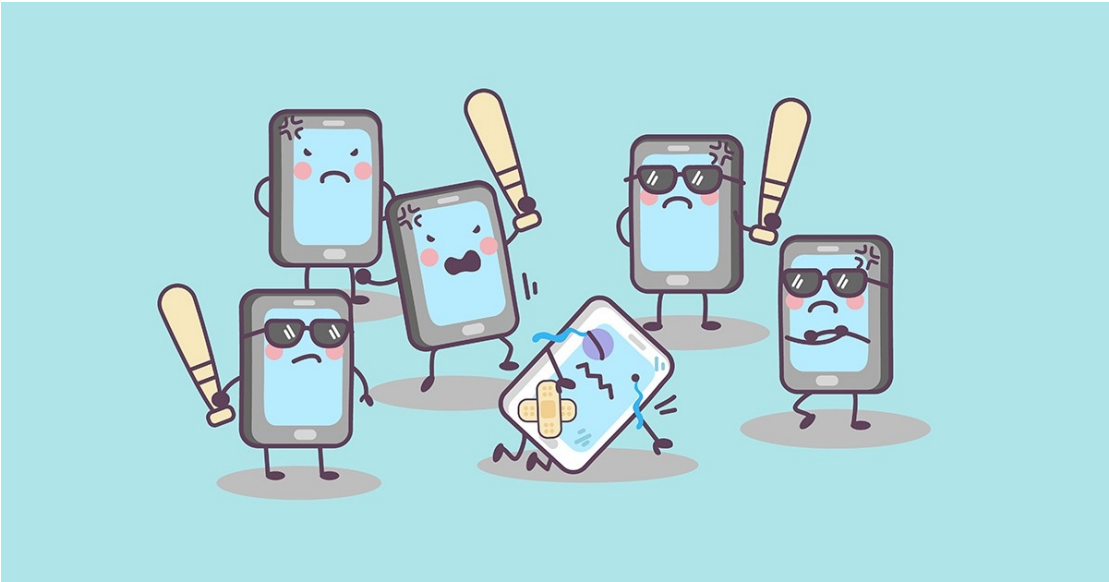
FLIP ROBO

Malignant comment classification



Submitted by:

ArunPrasad.k

Table of content

Abstract.....3

General Term.....3

Index Terms..... 3

1. Introduction.....3

2. Problem Statement.....4

3. DataSet.....4

4. Data Overview.....4

5. Approach.....4

6. Implementation.....5

6.1.Importing Training Dataset.....5

6.2.Cleaning Dataset 5

7. Exploratory Data Analysis 5

7.1.Visualization Prediction on count of each category.....5

7.2.Pie chart on percentage of each Label Combination.....5

7.3 Correlation matrix.....6

8. Feature Engineering 6

8.1. Cleaning the comments.....6

8.2. Lemmatizer.....6

8.3 Vectorization.....6

9. Data Training..... 7

9.1. Prediction.....7

10. Conclusion.....8

Future work 8

Reference..... 8

Abstract*—A large proportion of online comments present on public domains are usually constructive, however a signifificant proportion are Malignant in nature. Dataset is obtained online which are processed to remove noise from the dataset. The comments contain lot of errors which increases the number of features manifold, making the machine learning model to train the dataset by processing the dataset, in the form of transformation of raw comments before feeding it to the Classification models using a machine learning technique known as the term frequency-inverse document frequency (TF-IDF) technique. The logistic regression technique is used to train the processed dataset, which will differentiate Malignant comments from non-Malignant comments. The multi-headed model comprises Malignantity (severeMalignant, Rude, threat, abuse, and identity-hate) or Non-Malignantity Evaluation, using confusion metrics for their prediction.

***General Terms*: Data Analytics, Exploratory Data Analytics, Machine Learning, Model Evaluation , Data Science.**

***Keywords*- online comments, Malignantity, classifification models, TF-IDF technique, logistic regression.**

1. INTRODUCTION

Over the years, social media and social networking use have been increasing exponentially due to an upsurge in the use of the internet. Flood of information arises from online conversation in a daily basis as people are able to discuss, express themselves and air their opinion via these platforms. While this situation is highly productive and could contribute significantly to the quality of human life, it could also be destructive and enormously dangerous. While discussion or a conversation is opened, it is quite obvious that debates may arise due to differences in opinion.

But often these debates take a dirty side and may result in fights over the social media during which offensive language termed as Malignant comments may be used from one side. These Malignant comments may be threatening, Rude, abuseing or identity- based hatred. So,these clearly pose the threat of abuse and harassment online. Consequently, some people stop giving their opinions or give up seeking different opinions which result in unhealthy and unfair discussion. As a result, different platforms and communities find it very difficult to facilitate fair conversation and are often forced to either limit user comments or get dissolved by shutting down user comments completely. This study focuses on building a multi-headed model to detect different types of Malignantity like threats, obscenity, abuses, and identity-based hate. Detecting and controlling verbal abuse in an automated fashion is inherently a natural language processing task.

Natural Language Processing, (NLP), is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable.

Most NLP techniques rely on machine learning to derive meaning from human languages. Machine learning explores the construction and study of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model for example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions.

Malignant comment classification on online channels is conventionally carried out either by moderators or with the help of text classification tools. With recent advances in Deep Learning (DL) techniques, researchers are exploring if DL can be used for comment classification task. Text classification is a classic topic for natural language processing and an essential component in many applications, such as web searching, information filtering, topic categorization and sentiment analysis .

Text transformation is the very first step in any form of text classification. The online comments are generally in non-standard English and contain lots of spelling mistakes partly because of typos (resulting from small screens of the mobile devices) but more importantly because of the

deliberate attempt to write the abusive comments in creative ways to dodge the automatic filters.

2.PROBLEM STATEMENT

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.

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 Malignant 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.

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 abuses to third parties such as celebrities will be tagged as unoffensive, but “u are an idiot” is clearly offensive.

3.DATASET

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.

4.DATA OVERVIEW

- ✓ **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.

5.APPROACH

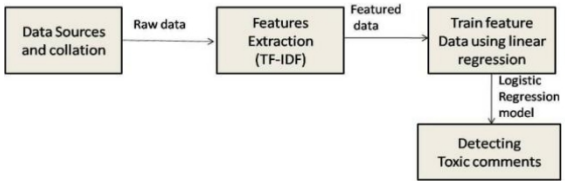


Fig. 1 System Design of the Study

The steps for document classification are:

i. Read Document

ii. Tokenization, where every word in the document is split into tokens

iii. Removing stop words, punctuation, or unwanted tokens

iv. Lemmatization/Stemming, Shorten words to their root stems, e.g. to take walk, walked, walking, walks as Walk etc

v. Feature vector representation

vi. Feature extraction

vii. Learning algorithm

6.IMPLEMENTATION

6.1. Importing Train Dataset:

Reading the data to plot the graphs:

```
from google.colab import files
uploaded=files.upload()
```

Choose Files train.csv

- **train.csv**(application/vnd.ms-excel) - 68597918 bytes, last modified: 7/4/2021 - 100% done

```
Saving train.csv to train (1).csv
```

```
import io
df_train = pd.read_csv(io.BytesIO(uploaded['train.csv']))
df_train.head(10)
```

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

6.2 Cleaning Dataset:

Before applying any type of data analytics on the data set, the data should be first cleaned.

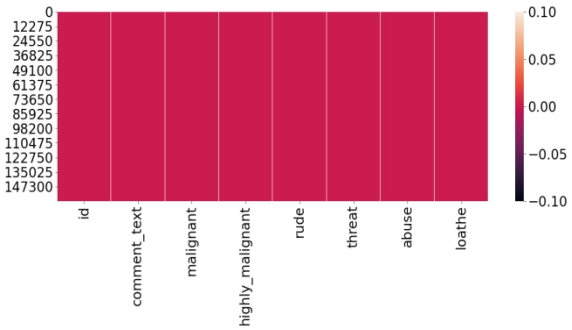
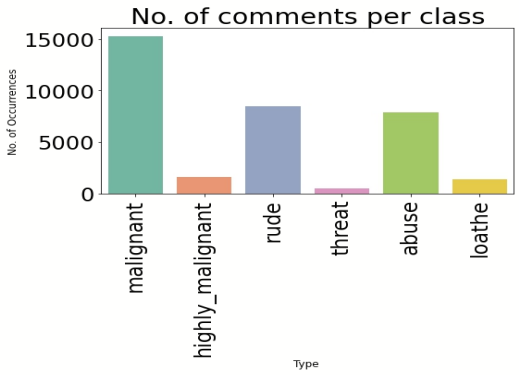


Figure 2: Heatmap on missing values

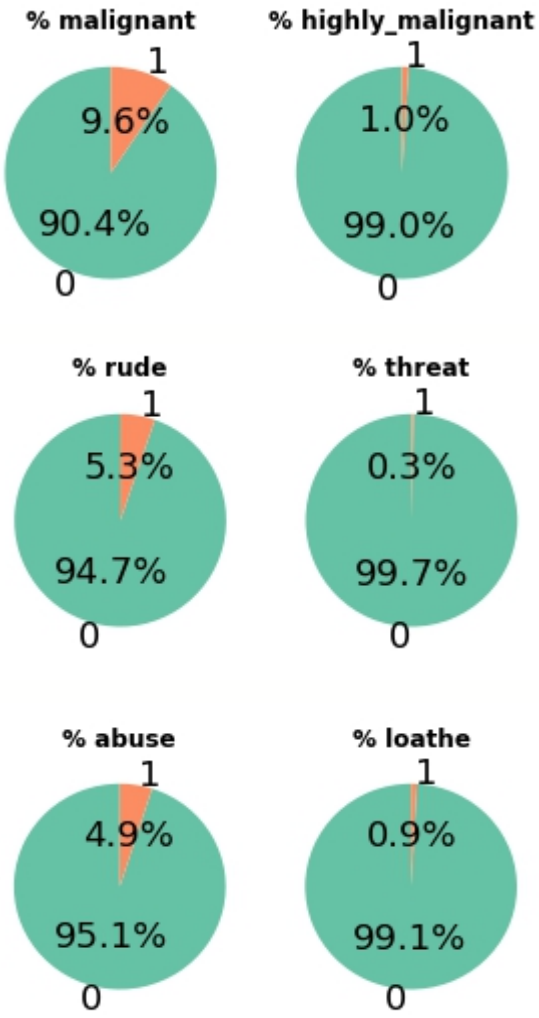
7. EXPLORATORY DATA ANALYSIS

We are going to perform exploratory data analysis for our problem in the first stage. In exploratory data analysis data set is explored to figure out the features which would influence the Houses sale price. The data is deeply analyzed by finding a relationship between each attribute and our target label.

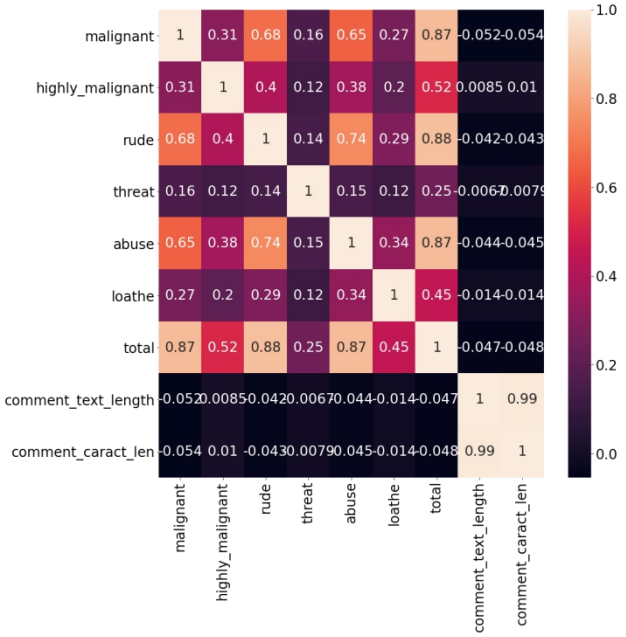
7.1 VISUALIZATION ON COUNT 0F EACH CATEGORY



7.2 PIE CHART ON PERCENTAGE OF EACH LABEL COMBINATION



7.3 CORRELATION MATRIX



Following can be inferred from above matrix:

- Malignant is highly correlated with Rude and abuse.
- abuse and Rude have highest correlation factor of 0.74

8.FEATURE ENGINEERING

8.1 CLEANING THE COMMENTS

Since, the comments in the dataset were collected from the internet they may contain various elements in them.

Removing Punctuations and other special characters

```
# Text preprocessing steps - remove numbers, capital letters, punctuation, '\n'
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
punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ',x)

# 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
df_train['comment_text'] = df_train['comment_text'].map(alphanumeric).map(punc_lower).map(remove_n).map(remove_non_ascii)
```

We then converted each comment into lower case and then split it into individual words. There were some words in the dataset which had length > 100, since there are no words in the English language whose length > 100, we remove such words.

First, we tried building the features removing stop words and then trained some models thinking that it may help the model in learning the semantics of Malignantity, but we found out that the model learns better if there are stop words in the comment.

Possible reason is, generally a hate/Malignant comment is used towards a person, seeing the data we found out that those persons are generally referred by pronouns, which are nothing but stop words.

8.2 LEMMATIZERS

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove endings only and to return the base or dictionary form of a word, which is known as the lemma.

We used **WordNet lemmatizer**. For grammatical reasons, documents are going to use different forms of a word, such as organizes, organize and organizing. But they all represent the same semantics. So, Lemmatizer for those three words gives a single word, which helps algorithm learn better.

```
#Lemmetizing
WL = WordNetLemmatizer()
df_train['comment_text'] = df_train['comment_text'].apply(lambda x: ' '.join(WL.lemmatize(i) for i in x.split()))
df_train.head(5)
```

8.3 VECTORIZATION

Python's scikit-learn deals with numeric data only. To convert the text data into numerical form, tf-idf vectorizer is used. TF-IDF vectorizer converts a collection of raw documents to a matrix of Tf-idf features.

We set the predictor variable on the dataset with tf-idf vectorizer, in two different ways. First, by setting the parameter analyzer as 'word'(select words) and the second by setting it to 'char'(select characters). Using 'char' was important because the data had many 'foreign languages' and they were di cult to deal with by considering only the 'word' analyzer.

We set the parameter n-gram range (an n-gram is a continuous sequence of n-items from a given sample of text or speech). After trying various values, we set the n-gram as (1, 1) for 'word' analyzer. We also set the max_features as 30000 for both word and character analyzer after many trails.

We then combined the word and character features and transformed the dataset into two sparse matrixes for train and test sets, respectively using tf-idf vectorizer.

9. DATA TRAINING

Training of the processed dataset is done by problem as a multi-label classification problem. Multi-label classification problem is transformed single-class classifier problem. This is known as problem transformation. This is achieved using the binary relevance method. The pre-processed dataset is trained using

1. **LogisticRegression**
2. **KNeighborsClassifier**
3. **MultinomialNB**
4. **BernoulliNB**
5. **LinearSVC**
6. **RandomForestClassifier**

```
def cv_tf_train_test(df,label,vectorizer,ngram):

    ''' Train/Test split'''
    # Split the data into X and y data sets
    x = df.comment_text
    y = df[label]

    # Split our data into training and test data
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)

    # Create a Vectorizer object and remove stopwords from the table
    cv1 = vectorizer(ngram_range=(ngram), stop_words='english')

    X_train_cv1 = cv1.fit_transform(x_train) # Learn the vocabulary dictionary and return term-document matrix
    X_test_cv1  = cv1.transform(x_test)      # Learn a vocabulary dictionary of all tokens in the raw documents.
```

```
#initializing all models
lr = LogisticRegression()
lr.fit(x_train_cv1, y_train)
print("Lr fit done")

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train_cv1, y_train)
print("KN fit done")

bnb = BernoulliNB()
bnb.fit(x_train_cv1, y_train)
print("BNB fit done")

mnb = MultinomialNB()
mnb.fit(x_train_cv1, y_train)
print("MNB fit done")

svm_model = LinearSVC()
svm_model.fit(x_train_cv1, y_train)
print("SVC fit done")

randomforest = RandomForestClassifier(n_estimators=100, random_state=42)
randomforest.fit(x_train_cv1, y_train)
print("RF fit done")
```

	Log Regression	KNN	BernoulliNB	MultinomialNB	SVM	Random Forest
F1 Score of malignant	0.718808	0.308465	0.646498	0.325547	0.763174	0.710162
F1 Score of highly_malignant	0.357955	0.209493	0.008016	0.000000	0.376658	0.119349
F1 Score of rude	0.731305	0.328698	0.301397	0.219673	0.773980	0.756349
F1 Score of threat	0.243902	0.117647	0.000000	0.000000	0.347368	0.093333
F1 Score of abuse	0.624805	0.292370	0.202050	0.088623	0.657965	0.609351
F1 Score of loathe	0.271186	0.183299	0.000000	0.000000	0.372881	0.099352

From the F1 score board we see LinearSVCand Logistic regression models perform best **LinearSVC as our final model.**

9.1 PREDICTION

Since we have evaluated all models by using **F1 SCORE** we will predict by using model which has highest accuracy.Here we can choose **LinearSVC** models to predict the malignant comment.The training accuracy for the various comments obtained from social media network is very high.

LABEL	TRAINING ACCURACY	TESTING ACCURACY
Malignant	98.53	95.95
Highly Malignant	99.60	99.01
Rude	99.29	97.85
Threat	99.92	99.74
Abuse	98.97	97.04
Loathe	99.76	99.22

10. CONCLUSION

Communication is one of the basic necessities of everyone's life. People need to talk and interact with one another to express what they think. Over the years, media and social networking have been increasing exponentially due to an upsurge (rise) in the use of the internet. Flood of information arises from online conversation on a daily basis, as people are able to discuss, express themselves and express their opinion While this situation is highly productive and could contribute significantly to the quality of human life, it could also be destructive and enormously dangerous. The responsibility lies on the social media administration, control and monitor these comments.

This research work focuses on developing a model that would automatically classify a comment as either Malignant or non-Malignant using logistic regression to develop a multi-headed model to detect different types of Malignantity like threats, obscenity, abuses, and loathe. By collecting and preprocessing Malignantity classified comments for training and testing using inverse document frequency(TF-IDF) algorithm, developing a multi-headed model will detect different types of Malignantity using logistic regression to train the dataset, and evaluate the model using F1 score.

Future Work

Future work on this study could be divided into four main areas to improve the result even further. Which can be done by:

I. The used pre-processing methods do help in the prediction accuracy. However, experimenting with different combinations of pre-processing methods to achieve better prediction accuracy.

II. Make use of the available features and if they could be combined as binning features has shown that the data got improved.

III. The correlation has shown the association in the testing data. Thus, attempting to enhance the Training data is required to make rich with features that vary and can provide a strong correlation relationship.

IV. Try more ways of vectorizing text data.

V. Go deeper on feature engineering : Spelling corrector, Sentiment scores, n-grams, etc.

VI. Advanced models (e.g., lightgbm).

VII. Advanced Ensemble model (e.g., stacking).

VIII. Deep learning model (e.g., LSTM).

REFERENCE

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

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

iii. Waseem, Z., Thorne, J., Bingel, J.,(2018). Bridgingthegaps:Multi task Learningfor Domain Transfer of Hate Speech Detection. Online Harassment.

iv. siva. 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.