FLIP ROBO

Malignant comment classifier

Submitted by:

Sabuj Ghosh

ACKNOWLEDGMENT

NLTK documentation

Sklearn Documentation.

INTRODUCTION

- Business Problem Framing

  Automated detection of malignant comments.

- Conceptual Background of the Domain Problem

  Insult, abuse, threat from strangers and known people on online social media is an everyday problem.

- Motivation for the Problem Undertaken
  So that people's harassments can be minimized.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  This is definitely a classification problem with one independent variable and multiple target variable.

- Data Sources and their formats

Provided by fliprobo in csv format.

- Data Preprocessing Done

  Stopwords removed, splitted, lower cased, lemmatized, built corpus and vectorized using TFIDF vectorizer.

- Hardware and Software Requirements and Tools Used

  Hardware: high RAM, how high is not known but at least 8 gb is not enough.

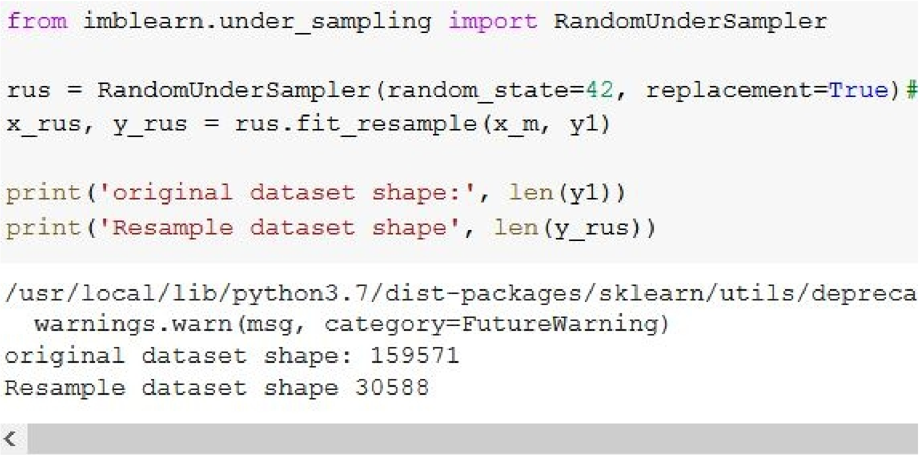
  Software: Sklearn, IMB learn, NLTK.

Model/s Development and Evaluation

- Testing of Identified Approaches (Algorithms)

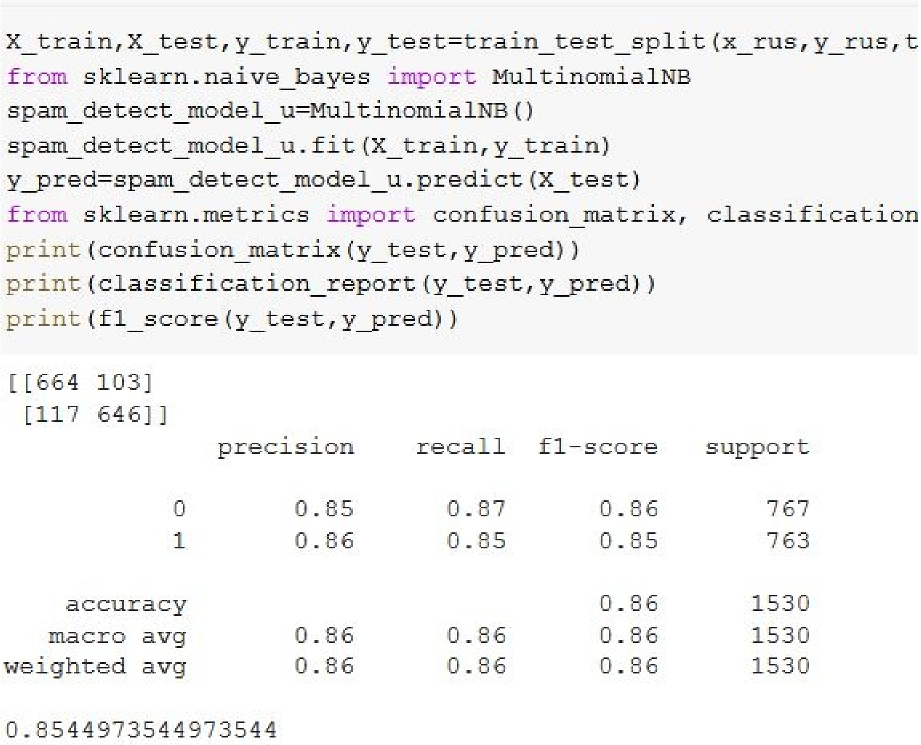
  Various classification algorithms like multinomial Gaussian NB, Logistic Regression, Decision tree, Random forest etc.

- Run and Evaluate selected models



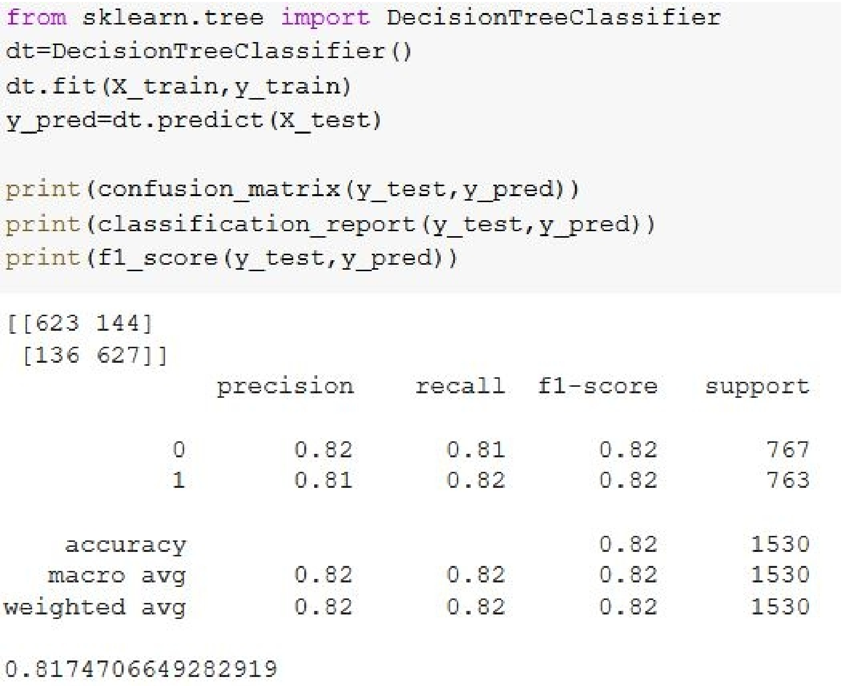
```
from imblearn.under_sampling import RandomUnderSampler

rus = RandomUnderSampler(random_state=42, replacement=True)#
x_rus, y_rus = rus.fit_resample(x_m, y1)

print('original dataset shape:', len(y1))
print('Resample dataset shape', len(y_rus))
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/depreca
  warnings.warn(msg, category=FutureWarning)
original dataset shape: 159571
Resample dataset shape 30588
```



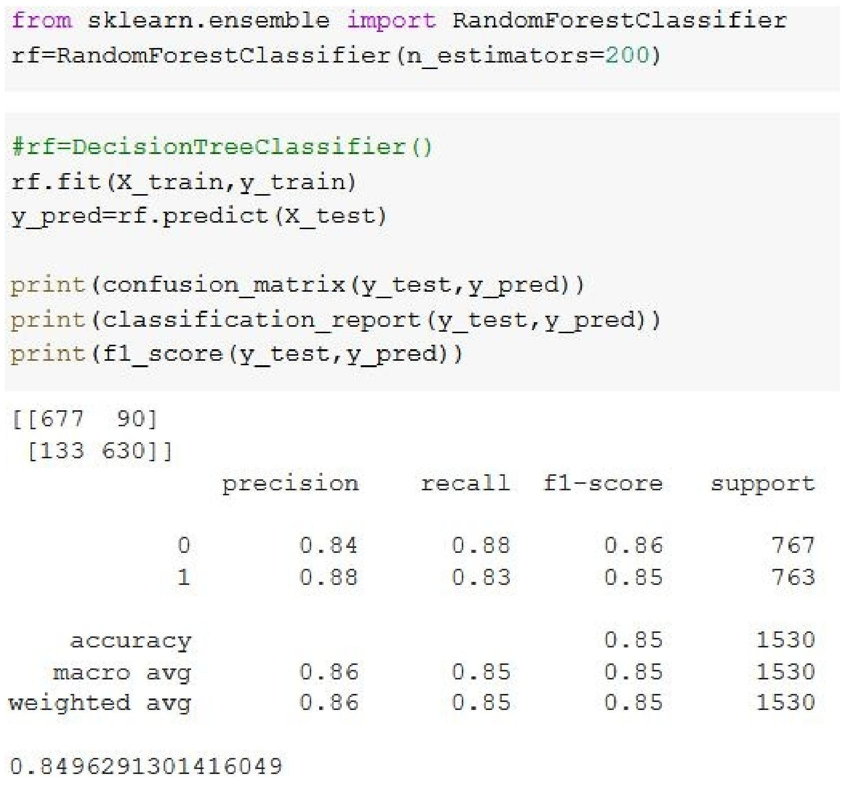
```
X_train,X_test,y_train,y_test=train_test_split(x_rus,y_rus,t
from sklearn.naive_bayes import MultinomialNB
spam_detect_model_u=MultinomialNB()
spam_detect_model_u.fit(X_train,y_train)
y_pred=spam_detect_model_u.predict(X_test)
from sklearn.metrics import confusion_matrix, classification
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(f1_score(y_test,y_pred))
```

```
[[664 103]
 [117 646]]
              precision    recall  f1-score   support

           0       0.85      0.87      0.86       767
           1       0.86      0.85      0.85       763

    accuracy                           0.86      1530
   macro avg       0.86      0.86      0.86      1530
weighted avg       0.86      0.86      0.86      1530

0.8544973544973544
```



```
from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
dt.fit(X_train,y_train)
y_pred=dt.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(f1_score(y_test,y_pred))
```

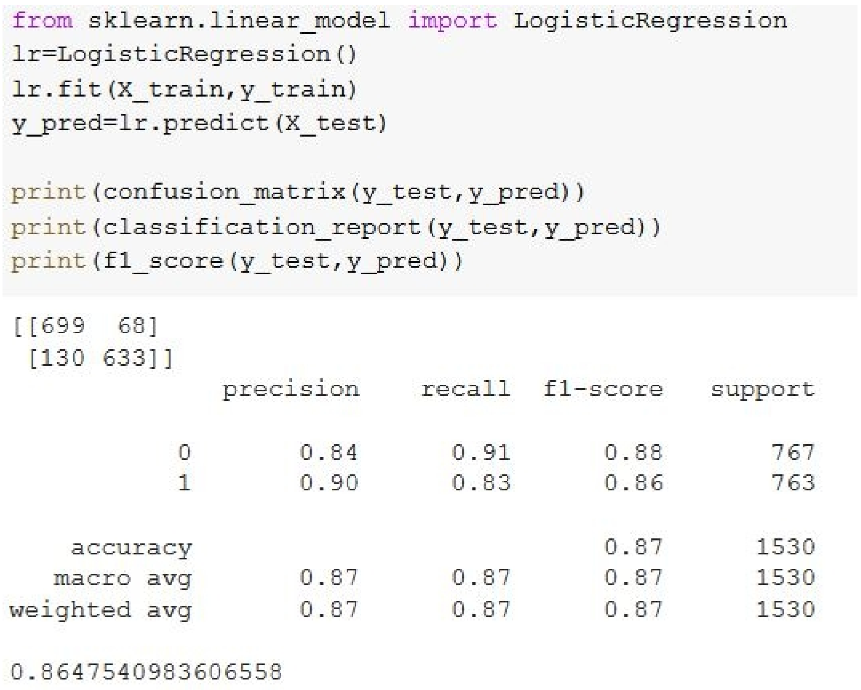
```
[[623 144]
 [136 627]]
              precision    recall  f1-score   support

           0       0.82      0.81      0.82       767
           1       0.81      0.82      0.82       763

    accuracy                           0.82      1530
   macro avg       0.82      0.82      0.82      1530
weighted avg       0.82      0.82      0.82      1530

0.8174706649282919
```



```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=200)
```

```
#rf=DecisionTreeClassifier()
rf.fit(X_train,y_train)
y_pred=rf.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(f1_score(y_test,y_pred))
```

```
[[677  90]
 [133 630]]
              precision    recall  f1-score   support

           0       0.84      0.88      0.86       767
           1       0.88      0.83      0.85       763

    accuracy                           0.85      1530
   macro avg       0.86      0.85      0.85      1530
weighted avg       0.86      0.85      0.85      1530

0.8496291301416049
```



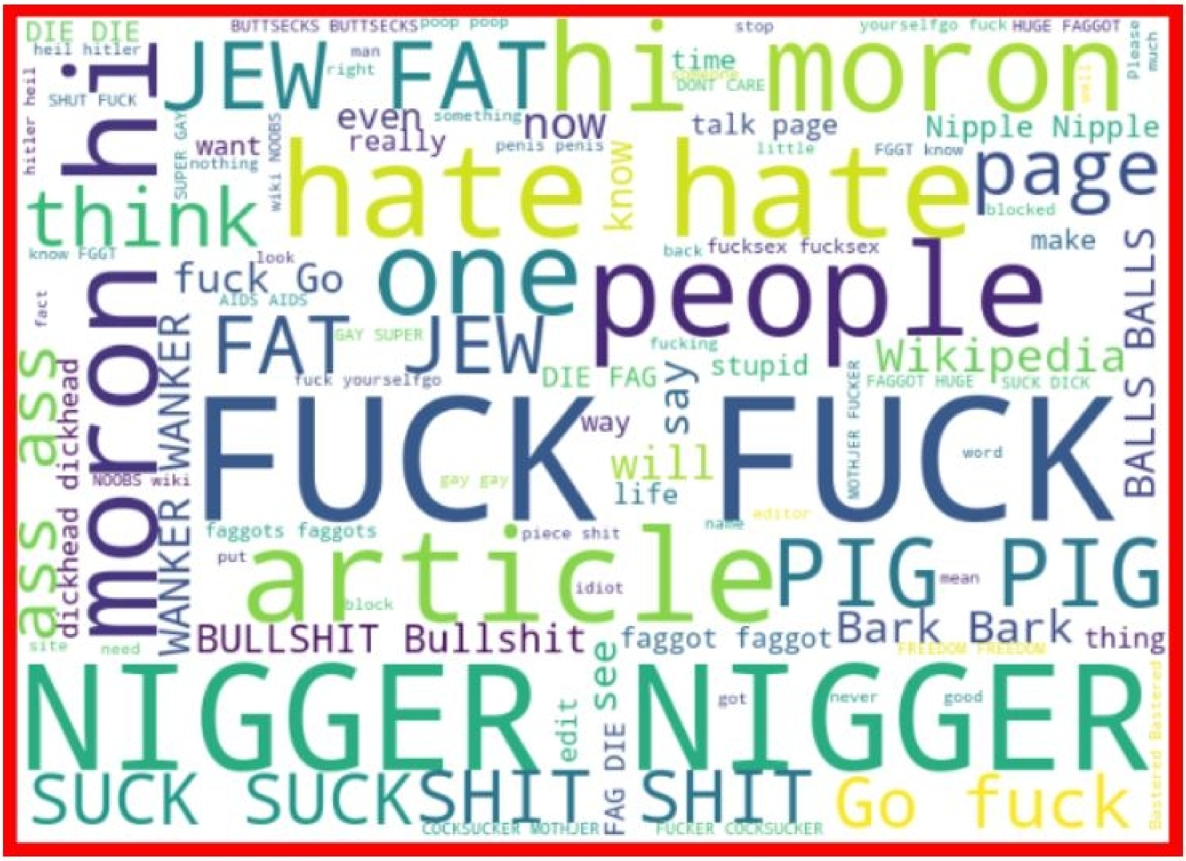
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(X_train,y_train)
y_pred=lr.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(f1_score(y_test,y_pred))
```

```
[[699  68]
 [130 633]]
              precision    recall  f1-score   support

           0       0.84      0.91      0.88       767
           1       0.90      0.83      0.86       763

    accuracy                           0.87      1530
   macro avg       0.87      0.87      0.87      1530
weighted avg       0.87      0.87      0.87      1530

0.8647540983606558
```

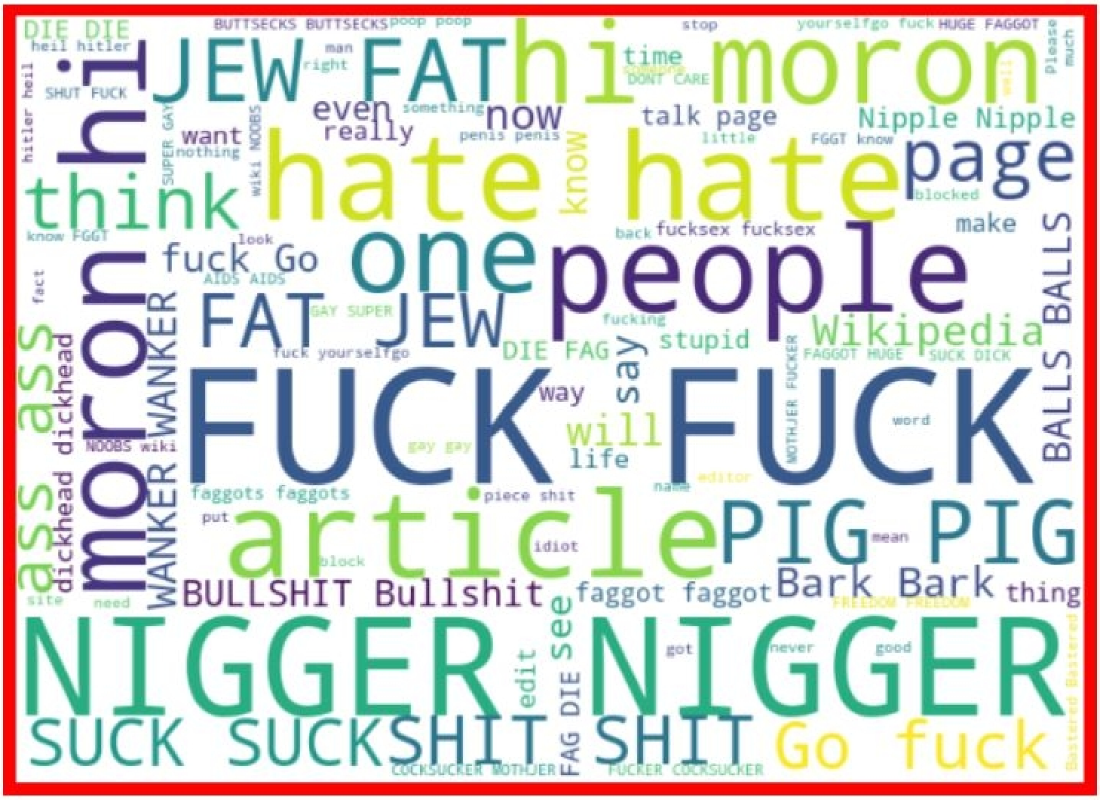
- Key Metrics for success in solving problem under consideration

  F1 score.

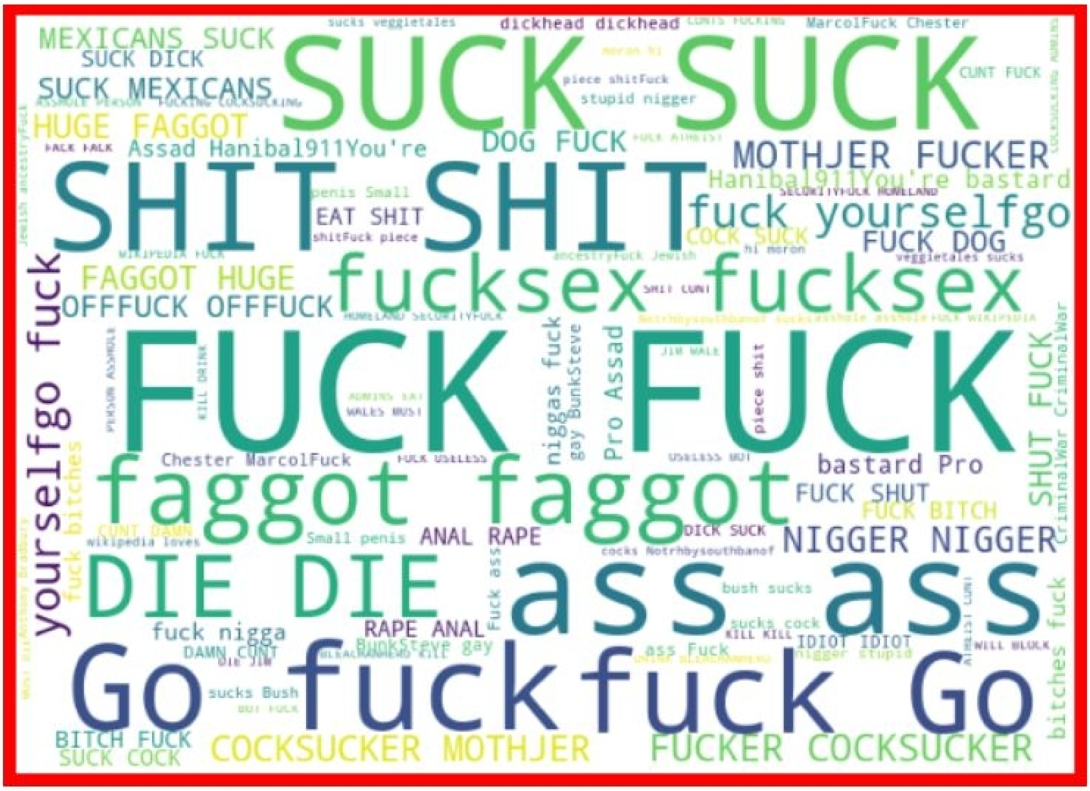
- Visualizations



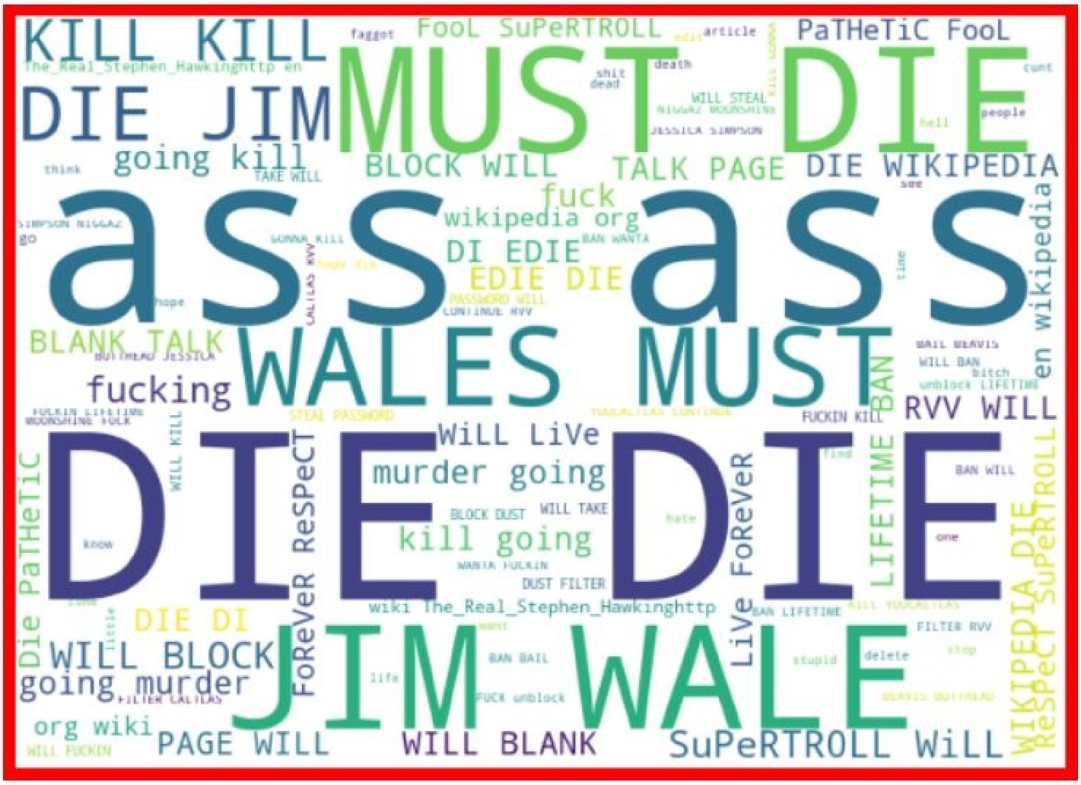
Word cloud for normal comments



Word cloud for malignant comments



Word cloud for highly malignant comments



Word cloud for threatening comments. And more...

Verbal malignancy primarily begins and ends with primordial absurdity towards human sexuality and moral degradation imagined around its sociopolitical boundaries.

The other aspect of malignancy is from body shaming in brute and patriarchal manner.

Words indicative of racism pops up very often.

Rude comments have relatively higher references to female body parts indicative of misogynistic behavioral pattern.

Threat comments tend to be indicative of life threatening words like ‘die’ and ‘murder’ along with body shaming.

Loathing comments tend more towards racial slurs.

- Interpretation of the Results

  Online hate speech can be stopped in automated way.

CONCLUSION

- Key Findings and Conclusions of the Study

  Online hate speech can be stopped in automated way.

- Learning Outcomes of the Study in respect of Data Science

Verbal malignancy primarily begins and ends with primordial absurdity towards human sexuality and moral degradation imagined around its sociopolitical boundaries.

The other aspect of malignancy is from body shaming in brute and patriarchal manner.

Words indicative of racism pops up very often.

Rude comments have relatively higher references to female body parts indicative of misogynistic behavioral pattern.

Threat comments tend to be indicative of life threatening words like ‘die’ and ‘murder’ along with body shaming.

Loathing comments tend more towards racial slurs.