Hate Speech Detection

Group Members:

Suet Wah CHU
Michael WARNER
Jing Fei XU
Gaylor BRUNNER
Yasser MARZOUKI
Purnama Sari SITORUS

TABLE OF CONTENTS

01 BUSINESS PROBLEM

02 DATA PREPARATION

03 MODEL BUILDING

04 RESULTS

01

Business Problem

BUSINESS PROBLEM

- Content Moderation:
  - Screen and monitor user-generated content

- Human Moderator:
  - Humans manually monitor and screen content
  - Go through thousands of visuals per day
  - Make super-quick decisions about the appropriateness of content

Negative Psychological Effects

BUSINESS PROBLEM

Some Facts :

- on average 500 million tweets per day in 2020
- manually moderating all of that traffic is close to impossible

Consequences :

- Problem for Twitter's reputation
- Problem for advertising

LEGAL PROBLEM FOR THE PLATFORM

Possibilities for victims in France :

- Sue the author of the attack
- **Sue the platform**

Possible even if **the platform is not hosted in France.**

Platforms are responsible of the content they accept to show.

Note : this problem deals with the same debate as the protected content such as films, musics etc.

LEGAL PROBLEM FOR THE PLATFORM

In the USA :

Since 1996 : platforms cannot be sued for the content there are showing
September 2020 : law project about the responsibility of the platforms on the internet.

The reason it did not exist :
It was technically impossible and there were less people on the internet than nowadays

How could AI optimize the *content moderation* processes for business and reduce harmful effects on *human moderators*?

02 Data Preparation

Datasets

01 Twitter with 3 Categories

- ★ hate speech, offensive language and neither
- ★ **~25k** tweets

02 Twitter with 2 Categories

- ★ Hate speech & non-hate speech
- ★ **2177** tweets of hate speech are added

03 Gab (Hate Speech Only)

- ★ American alt-tech social media known for its far-right user base
- ★ **7363** hate speech post are added

Unbalanced Dataset

Base Dataset:

<u>hate speech:</u>

Total: 24783

hate: **1430 (5.77% of total)**

<u>offensive speech:</u>

Total: 24783

Offensive: 19190 (77.43% of total)

<u>neither:</u>

Total: 24783

Neither: 4163 (16.80% of total)

+

~ 9500 datapoint

After Adding Extra Datasets:

<u>hate speech:</u>

Total: 34323

hate: **10970 (31.96% of total)**

<u>offensive speech:</u>

Total: 34323

Offensive: 19190 (55.91% of total)

<u>neither:</u>

Total: 34323

Neither: 4163 (12.13% of total)

Train-Test Split

80 %

Training

20 %

Testing

Data Cleaning

- **Step 1** Remove web links, retweet('RT') and username(@)
- **Step 2** Remove punctuation, numbers and stopwords
- **Step 3** Turn all words into lowercase, tokenize and lemmatize them

tweet
!!! RT @mayasolovely: As a woman you shouldn't...
!!!!! RT @mleew17: boy dats cold...tyga dwn ba...
!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
!!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...

clean_tweet
woman complain cleaning house man always take...
boy dat cold tyga dwn bad cuffin dat hoe st p...
dawg ever fuck bitch start cry confused shit
look like tranny
shit hear might true might faker bitch told ya

Term Frequency-Inverse Document Frequency

- ★ Calculation of how relevant a word in a series or corpus is to a text
- ★ The words with higher scores of weight are deemed to be more significant

abandon	abandonado	abandoned	abandoning	abased	abba	abbey	abbot	abby	abc	abcaustralia	abcnews	abd	abde	abdelka
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

03

Model Building

Models

Logistic Regression

Random Forest

Support Vector Machine

Naive Bayes

Neural Network

Modeling Process

1. Lemmatization + TF-IDF
2. Oversampling with SMOTE
3. Undersampling with Tomek Links
4. Grid Search

EVALUATE ALL MODELS

	Precision	Recall	F1_score
Baseline Random Forest	0.891396	0.891916	0.891604
Baseline Logistic Regression	0.899073	0.889294	0.891361
Logistic Regression - SMOTE	0.898562	0.892207	0.893589
Logistic Regression - TOMEK	0.895652	0.895266	0.895232
Logistic Regression - Grid Search	0.900083	0.893518	0.895046
Baseline Naive Bayes	0.813314	0.815732	0.806221
Baseline SVM	0.903907	0.897451	0.898848
SVM - SMOTE	0.899187	0.894829	0.895865
SVM - TOMEK	0.903273	0.896431	0.897903

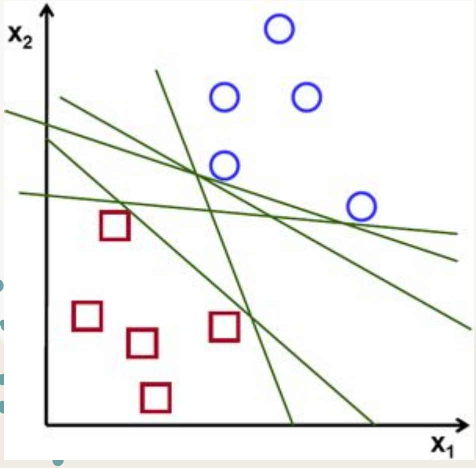
What is SVM ?

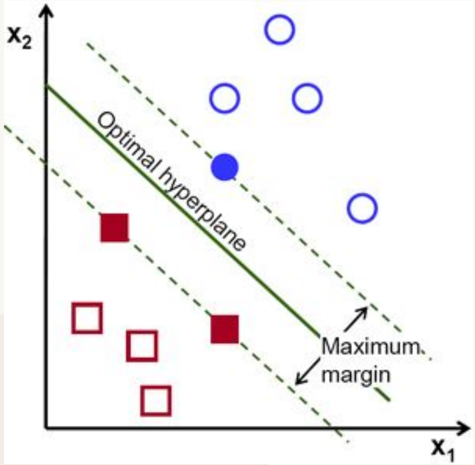
Like a lot of other ML algorithms, Support Vector Machine takes some data that is already classified, the training set, and tries to predict a set of unclassified data, the testing set. In our case we use the prepared tweets as our input and the class as the output.

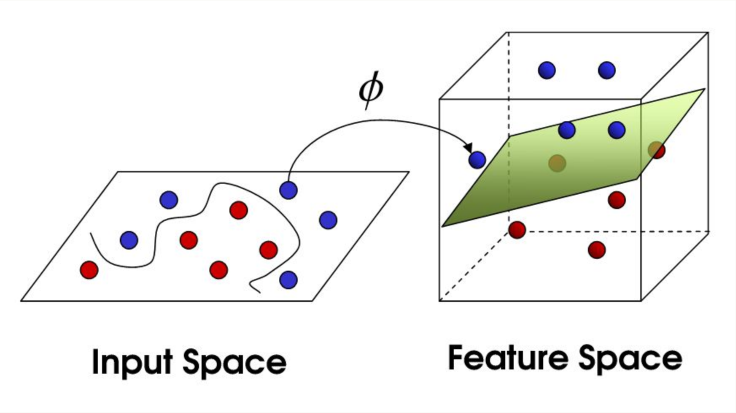
In simple terms, we can imagine every data item as a plotted point in space. The values of each feature being the coordinates.

SVM will put a "separation" in the best way possible by having as big a gap as possible between the closest data point from each of the groups and the separation. That is our optimal hyperplane.

Also, when we draw our line, we realize that only a few of our data items are actually useful in drawing the line : they are our support vectors !



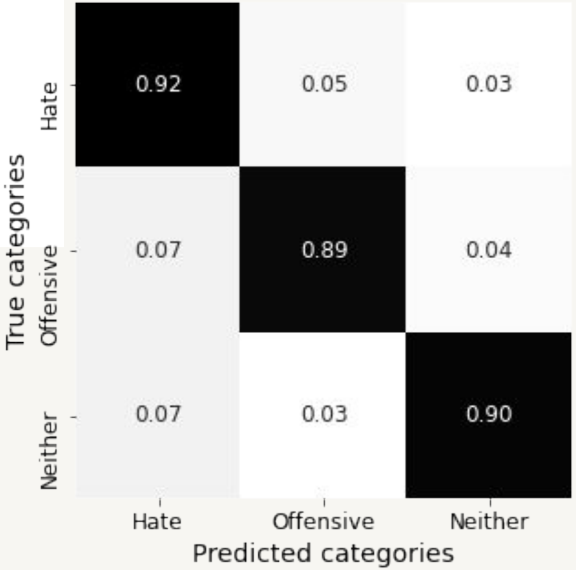




BEST MODEL - SVM

```
SVM_baseline = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto', class_weight='balance

SVM_baseline.fit(tfidf_data_train, y_train)
SVM_test_preds = SVM_baseline.predict(tfidf_data_test)

SVM_baseline_report = classification_report(y_test, SVM_test_preds)
print(SVM_baseline_report)
```

```
              precision    recall  f1-score   support

           0       0.86      0.92      0.89      2192
           1       0.96      0.89      0.92      3844
           2       0.77      0.90      0.83       829

    accuracy                           0.90      6865
   macro avg       0.86      0.90      0.88      6865
weighted avg       0.90      0.90      0.90      6865
```



Hate Speech:
Recall = 0.92
F1 = 0.89

Offensive Language:
Recall = 0.89
F1 = 0.92

04 Results

Hate Speech Detection Application

Streamlit

- Open-source app framework

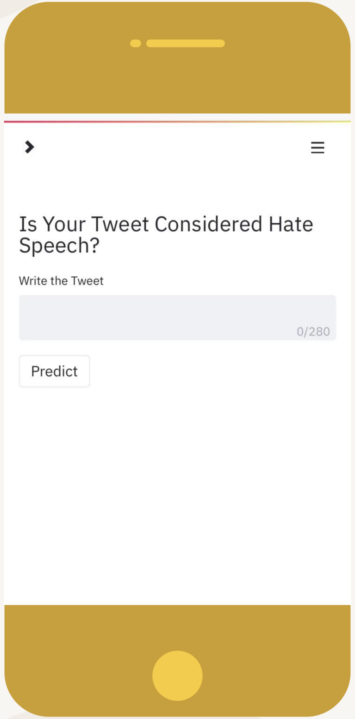
HEROKU

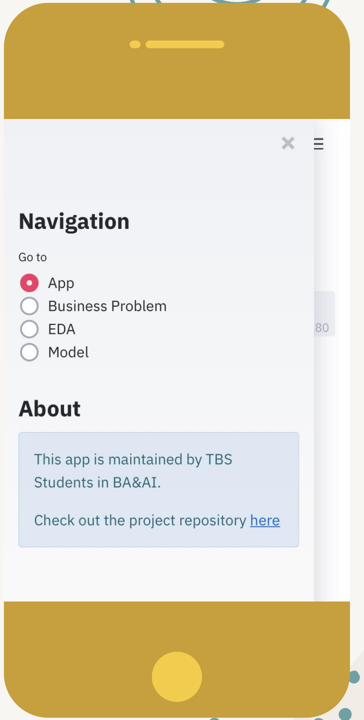
- Cloud application platform
- Build and deploy web apps

WEB APP

Link:

https://hate-speech-detection-tbs.herokuapp.com





CONCLUSIONS

Our app is able to label tweets as hate speech, offensive or neither. This will allow companies to save time and resources, along with protecting the mental health of its employees.

Room for improvement, as we have a limited labeled dataset, and it is subjective as to what is hate speech, this causes our model to declare some tweets as hate speech when it is actually not.

THANKS!

Do you have any questions?