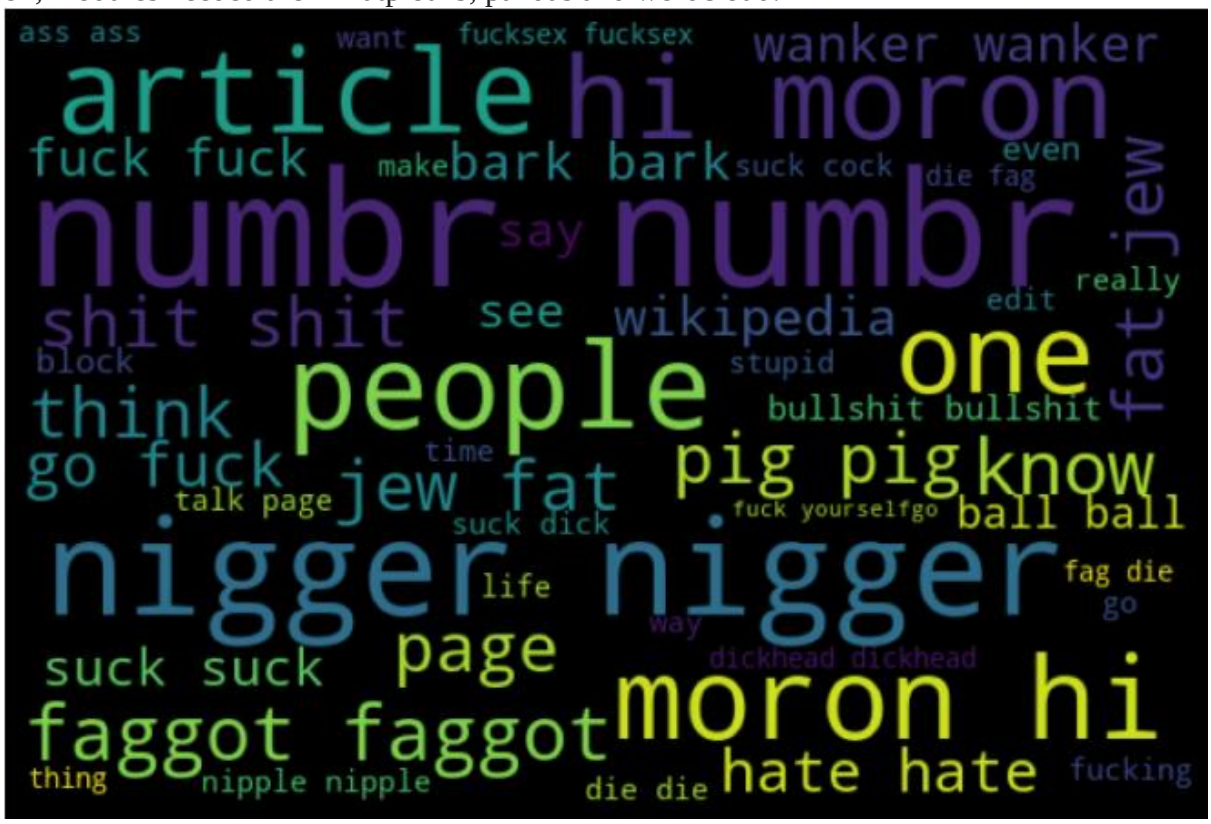
```
#import required library
from sklearn.feature_extraction.text import CountVectorizer

#create object supplying our custom stop words
count_vector = CountVectorizer(stop_words=stop_words)
#fitting it to converts comments into bag of words format
tf = count_vector.fit_transform(comments).toarray()
```

After doing all this process the clean comments are as follows and length column cont ains original length of comments and clean_length column contains length of clean co mments.

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	length	clean_length
0	0000997932d777bf	explanation edits made username hardcore metal...	0	0	0	0	0	0	264	180
1	000103f0d9cfb60f	d'aww! match background colour i'm seemingly s...	0	0	0	0	0	0	112	111
2	000113f07ec002fd	hey man, i'm really trying edit war. guy const...	0	0	0	0	0	0	233	149
3	0001b41b1c6bb37e	can't make real suggestion improvement wondere...	0	0	0	0	0	0	622	397
4	0001d958c54c6e35	you, sir, hero. chance remember page that's on?	0	0	0	0	0	0	67	47

5. WordCloud

Word Cloud is a data visualization technique used for representing text data in which t he size of each word indicates its frequency or importance. Word clouds are widely us ed for analyzing data from social network websites. For generating word cloud in Pyth on, modules needed are – matplotlib, pandas and wordcloud.



Feature Scaling

The engineered features are normalized from 0.0 to 1.0. The tf-idf features are not scaled.

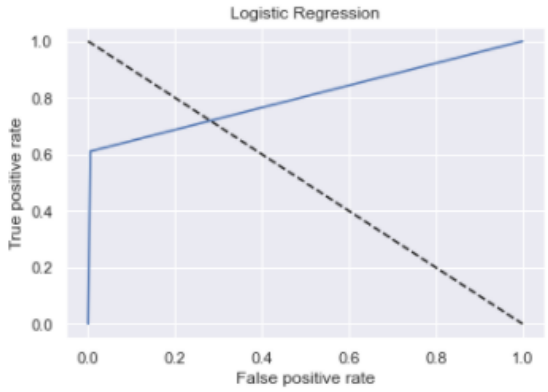
3. Methodology

Finding the Best Algorithm To compare the relative performances of each algorithm, I'm going to test them on the same preprocessed data, a tf-idf vectorized data with features. The target is any_label, and the scores are F1 scores with five-fold cross validation.

Logistic Regression

Logistic regression is **a statistical model that** in its basic form uses a logistic function to mod el a binary dependent variable, although many more complex extensions exist. In regression a nalysis, logistic regression is estimating the parameters of a logistic model. Using this we get
Training accuracy is 0.9595520103134316
Test accuracy is 0.9552974598930482

AUC ROC Curve



4. Classification Model Building:

Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

Splitting the Dataset

As usual for supervised machine learning problems, we need a training dataset to train our model and a test dataset to evaluate the model.

DecisionTreeClassifier

A decision tree is a simple representation for classifying examples. For this section, assume that all of the i nput features have finite discrete domains, and there is a single target feature called the "classification".
Training accuracy is 0.9988898736783678
Test accuracy is 0.9395053475935828

KNeighborsClassifier

The K in the name of this classifier **represents the k nearest neighbors**, where k is an integer value specified by the user.

Using KNeighborsClassifier and cross validation we got
Training accuracy is 0.922300110117369
Test accuracy is 0.9173629679144385

RandomForestClassifier

A random forest classifier. A random forest is **a meta estimator that fits a number of decision tree c lassifiers on various sub-samples of the dataset** and uses averaging to improve the predictive accura cy and control over-fitting. The number of trees in the forest.

We got Training accuracy is 0.9988451105202374

Test accuracy is 0.9551721256684492

The best model is Logistic Regression.

Since the difference between the percentages score of cross validation and r2_score is optimum. After that we save the best model using pickle method.

5. Conclusion

It is seen that the most effective model in predicting malignant comments is Logistic Regressi on. After predicting test dataset we get the following results.

And here is output for test dataset values.

	comment_text	Pred
0	Yo bitch Ja Rule is more succesful then you'll...	0
1	== From RfC == \n\n The title is fine as it is...	0
2	" \n\n == Sources == \n\n * Zawe Ashton on Lap...	0
3	:If you have a look back at the source, the in...	0
4	I don't anonymously edit articles at all.	0
...	...	...
153159	. \n i totally agree, this stuff is nothing bu...	0
153160	== Throw from out field to home plate. == \n\n...	0
153161	" \n\n == Okinotorishima categories == \n\n I ...	0
153162	" \n\n == ""One of the founding nations of the...	0
153163	" \n :::Stop already. Your bullshit is not wel...	0

153164 rows × 2 columns

6. Limitations of this work and Scope for Future Work

Our research has shown that harmful or toxic comments in the social media space have many negative impacts to society. The ability to readily and accurately identify comments as t oxic could provide many benefits while mitigating the harm. Also, our research has shown the capability of readily available algorithms to be employed in such a way to address this challen ge.

We suggest a plan to improve the NLP classifiers: first by using other algorithms which such as Support Vector Clustering (SVC) and Convolutional Neural Networks (CNN).

b) We also suggest using SVM for text processing and text classification. It requires a grid sea rch for hyper-parameter tuning to get the best results.