Malignant Comments Classifier

Submitted by

Steffin Varghese

Batch - 26

In partial fulfilment of Data Science – Internship

At

Flip Robo Technologies

AI and Software Development company

Flip Robo Technologies | Indiranagar, Bengaluru - 560 038, Karnataka, India

Month of Submission

September 2022

Acknowledgement

I am highly indebted to FlipRobo Technologies for giving me this opportunity to work on a project and for the guidance and persistent supervision, as well as for providing necessary project information and assistance in completing the project.

I would want to convey my gratitude to the members of FlipRobo Technologies for their kind encouragement and support in completing this project.

I would like to extend my heartfelt gratitude and appreciation to SME, Ms Khushboo Garg for spending such close attention to me and assisting me during the course of completion of project, as well as towards others who have volunteered to assist me with their skills.

I acknowledge my gratitude towards the authors of papers: “A Machine Learning Approach to Comment Toxicity Classification”, “ Detecting and Classifying Toxic Comments” and “Toxic Comment Classification” for the insights I could attain through the extensive research which enhanced my knowledge in development of the project.

CONTENTS

INTRODUCTION

Analytical Problem Framing

Model/s Development and Evaluation

CONCLUSION

INTRODUCTION

Business Problem Framing

Social media has spawned numerous job prospects in the twenty-first century while also developing into a distinctive forum for individuals to openly express their thoughts. There are some groups among these users, however, that abuse this framework and this freedom to spread their negative viewpoints (i.e., insulting, verbal sexual harassment, threads, Obscene, etc.).

Derogatory remarks are frequently uttered by people nowadays, not just in offline settings but also significantly in online settings like social networking sites and online communities. Therefore, it is essential that all social networking platforms and applications, as well as all online communities, include an Identification with Prevention System.

Conceptual Background of the Domain Problem

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.

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. Internet comments are bastions of hatred and vitriol.

Although researchers have found that hate is a problem across multiple platforms, there is a lack of models for online hate detection. 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. In such a system, the Identification Block would be responsible for seeing any unfavourable online behaviour and alerting the Prevention Block to take appropriate action.

Review of Literature

A Machine Learning Approach to Comment Toxicity Classification, (2012) Navoneel Chakrabarty

This study tries to analyse any text and identify several forms of toxicity, including obscenity, threats, insults, and hate speech motivated by identification. For this, Jigsaw's tagged Wikipedia Comment Dataset is utilised. The Mean Validation Accuracy of a 6-headed Machine Learning tf-idf Model was 98.08%, and the Absolute Validation Accuracy was 91.61% after it underwent separate training. Such an automated system needs to be used to promote positive internet discourse.

In this paper, a machine learning approach and natural language processing were used to identify the type of toxicity in user comments and to detect it. The Mean Validation Accuracy, as so obtained, is 98.08%, by far the best numerical accuracy yet attained by any Comment Toxicity Detection Model. The study conducted for this paper aims to promote fair online discourse and viewpoint exchange in social media. By using the Grid Search Algorithm on the same dataset instead of the Machine Learning Algorithms for each pipeline, which are used to provide more accurate classifications and better outcomes, a more resilient model may be created.

Detecting and Classifying Toxic Comments, (2018), Kevin Khieu, Neha Narwal

This study introduces multiple deep learning techniques used to categorise online comment toxicity. They investigate the effectiveness of word- and character-level embeddings in conjunction with Support Vector Machines (SVM), Long Short-Term Memory Networks (LSTM), Convolutional Neural Networks (CNN), and Multilayer Perceptron (MLP) approaches in detecting toxicity in text. The authors have evaluated their approaches on Wikipedia comments from the Kaggle Toxic Comments Classification Challenge dataset.

Their word-level analysis revealed that their forward LSTM model outperformed other models on both binary classification (classifying specific types of toxicity versus non-toxic) and multi-label classification tasks. Their greatest outcomes for character-level classification happened when they used a CNN model. However, overall, their word-level models performed noticeably better than their character-level models. In the future, they aim to improve performance by utilising more intricate deep learning models and better word/character representations.

Toxic Comment Classification, (2020), Sara Zaheri, Jeff Leath, David Stroud

The classification of unstructured text into dangerous and benign categories using unique applications of Natural Language Processing techniques is presented in this work. The study of the data revealed that LSTM has a true positive rate that is 20% greater than that of the well-known Naive Bayes method, and this might fundamentally alter the field of comment classification. These findings suggested that effective data science application can provide a healthier environment for virtual societies.

Motivation for the Problem Undertaken

There is no denying that social media is one of the main hallmarks of the twenty-first century. This is due to the development of IT technologies and the globalisation of virtualization. In the meantime, social media provides a platform to discuss ideas and express one's personal viewpoints in a way that helps build a secure space where everyone can exercise their rights as they see fit.

However, some people believe they may abuse and harass the beliefs and characters of other people while hiding behind the virtual barriers created by computers. So, the term "cyberbullying" has recently been used in jargon to describe similar practises.

So many of our fellow citizens are discouraged from expressing their ideas due to such internet harassment. To be able to identify and distinguish harassing comments and cyberbullying, which we term toxic comments, from regular comments, it is vital for data scientists to understand the studies and results pertaining to this type of online harassment.

For moderators of public platforms as well as users who could receive warnings and filter undesired contents, automatic recognition of poisonous contents in online forums and social media is a beneficial feature. With all the advancements in IT and data science, a well-designed method to identify and remove the harmful remarks is now necessary for the world.

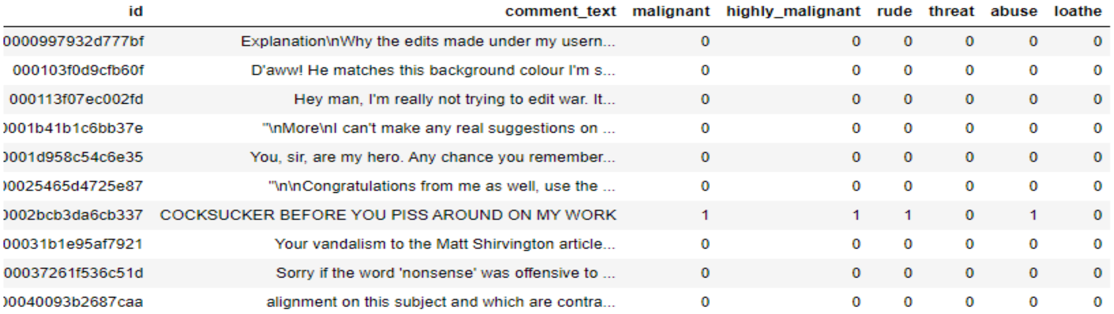
Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

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. The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains 1,53,000 samples.

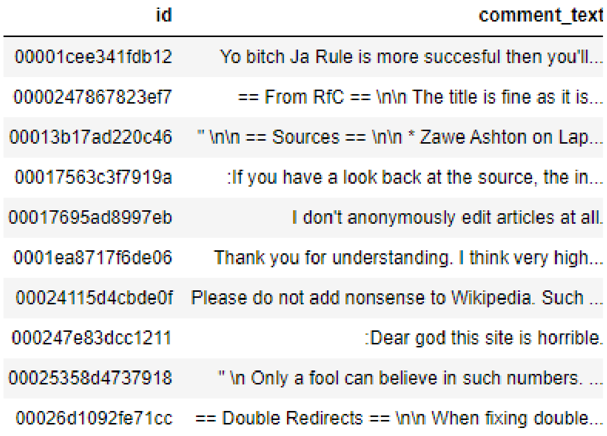
Data Sources and their formats

- We have 159,571 records and 8 columns including the target variable in the train dataset. In the test dataset, we have 153,164 records and two features.
- We have string and integer type of data in the train dataset. We only have string type of data in the test dataset.
- We have 159,571 non-null values in all the columns of train dataset and 153,164 non-null values in all the columns of test dataset.



id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
0025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
0040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

Train Dataset



id	comment_text
00001cee341fdb12	Yo bitch Ja Rule is more succesful then you'll...
0000247867823ef7	== From RfC == \n\n The title is fine as it is...
00013b17ad220c46	" \n\n == Sources == \n\n * Zawe Ashton on Lap...
00017563c3f7919a	:If you have a look back at the source, the in...
00017695ad8997eb	I don't anonymously edit articles at all.
0001ea8717f6de06	Thank you for understanding. I think very high...
00024115d4cbde0f	Please do not add nonsense to Wikipedia. Such ...
000247e83dcc1211	:Dear god this site is horrible.
00025358d4737918	" \n Only a fool can believe in such numbers. ...
00026d1092fe71cc	== Double Redirects == \n\n When fixing double...

Test Dataset

Description of the dataset

Features in Dataset (Independent Variable)

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:

`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.

Target in dataset (Dependent Variable)

`Malignant` : It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.

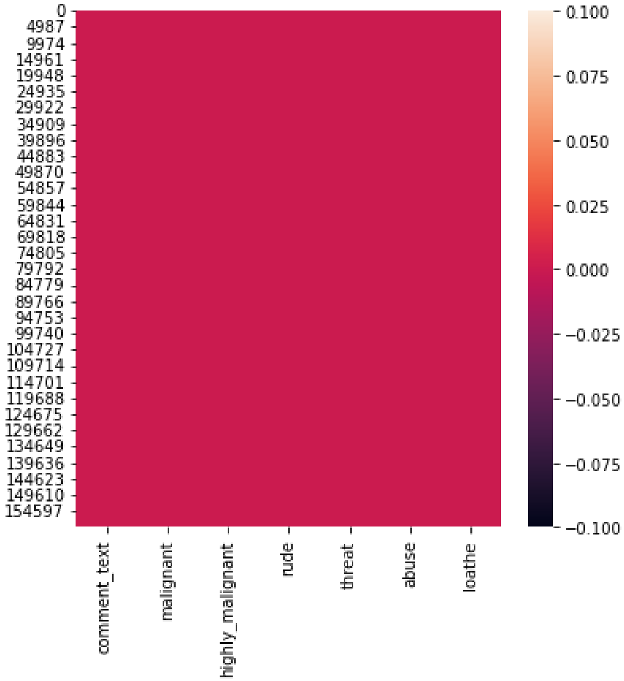
Data Pre-processing

Text Processing

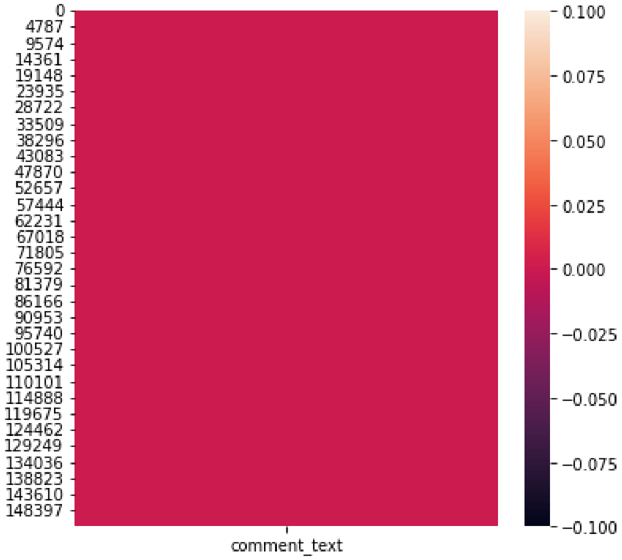
While exploring the categorical variables, there are many words, numbers, as well as punctuations which are not important for our predictions. So, we had to process the text.

- We have dropped the column 'id' from the train and test dataset as it was provided only for the identification of each comment.
- We used a user defined function to clean the text and we removed numbers, URLs, punctuations, and other unwanted characters from the train and test data and unshrink the words which were mentioned in short forms.
- Removed stop words from train and test dataset.
- Lemmatization – Using wordnet lemmatizer and word tokenize, we removed the inflected forms of words and converted them to their 'lemma' or dictionary form.

Checking for Missing Values in the Dataset



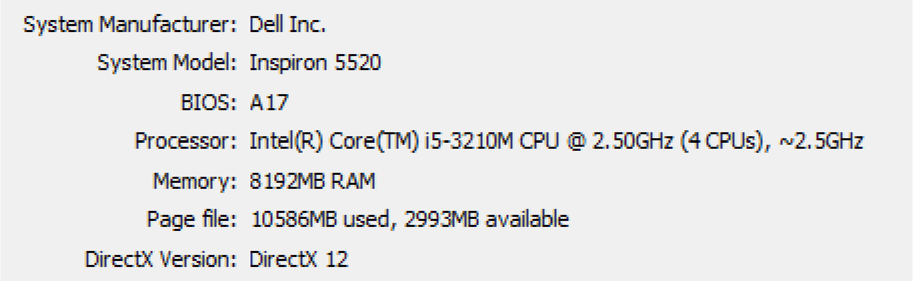
Heatmap of null values in train dataset.



Heatmap of null values in test dataset

Hardware and Software Requirements and Tools Used

Hardware Requirement:



```
System Manufacturer:  Dell Inc.
       System Model:  Inspiron 5520
               BIOS:  A17
          Processor:  Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz (4 CPUs), ~2.5GHz
             Memory:  8192MB RAM
          Page file:  10586MB used, 2993MB available
    DirectX Version:  DirectX 12
```

Hardware Configuration

Software Requirements:

- Windows Version: Windows 10 Pro
- Anaconda Navigator: 2.0.3
  Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine.
- Jupyter Notebook: 6.3.0
  Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine.
- Python3: Python 3.9.9
  Python3 is used as the base environment for performing the machine learning and data analysis.

  Python Libraries Used:
- Pandas: Data manipulation and analysis
- NumPy: Adding support for large, multi-dimensional arrays and matrices, along with an enormous collection of high-level mathematical functions to operate on these arrays.
- Matplotlib, Seaborn: For visualization of variable relations and data distribution, and analysis.
- Wordcloud : For plotting the word cloud
- Sklearn: Simple and efficient tools for predictive data analysis.
- SciPy: SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics, and many other classes of problems.
- Statsmodels: Provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
- Nltk : For natural language processing.
- Xgboost, catboost, lightgbm: Gradient boosting framework that uses tree-based learning algorithms.
- Pickle: Implements binary protocols for serializing and de-serializing

Model/s Development and Evaluation

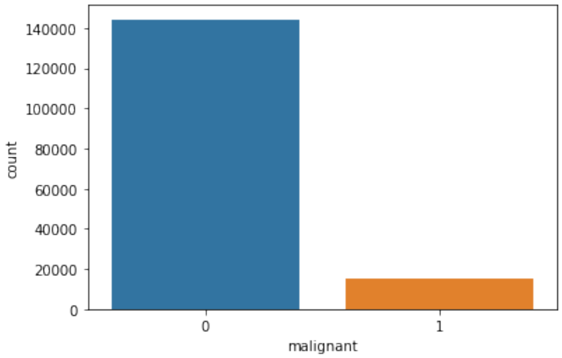
Identification of possible problem-solving approaches (methods)

Vectorizing the Text Data in Dataset

We used tfidf vectorizer for vectorizing the text data in the dataset.

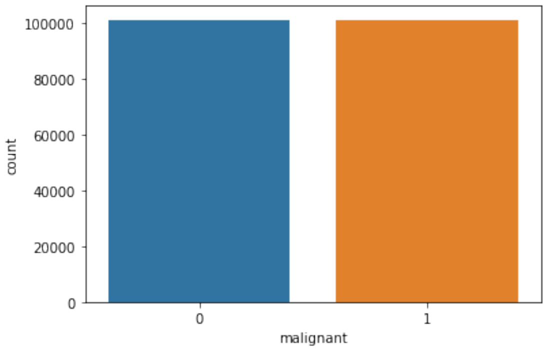
Balancing the dataset

We used smote technique to oversample the data in dataset to equalize the number of records in each category.



```
0     144277
1      15294
```

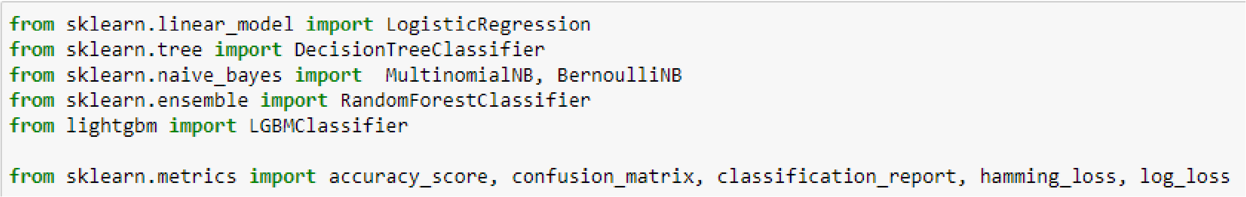
Unbalanced dataset



```
0     101100
1     101100
```

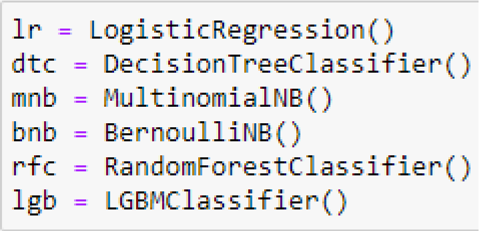
Balanced Dataset

Building the Model



```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import  MultinomialNB, BernoulliNB
from sklearn.ensemble import RandomForestClassifier
from lightgbm import LGBMClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, hamming_loss, log_loss
```

Algorithms and metrics for evaluation



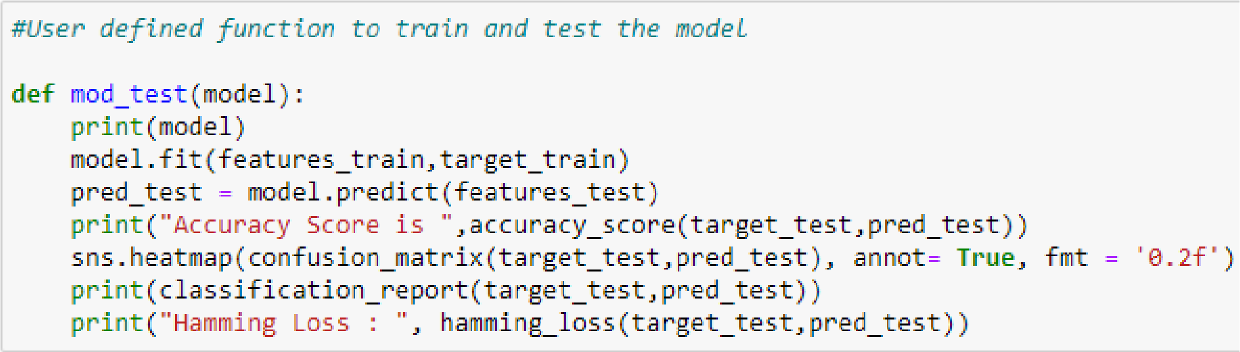
```
lr = LogisticRegression()
dtc = DecisionTreeClassifier()
mnb = MultinomialNB()
bnb = BernoulliNB()
rfc = RandomForestClassifier()
lgb = LGBMClassifier()
```

Instances of Algorithms

We have created two user defined functions to train, test and cross validate the models with the pre-processed train data.

- mod_test: Training and testing the model
- cross_val: Finding the best cross validation mean score for each model.

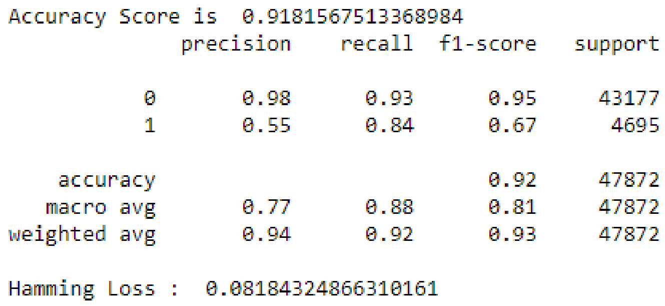
Testing of Identified Approaches (Algorithms) and evaluation of selected models



```
#User defined function to train and test the model

def mod_test(model):
    print(model)
    model.fit(features_train,target_train)
    pred_test = model.predict(features_test)
    print("Accuracy Score is ",accuracy_score(target_test,pred_test))
    sns.heatmap(confusion_matrix(target_test,pred_test), annot= True, fmt = '0.2f')
    print(classification_report(target_test,pred_test))
    print("Hamming Loss : ", hamming_loss(target_test,pred_test))
```

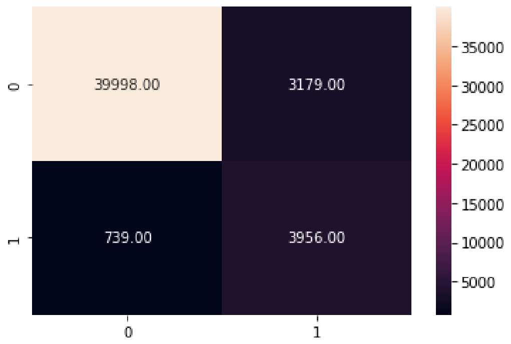
User Defined Function for Training and Testing the model

- **Logistic Regression**



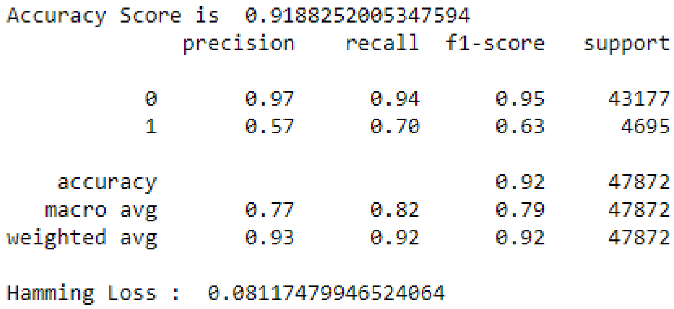
```
Accuracy Score is  0.9181567513368984
              precision    recall  f1-score   support

           0       0.98      0.93      0.95     43177
           1       0.55      0.84      0.67      4695

    accuracy                           0.92     47872
   macro avg       0.77      0.88      0.81     47872
weighted avg       0.94      0.92      0.93     47872

Hamming Loss :  0.08184324866310161
```

Test Performance



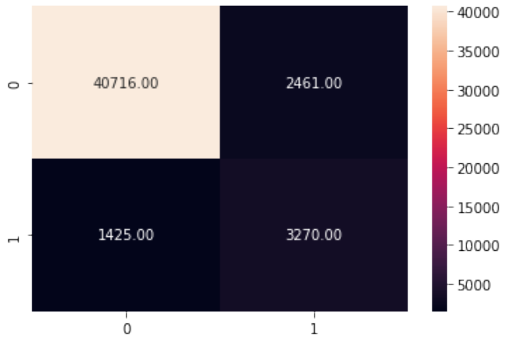
Confusion Matrix

- **DecisionTree Classifier**



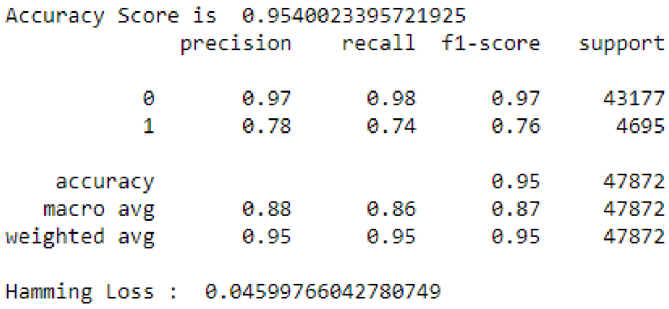
```
Accuracy Score is  0.9188252005347594
              precision    recall  f1-score   support

           0       0.97      0.94      0.95     43177
           1       0.57      0.70      0.63      4695

    accuracy                           0.92     47872
   macro avg       0.77      0.82      0.79     47872
weighted avg       0.93      0.92      0.92     47872

Hamming Loss :  0.08117479946524064
```

Test Performance



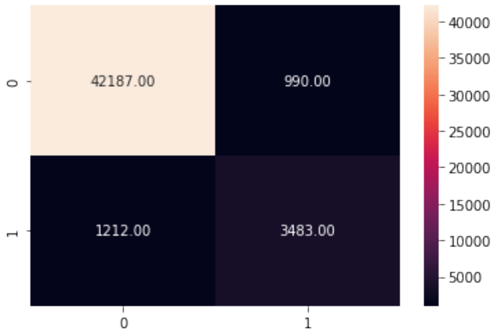
Confusion Matrix

- **Multinomial NB**



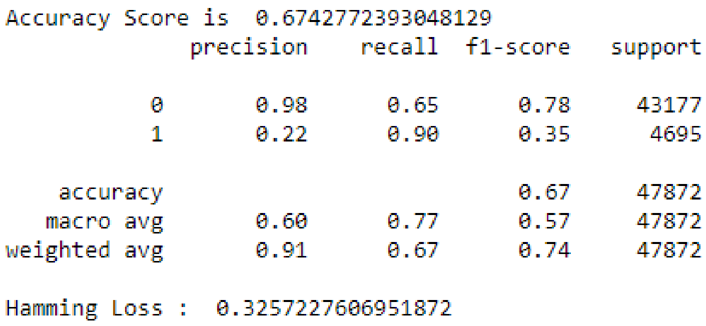
```
Accuracy Score is  0.9540023395721925
              precision    recall  f1-score   support

           0       0.97      0.98      0.97     43177
           1       0.78      0.74      0.76      4695

    accuracy                           0.95     47872
   macro avg       0.88      0.86      0.87     47872
weighted avg       0.95      0.95      0.95     47872

Hamming Loss :  0.04599766042780749
```

Test Performance



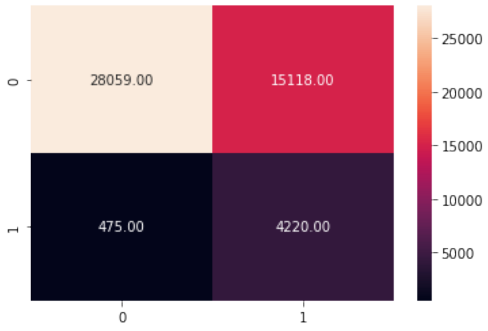
Confusion Matrix

- **Bernoulli NB**



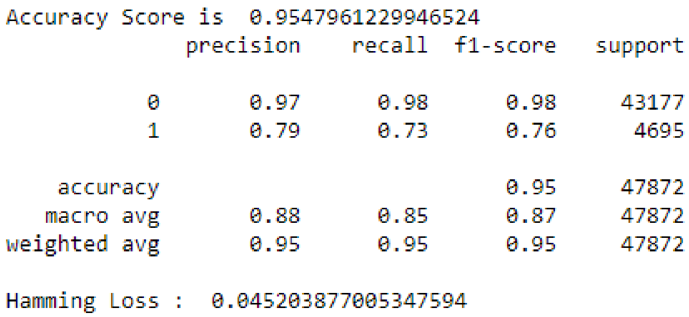
```
Accuracy Score is  0.6742772393048129
              precision    recall  f1-score   support

           0       0.98      0.65      0.78     43177
           1       0.22      0.90      0.35      4695

    accuracy                           0.67     47872
   macro avg       0.60      0.77      0.57     47872
weighted avg       0.91      0.67      0.74     47872

Hamming Loss :  0.3257227606951872
```

Test Performance



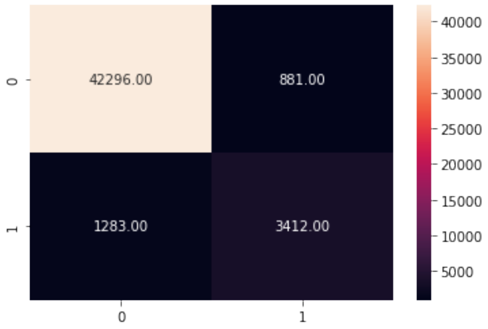
Confusion Matrix

- **RandomForest Classifier**



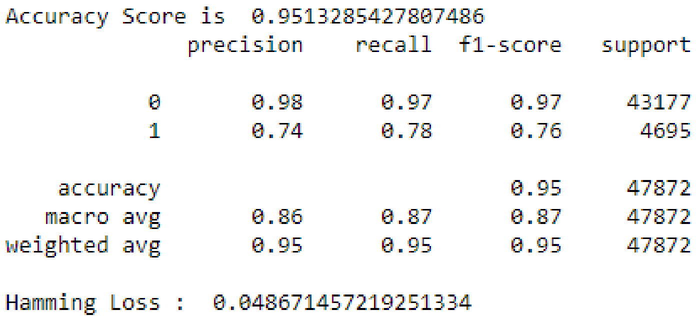
```
Accuracy Score is  0.9547961229946524
              precision    recall  f1-score   support

           0       0.97      0.98      0.98     43177
           1       0.79      0.73      0.76      4695

    accuracy                           0.95     47872
   macro avg       0.88      0.85      0.87     47872
weighted avg       0.95      0.95      0.95     47872

Hamming Loss :  0.045203877005347594
```

Test Performance



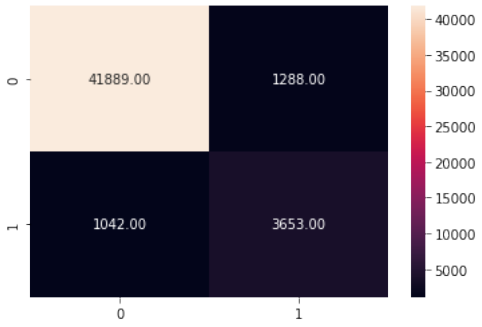
Confusion Matrix

- **LGBM Classifier**



```
Accuracy Score is  0.9513285427807486
              precision    recall  f1-score   support

           0       0.98      0.97      0.97     43177
           1       0.74      0.78      0.76      4695

    accuracy                           0.95     47872
   macro avg       0.86      0.87      0.87     47872
weighted avg       0.95      0.95      0.95     47872

Hamming Loss :  0.048671457219251334
```

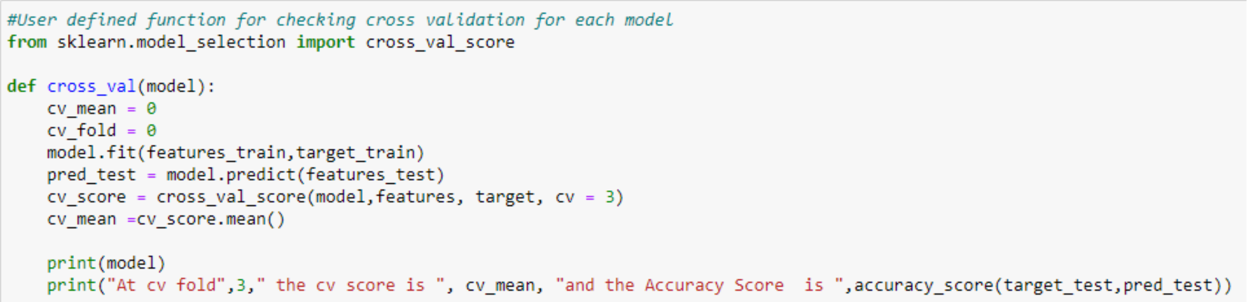
Test Performance



Confusion Matrix

After training and testing the models, the Multinomial NB (mnb), RandomForest Classifier (rfc) and the LGBM Classifier(lgb) are performing well and providing the maximum accuracy score. Now let us check the cross-validation score to find the best performing model.

Cross Validation



```
#User defined function for checking cross validation for each model
from sklearn.model_selection import cross_val_score

def cross_val(model):
    cv_mean = 0
    cv_fold = 0
    model.fit(features_train,target_train)
    pred_test = model.predict(features_test)
    cv_score = cross_val_score(model,features, target, cv = 3)
    cv_mean =cv_score.mean()

    print(model)
    print("At cv fold",3," the cv score is ", cv_mean, "and the Accuracy Score  is ",accuracy_score(target_test,pred_test))
```

Code Snippet for Cross Validation

- LogisticRegression
  At cv fold 3 the cv score is 0.961860239683861 and the Accuracy Score is 0.9181567513368984
- DecisionTree Classifier
  At cv fold 3 the cv score is 0.9463123000068568 and the Accuracy Score is 0.9201620989304813

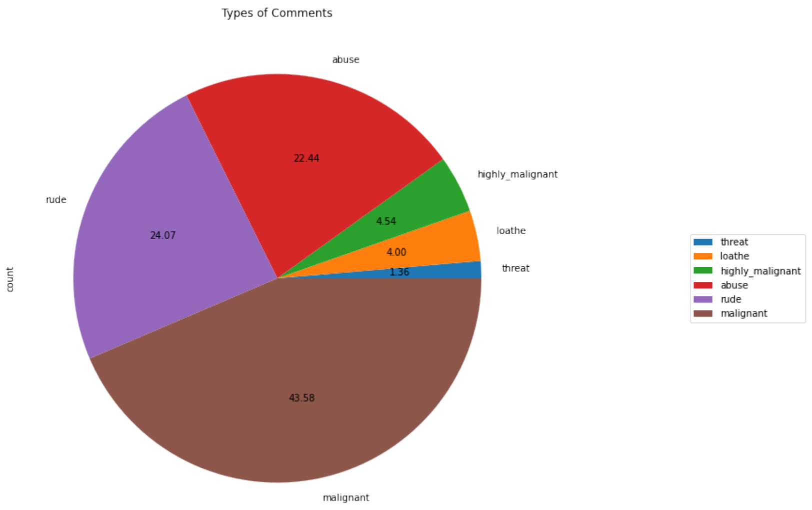
- MultinomialNB
  At cv fold 3 the cv score is 0.9609954241872941 and the Accuracy Score is 0.9537098930481284
- BernoulliNB
  At cv fold 3 the cv score is 0.7552750404097978 and the Accuracy Score is 0.6751128008021391
- RandomForestClassifier
  At cv fold 3 the cv score is 0.9621422438099153 and the Accuracy Score is 0.9551094585561497

- LGBMClassifier
  At cv fold 3 the cv score is 0.9630195989994622 and the Accuracy Score is 0.9513285427807486

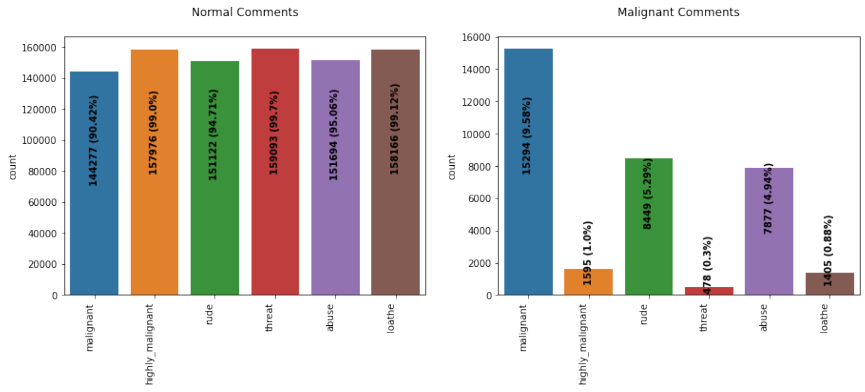
Key Metrics for success in solving problem under consideration

- Accuracy Score: Metric that summarizes the performance of a classification model as the number of correct predictions divided by the total number of predictions.
- Confusion Matrix: Performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.
- Classification Report: Displays your model's precision, recall, F1 score and support. It provides a better understanding of the overall performance of our trained model.
- Hamming Loss: It is the fraction of labels that are incorrectly predicted.
- Log Loss: It is indicative of how close the prediction probability is to the corresponding actual/true value

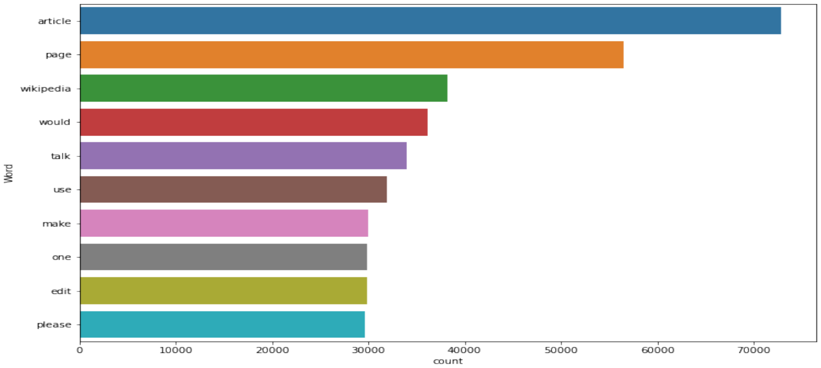
Visualization



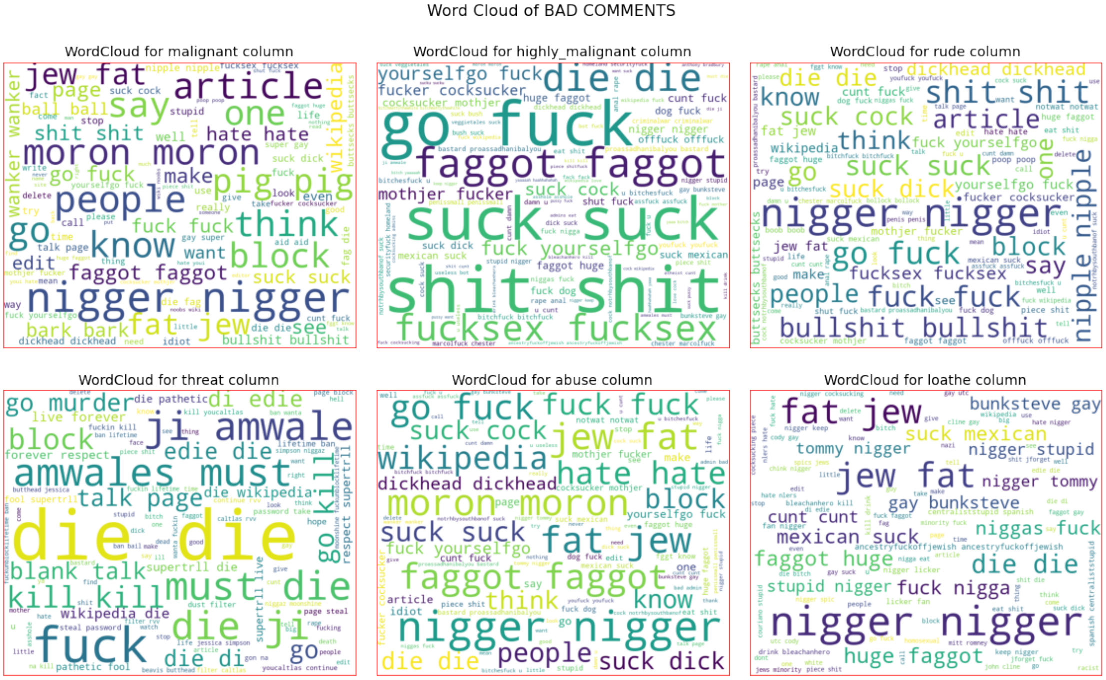
Segmentation of comments based on the nature.



Countplot of comments based on nature



Most frequent words used in comments



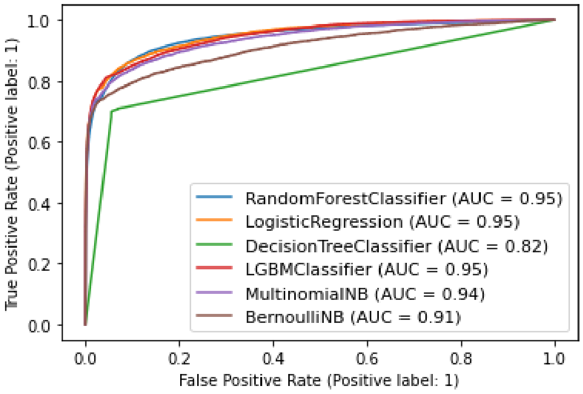
Word Cloud for loud words in bad comments

Observations:

- Malignant comments are comparatively higher than other features in the dataset. Rude and abusive comments are also more visible in the comments.
- Comments which are having threatening are comparatively low in the train dataset.
- As per the data, normal comments which are not malignant are having almost normal distribution for each type of features in the dataset.
- ['article', 'page', 'Wikipedia'] are the words which are most frequent in the comments.
- The word cloud of the bad comments shows the loud words which have used by the users. The presence of these words are making the comments malignant.

Interpretation of the Results

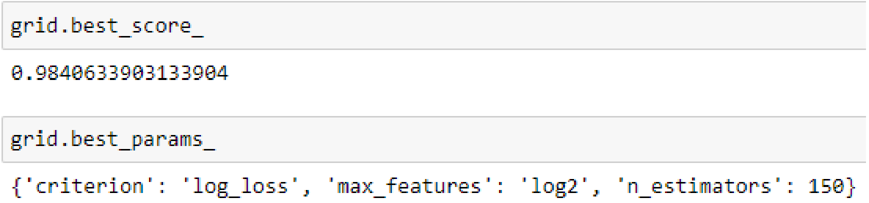
AUC ROC Curve



AUC ROC Curve

The models ['rfc','lr','lgb'] are performing well and providing the maximum ROC AUC Score. Since the RandomForest Classifier model is performing slightly better than all other models, we can consider the rfc model as the best performing model.

Hyperparameter Tuning



```
grid.best_score_
```

```
0.9840633903133904
```

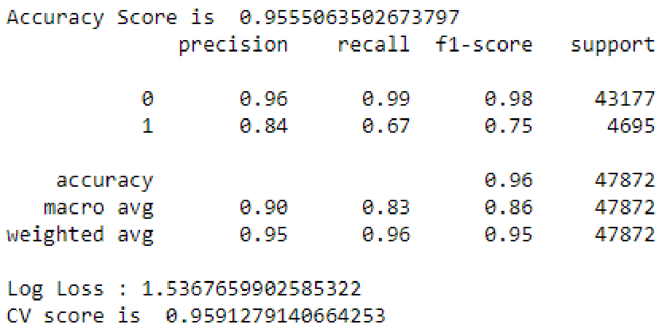
```
grid.best_params_
```

```
{'criterion': 'log_loss', 'max_features': 'log2', 'n_estimators': 150}
```

Best Score and Parameters

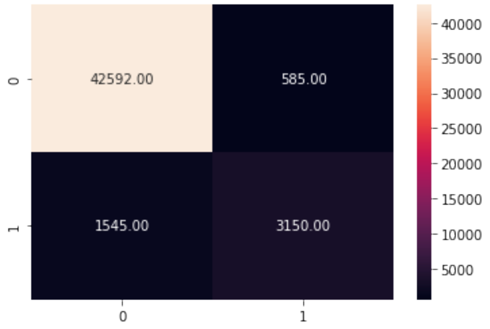
After the hyper parameter tuning the model is performing slightly better.

Final Model



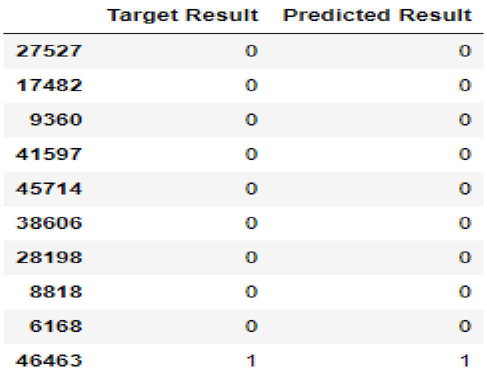
```
Accuracy Score is  0.9555063502673797
             precision    recall  f1-score   support

          0       0.96      0.99      0.98     43177
          1       0.84      0.67      0.75      4695

   accuracy                           0.96     47872
  macro avg       0.90      0.83      0.86     47872
weighted avg       0.95      0.96      0.95     47872

Log Loss : 1.5367659902585322
CV score is  0.9591279140664253
```

Test Performance



Confusion Matrix

Now we have trained our model and it is ready to test with the test data and actual data to cross verify the performance.

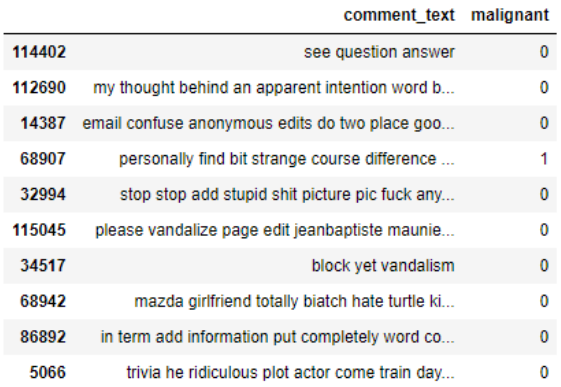


	Target Result	Predicted Result
27527	0	0
17482	0	0
9360	0	0
41597	0	0
45714	0	0
38606	0	0
28198	0	0
8818	0	0
6168	0	0
46463	1	1

Testing the model with prediction

Our model is performing well with predictions and provided accurate results.

Predicting the target for test data



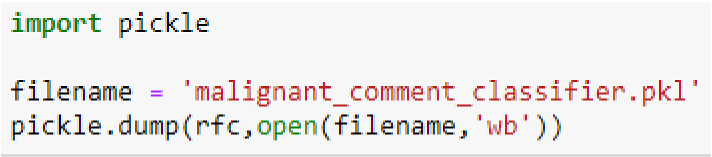
	comment_text	malignant
114402	see question answer	0
112690	my thought behind an apparent intention word b...	0
14387	email confuse anonymous edits do two place goo...	0
68907	personally find bit strange course difference ...	1
32994	stop stop add stupid shit picture pic fuck any...	0
115045	please vandalize page edit jeanbaptiste maunie...	0
34517	block yet vandalism	0
68942	mazda girlfriend totally biatch hate turtle ki...	0
86892	in term add information put completely word co...	0
5066	trivia he ridiculous plot actor come train day...	0

Predicted results for test dataset

We have saved the new dataset as ‘Prediction on Test Dataset of Malignant Comments.csv’.

The RandomForest Classifier (rfc) model is providing an accuracy score of 95.55% with a cross validation mean score of 95.91%.

Saving the best model



```
import pickle

filename = 'malignant_comment_classifier.pkl'
pickle.dump(rfc,open(filename,'wb'))
```

We have saved the machine learning model for future predictions. We have serialized and saved the binary file as “malignant_comment_classifier.pkl” using the pickle library.

CONCLUSION

Key Findings and Conclusions of the Study

With the help of data science and machine learning, we were able to create a machine learning model using RandomForest algorithm, which can predict whether a user comment is malignant or not.

Now this model can be used to analyse whether the comment is malignant or not.

Learning Outcomes of the Study in respect of Data Science

Due to the possibility of abuse or harassment, conversational toxicity is a problem that can make people stop speaking truthfully and stop asking for other people's thoughts. This project aims to utilise deep learning to identify toxicity in text, which might be used to help prevent users from submitting potentially harmful remarks, build more polite arguments when engaging in dialogue with others, and assess the toxicity of other users' comments. This was done using the help of data science, machine learning and NLP.

To be able to differentiate and distinguish harassing comments and cyberbullying, which we term toxic comments, from regular remarks, it is vital for data scientists to study and understand this type of online harassment. For users who could receive alerts and filter inappropriate content, as well as for moderators of public platforms, automatic recognition of hazardous contents in online forums and social media is a helpful feature. The current work was prompted by the need for more sophisticated procedures and approaches to enhance detection of various forms of online comments.

Limitations of this work and Scope for Future Work

Limitations

- The data was vast and included a large number of comments from users along with their nature as features, but it is still lacking many of the important features that could have been impacted detecting whether a comment is malignant or not.
- Even though we have identified the problem and proposed a powerful machine learning model, the technology is yet to integrate algorithms like these to the social media platforms as we don't have an on-time filtering system, which can detect the occurrence of abusive or hatred comments.
- We tried all the possible words and combination of words in analysing and detecting malignant comments. But still there are many other types of comments are left out which can potentially be considered as offensive such as comments which are malignant but conveyed in an indirect way, such as dark humour.
- Even with this powerful system, we can only detect and prevent the user from writing a malignant comment, but there are other platforms which are not yet advanced to implement systems like these. So, we can only limit the prevention of malignant comments arising on the social media platforms. We are not yet close to completely removing it from the system.

Scope

- With more meaningful data and impactful features, in future we will be able to build much more powerful machine learning model which can detect the nature of a comment made by user with more precision and accuracy.
- As the technology is advancing, we may be able to find ways to prevent the users from using malignant comments at the time when a user tries to write it instead of reporting or banning the comment once after it was posted.
- We should also work on making the users aware of the impact of using malignant comments on social media platforms and how it is impacting the people who are reading the comments. We should be able to remove the thought of a user to write malignant comments by making them aware of the consequences.

Thank You