Multi Label Text Classification

DS504 | NLP | Group 29

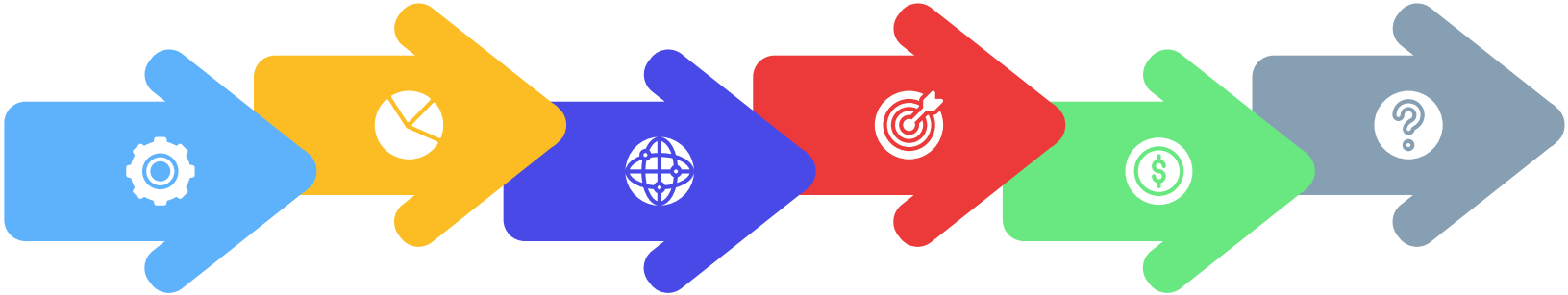
Shweta Chauhan -12041440
Siddharth Gupta -12041450
Piyush Pancholi -12041050
Akash Deep -12040060

Introduction

Inspect Data

Prepare Data

Building the model



Load Data

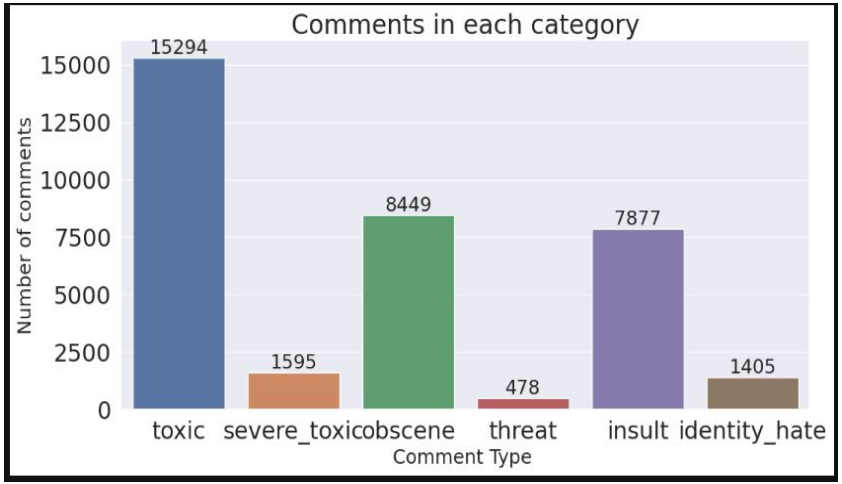
Loading data in test and training set>

Review Training Sample

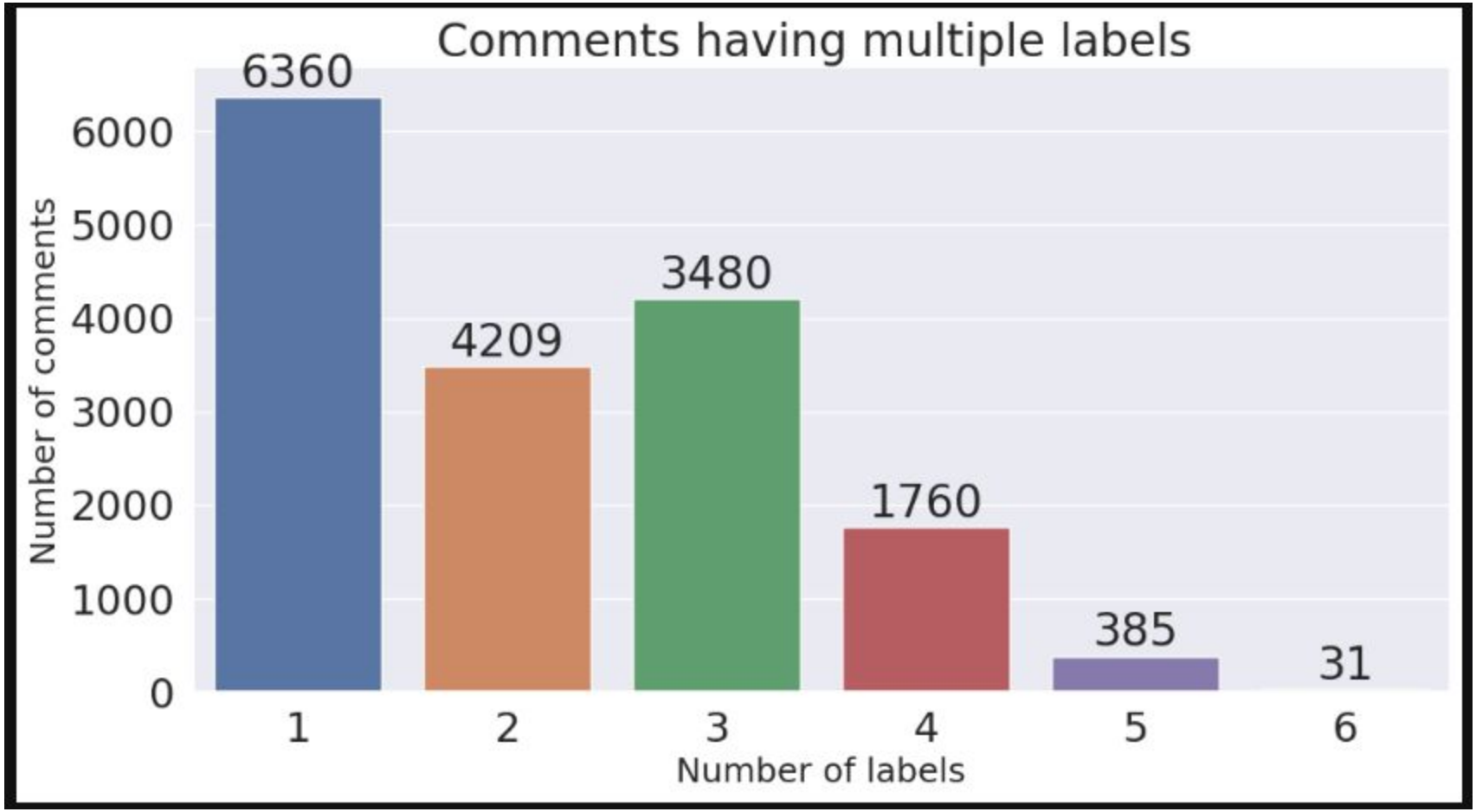
Label data

01 Load and inspect data

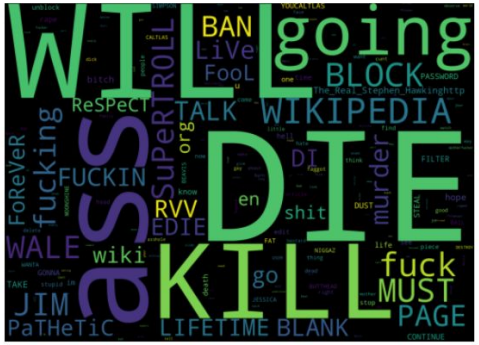
	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0



	category	number of comments
0	toxic	15294
1	severe_toxic	1595
2	obscene	8449
3	threat	478
4	insult	7877
5	identity_hate	1405


Comments having multiple labels
Number of comments
0
1000
2000
3000
4000
5000
6000
6360
4209
3480
1760
385
31
1
2
3
4
5
6
Number of labels

Threat



Identity Hate

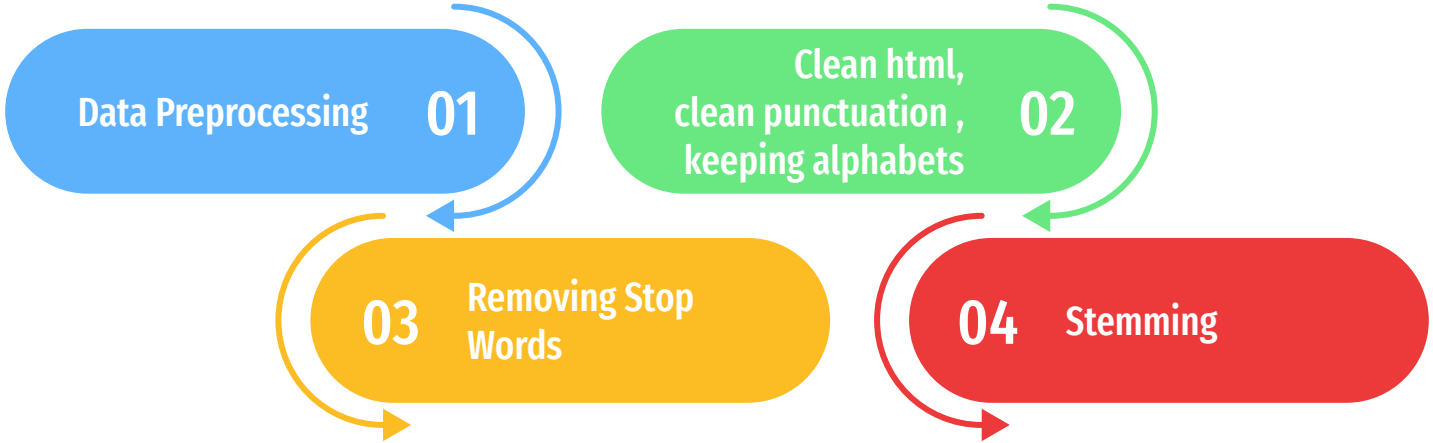
Analysis of Data 02

Severe Toxic

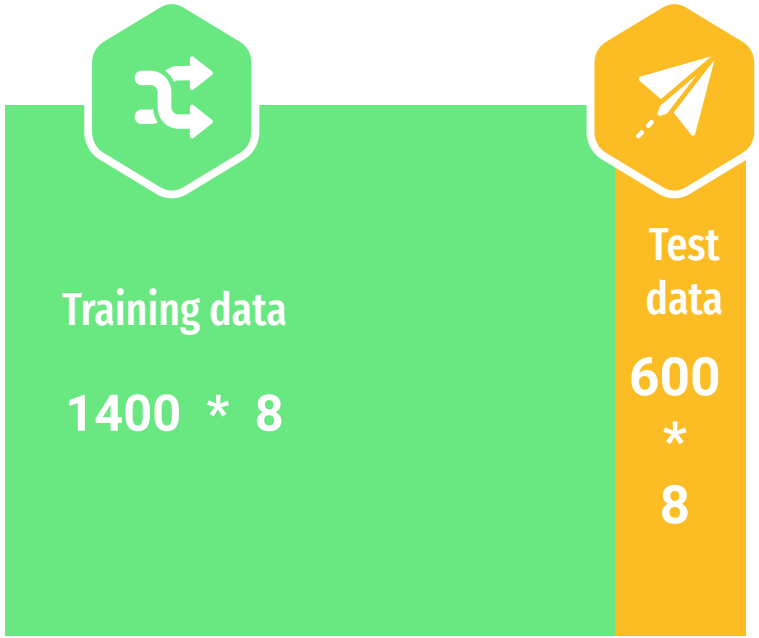
Obscene

Toxic

02 Data Cleaning



03 Train-Test Split



04 Multi - Label Classification

Processing toxic comments...

Test accuracy is 0.9133333333333333

Processing severe_toxic comments...

Test accuracy is 0.9916666666666667

Processing obscene comments...

Test accuracy is 0.9433333333333334

Processing threat comments...

Test accuracy is 0.995

Processing insult comments...

Test accuracy is 0.9516666666666667

Processing identity_hate comments...

Test accuracy is 0.9866666666666667

Multiple Binary Classifications and Classifier chains

05

```
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.naive_bayes import GaussianNB

classifier = BinaryRelevance(GaussianNB())

classifier.fit(x_train, y_train)

predictions = classifier.predict(x_test)

print("Accuracy = ",accuracy_score(y_test,predictions))
print("\n")
```

```
Accuracy =  0.8883333333333333


CPU times: user 4.26 s, sys: 3.83 s, total: 8.09 s
Wall time: 8.16 s
```

```
classifier = ClassifierChain(LogisticRegression())

classifier.fit(x_train, y_train)

predictions = classifier.predict(x_test)

print("Accuracy = ",accuracy_score(y_test,predictions))
print("\n")
```

```
Accuracy =  0.9066666666666666


CPU times: user 20 s, sys: 1.18 s, total: 21.2 s
Wall time: 21.3 s
```

```
classifier = LabelPowerset(LogisticRegression())

classifier.fit(x_train, y_train)

predictions = classifier.predict(x_test)

print("Accuracy = ",accuracy_score(y_test,predictions))
print("\n")
```

```
Accuracy =  0.9066666666666666
```

Applications of project - Novelty

--> In-game chat toxicity

--> Keeping e-market places like amazon reviews non offensive (amazon has its own toxicity detector which works for hindi words too).

--> Detecting bullying on online learning platforms like zoom.

--> Detecting insults hidden in between day to day texts for broadcast.

Thank you !