sabinabains / dsc-phase-4-project Public

0 stars 0 forks

Star Unwatch

Code Issues Pull requests Actions Projects Wiki Security

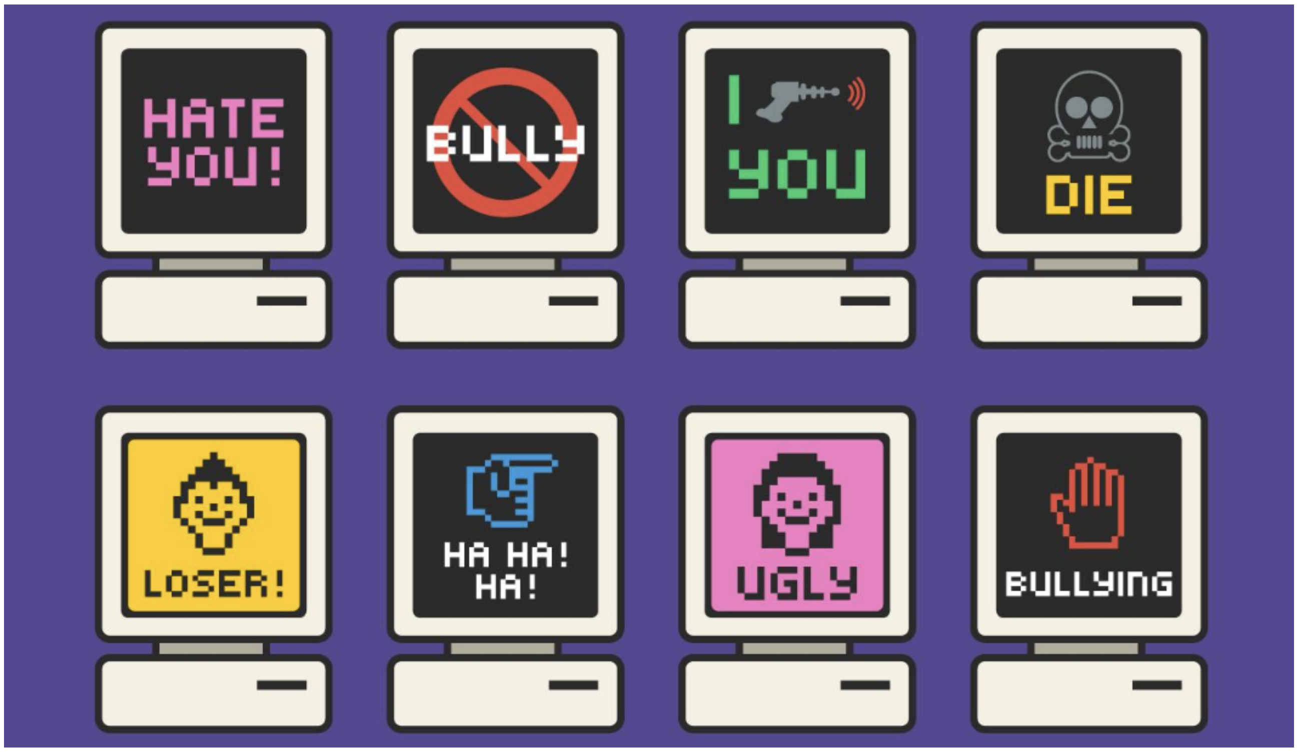
main

sabinabains Delete older_files directory ... 7 days ago 19

View code

README.md

Overview:



BeKind Org. would like help protecting users from hate speech.

Business Understanding

Over the past couple of months, Twitter has been in the headlines for the potential acquistion of their company from Elon Musk. This has upset many people, as Musk plans to reduce the level of moderation Twitter currently has on potentially harmful tweets.

This change will likely lead to a massive increase in hate speech and misinformation, and will negatively affect users. BeKind Org. would like our help to create the model behind a browser extension that can flag tweets that are considered cyberbullying. This way, users have the ability to hide tweets from their feed.

Data Understanding and Processing

We will take a supervised learning approach by using 47,000 tweets that have been manually flagged by cyberbullying type. This is a balanced dataset from Kaggle:.

The labels are as follows:

- Age
- Ethnicity
- Gender
- Religion
- Other type of cyberbullying
- Not cyberbullying

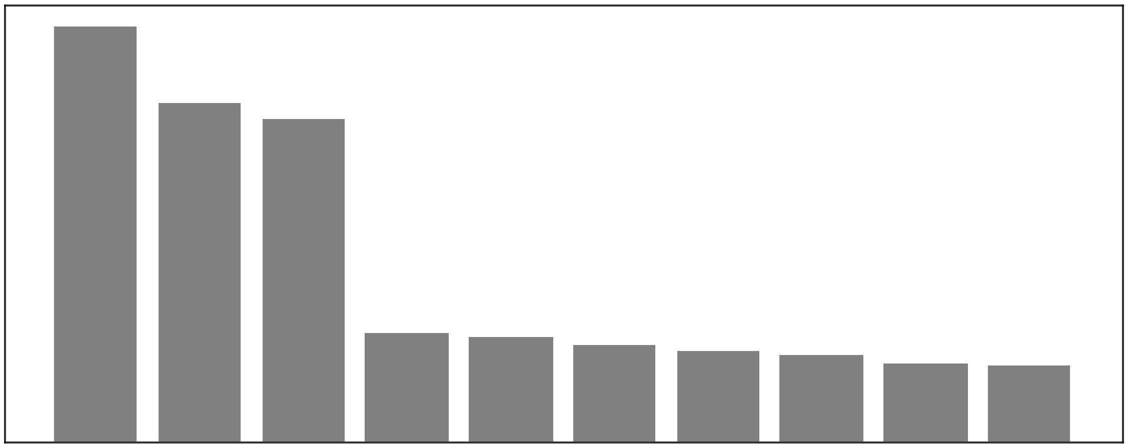
Each tweet has a maximum of 140 characters. Because this problem requires Natural Language Processing, One-Hot-Encoding must be performed to rearrange the dataset, where each column is a word, and each row has a binary output of 1 if the word is present in the tweet, and 0 if it is not.

Prior to One Hot Encoding, words are all standardized into lowercase only, and are stemmed. This unifies words that are in different tenses and therefore spelled differently, but have the same meaning. For example implementing our Stemmer to a dataset with the following words: ['CONNECT','CONNECTIONS','CONNECTED','CONNECTING','CONNECTION'] would return them all as the same word; CONNECT.

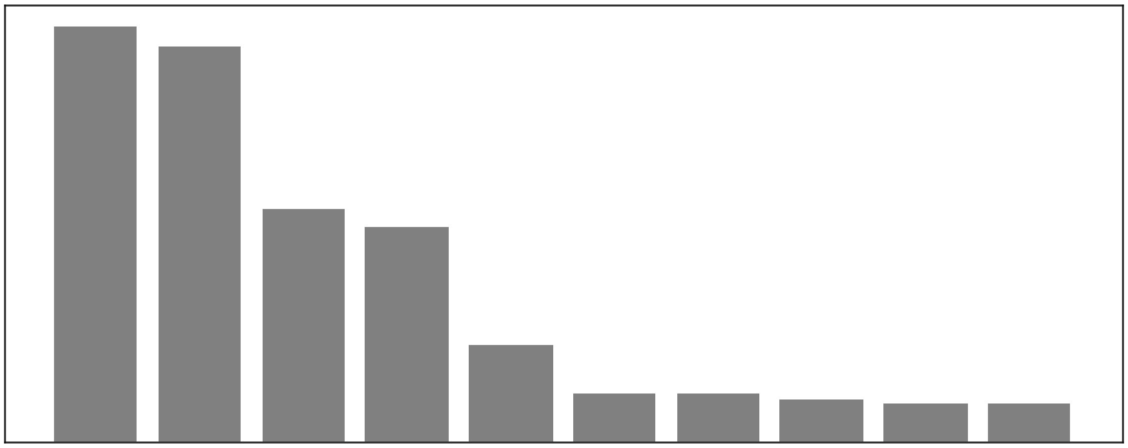
Stopwords (words such as "a", "is", "as", "the", etc.) have also been removed, as these words will have a high count even though they do not provide context to the sentence.

Once data cleaning is completed, our top tokens (or words) per label are as follows:

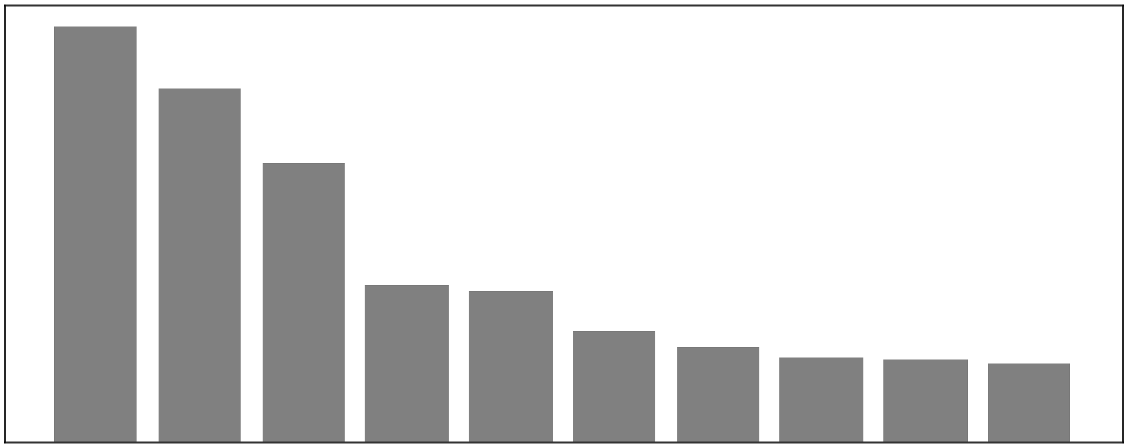
Top Tokens by Gender:



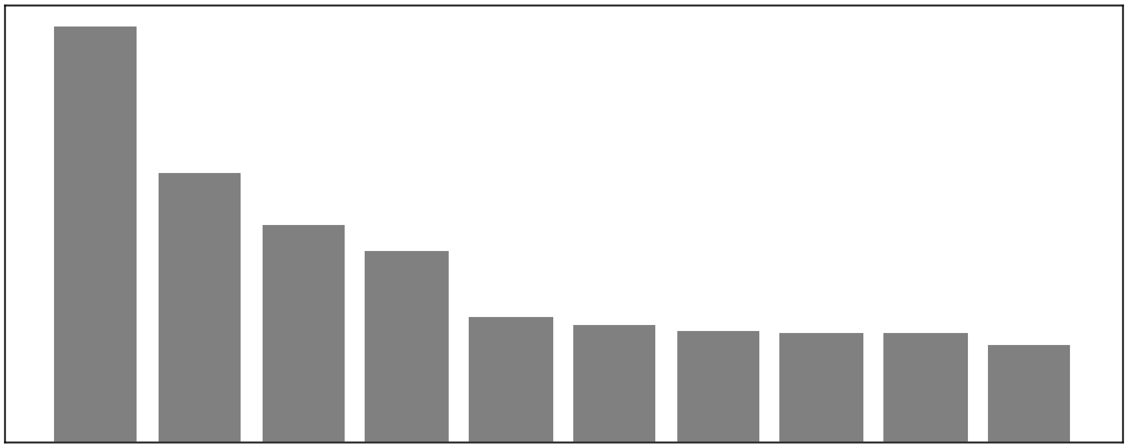
Top Tokens by Age:



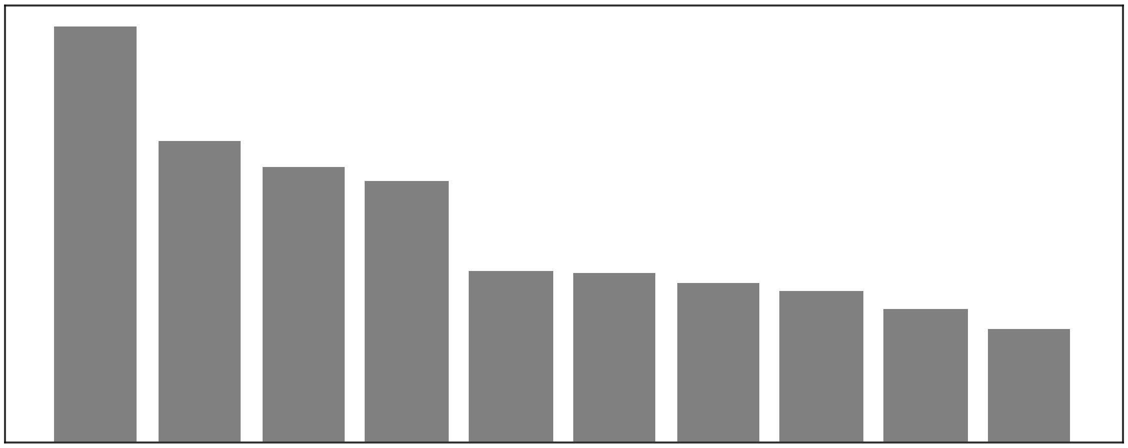
Top Tokens by Ethnicity:



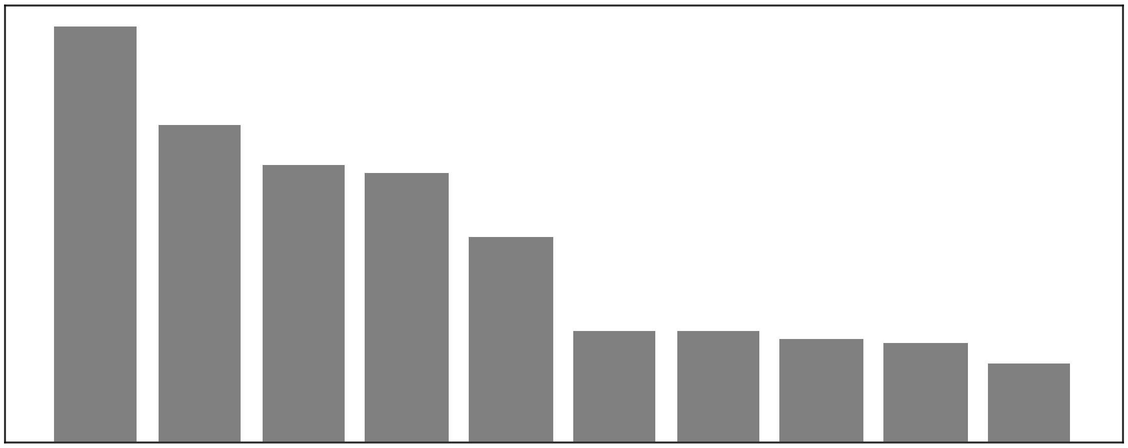
Top Tokens by Religion:



Top Tokens by Other Cyberbullying:



Top Tokens by No Cyberbullying:



An initial look tells us there are some heavy differences in the most common words used in different types of cyberbullying, however there is some overlap in words in the "other" and "none" cyberbullying categories.

Modeling

Multinomial Naive Bayes

Three iterations of Multinomial Naive Bayes were run. Our baseline model was run after only tokenizing the data. This generated the following accuracies.

```
* Training Accuracy: 56.1%
* Testing Accuracy: 55.9%
```

The second model was run after removing stopwords. This generated the following accuracies:

```
* Training Accuracy: 64.1%
* Testing Accuracy: 66.1%
```

The third model removed stopwords and stemmed words, generating the following results:

```
* Training Accuracy: 64.9%
* Testing Accuracy: 66.5%
```

The fourth model added bigrams and trigrams to the model, and also increased the features to 80. This generated a much higher accuracy:

```
* Training Accuracy: 69.3%
* Testing Accuracy: 69.7%
```

Lastly, by feature engineering columns were added based on whether a tweet was a reply or not, and had an associated link or not:

```
* Training Accuracy: 72.5%
* Testing Accuracy: 72.8%
```

While the model greatly improved and showed no signs of overfitting, trying a different model may improve the results further.

Recurrent Neural Networks

For RNNs, the data had to be reprocessed by turning tweets into sequence data.

Our first RNN model had a high accuracy of 94% on it's training data, but a significantly lower validation accuracy of 82%. this indicates we are indeed overfitting.

```
* Train Accuracy = 94.5%
* Test Accuracy = 82.8%
```

Our second RNN model closed the gap between training and testing accuracies, by adding regularization through ridge regression, but is still overfitting.

```
* Train Accuracy = 91.2%
* Test Accuracy = 80.5%
```

Our third RNN model does not overfit onto the training data, and has a testing accuracy of 80% because dropout layers were added.

```
* Train Accuracy = 87.6%
* Test Accuracy = 82.2%
```

Once we increased the coefficient our fourth iteration did not overfit as much, however our previous model with dropout laters still outperforms this model.

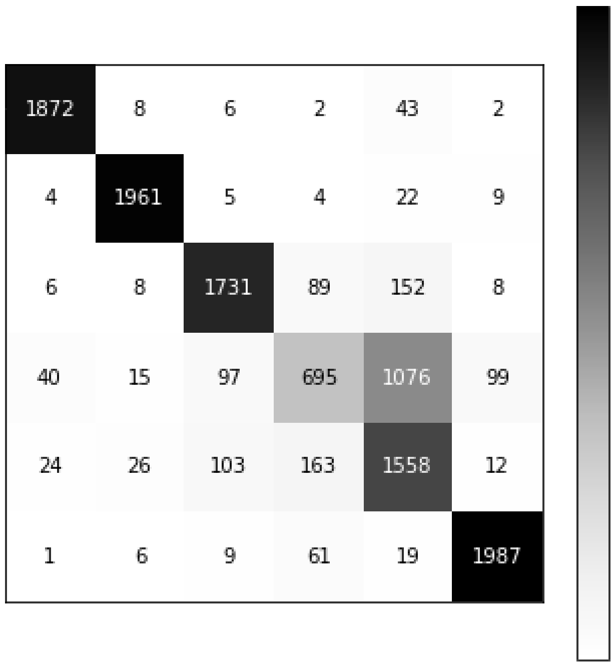
```
* Train Accuracy = 84.0%
* Test Accuracy = 76.4%
```

Evaluation

The final model chosen is our third iteration; RNN Model with 3 hidden layers and 3 dropout layers to prevent overfitting. Testing Accuracy was 82.2% and Training Accuracy was 87.6%. This model had the highest testing accuracy, and minimal overfitting.

Since it is a recurrent model, the order of words are incorporated into the model. It also uses Long Term Short Memory cells which teaches the model which words are important and which ones we can forget. This will help our algorithm as we incorporate more and more data.

However, there are ways to improve this model. as seen in the confusion matrix, there is incorrect labeling between non-cyberbullying and other. Non-cyberbullying was only predicted 35% of the time. This would lead to many flagged tweets, therefore drasticly affecting the purpose of browsing twitter.



1872	8	6	2	43	2
4	1961	5	4	22	9
6	8	1731	89	152	8
40	15	97	695	1076	99
24	26	103	163	1558	12
1	6	9	61	19	1987

Conclusion and Next Steps

In the end, our Recurrent Neural Network Model outperformed our Multinomial Naive Bayes Model in terms of accuracy on both Training and Testing data. Our first Sequential model had an accuracy of 95% on it's training data, but was severely overfitting as we saw with our low Testing data score. With each iteration we worked towards reducing overfitting, through the dropout method and L2 regularization. In the end, we settled on our final model with a Training Accuracy of 87.6% and Testing Accuracy of 82.2%.

Next steps to consider when working on this project would be to create a pipeline to find the optimal testing accuracy, as it could be higher. I would also look into the low accuracy score between "No Cyberbullying" and "Other Cyberbullying", as our confusion matrix shows that the model could not identify differences in these tweets as well. This would involve potentially gathering more data, or looking more into the accuracy of labeling for those two categories. I would also incorporate data into this model that would have the ability to flag fake news, as these types of tweets could increase with less moderation on twitter, and are dangerous to our society.

Repository Structure

```
├── data : data used for modeling
├── images : images used in PPT and readme
├── README.md : project information and repository structure
├── dsc-phase-4-project-presentation.pptx : (Presentation for
Stakeholders)
└── dsc-phase-4-project.ipynb (jupyter notebook used for modeling)
```

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● **Jupyter Notebook** 100.0%