MALIGNANT COMMENT PREDICTION REPORT

Submitted by:

NEETAL TIWARI

ACKNOWLEDGMENT

I would like to express my special thank of gratitude to my SME (Mohd Kashif) as well as my company (Flip Robo Technologies) who gave me the golden opportunity to do this wonderful project on the (malignant comment prediction project) which also helped me to doing lots of research and I came to know about so many things. I am really thankful to them.

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 un offensive, 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

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

Review of Literature

The data set includes:

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.

Motivation for the Problem Undertaken

Over a decade, social networking and social media have been growing in leaps and bounds. Today, people are able to express themselves and their opinions and also discuss among others via these platforms. In such a scenario, 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 toxic comments may be used from one side. These toxic comments may be threatening, obscene, insulting 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.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

```
n [51]: train_df.describe()
```

ut[51]:

	malignant	highly_malignant	rude	threat	abuse	loathe
count	159621.000000	159621.000000	159621.000000	159621.000000	159621.000000	159621.000000
mean	0.095783	0.009999	0.052900	0.002995	0.049336	0.008802
std	0.294295	0.099493	0.223835	0.054641	0.216568	0.093406
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

We can see there are not much information in data describe. There in only count of rows such as 159621, it means there is null values in train dataset.

```
In [52]: test_df.describe()
Out[52]:
```

	id	comment_text
count	153186	153186
unique	153186	153032
top	00001cee341fdb12	#NAME?
freq	1	155

In test dataset both data are object so we can see not much information about the dataset in describe. Only we can see the counts of rows which is 153186 and we can say that there are no null values in test dataset.

Data Sources and their formats

	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	0.0	0.0	0.0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0.0	0.0	0.0	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	0.0	0.0	0.0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0.0	0.0	0.0	0.0	0.0	0.0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0.0	0.0	0.0	0.0	0.0	0.0

We can see in train dataset 8 columns are there.

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.

Data Pre-processing Done

In train dataset there are only 0 and 1 type of data so we can not deal with skewness and outlies.

In data pre-processing I had added two columns length of comments and clean-length of comments, so that we can analysis the data easily. Because it Is not possible to analysis the data by using label encoding in this particular dataset.

ut[83]:

	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	length	clean_length
0	explanation why the edits made under my userna...	0.0	0.0	0.0	0.0	0.0	0.0	264	263
1	d'aww! he matches this background colour i'm s...	0.0	0.0	0.0	0.0	0.0	0.0	112	121
2	hey man, i'm really not trying to edit war. it...	0.0	0.0	0.0	0.0	0.0	0.0	233	233
3	more i can't make any real suggestions on impr...	0.0	0.0	0.0	0.0	0.0	0.0	622	611
4	you, sir, are my hero. any chance you remember...	0.0	0.0	0.0	0.0	0.0	0.0	67	67

n [84]:
```
# Total length removal
print ('Origian Length', train_df.length.sum())
print ('Clean Length', train_df.clean_length.sum())
```

```
Origian Length 62979376
Clean Length 62601017
```

Data Inputs- Logic- Output Relationships

In [53]:
```
train_df.corr().sum().sort_values(ascending=False)
```

Out[53]:
```
abuse               3.252177
rude                3.248776
malignant           3.055547
highly_malignant    2.411608
loathe              2.207477
threat              1.687056
dtype: float64
```

We can see abuse and rude columns are highly correlated and threat is very less correlated.

Model/s Development and Evaluation

Testing of Identified Approaches (Algorithms)

For data analysis I had used five algorithms such as DecisionTreeClassifier, KNeighborsClassifier, RandomForestClassifier, AdaBoostClassifier, LogisticRegression.

Run and evaluate selected models

```
In [99]: # LogisticRegression
LG = LogisticRegression()
LG.fit(x_train, y_train)

y_pred_train = LG.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
y_pred_test = LG.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
Training accuracy is 0.9599495229742065
Test accuracy is 0.9547685175517364
[[42707   247]
 [ 1919  3014]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42954
           1       0.92      0.61      0.74      4933

    accuracy                           0.95     47887
   macro avg       0.94      0.80      0.86     47887
weighted avg       0.95      0.95      0.95     47887
```

In logistic regression accuracy score is 95.47%.

```
[100]: # DecisionTreeClassifier
DT = DecisionTreeClassifier()
DT.fit(x_train, y_train)

y_pred_train = DT.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
y_pred_test = DT.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
Training accuracy is 0.9989170709005316
Test accuracy is 0.9405057740096477
[[41600  1354]
 [ 1495  3438]]
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     42954
           1       0.72      0.70      0.71      4933

    accuracy                           0.94     47887
   macro avg       0.84      0.83      0.84     47887
weighted avg       0.94      0.94      0.94     47887
```

In Decision Tree Classifier accuracy score is 94.05%

```
In [101]: #RandomForestClassifier
RF = RandomForestClassifier()
RF.fit(x_train, y_train)

y_pred_train = RF.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
y_pred_test = RF.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
Training accuracy is 0.9989170709005316
Test accuracy is 0.9565226470649655
[[42404   550]
 [ 1532  3401]]
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     42954
           1       0.86      0.69      0.77      4933

    accuracy                           0.96     47887
   macro avg       0.91      0.84      0.87     47887
weighted avg       0.95      0.96      0.95     47887
```

In Random Forest classifier accuracy score is 95.65%.

```
In [102]: #AdaBoostClassifier
ada=AdaBoostClassifier(n_estimators=100)
ada.fit(x_train, y_train)
y_pred_train = ada.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
y_pred_test = ada.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
Training accuracy is 0.9511428929421663
Test accuracy is 0.9491511266105623
[[42528   426]
 [ 2009  2924]]
              precision    recall  f1-score   support

           0       0.95      0.99      0.97     42954
           1       0.87      0.59      0.71      4933

    accuracy                           0.95     47887
   macro avg       0.91      0.79      0.84     47887
weighted avg       0.95      0.95      0.94     47887
```

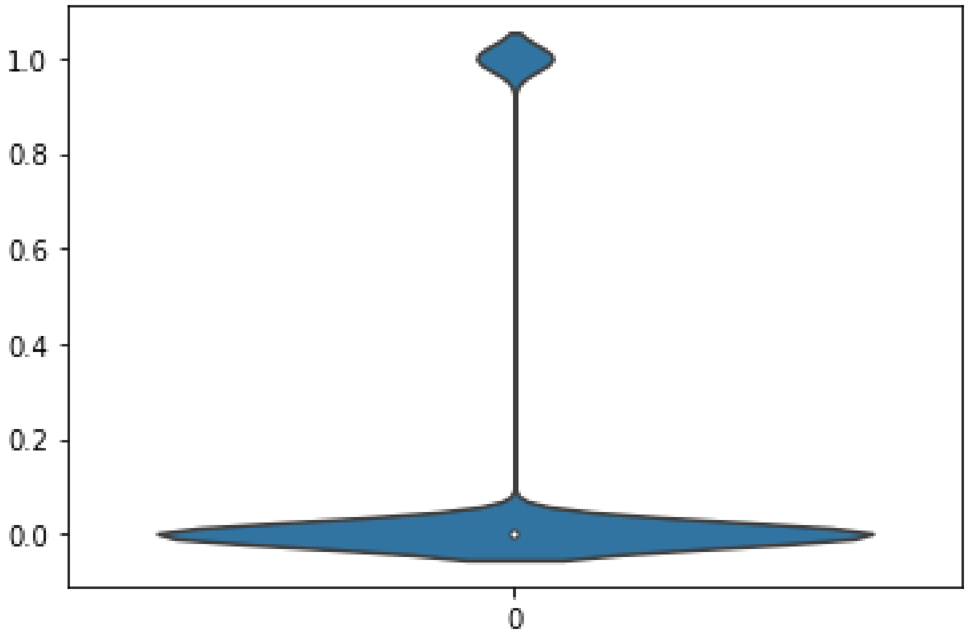
In Ada Boost Classifier accuracy score is 94.91%.

```
In [103]: #KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=9)
knn.fit(x_train, y_train)
y_pred_train = knn.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
y_pred_test = knn.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

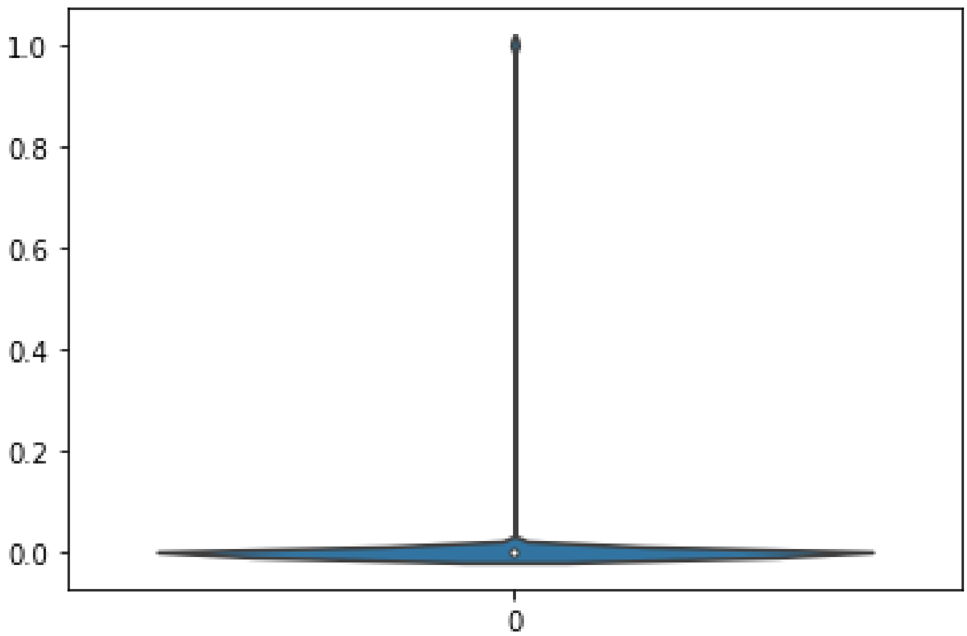
```
Training accuracy is 0.9220112051837399
Test accuracy is 0.9156556059055694
[[42772   182]
 [ 3857  1076]]
              precision    recall  f1-score   support

           0       0.92      1.00      0.95     42954
           1       0.86      0.22      0.35      4933

    accuracy                           0.92     47887
   macro avg       0.89      0.61      0.65     47887
weighted avg       0.91      0.92      0.89     47887
```

IN KNeighbors Classifier accuracy score is 91.56%.

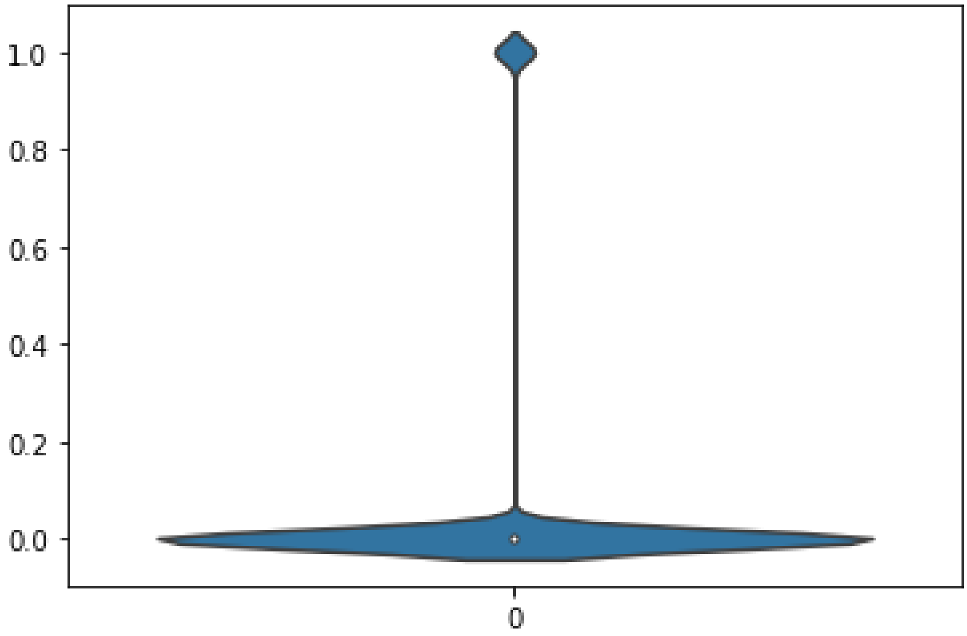
Visualizations



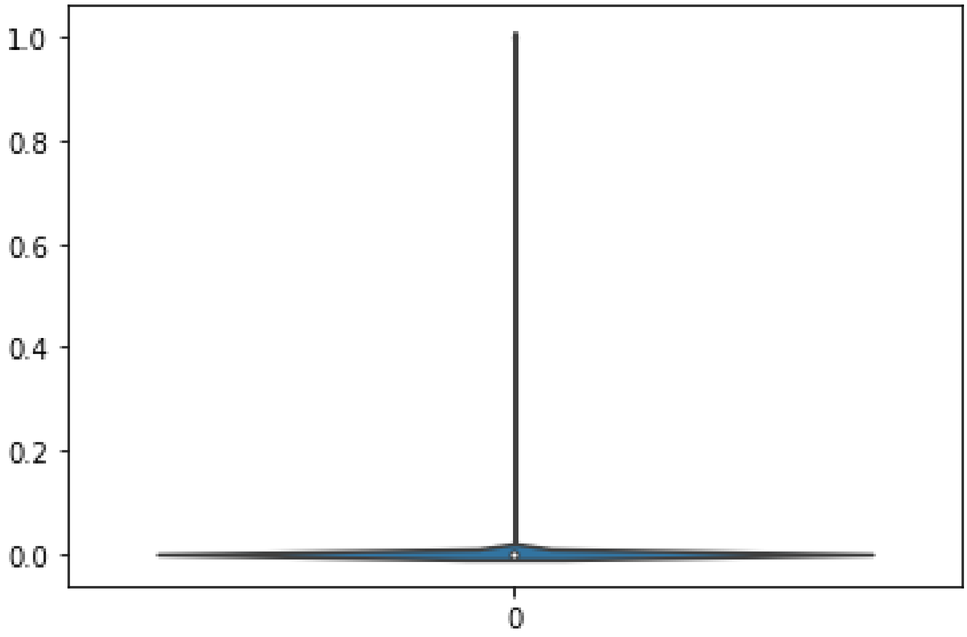
Out of 159621, 144332 comments are not malignant and 15289 comments are malignant.



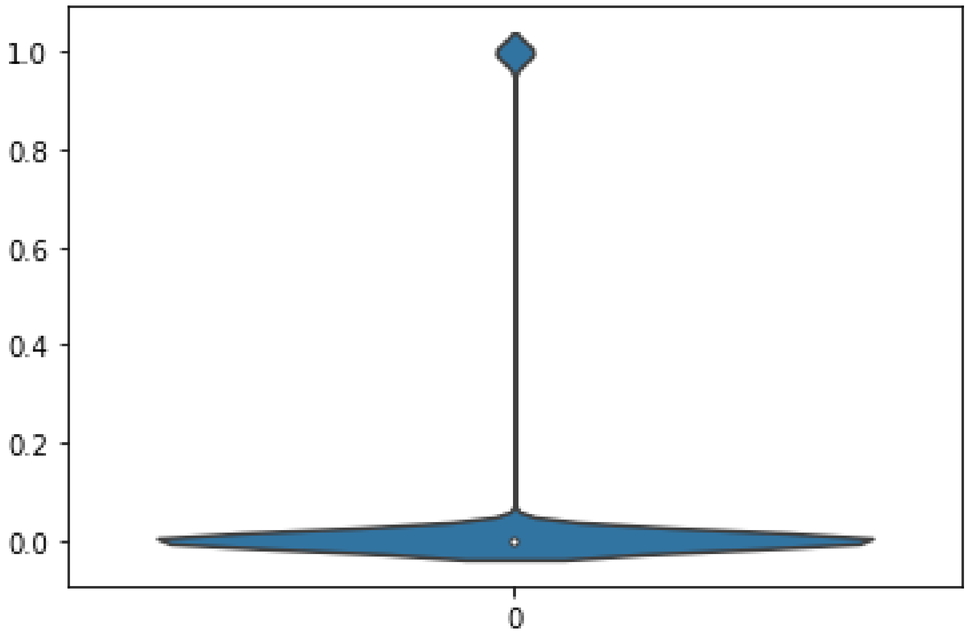
158025 comments are not highly malignant and 1596 comments are highly malignant.



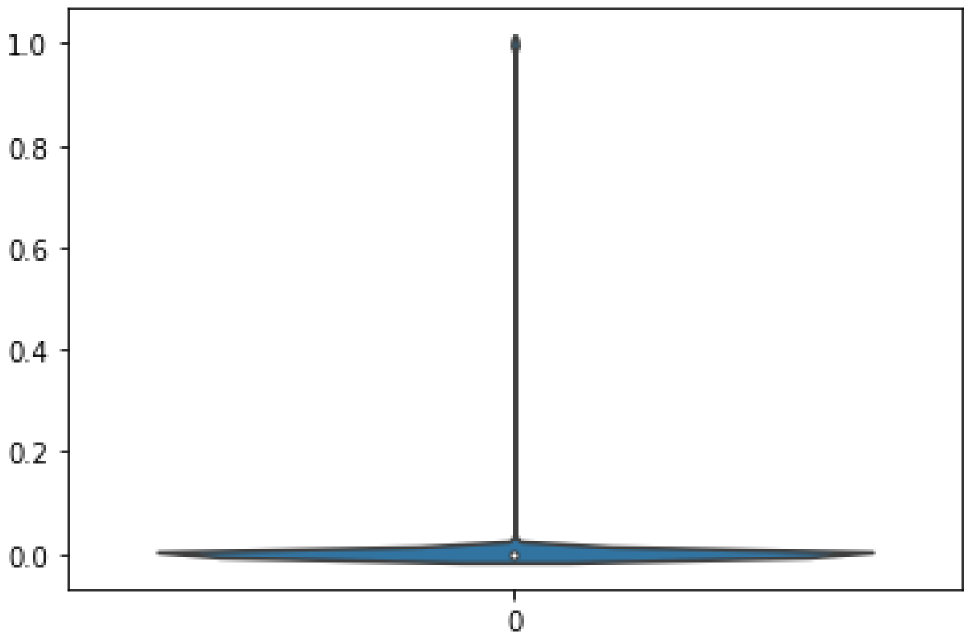
Out of 159621, 8444 comments are rude.



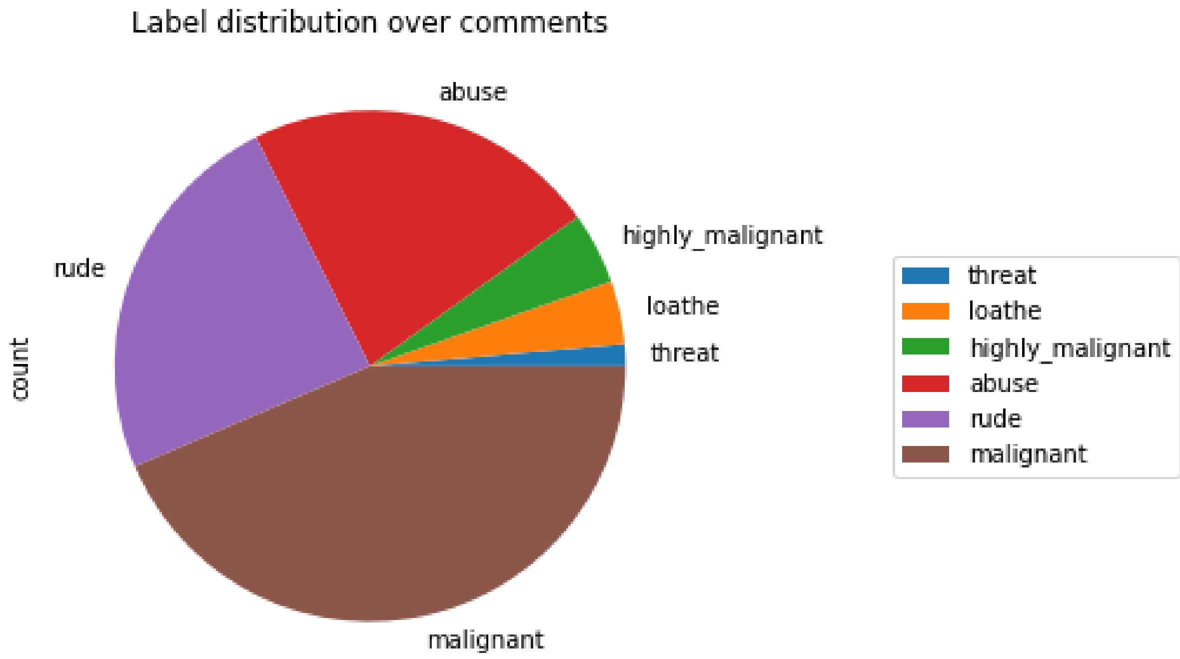
Out of 159621, 478 comments are threat.



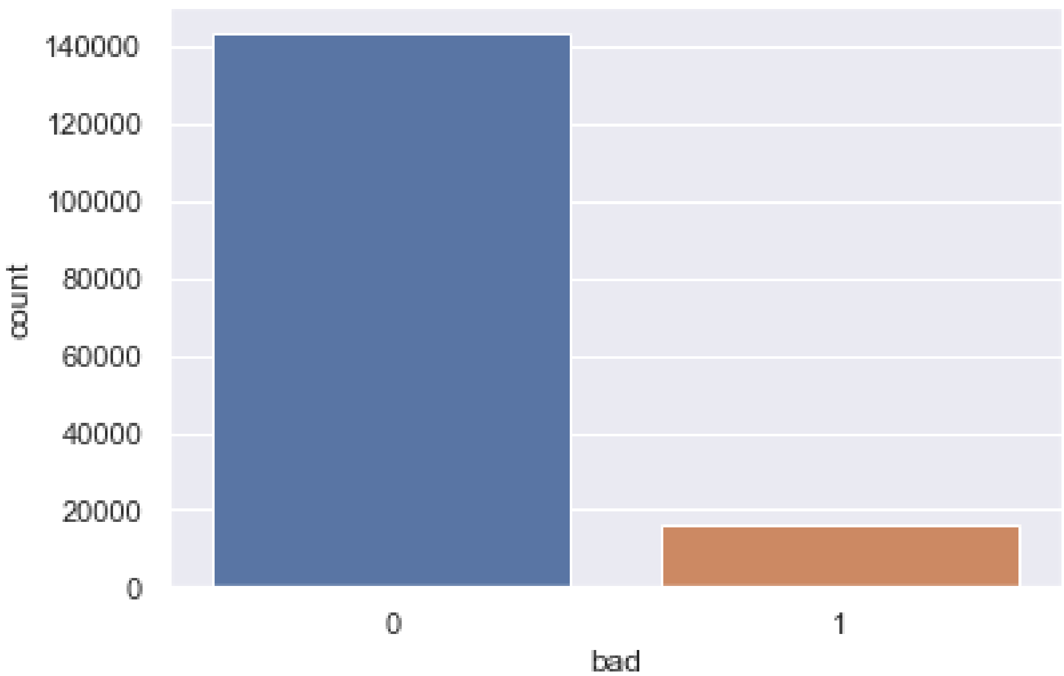
7875 comments are abuse.



1405 comments are loathe.



We can see rude and abuse comments are higher than others which comments are malignant.



143400 comments are not malignant and 16221 comments are malignant.

CONCLUSION

Walk down any busy street and you'll soon see someone head-down, absorbed by their phone, tapping away to someone unseen. There's a good chance they're on Twitter or Facebook, or chatting via WhatsApp or SMS (or one of many other messaging apps). Using these services now accounts for much of the writing we do each day.

In the eyes of some, this is a kind of malignant disease that's eroding the quality of writing everywhere. They may point to a perceived decline in standards of spelling, grammar and punctuation – threats to even the most fundamental features of writing, like the full stop. Or they'll complain bitterly about emoticons and emojis slipping into business emails. Other people take a different view: we're spending more time writing than ever, and that's good, not bad. After all, you'd expect a population where everyone was constantly throwing and catching balls to be good at cricket, even if some bits of their technique would make cricket coaches wince. So maybe our daily writing practice – even in the form of writing text messages and on social media – is similarly positive.

Whatever side you take, this is much more than just an academic argument – it matters for everyone, whether you've just joined your first company or you run one. If social media and SMS are making us incapable of stringing together persuasive arguments, producing coherent reports or writing effective emails, then we need to do something to resolve this.