

# Toxic Comment Classification

**KATTAK Abdullah, DELAMARE Irina, SAVELYEVA Natasha**



**Bullying, including cyberbullying, is a serious impediment to achieving the Sustainable Development Goals (SDGs).**

Book: Developmental Science and Sustainable Development Goals for Children and Youth - Tracing the Connections Between Sustainable Development, Bullying, and Cyberbullying

R. Sittichai, T. Ojanen, J. Burford - 2018

# Bullying has documented impacts on:

Educational access

Mental health

Depression and  
suicidal rate



Multiple SDGs and  
their targets

# Wikipedia Toxic Comment Classification

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	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
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...
159566	ffe987279560d7ff	"::::And for the second time of asking, when ...	0	0	0	0	0	0
159567	ffea4adeee384e90	You should be ashamed of yourself \n\nThat is ...	0	0	0	0	0	0
159568	ffee36eab5c267c9	Spitzer \n\nUmm, theres no actual article for ...	0	0	0	0	0	0
159569	fff125370e4aaaf3	And it looks like it was actually you who put ...	0	0	0	0	0	0
159570	fff46fc426af1f9a	"\nAnd ... I really don't think you understand...	0	0	0	0	0	0
159571 rows × 8 columns								

**15971 rows, 8 columns**

Source: <https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview>

# Dataset

## Comments from Wikipedia's discussions page

### 6 classes of comment:

- Toxic
- Severe toxic
- Obscene
- Threat
- Insult
- Identity hate

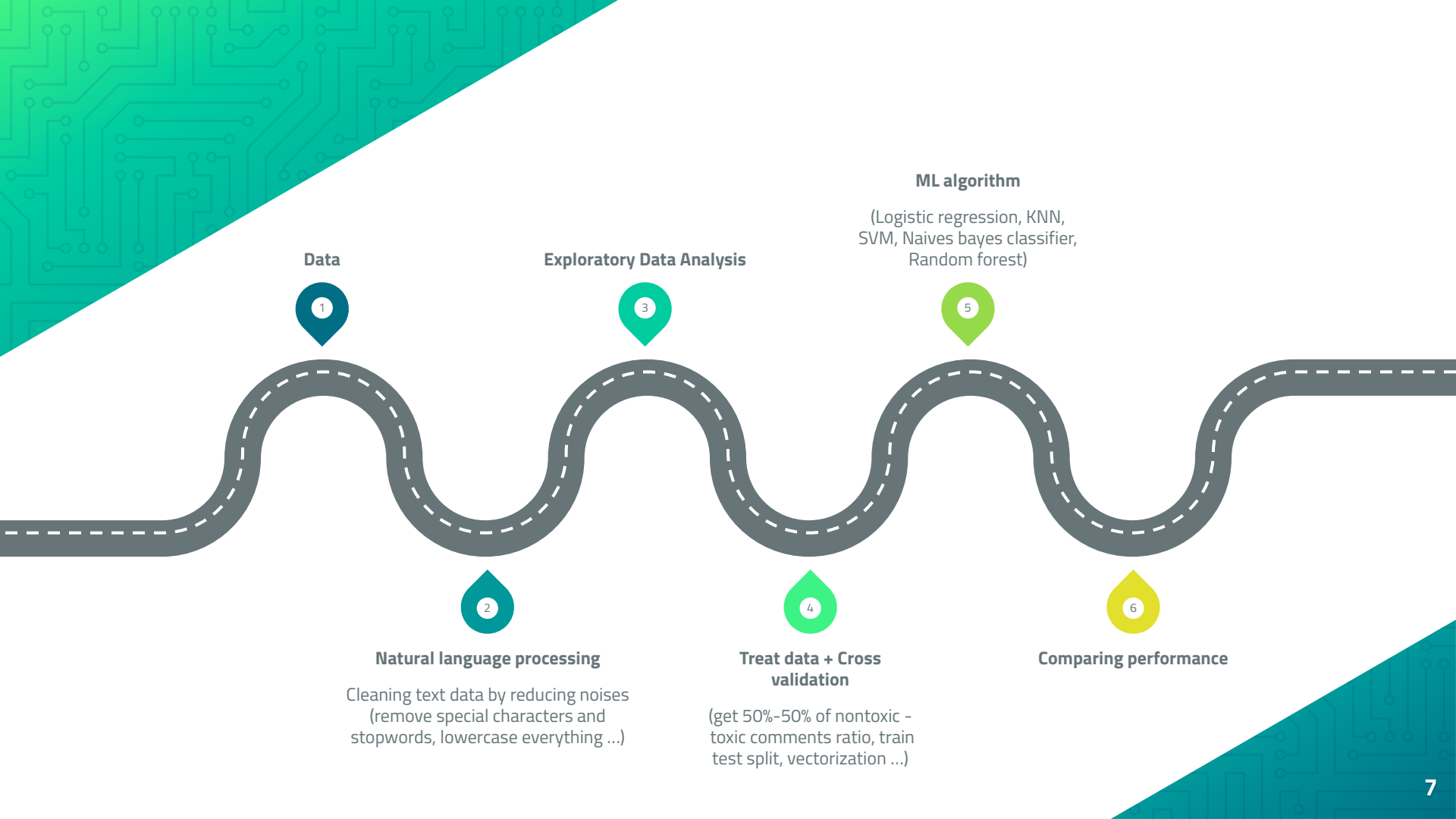
### 2 labels:

- **0 : False**
- **1 : True**



## Objective

- **Build a classification model that is able to predict the toxicity of comment**
- **Evaluate and compare the performance between ML models**



# Data analysis

- No null values to deal with
- Read comments properly using **neattext** library



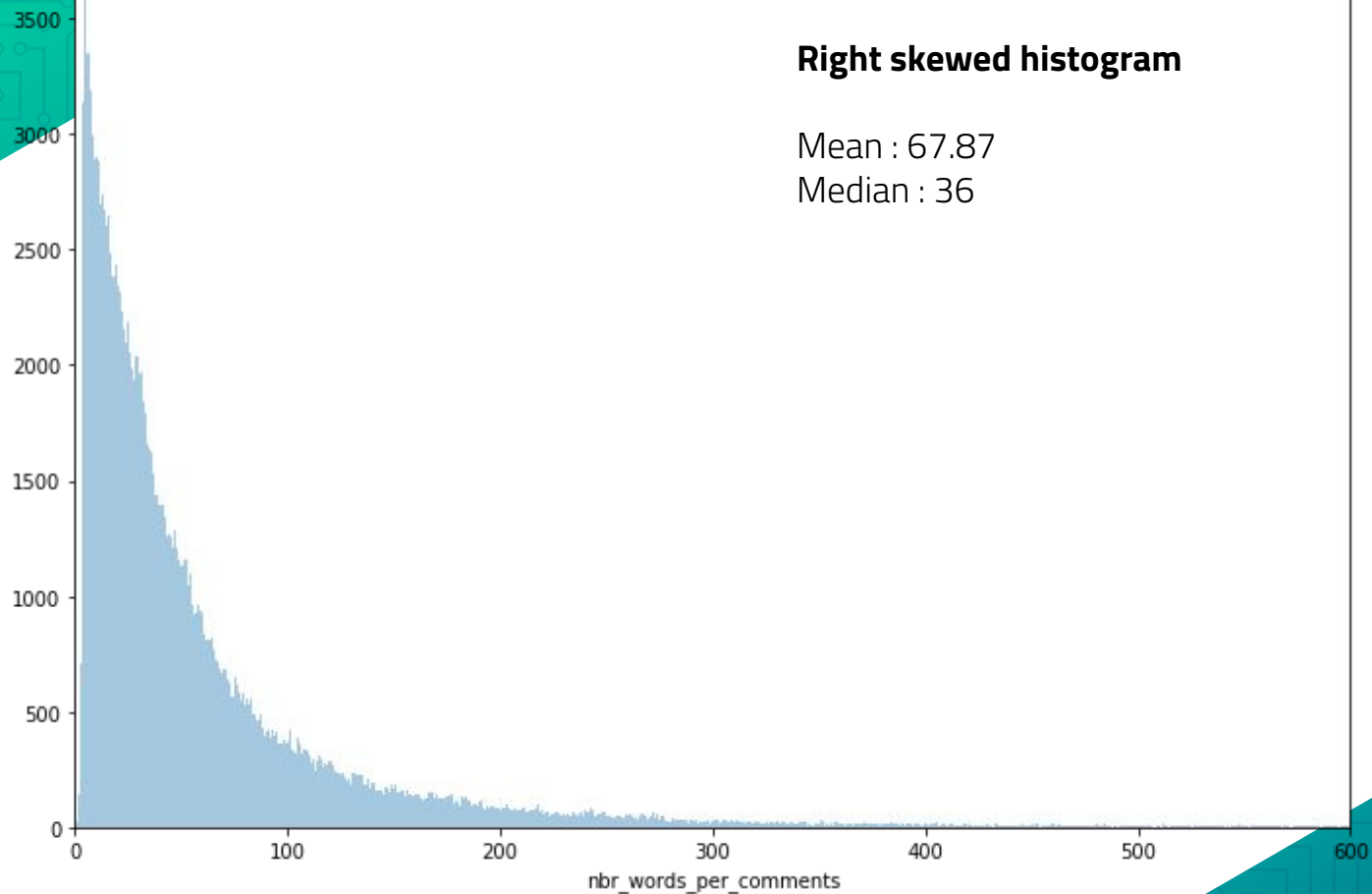
# Comment

```
'"\nMore\nI can\'t make any real suggestions on improvement - I wondered if the section statistics should be later on, or a subsection of "types of accidents" -I think the references may need tidying so that they are all in the exact same format ie date format etc. I can do that later on, if no-one else does first - if you have any preferences for formatting style on references or want to do it yourself please let me know.\n\nThere appears to be a backlog on articles for review so I guess there may be a delay until a reviewer turns up. It\'s listed in the relevant form eg Wikipedia:Good_article_nominations#Transport  "'
```

## Description:

Key	Value
Length	: 622
vowels	: 196
consonants	: 290
stopwords	: 62
punctuations	: 20
special_char	: 21
tokens(whitespace)	: 113
tokens(words)	: 116

Number of Words frequency Wikipedia comments



[illegible]

## Most frequent words in all Wikipedia comments



# Data analysis

- No null values to deal with
- Read comments properly using **neattext** library
- NLP and pre-processing: Clean data by removing punctuation, stopwords, lower casing ...

# Natural language processing (NLP)

*linguistics, computer science, artificial intelligence*

**Interactions between computers and human language. Allow computer to process and analyze large amounts of natural language data (human language).**

NLP tools and process :

- Normalization: take into account all the form of the same word (*have, having*)
- Stemming: finding a root of the words (*have, having*)  $\Rightarrow$  *hav*
- Tokenization: splitting a sentence into tokens (words, punctuation ... )
- Vectorization: process of converting text into numerical representation

# Data analysis

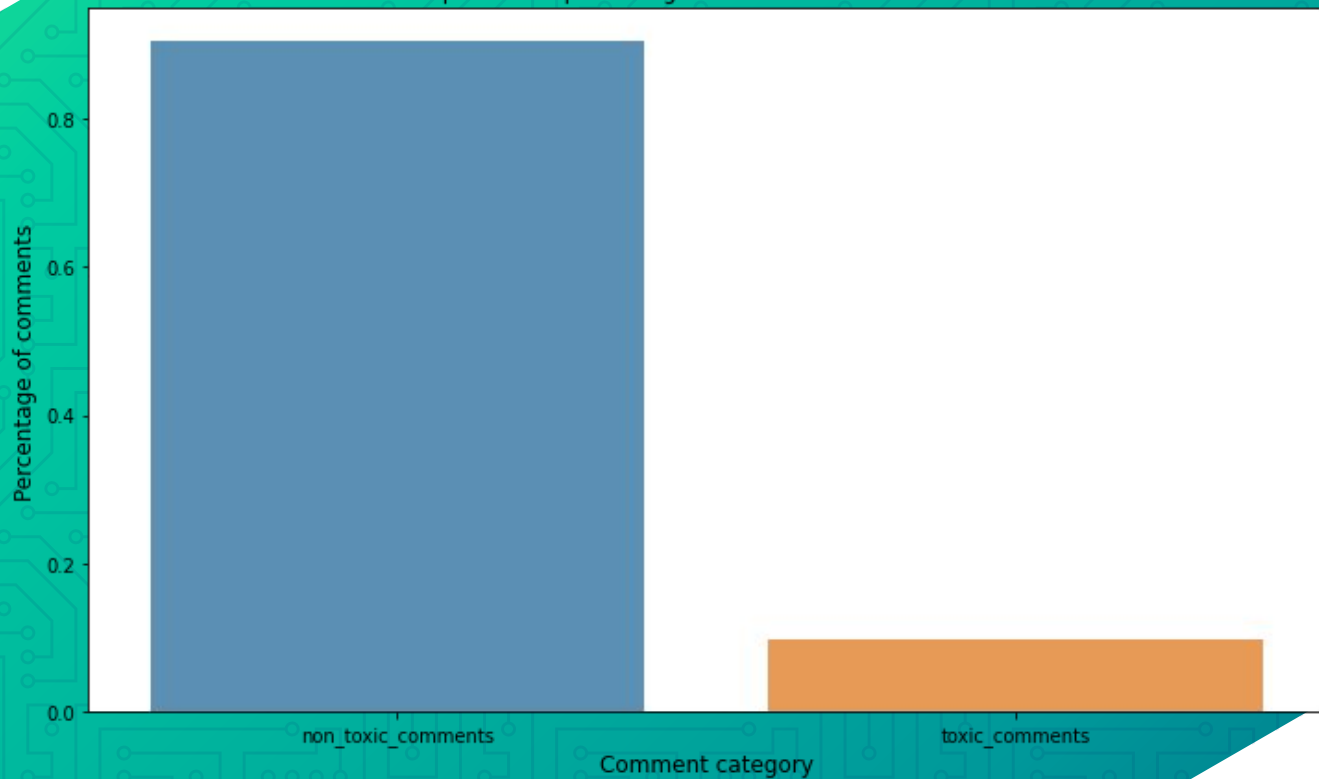
Number of toxic comment for each category:

toxic	15294
severe_toxic	1595
obscene	8449
threat	478
insult	7877
identity_hate	1405

- No null values to deal with
- Read comments properly using **neattext** library
- NLP and pre-processing: Clean data by removing punctuation, stopwords, lower casing ...
- Check repartition of toxic comments:
  - 90% of the comments have no form of toxicity at all.

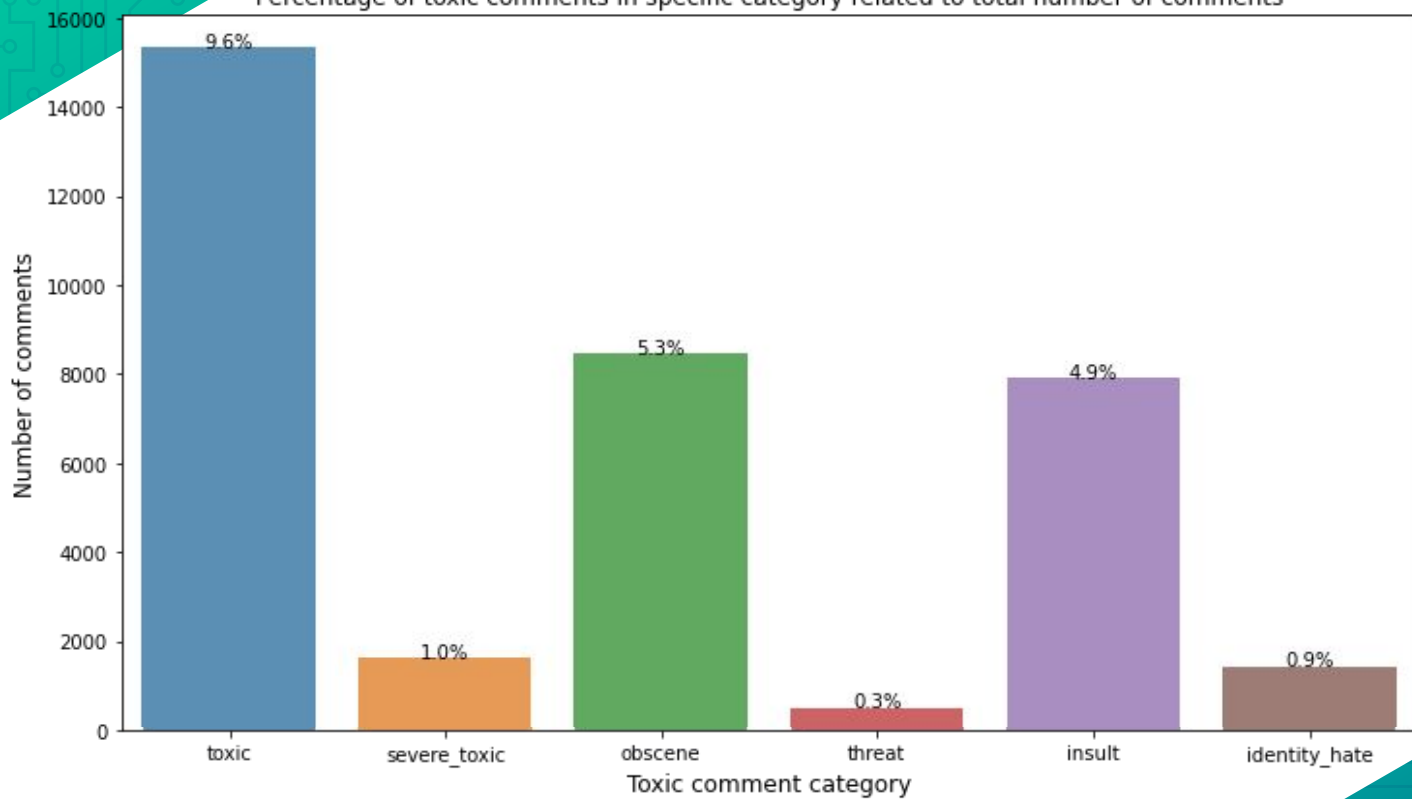
# Only around 10% comments have a form of toxicity

Barplot of the percentage of Clean VS Toxic comments





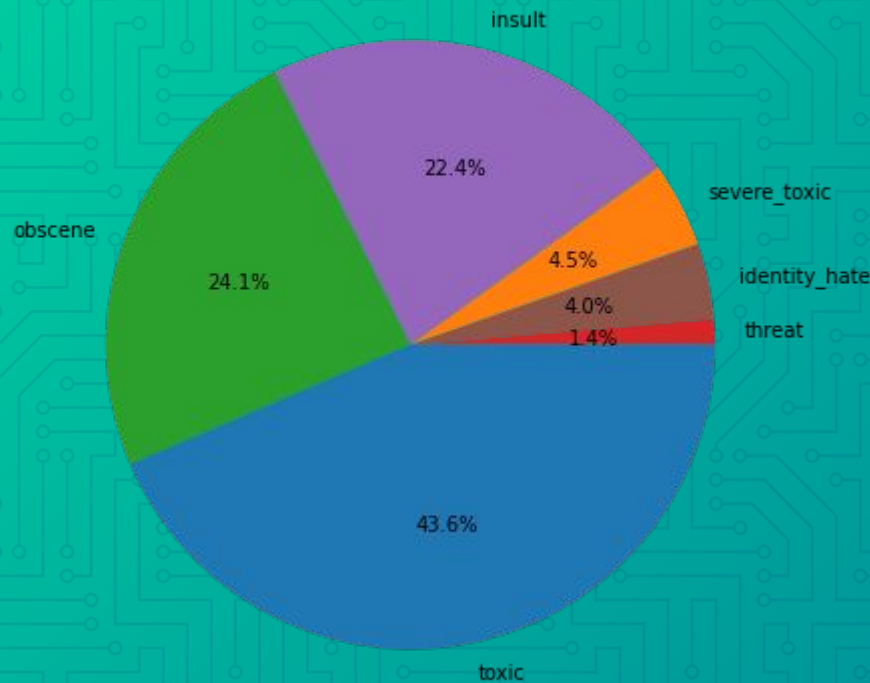
Barplot of number of comments per toxic category &  
Percentage of toxic comments in specific category related to total number of comments



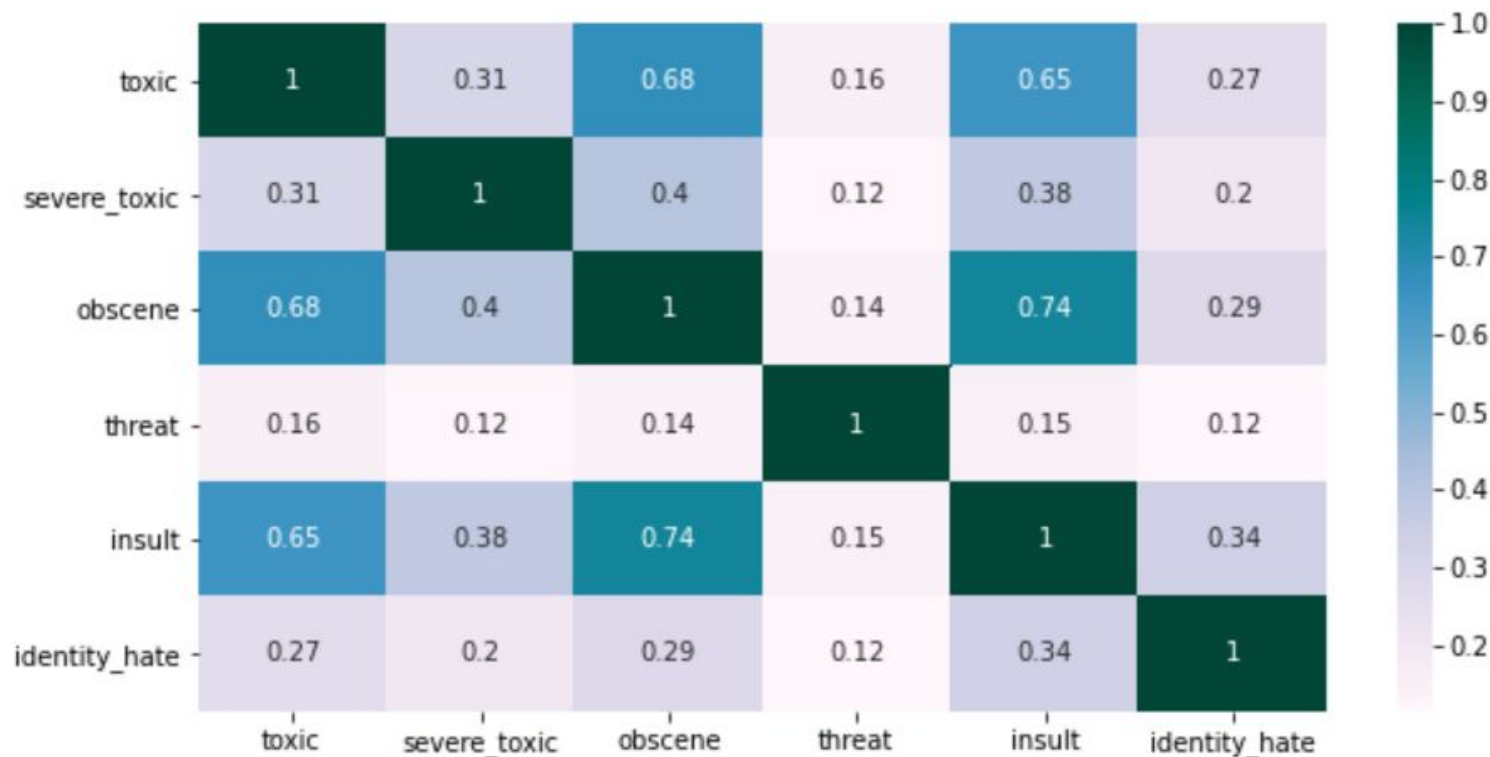


# Toxic, Insults & Obscene comments compose the majority of toxic comments

Pie chart of the percentage of toxic comments in each category related to the total number of toxic comments

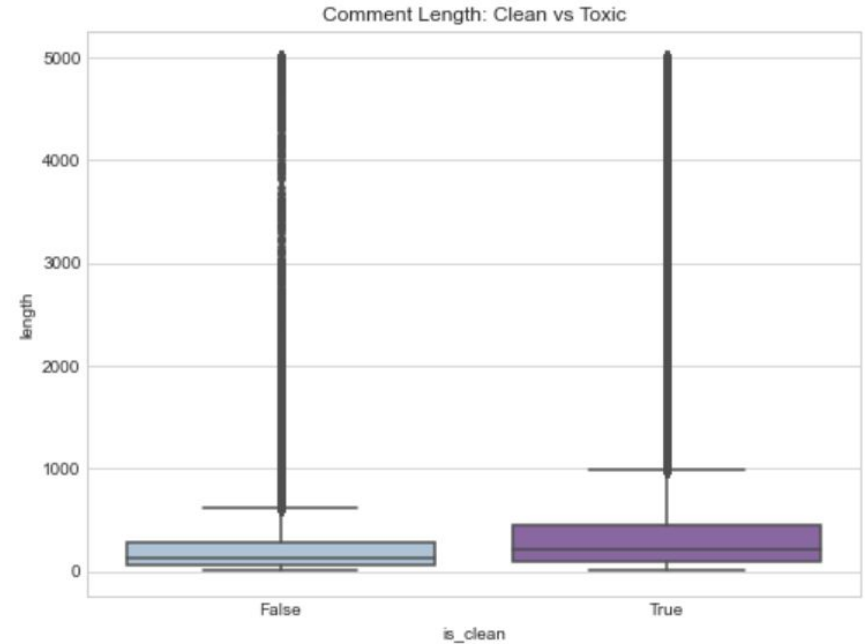
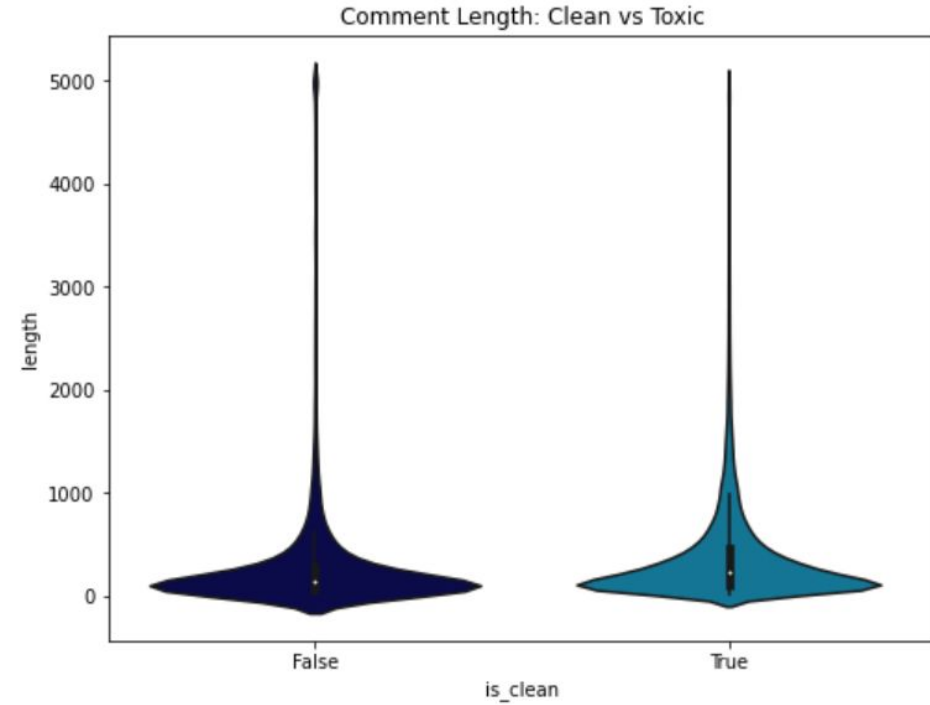


# Correlation Matrix of toxic categories

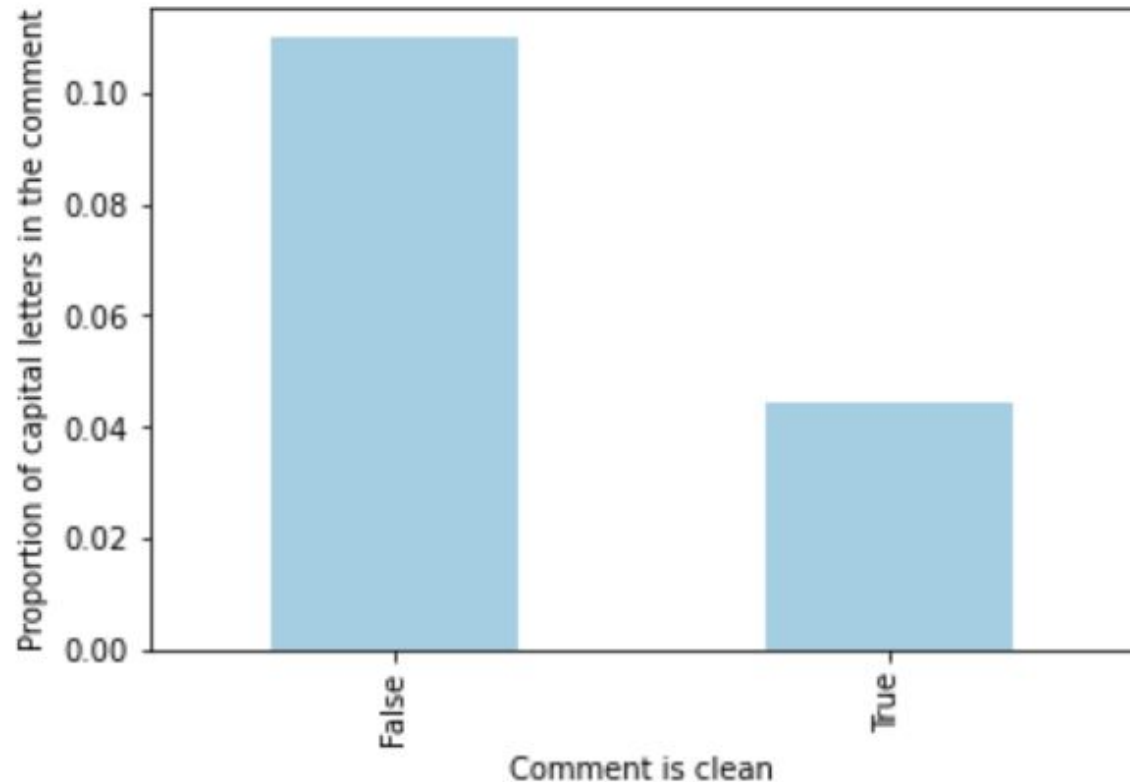


[illegible][illegible][illegible]

# Length and toxicity

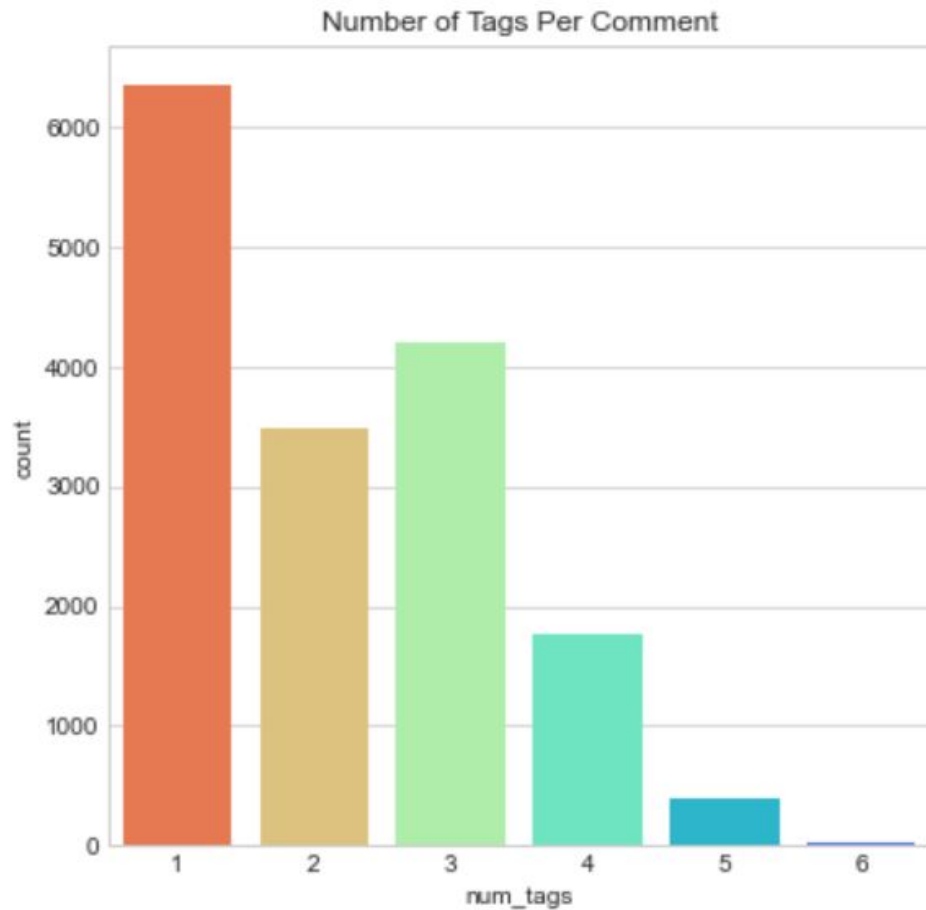


# Capital letters in toxic comments



	length	capitals	proportion
is_clean			
False	305.672604	42.980770	0.109680
True	406.885138	14.236993	0.044497

# Number of tags for each toxic comment



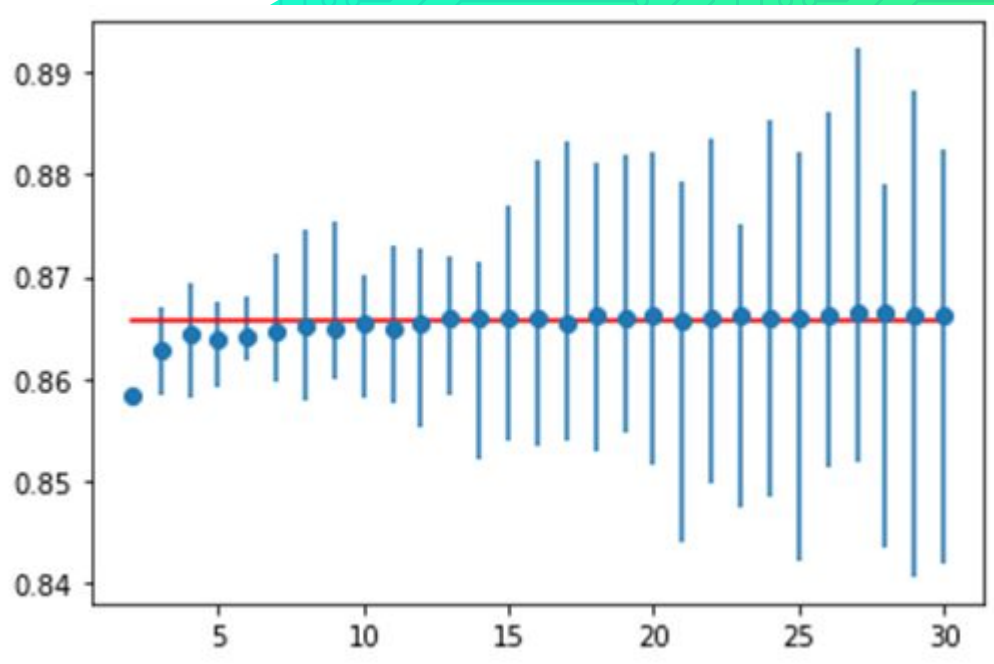


**We took a subset of the dataset to create 6 datasets (one for each level of toxicity)**

**We balanced each datasets to have a 50-50 frequency of toxic-nontoxic words**



# Logistic regression. Cross-validation



```
Ideal: 0.866
> folds=2, accuracy=0.858 (0.858,0.859)
> folds=3, accuracy=0.863 (0.858,0.867)
> folds=4, accuracy=0.864 (0.858,0.869)
> folds=5, accuracy=0.864 (0.859,0.868)
> folds=6, accuracy=0.864 (0.862,0.868)
> folds=7, accuracy=0.865 (0.860,0.872)
> folds=8, accuracy=0.865 (0.858,0.874)
> folds=9, accuracy=0.865 (0.860,0.875)
> folds=10, accuracy=0.865 (0.858,0.870)
> folds=11, accuracy=0.865 (0.858,0.873)
> folds=12, accuracy=0.866 (0.855,0.873)
> folds=13, accuracy=0.866 (0.858,0.872)
> folds=14, accuracy=0.866 (0.852,0.871)
> folds=15, accuracy=0.866 (0.854,0.877)
> folds=16, accuracy=0.866 (0.853,0.881)
> folds=17, accuracy=0.865 (0.854,0.883)
> folds=18, accuracy=0.866 (0.853,0.881)
> folds=19, accuracy=0.866 (0.855,0.882)
> folds=20, accuracy=0.866 (0.852,0.882)
> folds=21, accuracy=0.866 (0.844,0.879)
> folds=22, accuracy=0.866 (0.850,0.883)
> folds=23, accuracy=0.866 (0.847,0.875)
> folds=24, accuracy=0.866 (0.849,0.885)
> folds=25, accuracy=0.866 (0.842,0.882)
> folds=26, accuracy=0.866 (0.851,0.886)
> folds=27, accuracy=0.867 (0.852,0.892)
> folds=28, accuracy=0.866 (0.843,0.879)
> folds=29, accuracy=0.866 (0.841,0.888)
```



# Logistic regression.

## AUC (Area Under the Curve):

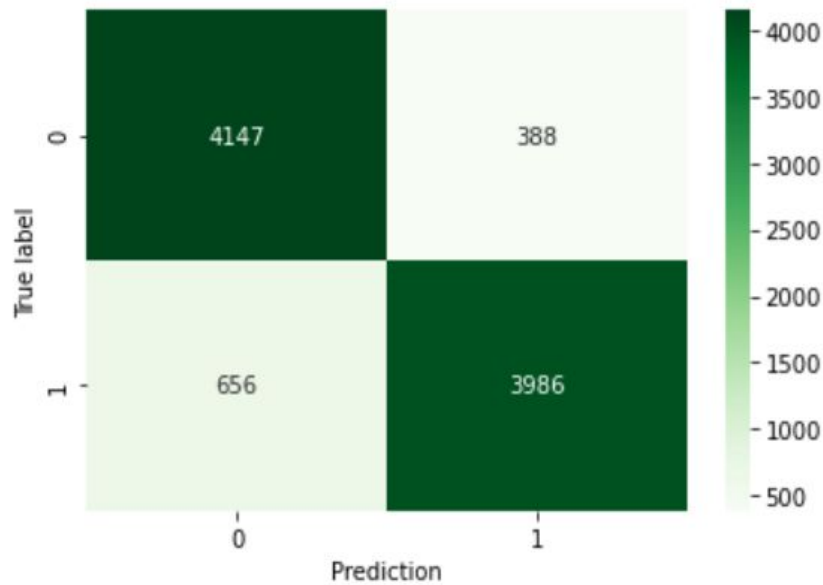
- measure of the ability of a classifier to distinguish between classes.
- summary of the ROC curve.

*The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.*

## Log Loss:

- applies to the prediction process in machine learning related to probabilities

*The lower the log loss, the more accurate predictions your AI will make, meaning its overall accuracy and functionality will rise.*



Accuracy: 0.886

AUC: 0.947

AUC2: 0.887

0.887

Log loss: 0.381

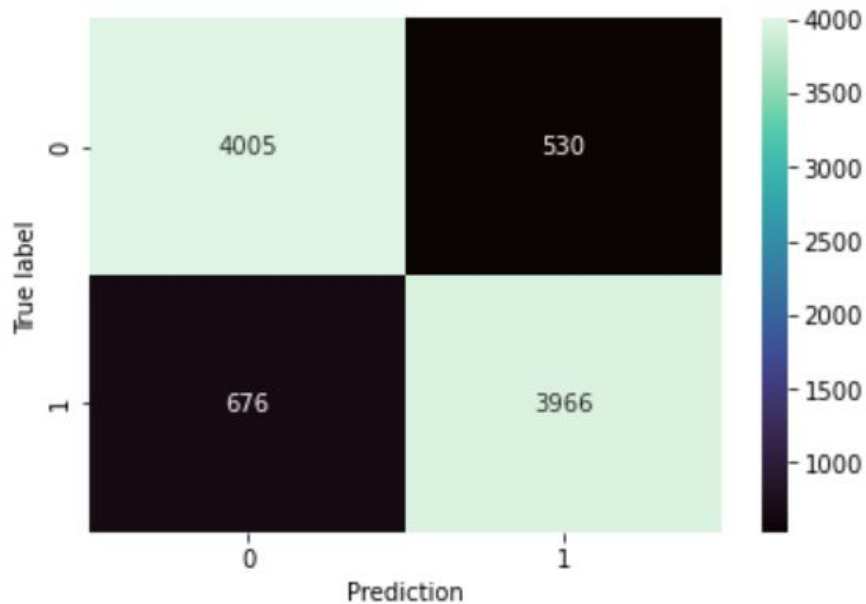
TN 4147

FP 388

FN 656

TP 3986

# SVM



Accuracy: 0.869

AUC2: 0.869

0.869

Log loss: 0.381

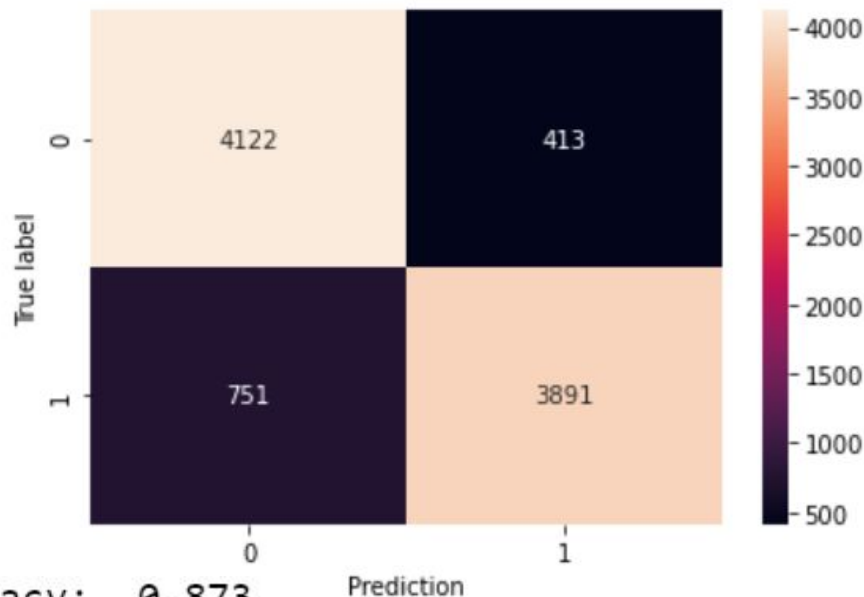
TN 4005

FP 530

FN 676

TP 3966

# Multinomial Naive Bayes



Accuracy: 0.873

AUC: 0.928

AUC2: 0.874

0.874

Log loss: 0.797

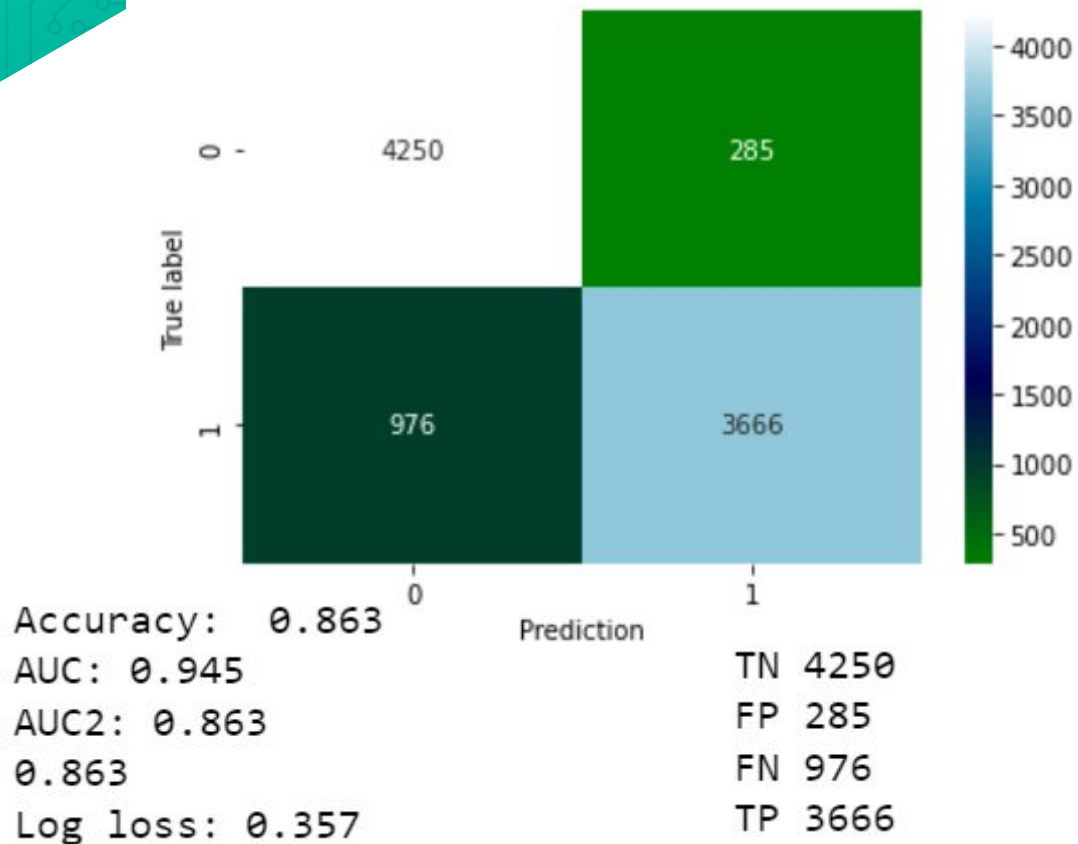
TN 4122

FP 413

FN 751

TP 3891

# Random Forest



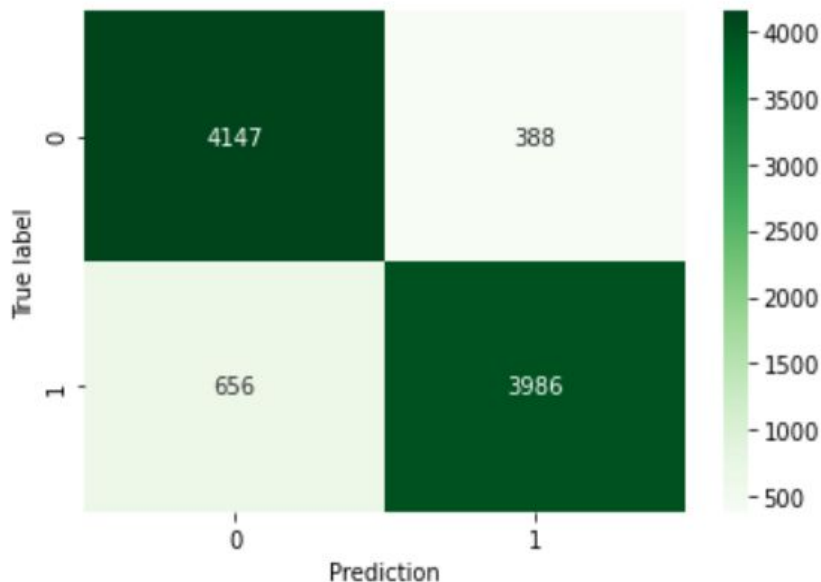
# Comparison for Toxic column

Algorithm	KNN	Logistic regression	SVM	Naive Bayes	Random Forest
Accuracy	0.728	0.886	0.869	0.873	0.863
AUC	0.814	0.887	0.869	0.874	0.863
Log loss		0.381	0.381	0.797	0.357

# CountVectorizer vs TFDIF

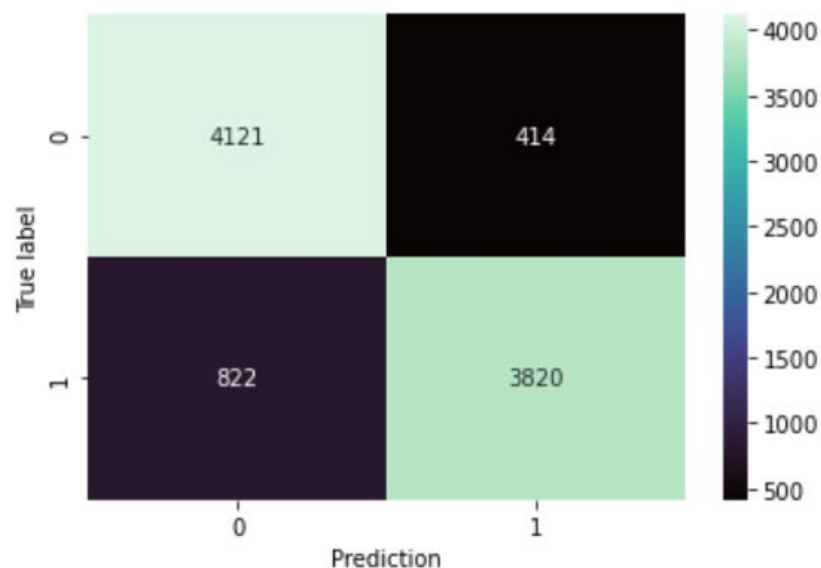
*word frequency*

*word frequency & importance/weight*



Accuracy: 0.886  
AUC: 0.947  
AUC2: 0.887  
0.887  
Log loss: 0.381

TN 4147  
FP 388  
FN 656  
TP 3986



Accuracy: 0.865  
AUC: 0.943  
AUC2: 0.866  
0.866  
Log loss: 0.301

TN 4121  
FP 414  
FN 822  
TP 3820

# Classifier Chain

	Model	Accuracy	AUC	Log loss
0	Chain-LogisticRegression	0.918	0.698	2.838

# To go further.

More time for  
tuning:

- Find the best **parameters**
- Find the best **vectorization** library

Compare the models  
accuracy of deep  
NLP VS no NLP

Build an app to  
detect and erase  
toxic comments



**Thank you for your Time !**  
**If you have any questions?**

