A MACHINE LEARNING APPROACH TO IDENTIFYING HATE SPEECH

Jarod DeWeese

Inspiration for project

- People infiltrating KSUnite and spewing hate speech.
- A friend of mine was specifically targeted by this.

Challenges

- Class imbalance
- Obtaining data via Twitter API
- Runtime environment
- Emojis, and other markers in data
- Team of 1
- Creating report

Runtime/Tooling/Libraries

- PyCharm
- Pandas, but if I take this farther, it will converge towards PySpark
- NLTK – stemming, lemmatizing, P-POS
- Sklearn
- vaderSentiment – minimal configuration needed, Valence Aware.
- textStat

Cleaning Tweet Text

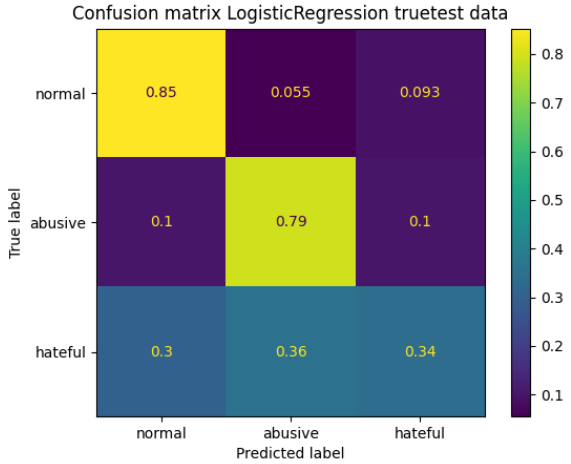
- Strip URL, replace with token
- Strip mentions, replace with token
- Strip hashtags
- lowercase
- Normalize whitespace
- Take out emojis
- Stem results

Feature Selection

- TFIDF of vocab -> TFIDF of characters and TFIDF of POS
- Emojis
- Number URLS and mentioning
- Flesch-Kincaid readability score
- VADER Sentiment Score – default configuration of the package, met needs, and
- Multi-class or Binary?

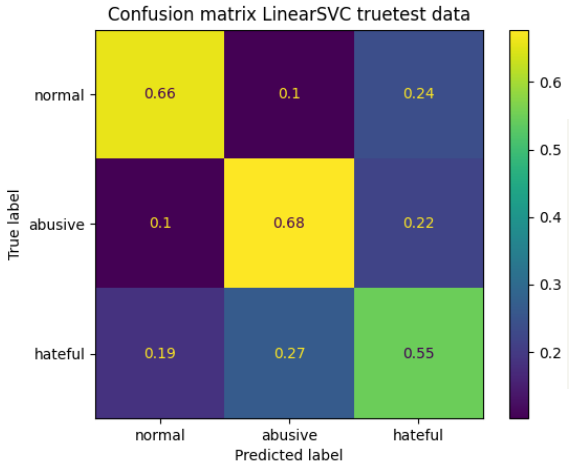
Which model and method?

- It depends.
- What if building a system to notify admins of a Zoom account about possible influx of hate group? You may care more about about precision, or else the admins may get many false positives, and turn the system off.
- What if this system was deployed to filter out hateful tweets from a timeline? You may not care if some normal tweets don't show up, as long as you don't have to see hateful content. (Recall)



LR

label	precision	recall	f1-score	support
normal	1.00	0.85	0.92	33113
abusive	0.16	0.79	0.27	453
hateful	0.02	0.34	0.04	188
accuracy			0.85	33754
macro avg	0.39	0.66	0.41	33754
weighted avg	0.98	0.85	0.91	33754



SVC label

	precision	recall	f1-score	support
normal	1.00	0.66	0.79	33113
abusive	0.08	0.68	0.15	453
hateful	0.01	0.55	0.02	188
accuracy			0.66	33754
macro avg	0.36	0.63	0.32	33754
weighted avg	0.98	0.66	0.78	33754

Best natural distribution multiclass performers

Normalized by row (True values)

Test data was (fairly) representative of the class imbalance in the real world.

Binary

label	precision	recall	f1-score	support
nice	1.00	0.77	0.87	33113
unpleasant	0.07	0.88	0.13	641
accuracy			0.77	33754
macro avg	0.53	0.82	0.50	33754
weighted avg	0.98	0.77	0.86	33754

Best 2 class natural distribution performer

Normalized by row (True values)

Test data was (fairly) representative of the class imbalance in the real world.

Model results

- Tend to predict that tweets are more hateful than they actually are. This is consistent with what one of the other papers found.
- Caveats:
  - Bias in data collection, modeling, etc
  - This system may not identify hate speech that uses words that are not in the train data.
    - May not be effective on different writing styles as well.
    - The performance of the system decreases as the % of unsavory tweets decreases (towards reality)

Further work

- Neural approaches could be more effective… with the tradeoff of time and energy.
- What combination (and weights) of features are the most effective?
- Changes in the types of hate speech as it relates to time.
- Translate to PySpark
- More advanced analysis
- Being able to react in real time

Special Thanks to:

- Zeerak Waseem – University of Sheffield's NLP Group
- My friend who was targeted in the KSUnite sessions. He let me interview him and gain insight into what it's like to experience this.