Fliprobo

Malignant Comments Classification Model

Report

Submitted by:

Arjun Verma,

Intern Data Scientist

ACKNOWLEDGEMENT

I would like to express my greatest appreciation to the all individuals who have helped and supported me throughout the project. I am thankful to Fliprobo team for their ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project.

A special acknowledgement goes to my institute Datatrained who helped me in completing the project and learning concepts.

I wish to thank my parents as well for their undivided support and interest who inspired me and encouraged me to go my own way, without whom I would be unable to complete my project.

Below following are the other references:

www.towardsdatascience.com

www.medium.com

www.stackoverflow.com

Datatrained lectures

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, cyber bullying, 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 cyber bullying 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 inoffensive, 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 cyber bullying.

➢ Conceptual Background of the Domain Problem

There are various many platform where people tried to find out good friends by posting content and some of send comments by checking his/her post but some of comments make 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.

➢ Review of Literature

Data has been provided by Fliprobo to make model with having dataset constraint 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.

➢ Motivation for the Problem Undertaken

Genuinely it's a need of the any social media site to complete their goal with many people with people satisfaction. Hence this model can brings higher revenue because we can detect hatred and malignant comments through this model.

➢ Mathematical/ Analytical Modeling of the Problem

Data is statistically analysed through TFIDF vectorization techniques.. Graphical modelling done through seaborn and matplotlib to understanding how different features impact dataset.

Statistical models used

➢ Logistics Regression

➢ Multinomial Naïve Bayes

➢ Data Sources and their formats

Dataset 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.

```
'id', 'comment_text', 'malignant', 'highly_malignant',
'rude', 'threat', 'abuse', 'loathe'
```

Dataframe Description:

- **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.

	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

<u>We have done feature engineering for preprocessing dataset and get below informations</u>

Dataset Information

'id', 'comment_text' `are objects columns while rest are predictors columns having binary classifications.`

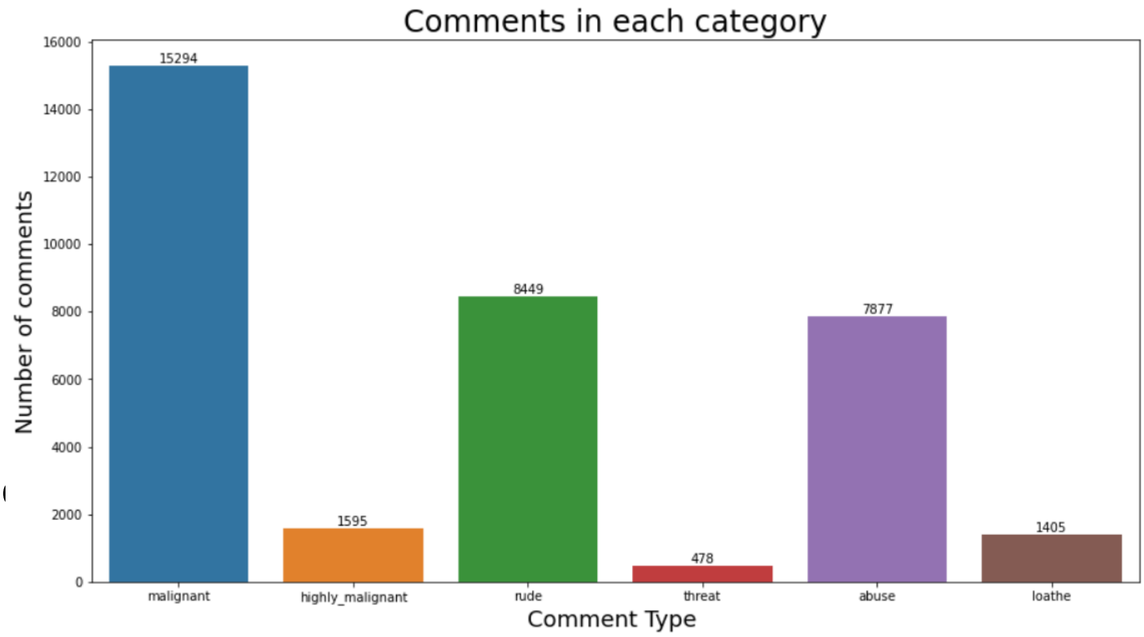
Checking Null Values of the dataset

Dataset having no null values.

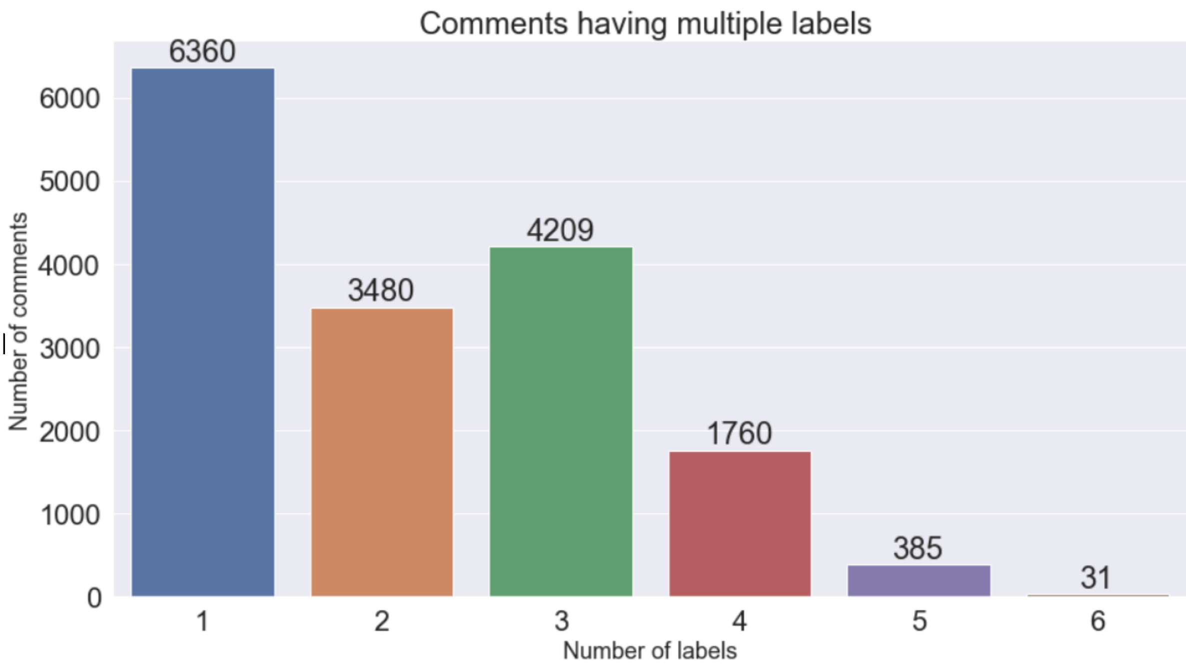
Dataset having 0 duplicated values.

Visualization of important features for understanding

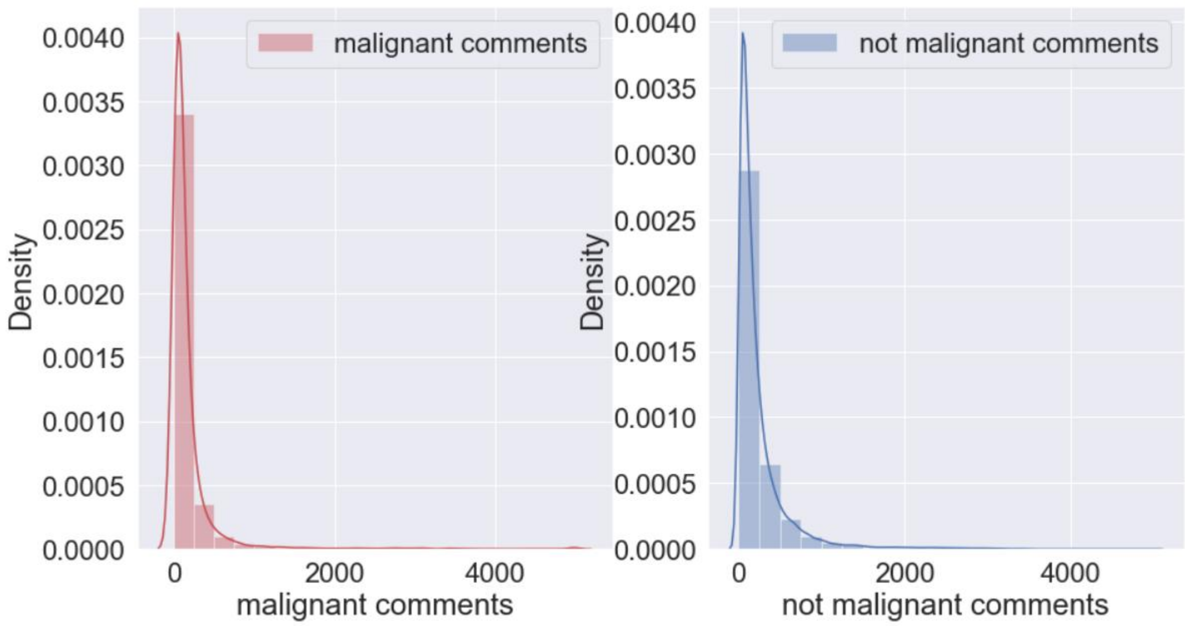
Comments in each category



Comments which are having multiple labels count.



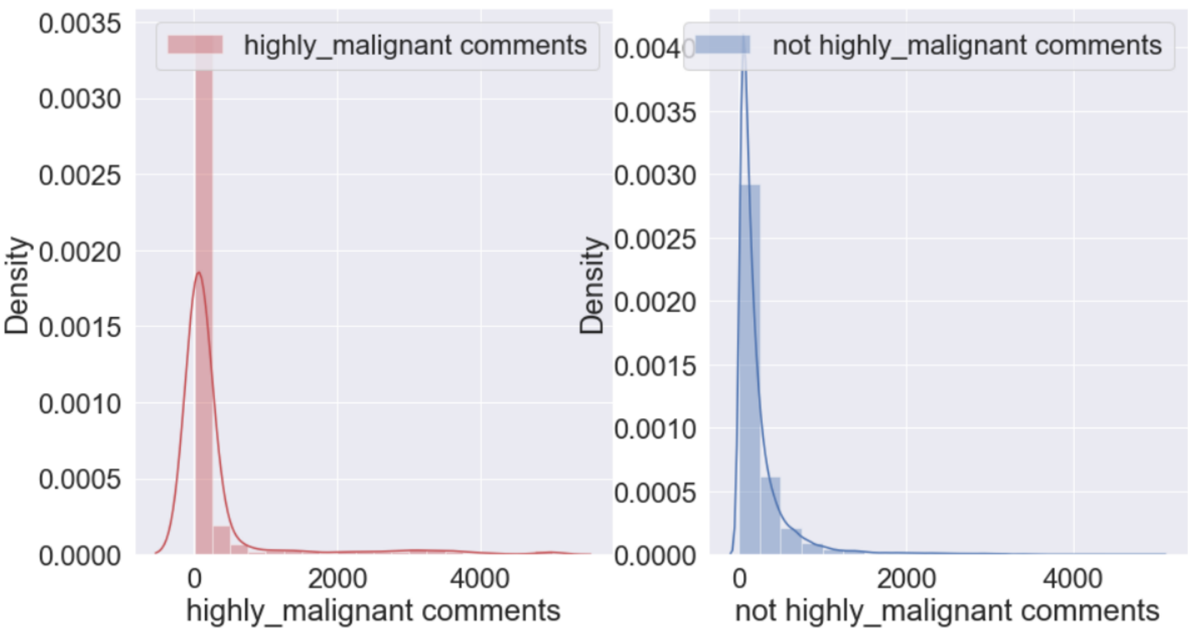
Malignant comments vs not malignant comments



Observations:

Malignant comments are low in numbers as per not malignant comments with average ratio 0.8:1.0.

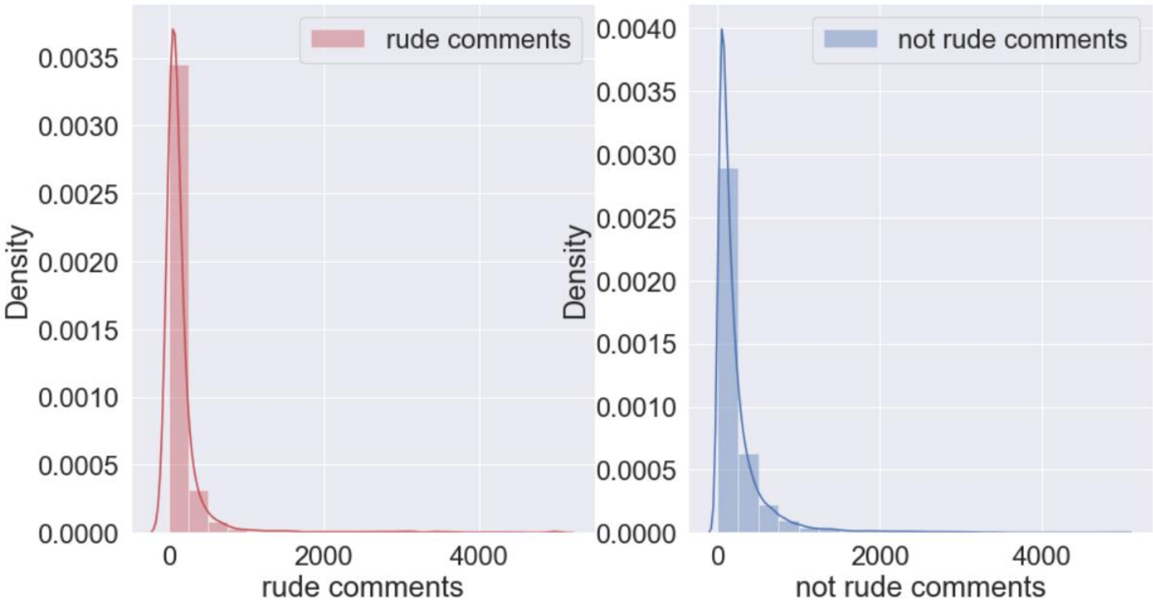
Highly Malignant comments vs not highly malignant comments



Observations:

Malignant comments are low in numbers as per not malignant comments with average ratio 0.5:1.0.

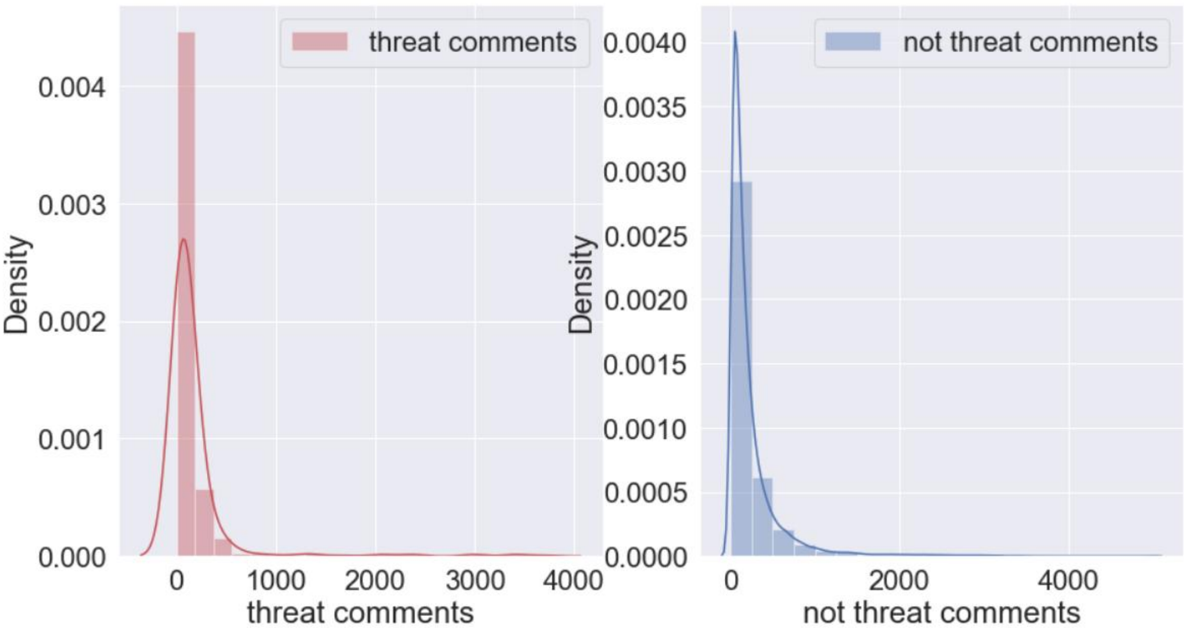
Rude comments vs not rude comments



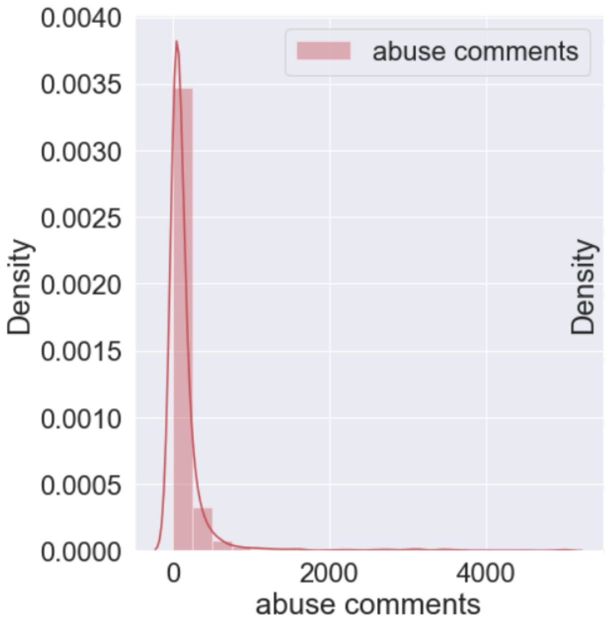
Observations:

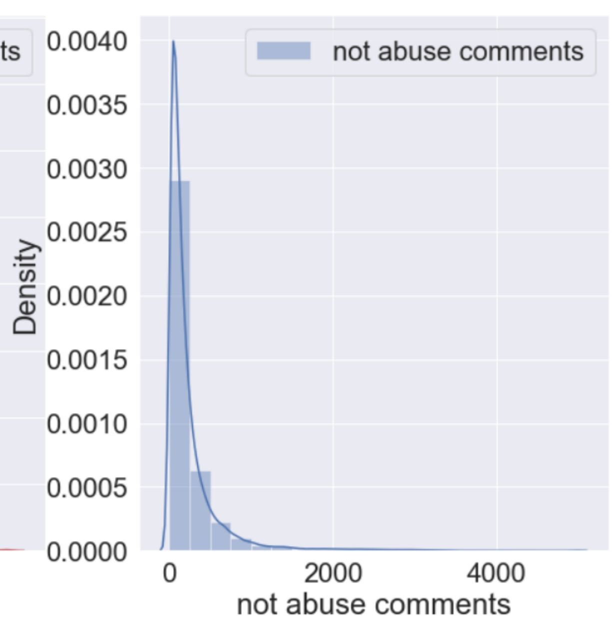
Rude comments having higher counts as per density of not rude comments.

Threat comments vs not threat comments

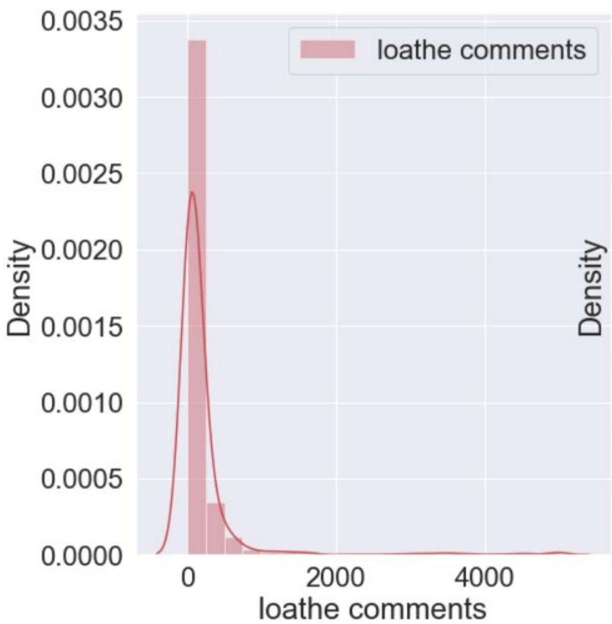


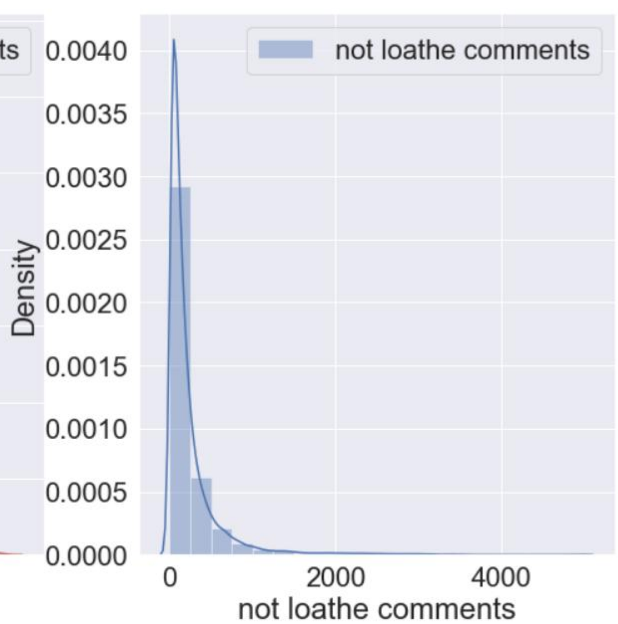
Abuse comments vs not abuse comments



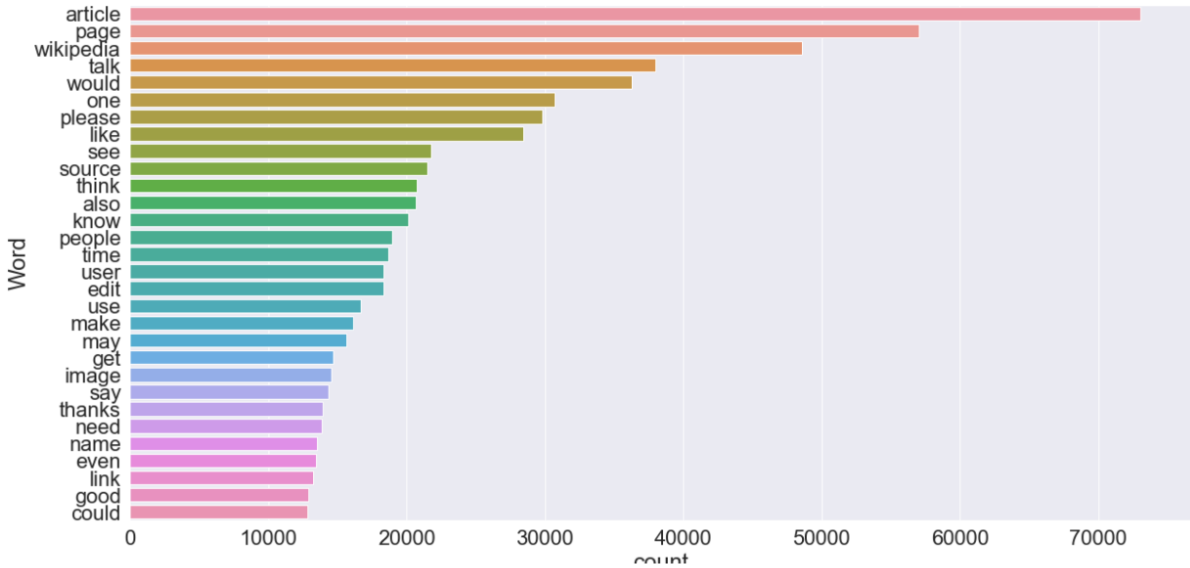


Loathe comments vs not loathe comments

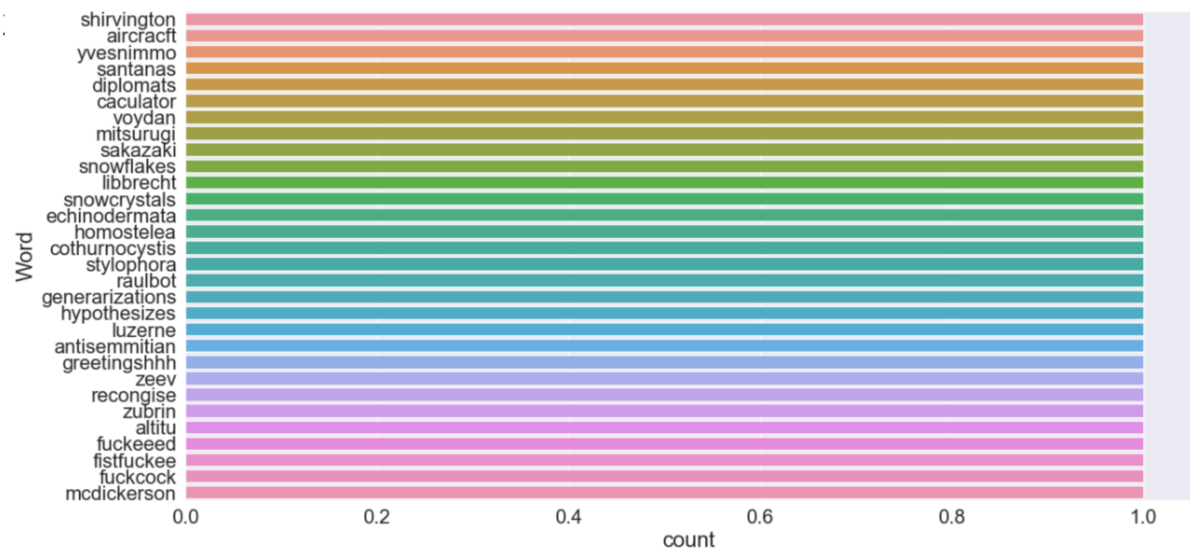




30 most used words



30 least used words



Word clouds of label 1 comments

Word clouds of label 0 comments

Model Building and Predictions

LogisticRegression

```
: model = OneVsRestClassifier(LogisticRegression()).fit(x_train,y_train)
  y_pred = model.predict(x_test)
  print('Accuracy Score',accuracy_score(y_pred, y_test))
  print('Classification Report: \n', classification_report(y_pred, y_test) )
  print('Confusion Matrix: \n', multilabel_confusion_matrix(y_test,y_pred) )
```

```
Accuracy Score 0.9175043240668789
Classification Report:
               precision    recall  f1-score   support

           0       0.57      0.92      0.70      2369
           1       0.20      0.58      0.29       125
           2       0.60      0.91      0.72      1359
           3       0.09      0.56      0.15        16
           4       0.47      0.82      0.60      1123
           5       0.15      0.64      0.25        76

   micro avg       0.52      0.88      0.65      5068
   macro avg       0.35      0.74      0.45      5068
weighted avg       0.54      0.88      0.67      5068
 samples avg       0.04      0.05      0.05      5068

Confusion Matrix:
 [[[35863   193]
  [ 1661  2176]]

 [[39474    53]
  [  294    72]]

 [[37699   124]
  [  835  1235]]

 [[39781     7]
  [   96     9]]

 [[37744   205]
  [ 1026   918]]

 [[39546    27]
  [  271    49]]]
```

Multinomial NB

```
model = OneVsRestClassifier(MultinomialNB()).fit(x_train,y_train)
y_pred = model.predict(x_test)
print('Accuracy Score',accuracy_score(y_pred, y_test))
print('Classification Report: \n', classification_report(y_pred, y_test) )
print('Confusion Matrix: \n', multilabel_confusion_matrix(y_test,y_pred) )
```

```
Accuracy Score 0.8996565813551249
Classification Report:
               precision    recall  f1-score   support

           0       0.17      0.99      0.30       672
           1       0.00      0.00      0.00         0
           2       0.10      0.98      0.18       208
           3       0.00      0.00      0.00         0
           4       0.04      0.96      0.07        77
           5       0.00      0.00      0.00         0

   micro avg       0.11      0.99      0.20       957
   macro avg       0.05      0.49      0.09       957
weighted avg       0.15      0.99      0.25       957
 samples avg       0.01      0.02      0.01       957

Confusion Matrix:
 [[[36051     5]
  [ 3170   667]]

 [[39527     0]
  [  366     0]]

 [[37819     4]
  [ 1866   204]]

 [[39788     0]
  [  105     0]]

 [[37946     3]
  [ 1870    74]]

 [[39573     0]
  [  320     0]]]
```

BernouliNB

```
BernoulliNB()
Training accuracy is :  0.5592266462480857
Testing accuracy is : 0.5212148280482358
-------------------------------------------------------------------------------
Classification Report:
               precision    recall  f1-score   support

           1       0.56      0.74      0.64      1714
           2       0.50      0.95      0.66      1195
           3       0.44      0.82      0.57      1186
           4       0.38      0.65      0.48      1289
           5       0.72      0.28      0.40      5811

    accuracy                           0.52     11195
   macro avg       0.52      0.69      0.55     11195
weighted avg       0.60      0.52      0.49     11195

Confusion Matrix:
 [[1265   87  108  119  135]
 [  18 1131    6   18   22]
 [  15    0  975   74  122]
 [  58    0   35  839  357]
 [ 909 1025 1089 1163 1625]]
--------------------------------------------------------------------------------
Cross value score
cv score 0.5069014797046475 at 2 cross fold
cv score 0.5171926507385952 at 3 cross fold
cv score 0.5203008961799894 at 4 cross fold
--------------------------------------------------------------------------------
```

Model Building Results

Best models

- Logistics Regression : Model shows highest accuracy score in training and testing accuracy hence we can consider it.

- MultinomialNB: Model shows little lower accuracy with respect to logistics regression hence we cannot consider it.

- Bernouli: Model shows low accuracy score in training and testing accuracy hence we cannot consider it.

Final Model Logistics Regression

Hyper Parameter Tuning is applied to Logistic Regression model as it is giving best accuracy in all used ML algorithms

```
params ={"estimator__penalty":["l2","none"],
          "estimator__fit_intercept":[True,False],
          "estimator__solver":["newton-cg","lbfgs","liblinear","sag","saga"]}
```

```
model_tunning = GridSearchCV(estimator = model_to_set, param_grid=params, cv = 3)
model_tunning.fit(x_train, y_train)
model_tunning.best_params_
```

```
{'estimator__fit_intercept': True,
 'estimator__penalty': 'l2',
 'estimator__solver': 'liblinear'}
```

Tuning with Parameters

```
model = LogisticRegression(fit_intercept = 'True', penalty = 'l2', solver = 'liblinear')
final_model = OneVsRestClassifier(model).fit(x_train, y_train)
prediction = final_model.predict(x_test)
prediction2 = final_model.predict(x_train)
print('Accuracy of Testing ',accuracy_score(prediction, y_test))
print('Accuracy of Training ',accuracy_score(prediction2, y_train))
print('Classification Report: \n', classification_report(prediction, y_test) )
print('Confusion Matrix: \n', multilabel_confusion_matrix(y_test,prediction) )
```

```
Accuracy of Testing  0.9175043240668789
Accuracy of Training  0.9240879693845151
Classification Report:
               precision    recall  f1-score   support

           0       0.57      0.92      0.70      2369
           1       0.20      0.58      0.29       125
           2       0.60      0.91      0.72      1359
           3       0.09      0.56      0.15        16
           4       0.47      0.82      0.60      1123
           5       0.15      0.64      0.25        76

   micro avg       0.52      0.88      0.65      5068
   macro avg       0.35      0.74      0.45      5068
weighted avg       0.54      0.88      0.67      5068
 samples avg       0.04      0.05      0.05      5068

Confusion Matrix:
 [[[35863   193]
  [ 1661  2176]]

 [[39474    53]
  [  294    72]]

 [[37699   124]
  [  835  1235]]

 [[39781     7]
  [   96     9]]

 [[37744   205]
  [ 1026   918]]

 [[39546    27]
  [  271    49]]]
```

Model Deployment

4.4 Deploying the model

```
import pickle
filename = 'comment_project.pkl'                    # model name
pickle.dump(final_model, open(filename, 'wb'))          # operation to deploy model
```

4.5 Loading model

```
load_model =  pickle.load(open('comment_project.pkl', 'rb'))    # loading deployed model
result = load_model.score(x_test, y_test)
print(result)
```

```
0.9175043240668789
```

4.6 Conclusion

```
original = np.array(y_test)
predicted = np.array(load_model.predict(x_test))
# convert columns in to np.array
```

Model testing with test data

5. For Test data

```
x_test = df_test['pre_test_comments']
```

```
#converting text to numerical through n-gram tfidf vectorizer
train_word_features_test=word_vectorizer.transform(x_test)
```

```
train_word_features_test.shape, train_word_features.shape
```

```
((153164, 165384), (159571, 165384))
```

```
pred_for_test = np.array(load_model.predict(train_word_features_test))
```

```
pred_for_test
```

```
array([[1, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0],
       ...,
       [0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 0, 0]])
```

```
malignant = []
highly_malignant = []
rude = []
threat = []
abuse = []
loathe = []

for i in range(pred_for_test.shape[0]):
    malignant.append(pred_for_test[i][0])
    highly_malignant.append(pred_for_test[i][1])
    rude.append(pred_for_test[i][2])
    threat.append(pred_for_test[i][3])
    abuse.append(pred_for_test[i][4])
    loathe.append(pred_for_test[i][5])

print(len(malignant))
print(len(highly_malignant))
print(len(rude))
print(len(threat))
print(len(abuse))
print(len(loathe))
```

```
153164
153164
153164
153164
153164
153164
```

```
df_test['malignant'] = malignant
df_test['highly_malignant'] = highly_malignant
df_test['rude'] = rude
df_test['threat'] = threat
df_test['abuse'] = abuse
df_test['loathe'] = loathe
```

```
df_test.head()
```

	comment_text	pre_test_comments	malignant	highly_malignant	rude	threat	abuse	loathe
0	Yo bitch Ja Rule is more succesful then you'll...	yo bitch ja rule succesful ever whats hating s...	1	0	1	0	1	0
1	== From RfC == \n\n The title is fine as it is...	rfc title fine imo	0	0	0	0	0	0
2	" \n\n == Sources == \n\n * Zawe Ashton on Lap...	sources zawe ashton lapland	0	0	0	0	0	0
3	:If you have a look back at the source, the in...	look back source information updated correct f...	0	0	0	0	0	0
4	I don't anonymously edit articles at all.	anonymously edit article	0	0	0	0	0	0

```
df_test.to_csv("test.csv")
```

➢ Hardware and Software Requirements and Tools Used

Operating System: Window 11

RAM: 8 GB

Processor: i5 10th Generation

Software: Jupyter Notebook

Python Libraries: Mainly

Pandas: This library used for dataframe operations .

Numpy: This library gives statistical computation for smooth functioning .

Matplotlib: Used for visualization.

Seaborn: This library is also used for visualization.

Sklearn: This library having so many machine learning module and we can import them from this library.

Pickle: This is used for deploying the model.

CONCLUSION

➢ Key Findings and Conclusions of the Study

This project has built a model that can predict malignant comments of the people from various social media sites.

➢ Learning Outcomes of the Study in respect of Data Science

Data cleaning is the most important part in this model building as we see above there are so many stopwords, punkt and symbols values we remove it from the dataset dataset for better observations. This project has gives so much information about parameters that how a single parameter can hit comments.

➢ Limitations of this work and Scope for Future Work

Model work with similar parameters as we build the whole model if some of the parameters missed then we need to train model with remains parameter after that we can predict upcoming comments of people hence we need to up to date all the parameters as per training dataset.