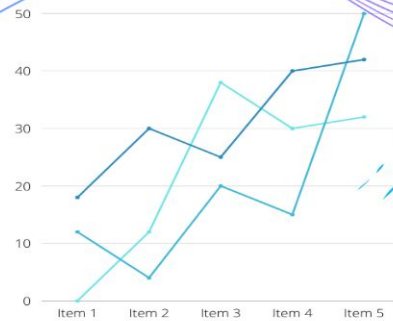


Malignant Commentes Classifier - Multi Label Classification Project using NLP

Author :
Wesley Sirra

Malignant Comment Classification



Malignant Commentes Classifier - Multi Label Classification Project using NLP

- ◆ 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 behavior.

Malignant Commentes Classifier - Multi Label Classification Project using NLP

- ◆ 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.

Problem Statement

Our goal is to build a prototype of online hate and abuse comment classifier which can be used to classify hate and offensive comments so that it can be controlled and restricted from spreading hatred and cyberbullying.

Multi –Label Classification Problem

- ◆ **Difference between multi-class classification & multi-label classification** is that in multi-class problems the classes are mutually exclusive, whereas for multi-label problems each label represents a different classification task, but the tasks are somehow related.
- ◆ For example, ***multi-class classification*** makes the assumption that each sample is assigned to one and only one label: a fruit can be either an apple or a pear but not both at the same time. Whereas, an instance of ***multi-label classification*** can be that a text might be about any of religion, politics, finance or education at the same time or none of these.



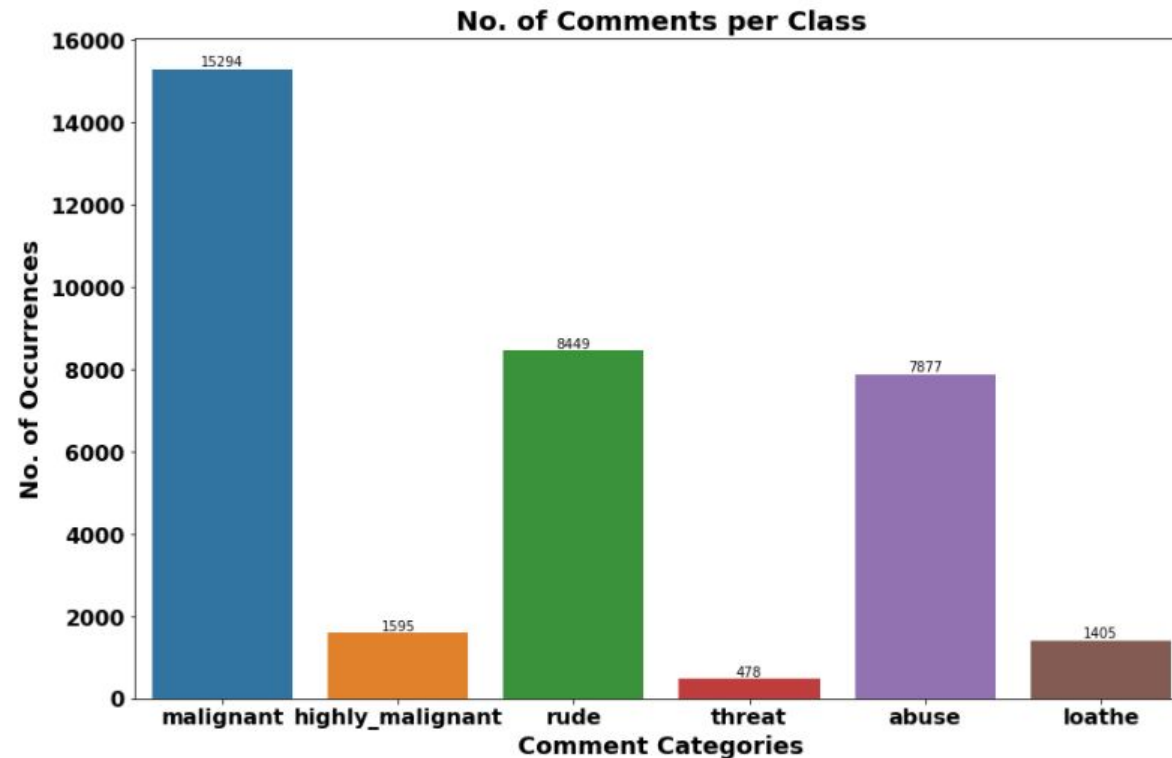
Horror: 0.02%
Romance: 0.02%
Adventure: 99.96%
Documentary: 0.0%

Multi-label classification of textual data is an important problem. Examples range from news articles to emails.

For instance, this can be employed to find the genres that a movie belongs to, based on the summary of its plot.

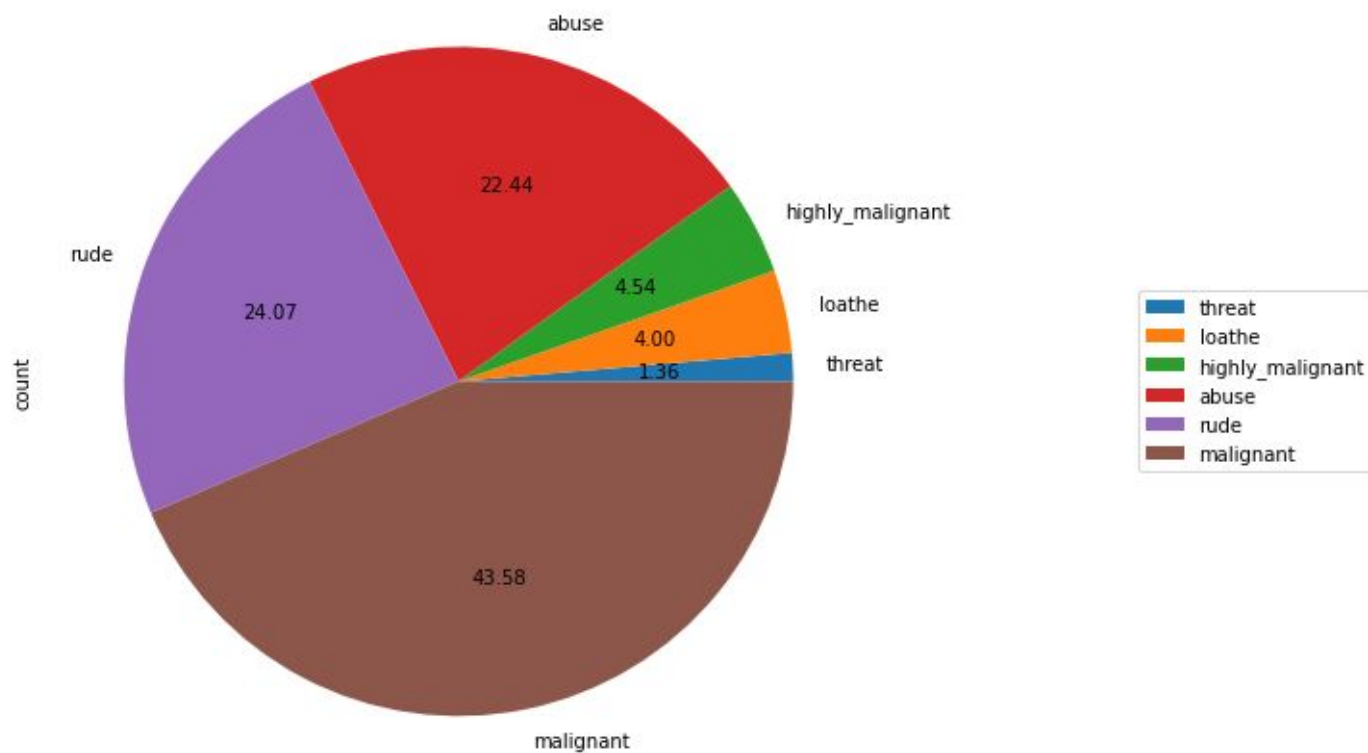
Fig-2: Multi-label classification to find genres based on movie posters.

Exploration of Target Variable Ratings

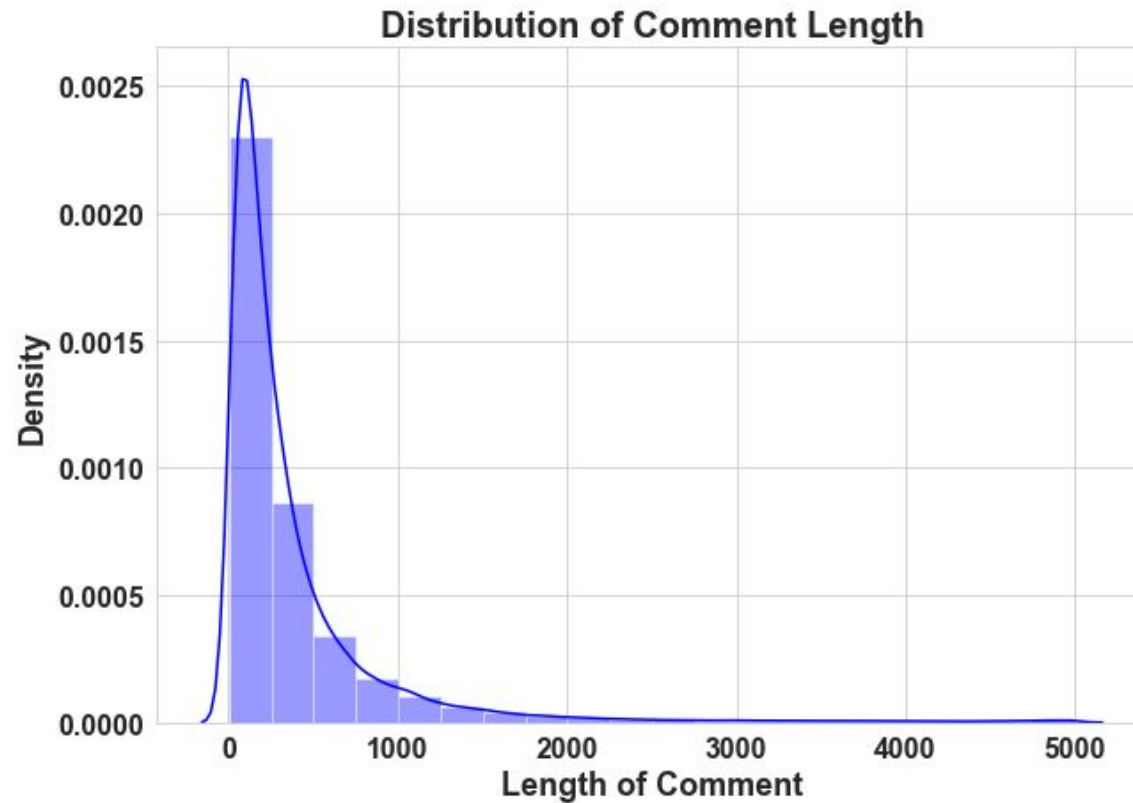


- Out of total Negative comments the maximum negative comments come with Malignant in nature followed by rude categories.
- Around 90% comments are Good/Neutral in nature while rest 10% comments are Negative in nature.
- Very few comments come with threatening nature.

Label distribution over comments



Out of total negative comments around 43.58% are malignant in nature followed by 24.07% are rude comments



- Above is a plot showing the comment length frequency. As noticed, most of the comments are short with only a few comments longer than 1000 words.
- Majority of the comments are of length 500, where maximum length is 5000 and minimum length is 5. Median length being 250.

Data Pre Processing

- Convert the text to lowercase
- Remove the punctuations, digits and special characters
- Tokenize the text, filter out the adjectives used in the review and create a new column in data frame
- Remove the stop words
- Stemming and Lemmatising
- Applying Text Vectorization to convert text into numeric

Multi-Label Classification Techniques

- ◆ One Vs Rest
- ◆ Binary Relevance
- ◆ Classifier Chains
- ◆ Label Powerset
- ◆ Adapted Algorithm

The background features decorative blue geometric shapes in the corners. The top-left corner has a dark blue shape with a white border. The top-right corner has a light blue shape. The bottom-left corner has two light blue shapes. The bottom-right corner is empty white space.

Word Cloud

Word Cloud for getting word sense

- . Word Cloud is a visualization technique for text data wherein each word is picturized with its importance in the context or its frequency.
- . The more commonly the term appears within the text being analysed, the larger the word appears in the image generated.
- . The enlarged texts are the greatest number of words used there and small texts are the smaller number of words used.



From word cloud of Highly malignant comments, it is clear that it mostly consists of words like fuck, stupid, fucking, bitch, crow, shit, cocksucker etc



From word cloud of Rude comments, it is clear that it mostly consists of words like fucking, shit, white, piece, edits, stuff, absurd etc.

[illegible]

From word cloud of Loathe comments, it is clear that it mostly consists of words like fuck, gay, kill, think, jew, u etc.

```
#Importing warning library to avoid any warnings
import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # plotting package
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')
```

```
#Importing required libraries
import re
import string
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from wordcloud import WordCloud
```

```
#Importing Machine Learning Model Library
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.svm import SVC, LinearSVC
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.metrics import roc_auc_score, roc_curve, auc
from sklearn.metrics import hamming_loss, log_loss
```




Machine Learning Model Building

Machine Learning Model Building

The different classification algorithm used in this project to build ML model are as below:

- ❖ Random Forest classifier
- ❖ Support Vector Classifier
- ❖ Logistics Regression
- ❖ AdaBoost Classifier

Machine Learning Evaluation Matrix

- ◆ Support Vector Classifier gives maximum Accuracy Score: 91.1508 % and Hamming Loss: 2.0953% than the other classification models.
- ◆ Hyper parameter Tuning is perform over this best model using best param shown below :

```
Out[69]: {'estimator__loss': 'hinge',  
          'estimator__multi_class': 'ovr',  
          'estimator__penalty': 'l2',  
          'estimator__random_state': 42}
```

Final ML Model

Final Model

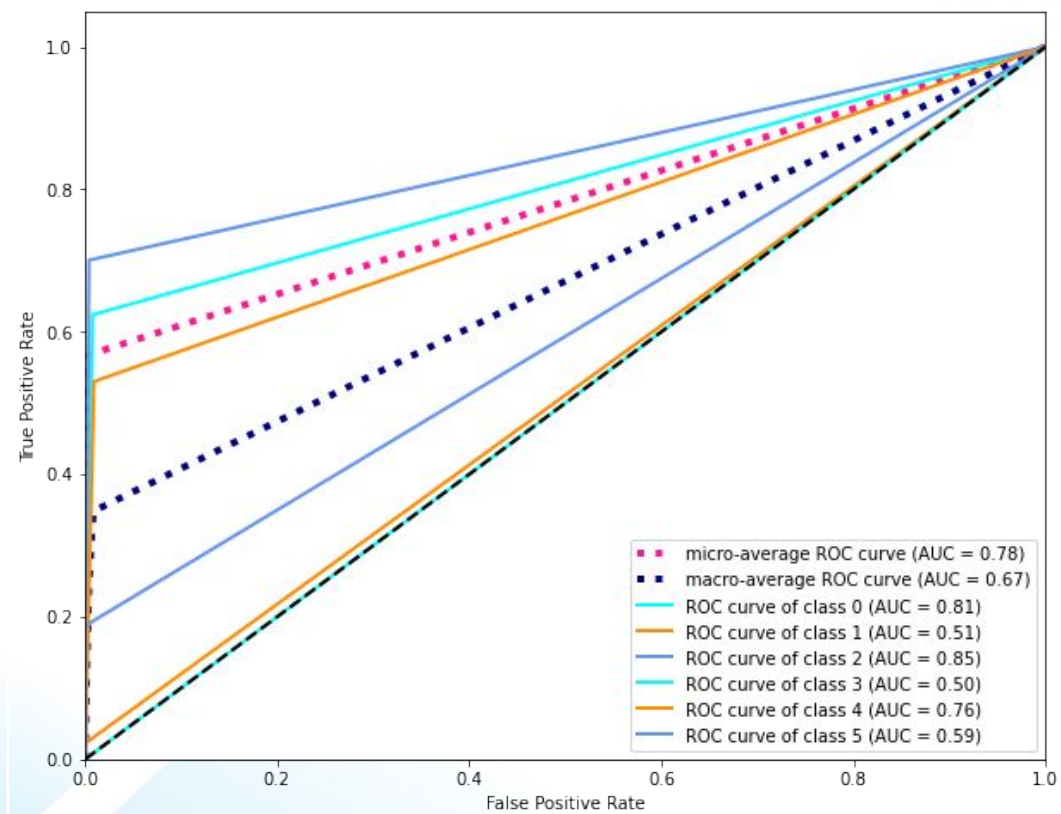
```
Final_Model = OneVsRestClassifier(LinearSVC(loss='hinge',  
                                           multi_class='ovr', penalty='l2', random_state=42))
```

```
Classifier = Final_Model.fit(x_train, y_train)  
fmod_pred = Final_Model.predict(x_test)  
fmod_acc = (accuracy_score(y_test, fmod_pred))*100  
print("Accuracy score for the Best Model is:", fmod_acc)  
h_loss = hamming_loss(y_test, fmod_pred)*100  
print("Hamming loss for the Best Model is:", h_loss)
```

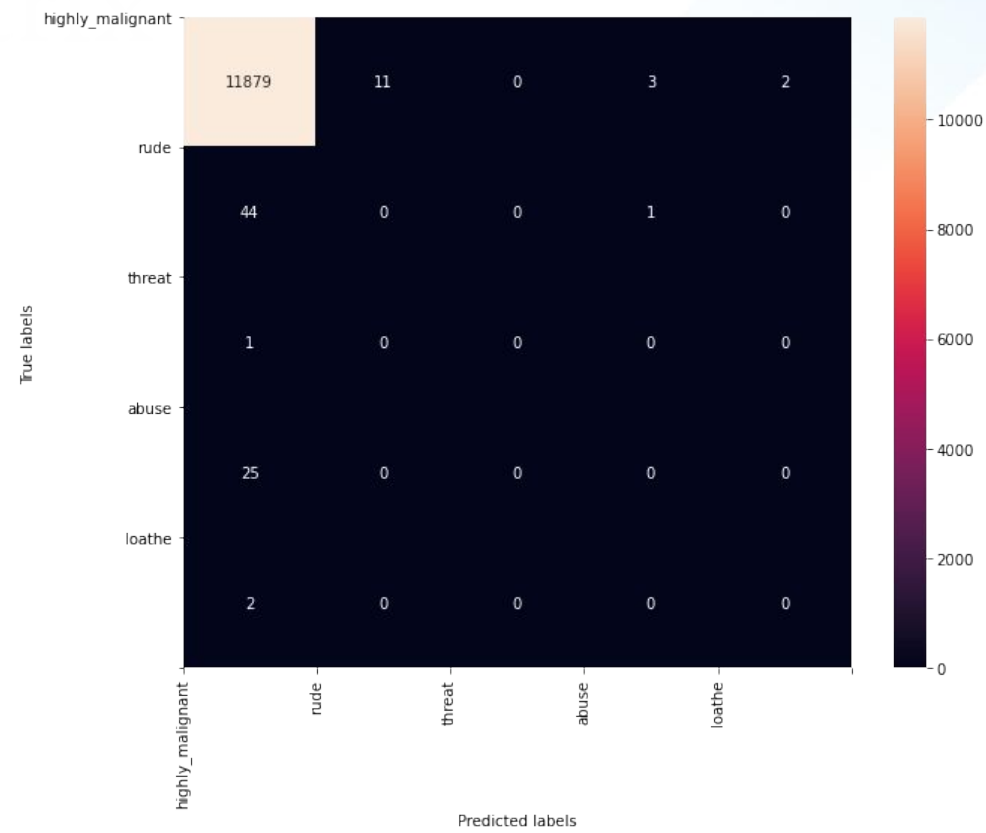
```
Accuracy score for the Best Model is: 91.26002673796792  
Hamming loss for the Best Model is: 2.0819407308377897
```

Final Model is giving us Accuracy score of 91.26% which is slightly improved compare to earlier Accuracy score of 91.15%.

Receiver operating characteristic (ROC) and Area under curve (AUC) for multiclass labels



Confusion Matrix for the Final Classification Model



Machine Learning Evaluation Matrix

Algorithm	Accuracy Score	Recall (Micro)	Precision (Micro)	F1 Score (Micro)	Hamming Loss
Logistics Regression	0.9123	0.89	0.94	0.61	0.02206
Random Forest Classifier (RFC)	0.9074	0.56	0.79	0.66	0.02191
Support Vector Classifier	0.9115	0.56	0.82	0.67	0.02952
Ada Boost Classifier	0.9057	0.50	0.80	0.61	0.9057

CONCLUSION

- Linear Support Vector Classifier performs better with Accuracy Score: 91.15077857956704 % and Hamming Loss: 2.0952019242942144 % than the other classification models.
- Final Model (Hyperparameter Tuning) is giving us Accuracy score of 91.26% which is slightly improved compare to earlier Accuracy score of 91.15%.
- SVM classifier is fastest algorithm compare to others.

Limitations of this work and Scope for Future Work

- The Maximum feature used while vectorization is 2000. Employing more feature in vectorization lead to more accurate model which I not able to employed due computational resources.
- Data is imbalanced in nature but due to computational limitation we have not employed balancing techniques here.
- Deep learning CNN, ANN can be employed to create more accurate model.