MALIGNANT COMMENTS CLASSIFICATION

Submitted by:

Utsav Rastogi

ACKNOWLEDGMENT

This project was made possible with the help of my supervisor Swati Mahaseth. Under her guidance, I was able to complete the project on time and ample learning possibilities this project provided me with.

Dataset was provided by Flip robo Technologies, and was scrapped from various social media platforms.

Data analysis and preprocessing was done keeping in mind how to make the code run efficiently and quickly with utilizing minimal memory. I had to refer to various articles on Medium, Analytics Vidhya, Kaggle and stack exchange to look for potential debugging solutions and visualization plots.

INTRODUCTION

- ## Business Problem Framing

  Our goal of the business to understand a comment posted in socialmedia can be harassing.

  The objective of the project is to create a prototype of online malignant comment classifier which can used to classify hate and offensive comments to the number of categories so that it can be restricted from expanding hatred and cyberbullying on social media.

- ## Conceptual Background of the Domain Problem

  The spread of hate and angst amount the youth have been especially fuelled by the aggressive tweets, comments and videos. Behind the façade of anonymity, anti-social elements of the society have received an enormous ability to hurt anyone without caring for any consequences. Increase in Internet penetration have drastically increased such instances. This project is an attempt to classify online comments into different categories and make it easier for the algorithms which type of comments to tag as hateful and it belongs to how many categories (0,1,2,3,4,5,6) as there are 6 categories provided. And 0 indicating a clean comment.

- ## Motivation for the Problem Undertaken

  Hate comments have made several communities target of attacks worldwide. It's the easiest method to demotivate someone, Incite people to commit crime and inspire a great deal of violence. Today's youth is heavily influenced by the social media platforms and this interconnectedness is an advantage but also can prove to be breeding ground for various nefarious

activities. ISIS and Taliban, the terrorist organizations have been using social media to recruit young people who can simply be brainwashed into committing atrocious crimes against humanity. Hateful comments and cyberbullying can result in severe depression or in some cases death by suicide which has been increasing in the last few years.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

Visual analysis	Libraries used: Matplotlib and Seaborn
Algorithms	Different Regression algorithms such as AdaBoost Classifier& Logistic Regression were used which are all provided by sklearn library.

- Data Sources and their formats
  - ❖ There is are 2 features id and comment_text .
  - ❖ Comment_text is an uncleaned data column which contains data that has been scraped from different social media websites.
  - ❖ There are 6 different labels under which comments can be tagged.

malignant	Binary column with comment labeled as malignant
highly_malignant	Binary column with labels for highly malignant text.
rude	Binary column with comment labeled as rude
threat	Binary column with comment labeled as threat
abuse	Binary column with comment labeled as abusive
loathe	Binary column with comment labeled as loathe and hateful

- Data Preprocessing Done
  1. Checking null values
  2. Checking the number of comments in under each label
  3. Count the number of comments having multiple labels
  4. Word Cloud representation
  5. Removing stop words, Lemmatization, lower casing using NLTK

- State the set of assumptions (if any) related to the problem under consideration

  “Stopwords” corpus is inclusive of every word used today. Sarcasm and hidden meaning comments were not considered

- Hardware and Software Requirements and Tools Used

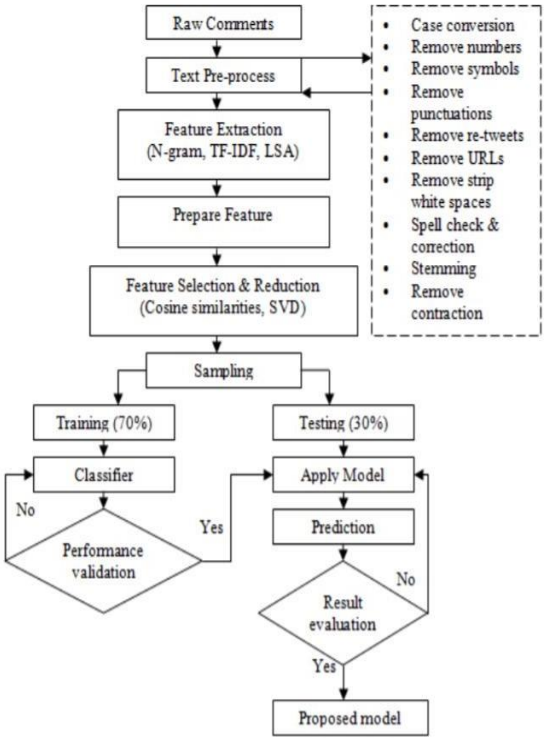
  Hardware: Windows 11

  Software: Jupyter notebook

  Libraries: seaborn, matplotlib, nltk, numpy, pandas, sklearn, joblib

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)



We merge all the 6 categories into a single target column to transform the problem into a simple classification problem to save time and computation involved in a muti label classification problem.

- Testing of Identified Approaches (Algorithms)

1. Logistic Regression
2. Linear SVC
3. Multinomial Naïve Bayes
4. AdaBoost Classifier

- Run and Evaluate selected models

Logistic Regression

```
Logr = LogisticRegression(C=1, max_iter = 3000)

Logr.fit(x_train, y_train)

y_pred_train = Logr.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
y_pred_test = Logr.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
cvscore_LR = cross_val_score(Logr,X,y,cv = 10 )
print("Cross val score of Logistic Regression is :", round(cvscore_LR.mean(),4)*100,'%')
print(classification_report(y_test,y_pred_test))
```

Linear SVC

```
svc = LinearSVC()
svc.fit(x_train,y_train)
# Performing Evaluation metrics for our model
predsvc= svc.predict(x_test)
print('Accuracy Score :',round(accuracy_score(y_test,predsvc),4)*100,'% \n')
CVscore_svc = cross_val_score(svc,X,y,cv = 10 )
print("Cross validation score :", round(CVscore_svc.mean(),4)*100,'%')
print('Classification : \n',classification_report(y_test,predsvc))
print("Hamming Loss : ",hamming_loss(y_test,predsvc))
```

AdaBoost Classifier

```
ada = AdaBoostClassifier()
ada.fit(x_train,y_train)
# Performing Evaluation metrics for our model
predada= ada.predict(x_test)
print('Accuracy Score  :',round(accuracy_score(y_test,predada),4)*100,'% \n')
CVscore_adaboost = cross_val_score(ada,X,y,cv = 10 )
print("Cross validation score of AdaBoost Classifier is :",round(CVscore_adaboost.mean(),4)*100,'%')
print('Classification  : \n',classification_report(y_test,predada))
print("Hamming Loss  : ",hamming_loss(y_test,predada))
```

Multinomial Naïve Bayes

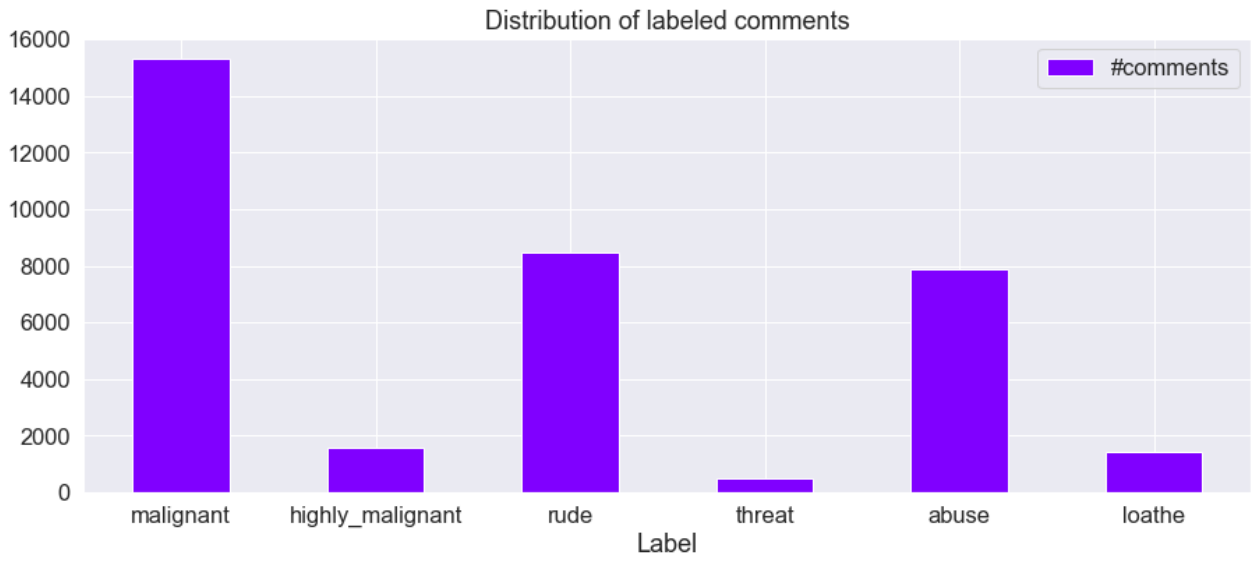
```
MNB = MultinomialNB()
MNB.fit(x_train,y_train)

predmnb= MNB.predict(x_test)
print('Accuracy Score for Multinomial Naive Bayes Classifier is :',round(accuracy_score(y_test,predmnb),4)*100,'% \n')
CVscore_mnb = cross_val_score(MNB,X,y,cv = 10 )
print("Cross validation score :",round(CVscore_mnb.mean(),4)*100,'%')
print('Classification Multinomial Naive Bayes Classifier : \n',classification_report(y_test,predmnb))
print("Hamming Loss for our Multinomial Naive Bayes Classifier model is : ",hamming_loss(y_test,predrf))
```

- Key Metrics for success in solving problem under consideration

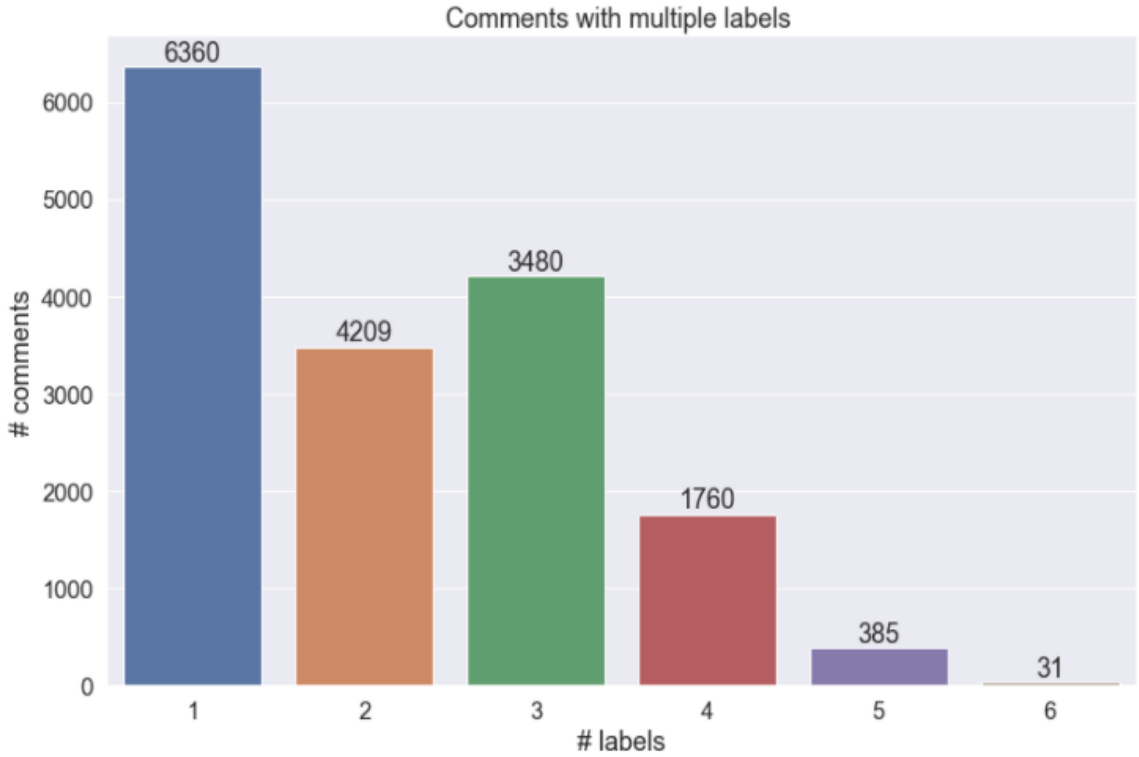
Model	Accuracy score	Cross Val Score	Hamming Loss
Logistic Regression	91.87	91.98	0.0812
Multinomial Naïve Bayes	90.49	90.74	0.0838
Linear SVC	91.69	91.82	0.0831
AdaBoost Classifier	90.7	90.94	0.0929

- Visualizations

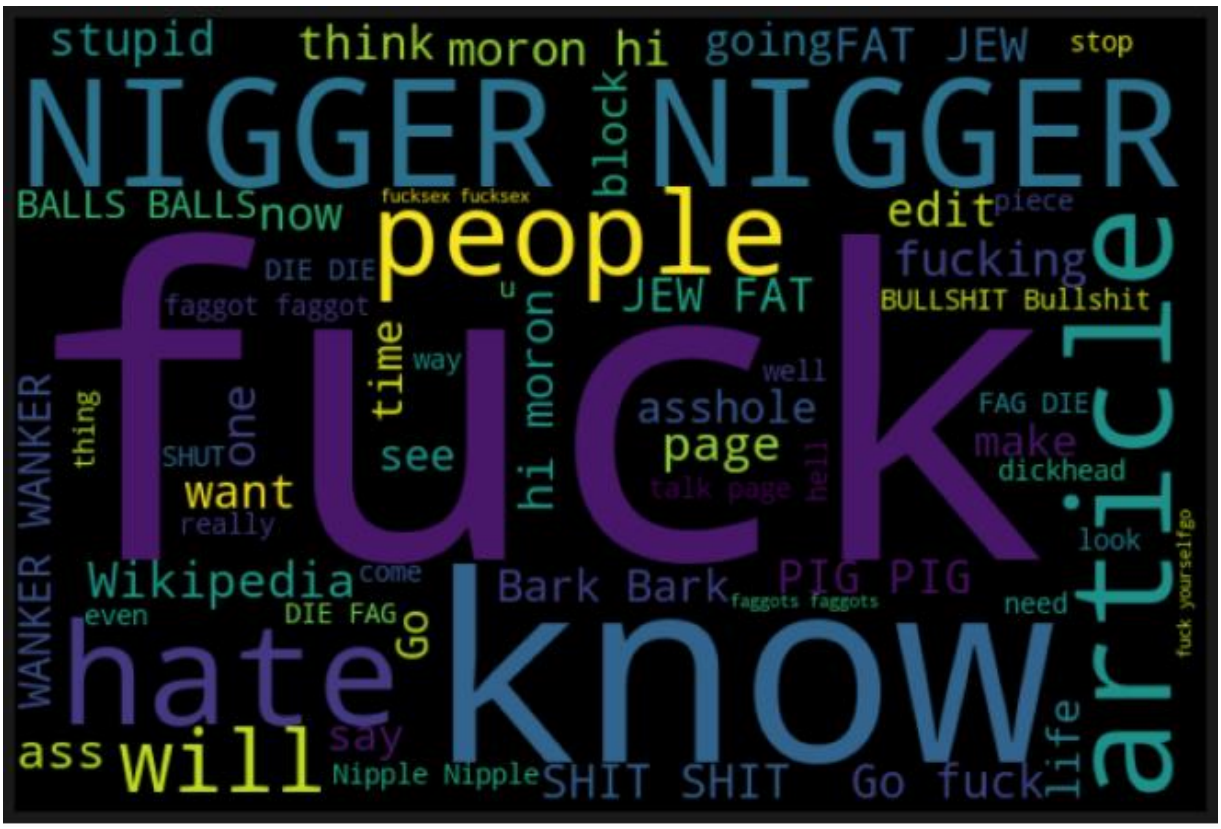


- Total number of comments under each label

	Label	#comments
0	malignant	15294
1	highly_malignant	1595
2	rude	8449
3	threat	478
4	abuse	7877
5	loathe	1405

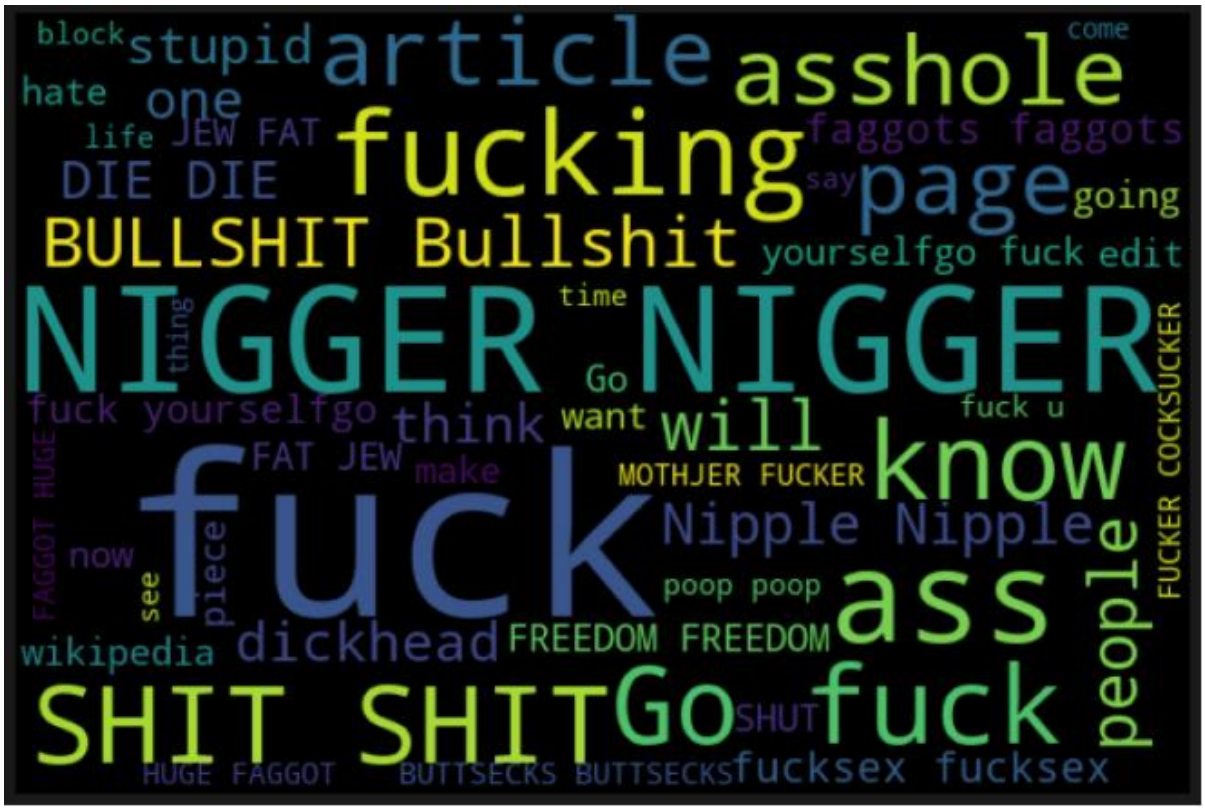


- Frequent words used in comments labelled “malignant”

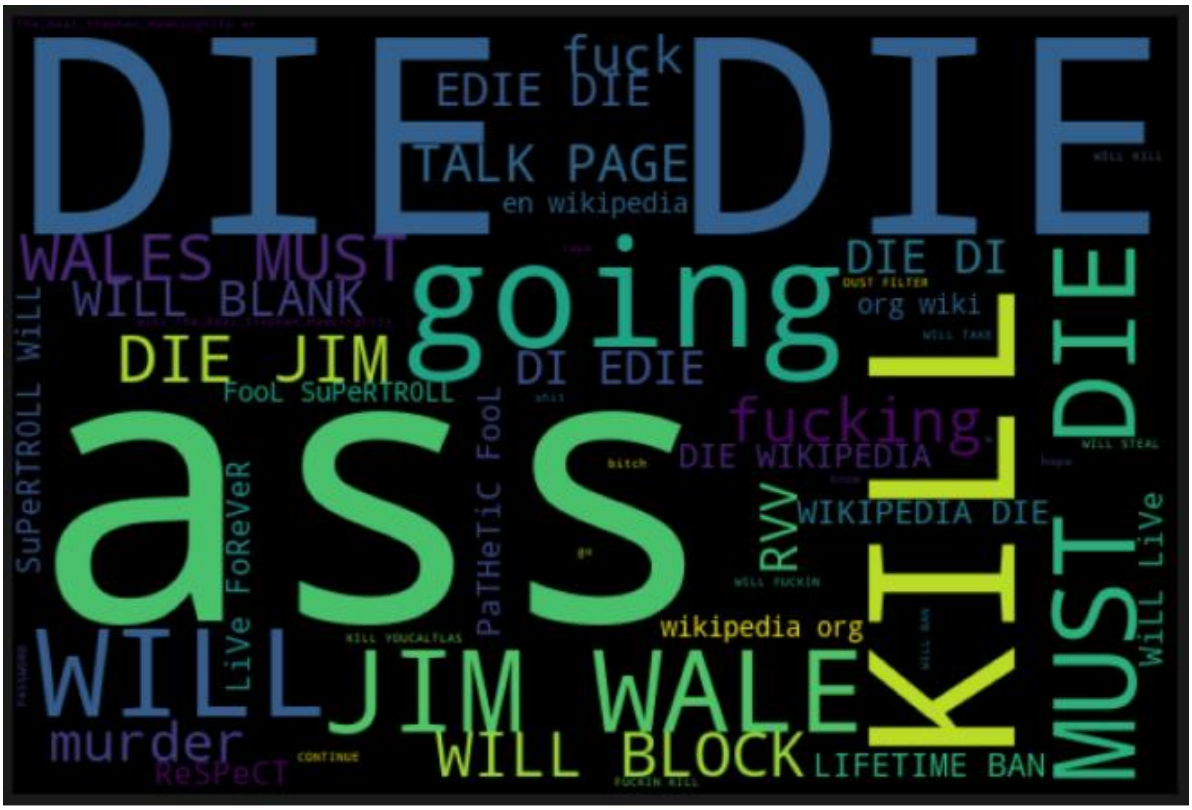


- Frequent words used in comments labelled “highly_malignant”

- Frequent words used in comments labelled “rude”

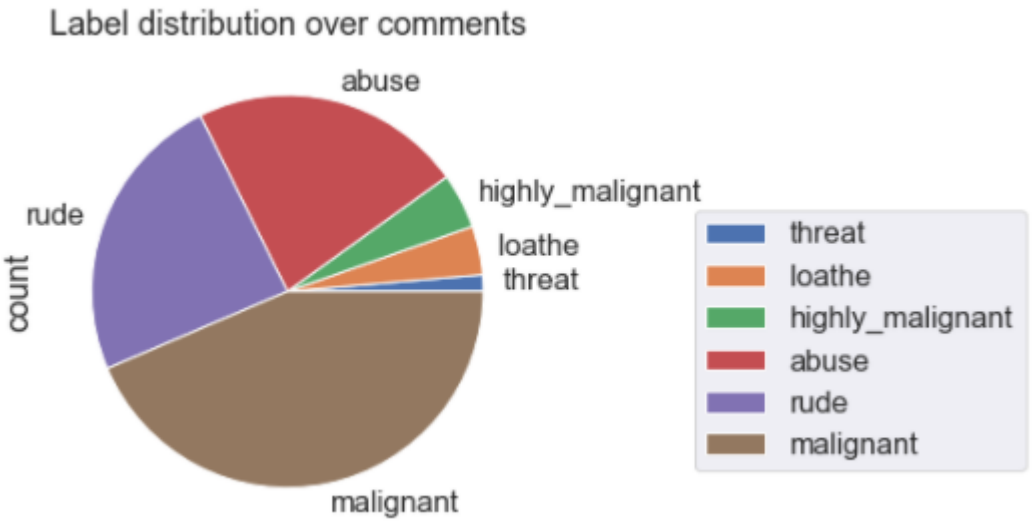


- Frequent words used in comments labelled “threat”



- Frequent words used in comments labelled “loathe”

- Frequent words used in comments labelled “abuse”



CONCLUSION

- Key Findings and Conclusions of the Study
  - The plots above are self-explanatory.
  - The number of malignant comments is the highest.
  - Majority of the comments have only one label associated with them.
  - The problem is multi label classification problem which I converted to single target column classification due to the laptop computing constraint.
  - The model interprets a comment and decides its range of malign (whether it belongs to only one label or have multiple labels associated with it).
- Limitations of this work and Scope for Future Work

  Better computing power could have yielded a much better output as it could have been possible to check the accuracy for each comment for every label. We had to group all the labels together for each comment and then use classification algorithms.