Malignant Comments Classifier

Submitted by:

Ashish Modi

Acknowledgement

I would like to express my gratitude to my guide Shubham Yadav (SME, Flip Robo) for his constant guidance, continuous encouragement and unconditional help towards the development of the project. It was he who helped me whenever I got stuck somewhere in between. The project would have not been completed without his support and confidence he showed towards me.

Lastly, I would l like to thank all those who helped me directly or indirectly toward the successful completion of the project.

Ashish Modi

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.
- 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.
- 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.
- 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

- In the past few years its seen that the cases related to social media hatred have increased exponentially. The social media is turning into a dark venomous pit for people now a days. Online hate is the result of difference in opinion, race, religion, occupation, nationality etc.
- In social media the people spreading or involved in such kind of activities uses filthy languages, aggression, images etc. to offend and gravely hurt the person on the other side. This is one of the major concerns now.
- The result of such activities can be dangerous. It gives mental trauma to the victims making their lives miserable. People who are not well aware of mental health online hate or cyber bullying become life threatening for them. Such cases are also at rise. It is also taking its toll on religions. Each and every day we can see an incident of fighting between people of different communities or religions due to offensive social media posts.

- Online hate, described as abusive language, aggression, cyberbullying, hatefulness, insults, personal attacks, provocation, racism, sexism, threats, or toxicity has been identified as a major threat on online social media platforms. These kinds of activities must be checked for a better future.

Review of Literature

Nowadays users leave numerous comments on different social networks, news portals, and forums. Some of the comments are toxic or abusive. Due to numbers of comments, it is unfeasible to manually moderate them, so most of the systems use some kind of automatic discovery of toxicity using machine learning models. In this work, we performed a systematic review of the state-of-the-art in toxic comment classification using machine learning methods. First, we have investigated when and where the papers were published and their maturity level. In our analysis of every primary study we investigated: data set used, evaluation metric, used machine learning methods, classes of toxicity, and comment language.

Motivation for the Problem Undertaken

The project was the first provided to me by FlipRobo as a part of the internship program. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary objective. However, the motivation for taking this project was that it is relatively a new field of research. Here we have many options but less concrete solutions. The main motivation 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.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

We would perform one type of supervised learning algorithms: Classification. While it seems more reasonable to perform Classification since we have 5-6 class to predict. Here, we will only perform classification. Since there only 1 feature in the dataset, filtering the words is needed to prevent overfit. In order to determine the regularization parameter, throughout the project in classification part, we would first remove email, phone number, web address, spaces and stops words etc. In order to further improve our models, we also performed TFID in order to convert the tokens from the train documents into vectors so that machine can do further processing.

Data Sources and their formats

We have 1 input field i.e. comments and 6 output classes.
Comments field is type String.
6 other classifications are in 1 0 forms.

Data Pre-processing Done

Cleaned the data from junk values. Replace multiple spaces with single space So that it'll easy to classify it.

Hardware and Software Requirements and Tools Used

The system requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view. System requirements are all of the requirements at the system level that describe the functions which the system as a whole should fulfil to satisfy the stakeholder needs and requirements, and is expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

Hardware requirements: -

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software requirements: -

Anaconda

Libraries: -

From sklearn.preprocessing import StandardScaler

As these columns are different in **scale**, they are **standardized** to have common **scale** while building machine learning model. This is useful when you want to compare data that correspond to different units.

from sklearn.preprocessing import Label Encoder

Label Encoder and One Hot Encoder. These two encoders are parts of the SciKit Learn library in Python, and they are used to convert categorical data, or text data, into numbers, which our predictive models can better understand.

from sklearn.model_selection import train_test_split,cross_val_score

Train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train_test_split will make random partitions for the two subsets.

The algorithm is trained and tested K times, each time a new set is used as testing set while remaining sets are used for training. Finally, the result of the K-Fold Cross-Validation is the average of the results obtained on each set.

from sklearn.neighbors import KNeighborsClassifier

K Nearest Neighbor(KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition

from sklearn.linear_model import LogisticRegression

The library sklearn can be used to perform logistic regression in a few lines as shown using the LogisticRegression class. It also supports multiple features. It requires the input values to be in a specific format hence they have been reshaped before training using the fit method.

from sklearn.tree import DecisionTreeClassifier

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy.

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

Just make the comments more appropriate so that we'll get less word to process and get more accuracy. Removed extra spaces, converted email address into email keyword, likely wise phone number etc. Tried to make Comments small and more appropriate as much as it was possible.

Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing.

- KNN = KNeighborsClassifier()
- LR = LogisticRegression()
- BNB = BernoulliNB()
- DT = DecisionTreeClassifier()
- RF = RandomForestClassifier()

I applied all these algorithms in the dataset.

Run and Evaluate selected models

```
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> LogisticRegression()
0.9551721256684492
[[42809   196]
 [ 1950  2917]]
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     43005
           1       0.94      0.60      0.73      4867

    accuracy                           0.96     47872
   macro avg       0.95      0.80      0.85     47872
weighted avg       0.95      0.96      0.95     47872

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> KNeighborsClassifier()
0.9017379679144385
[[42343   662]
 [ 4042   825]]
              precision    recall  f1-score   support

           0       0.91      0.98      0.95     43005
           1       0.55      0.17      0.26      4867

    accuracy                           0.90     47872
   macro avg       0.73      0.58      0.60     47872
weighted avg       0.88      0.90      0.88     47872

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> BernoulliNB()
0.7941385360962567
[[33774  9231]
 [  624  4243]]
              precision    recall  f1-score   support

           0       0.98      0.79      0.87     43005
           1       0.31      0.87      0.46      4867

    accuracy                           0.79     47872
   macro avg       0.65      0.83      0.67     47872
weighted avg       0.91      0.79      0.83     47872

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> DecisionTreeClassifier()
0.9416151403743316
[[41633  1372]
 [ 1423  3444]]
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     43005
           1       0.72      0.71      0.71      4867

    accuracy                           0.94     47872
   macro avg       0.84      0.84      0.84     47872
weighted avg       0.94      0.94      0.94     47872

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
```

```
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> RandomForestClassifier()
0.9564672459893048
[[42664   341]
 [ 1743  3124]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     43005
           1       0.90      0.64      0.75      4867

    accuracy                           0.96     47872
   macro avg       0.93      0.82      0.86     47872
weighted avg       0.95      0.96      0.95     47872

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> KNeighborsClassifier(n_neighbors=3)
0.9144593917112299
[[42888   117]
 [ 3978   889]]
              precision    recall  f1-score   support

           0       0.92      1.00      0.95     43005
           1       0.88      0.18      0.30      4867

    accuracy                           0.91     47872
   macro avg       0.90      0.59      0.63     47872
weighted avg       0.91      0.91      0.89     47872

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
```

Saving the Model

```
import joblib
joblib.dump(rfc,"Malignant_Comment_Classifier_Predict.obj")
```

Key Metrics for success in solving problem under consideration

Precision: can be seen as a measure of quality, **higher precision** means that an algorithm returns more relevant results than irrelevant ones.

Recall is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.

Accuracy score is used when the True Positives and True negatives are more important. **Accuracy** can be used when the class distribution is similar.

F1-score is used when the False Negatives and False Positives are crucial. While F1-score is a better metric when there are imbalanced classes.

Visualizations

Word Clouds -

Malignant

Rude


Nipple Nipple
yourselfgo fuck
faggots faggots
asshole
JEW FAT
FREEDOM FREEDOM
SHIT SHIT
Go fuck
poop poop
SUCK SUCK
wikipedia
NIGGER NIGGER
fucksex
fuck yourselfgo
BULLSHIT
Bullshit
ass ass
think
fucking
fuck fuck
FAT JEW
dickhead dickhead
fuck Go
DIE DIE

Treat

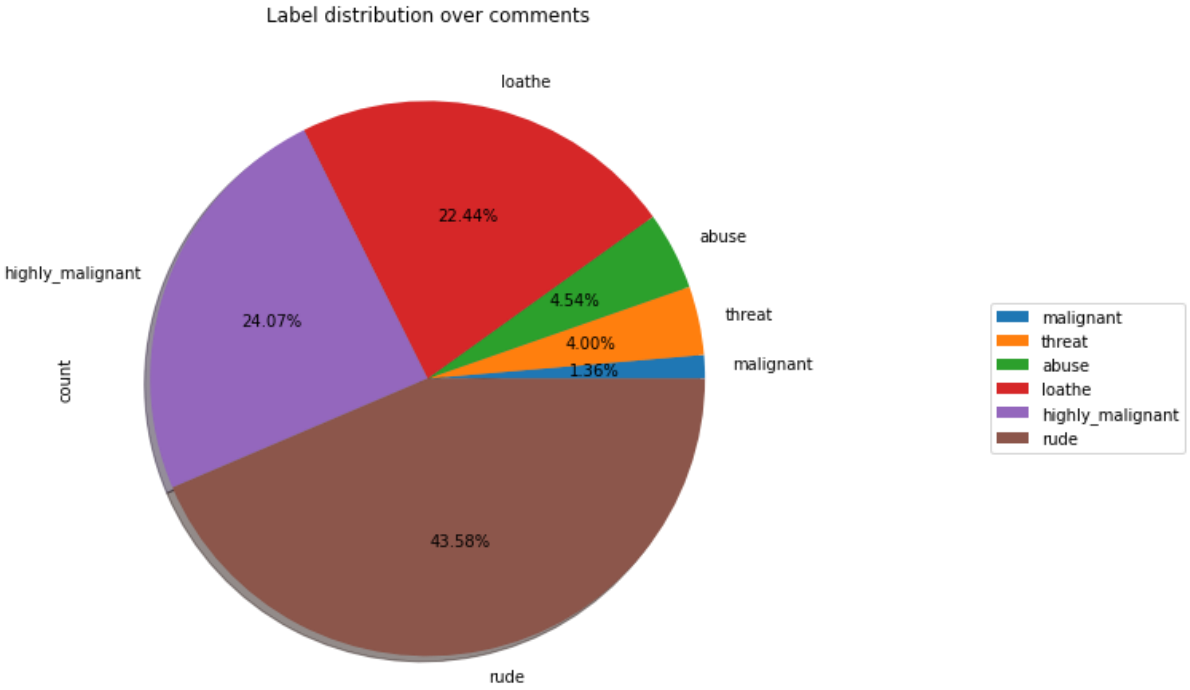

fucking
MUST DIE
DIE WIKIPEDIA
DIE JIM
going murder
WIKIPEDIA DIE
WILL BLANK
ass ass
kill going
EDIE DIE
WILL BLOCK
SuPeRTRnumbrLL
WiLL
LIFETIME BAN
DIE DIE
KILL
going kill
DI EDIE
murder going
BLOCK WILLTALK PAGE
WALES MUST
PAGE WILL
JIM WALE
fuck

Abuse

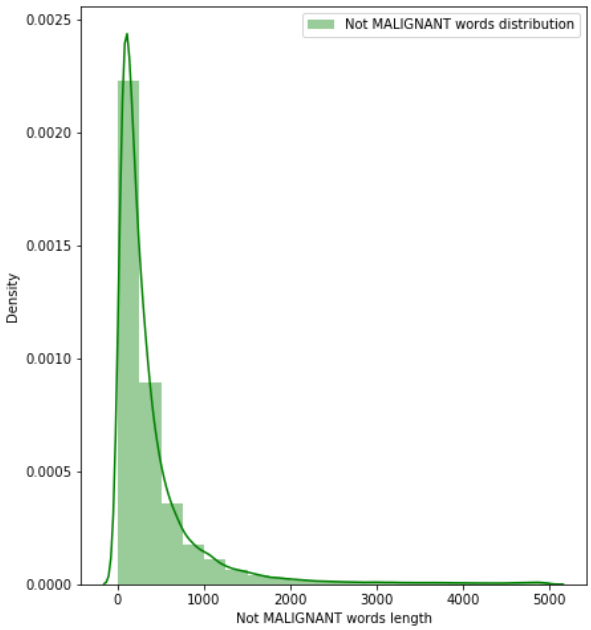
Loathe

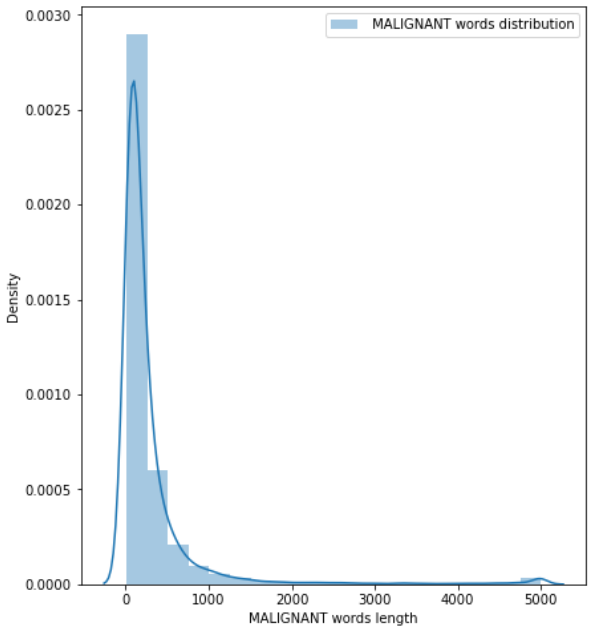
Highly_Malignant

Label Distribution over Comments

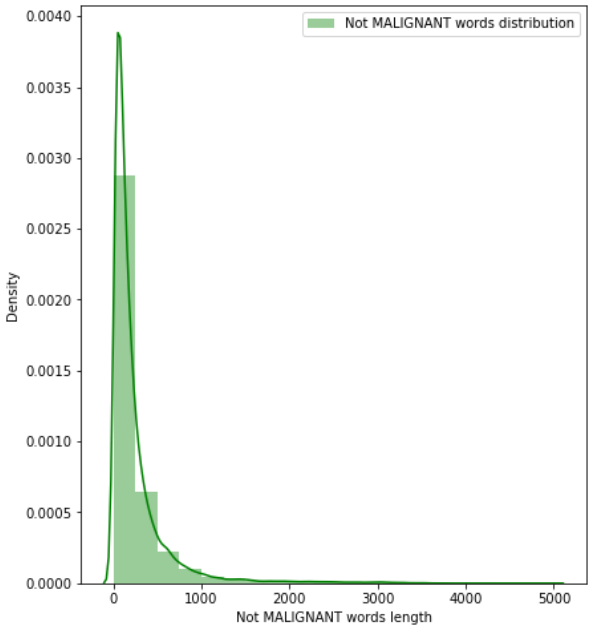


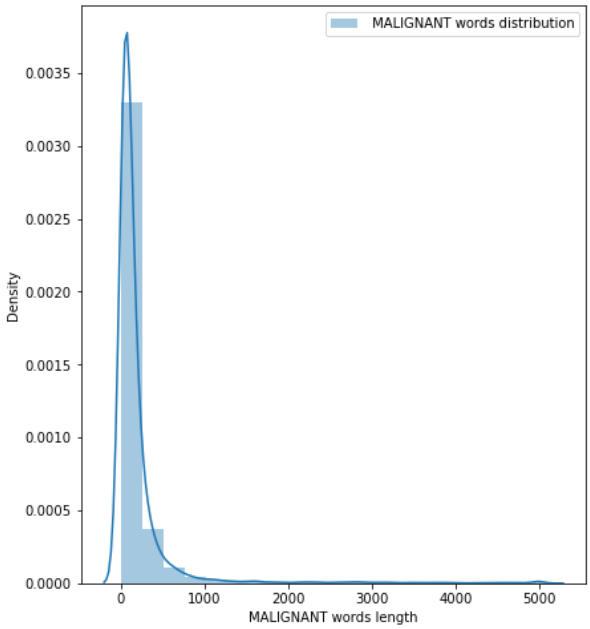
Not Malignant and Malignant comments with original comments





Not Malignant and Malignant comments with filtered comments





Interpretation of the Results

From Random Forest Classifier R2 score was 95.64%, means 96% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

Higher the R2 score means the model is well fit for the data. However, if R2 score is very high, it might be a case of overfitting. Other metrics Mean Absolute Error, Mean Squared Error and Root Mean Squared Error, with gradient boosting these scores are less then compared to other models. If these errors are less that means the model shows less errors.

Conclusion

Key Findings and Conclusions of the Study

Here we saw 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 toxic could provide many benefits while mitigating the harm. Also, we have seen the capability of readily available algorithms to be employed in such a way to address this challenge. In our specific study, it was demonstrated that an Random Forest Classifier solution provides substantial improvement in classification versus any other algorithm.

Learning Outcomes of the Study in respect of Data Science

The power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important steps, tried to make comments shorter and all the necessary keywords in it.
Various algorithms I used in this dataset and to get out best result and save that model. The best algorithm is Random Forest Classification.

Limitations of this work and Scope for Future Work

Additionally, the followings are some suggested studies to be considered as future work in this area:

a) 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); secondly, extend the classifiers to the overall goal which is multi-label classifiers. in the current study, the problem simplified into two classes but it worth to pursue a main goal which is 6 classes of comments.

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

c) using Other DNN techniques (CNN)) because some recently published papers have shown that CNN proves to have a very high performance for various NLP tasks.