Moderation System for Hate Speech Detection

AGENDA

- Problem
- Proposed Solution
- Literature Survey
- More about Dataset
- Implementation
- Demo
- Impact
- Future work

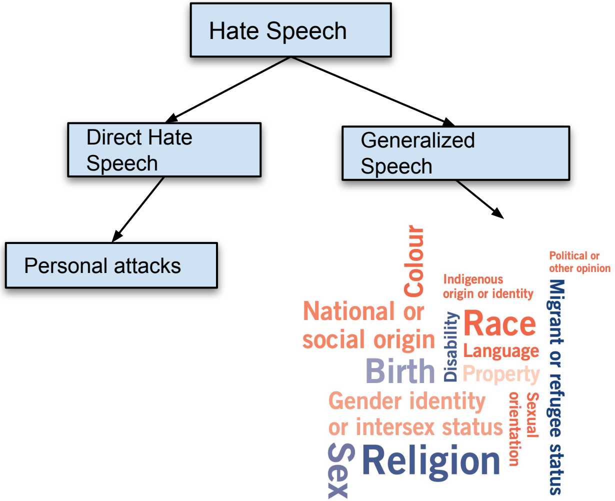
What is Online Hate Speech?

- Hate speech is a speech that attacks a person or group on the basis of attributes such as race, religion, ethnic origin, national origin, gender, disability, sexual orientation.

Intense and irrational emotion of opprobrium, enmity and detestation towards an individual or group.

Any expression of hate towards an individual or group defined by a protected characteristic.

Any expression imparting opinions or ideas – bringing an internal opinion or idea to an external audience. It can take many forms: written, non-verbal, visual, artistic, etc, and may be disseminated through any media, including internet, print, radio, or television.


Hate Speech

Direct Hate Speech

Generalized Speech

Personal attacks

Colour
Indigenous origin or identity
Political or other opinion
National or social origin
Disability
Race
Language
Migrant or refugee status
Birth
Property
Gender identity or intersex status
Sexual orientation
Sex
Religion

Motivation

- With increasing anonymity and flexibility provided by the Internet, it has made it easy for users to communicate in an aggressive manner

- Hate speech on social media could also lead to harassment, bullying, depression

  Fact : The most common type of online bullying is **mean comments 22.5%**.

Cyberbullying on social media is considered a much bigger threat than in-person bullying

Can happen around the clock, 24/7

Tends to be more permanent

Difficult to pinpoint as typically not in places easily seen

Problem Statement

With increase in amount of aggressive content, methods that **Automatically detect hate speech** are very much required

Education on media ethics and awareness about the impact of hate speech could contribute reducing the hate content on social media

Proposed solution

To develop a ‘Moderation System for Hate Speech Detection’ which can be embedded in the post section of any social media platform

The model alerts users on **Hate Speech Content before posting** and allows them to rethink before **publishing it on social media platforms**

Educate users on social media policies on hate speech

Literature Survey

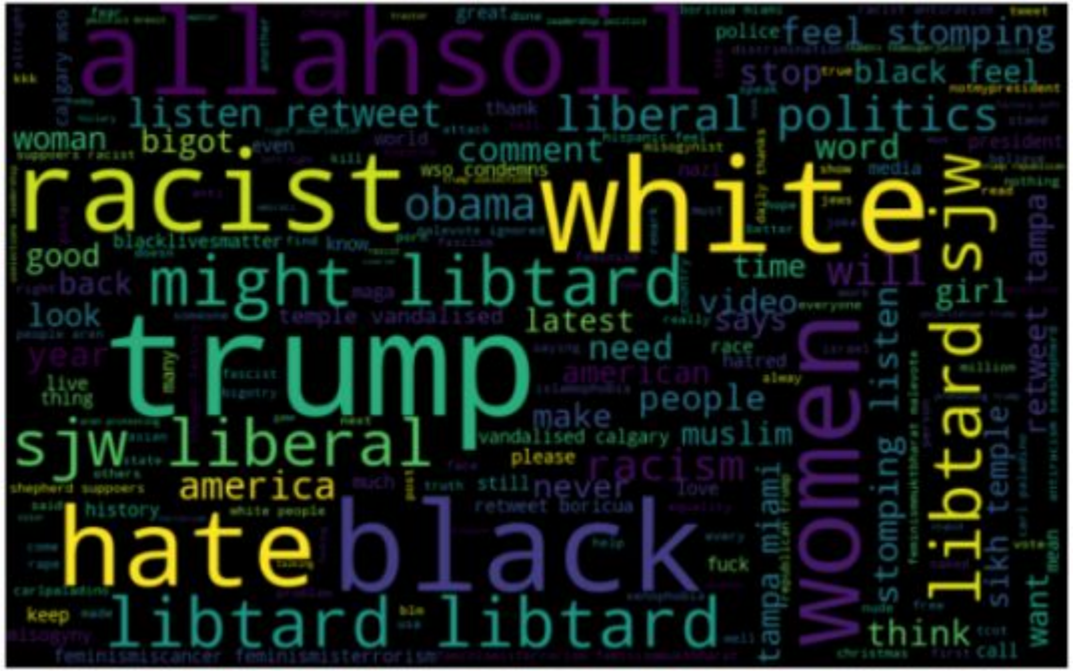
Reference	**Dataset**	**Technique**	**Results**
Greevy Edel (2004)	**PRINCIP Corpus** **Size:** 3M words from tweets	**Model**: SVM **Feature Extraction**: BOW, Bi-gram	**BOW:** Precision: 92.5% Recall: 87% **Bi-gram** Precision: 92.5% Recall: 87%
Waseem and Hovy (2016)	**Total Annotated tweets:** 16,914 #Sexist tweets: 3,383 #Racist tweets: 1,972 #tweets Neither racist nor sexist: 11,559	**Model**: Char n-grams Word n-grams	**Char n-gram:** Precision: 73.89% Recall: 77.75% F1 score: 72.87% **Word n-grams:** Precision: 64.58% Recall: 71.93% F1 score: 64.58%
Akshita et al (2016)	Waseem and Hovy, 2016 **Size**: 22,142 tweets **Class**: Benevolent, Hostile, others	**Model**: SVM, Seq2Seq (LSTM), FastText Classifier(by Facebook AI research) **Feature Extraction**: TF-IDF, Bag of n-words	Average F1 score among classes: 0.723(SVM), 0.74(Seq2Seq) Overall F1 Score: 0.84(FastText)

The Dataset

	Attributes	**Description**
Train data	Id	Unique number assigned to each tweet
	Label	Contains label’s data (1 : Hate , 0 : Not-Hate)
	Tweet	Unique Sentences
Test data	Id	Unique number assigned to each tweet
	Tweet	Unique Sentences

- **Dataset**: Twitter tweets data to do sentiment analysis (https://www.kaggle.com/nitin194/twitter-sentiment-analysis)
- **Number of tweets:** 31,935
- **Classes (%):** Not-Hate Labeled(93%), Hate Labeled(7%)
- **Target Class**: Hate, Offensive, Abusive

Word Cloud of Hate Speech Tweets



Implementation

Techniques used

Data Cleaning

Lemmatization,Stemming, Tokenization, Removal of stopwords, emoji, URL, orphaned characters and slang words, replace shorthand words

Word Embedding Techniques and Bag of Words

Word2Vec with genism, TF IDF Vectorizor

Feature Selection

Chi-Square Test, Lime Text Explainer

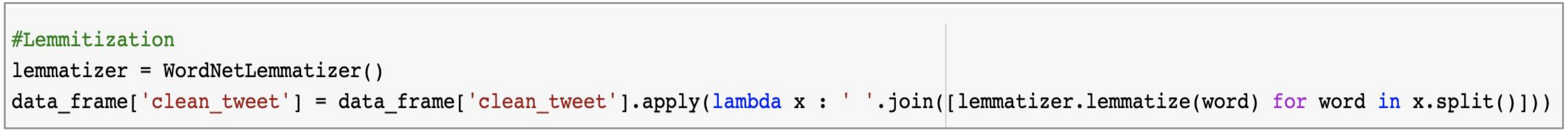
OverSampling and Classification Algorithms:

RandomOverSampler, Best model adoption using Autogluon

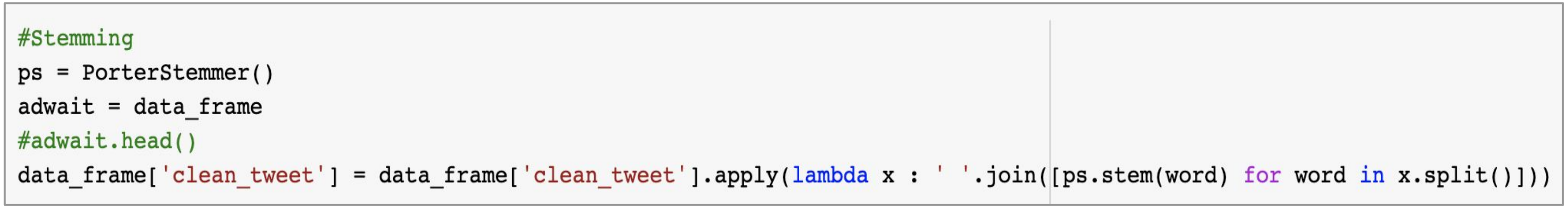
Language Modelling

BERT, DistilBERT

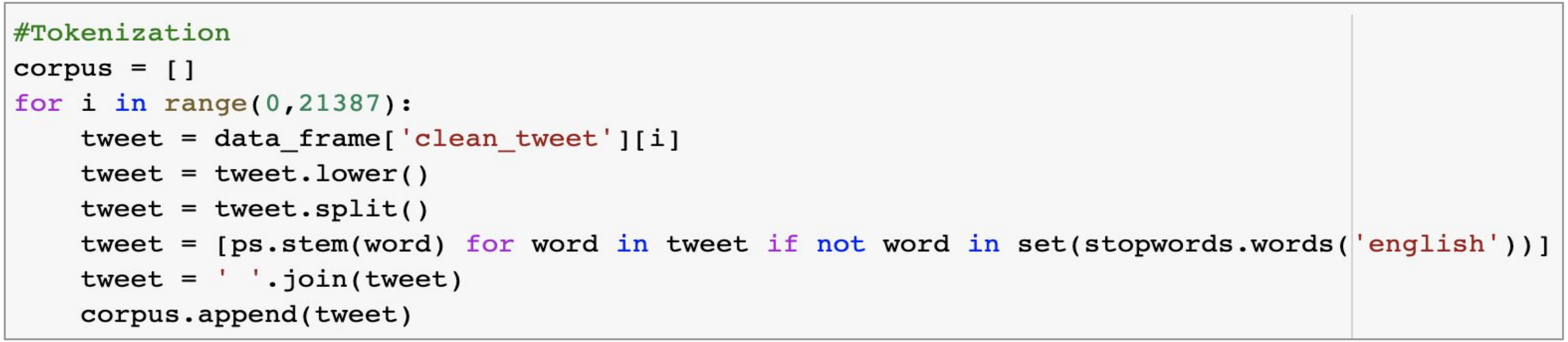
Procedure



```
#Lemmitization
lemmatizer = WordNetLemmatizer()
data_frame['clean_tweet'] = data_frame['clean_tweet'].apply(lambda x : ' '.join([lemmatizer.lemmatize(word) for word in x.split()]))
```

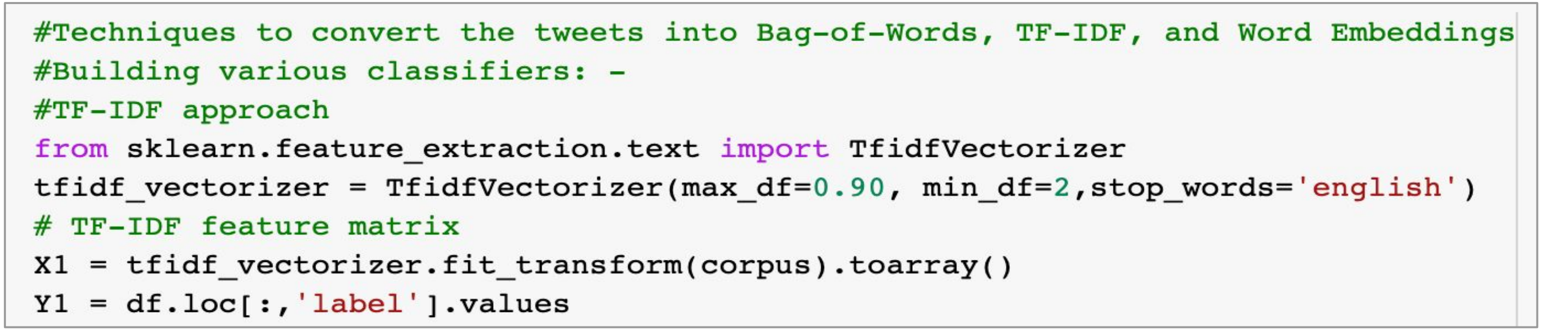


```
#Stemming
ps = PorterStemmer()
adwait = data_frame
#adwait.head()
data_frame['clean_tweet'] = data_frame['clean_tweet'].apply(lambda x : ' '.join([ps.stem(word) for word in x.split()]))
```

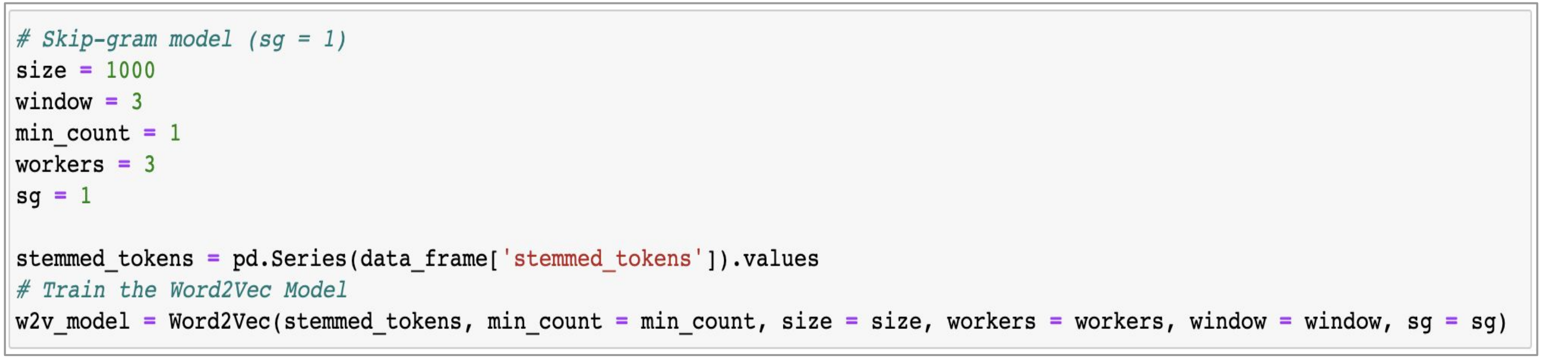


```
#Tokenization
corpus = []
for i in range(0,21387):
    tweet = data_frame['clean_tweet'][i]
    tweet = tweet.lower()
    tweet = tweet.split()
    tweet = [ps.stem(word) for word in tweet if not word in set(stopwords.words('english'))]
    tweet = ' '.join(tweet)
    corpus.append(tweet)
```

Procedure (cntd.)



```
#Techniques to convert the tweets into Bag-of-Words, TF-IDF, and Word Embeddings
#Building various classifiers: -
#TF-IDF approach
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(max_df=0.90, min_df=2,stop_words='english')
# TF-IDF feature matrix
X1 = tfidf_vectorizer.fit_transform(corpus).toarray()
Y1 = df.loc[:,'label'].values
```



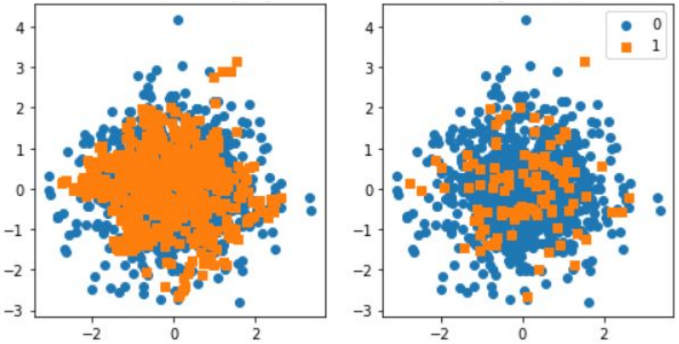
```
# Skip-gram model (sg = 1)
size = 1000
window = 3
min_count = 1
workers = 3
sg = 1

stemmed_tokens = pd.Series(data_frame['stemmed_tokens']).values
# Train the Word2Vec Model
w2v_model = Word2Vec(stemmed_tokens, min_count = min_count, size = size, workers = workers, window = window, sg = sg)
```

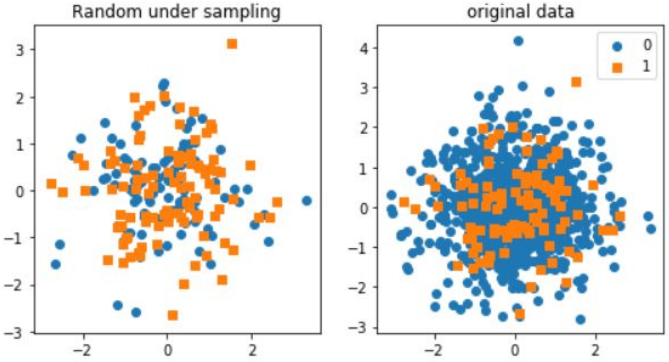
Random Oversampling and UnderSampling

0 : Not- Hate
1 : Hate

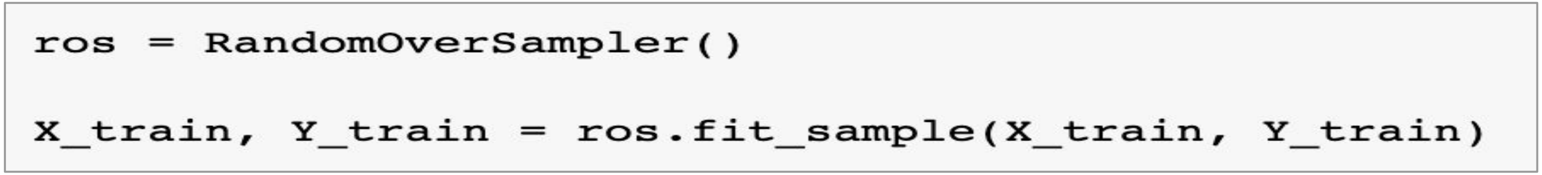
Random Over Sampling

Original Data

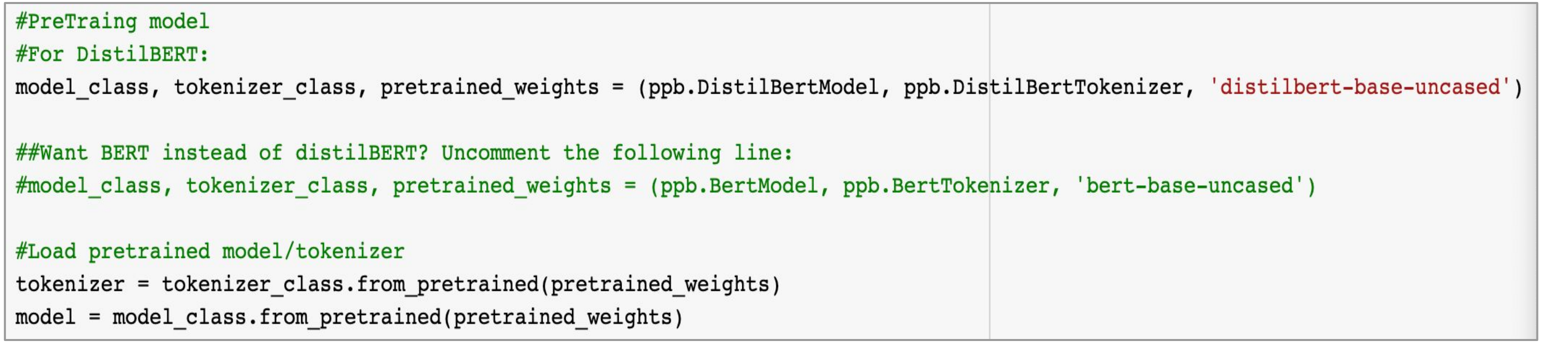




Procedure (cntd.)



```
ros = RandomOverSampler()

X_train, Y_train = ros.fit_sample(X_train, Y_train)
```



```
#PreTraing model
#For DistilBERT:
model_class, tokenizer_class, pretrained_weights = (ppb.DistilBertModel, ppb.DistilBertTokenizer, 'distilbert-base-uncased')

##Want BERT instead of distilBERT? Uncomment the following line:
#model_class, tokenizer_class, pretrained_weights = (ppb.BertModel, ppb.BertTokenizer, 'bert-base-uncased')

#Load pretrained model/tokenizer
tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)
```

Performance

Results

	Model	Class	Precision	Recall	F1 Score	Accuracy
1	LightGBM ClassifierCustom with AutoGluon	0	0.95	0.99	0.95	95%
		1	0.84	0.44	0.58	
2	RandomForestClassifier with TfidfVectorizer	0	0.96	1.00	0.98	96%
		1	0.93	0.49	0.64	
3	RandomForestClassifier with Word2Vec	0	0.93	1.00	0.96	93%
		1	0.91	0.34	0.51	
4	distilBERT	0	0.67	0.017	0.016	94%
		1	0.51	0.012	0.23	

Impact

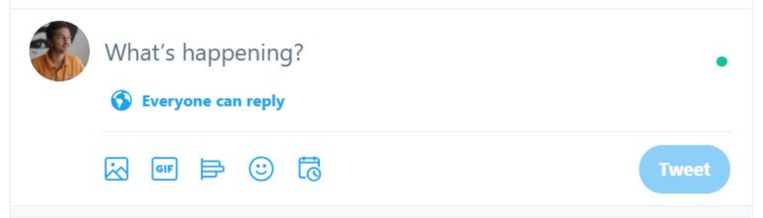
Restricting spread of hate messages

Reduction in cyber bullying and harassment.

Building a peaceful community

Giving users second chance

Digital media Literacy

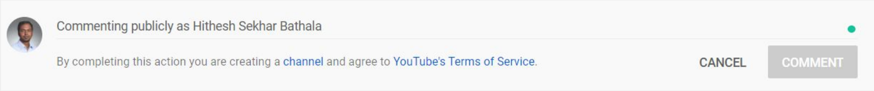


What's happening?

Everyone can reply

Your Tweet might hurt people! Click here to know more...

Tweet Anyway | Delete Tweet



Commenting publicly as Hithesh Sekhar Bathala

By completing this action you are creating a channel and agree to YouTube's Terms of Service.

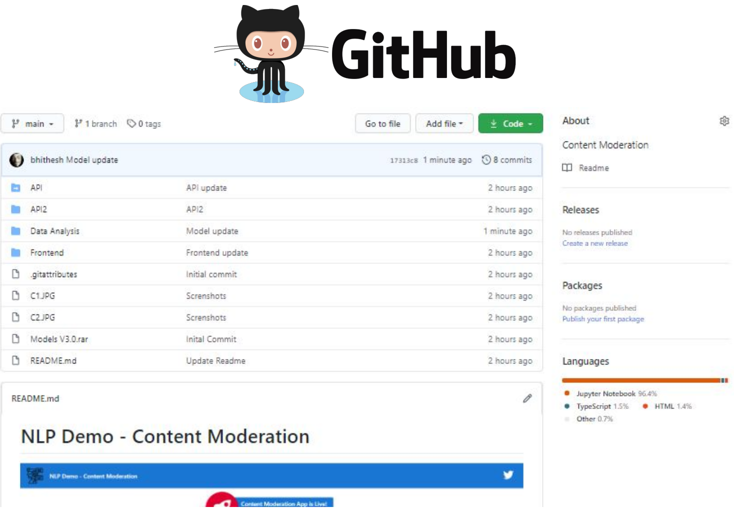
What's Happening !!

Tweet#

Tweet#

Future Work

- Further fine tuning of the hyperparameters to improve accuracy on the dataset.
- Add more features to the dataset: Number of followers, location, age, etc.
- Use Multi-class classification to categorize the sentiment of the tweets.
- Include tweets in other languages: French, Hindi, etc.



bhithesh Model update 17313c8 1 minute ago 8 commits

API	API update	2 hours ago
API2	API2	2 hours ago
Data Analysis	Model update	1 minute ago
Frontend	Frontend update	2 hours ago
.gitattributes	Initial commit	2 hours ago
C1.JPG	Screnshots	2 hours ago
C2.JPG	Screnshots	2 hours ago
Models V3.0.rar	Inital Commit	2 hours ago
README.md	Update Readme	2 hours ago

README.md

NLP Demo - Content Moderation

NLP Demo - Content Moderation

Content Moderation App is Live!

About

Content Moderation

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- Jupyter Notebook 96.4%
- TypeScript 1.5%
- HTML 1.4%
- Other 0.7%

Contribute to our project Pull Today !

https://github.com/bhithesh/NLP-Demo

References

- ML Class Notes: https://srdas.github.io/MLBook2/
- https://scikit-learn.org/stable/
- https://huggingface.co/transformers/model_doc/distilbert.html
- https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/
- https://towardsdatascience.com/end-to-end-deployment-of-a-machine-learning-model-using-flask-dc456abcc6da
- https://medium.com/@tenzin_ngodup/simple-text-classification-using-random-forest-fe230be1e857
- https://www.kaggle.com/shahules/tackling-class-imbalance
- https://stackabuse.com/text-classification-with-python-and-scikit-learn/
- https://www.analyticsvidhya.com/blog/2018/07/hands-on-sentiment-analysis-dataset-python/
- https://towardsdatascience.com/another-twitter-sentiment-analysis-bb5b01ebad90
- https://auto.gluon.ai/stable/tutorials/tabular_prediction/tabular-quickstart.html
- https://www.kaggle.com/c/detecting-insults-in-social-commentary/data
- https://marcotcr.github.io/lime/tutorials/Lime%20-%20basic%20usage%2C%20two%20class%20case.html
- https://rstudio-pubs-static.s3.amazonaws.com/343661_dc127bbf141845b083b2dfa2cc75c9d2.html
- https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data
- https://www.researchgate.net/publication/29651698_Classifying_racist_texts_using_a_support_vector_machine

Thank you !