Abusive Language Detection in Social Media

STOR 565 Final Project

Team: MLFs
Sathvik Chatta, Lillian Hurban, Elizabeth Singletary, Grace Sosa, Rujula Yete

Introduction

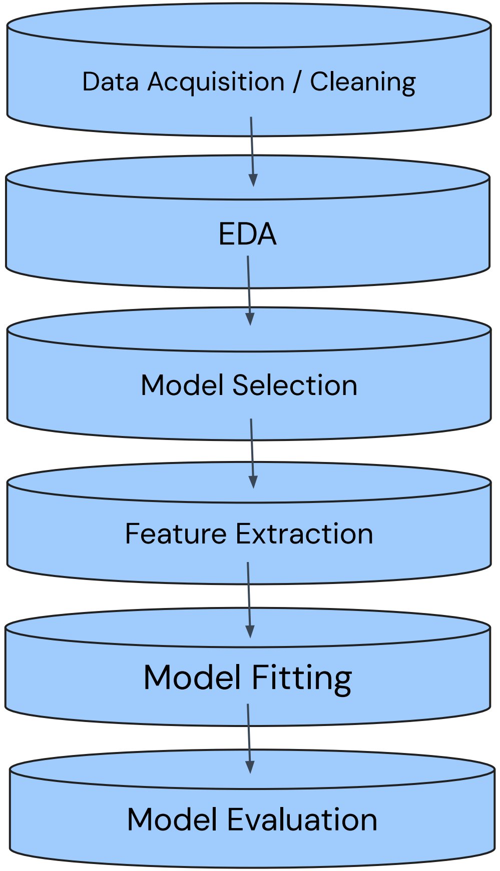
Motivations:

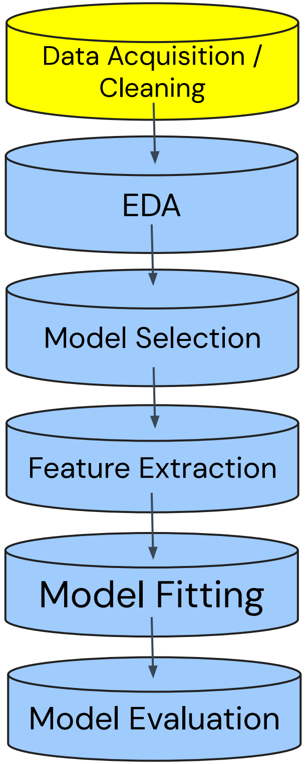
- Children are increasingly exposed to social media
- Age restrictions are not enough to prevent children from seeing harmful material

Goals:

- Develop machine learning model to classify text as abusive and non-abusive language
- Can be implemented as a browser extension to filter online content

Project Workflow





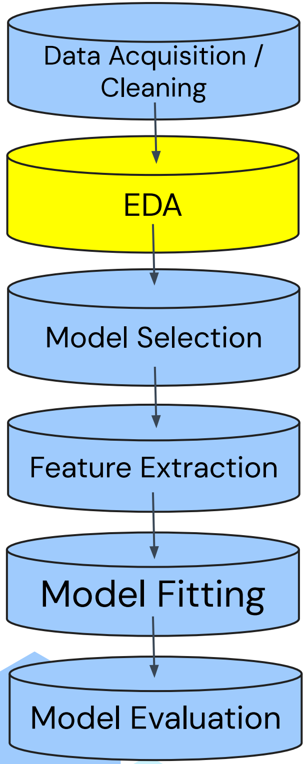
Original Dataset

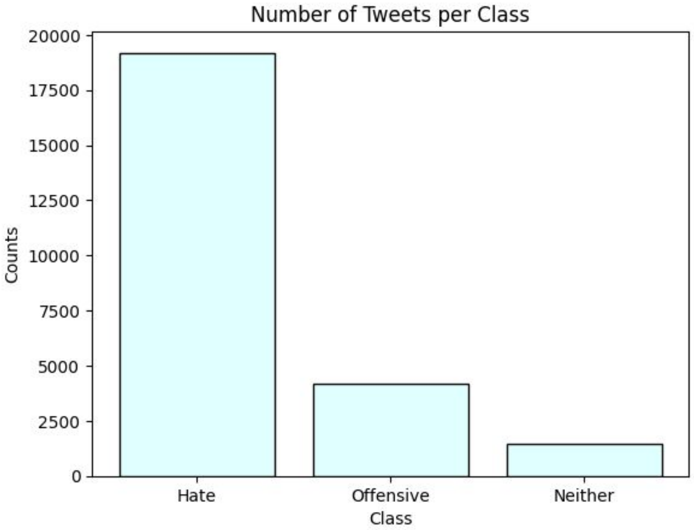
- Compiled by Cornell University researchers based on Twitter API searches using terms from an established offensive language lexicon[1]
- 24,783 randomly sampled tweets, manually classified:
  - 0: Hate Speech (1430)
  - 1: Offensive but not Hate Speech (19190)
  - 2: Neither Offensive nor Hate Speech (4163)

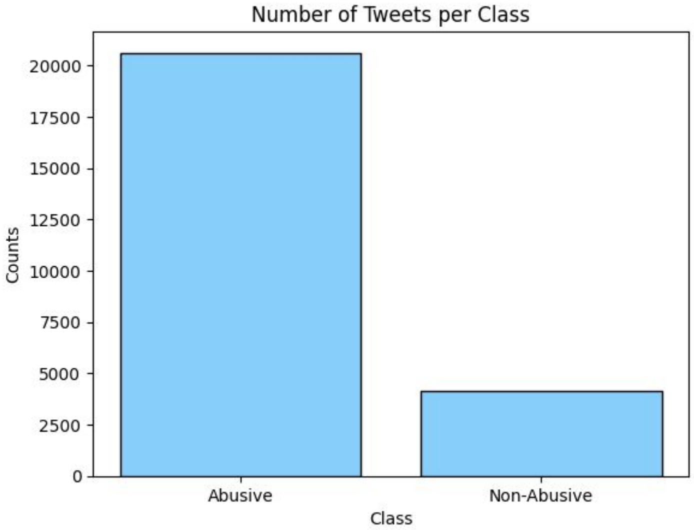
Old Label	New Label
2: Neither Offensive nor Hate Speech (4163)	**0: Non-Abusive Speech (4163)**
0: Hate Speech (1430) 1: Offensive but not Hate Speech (19190)	**1: Abusive Speech (20620)**

[1] https://huggingface.co/datasets/tdavidson/hate_speech_offensive

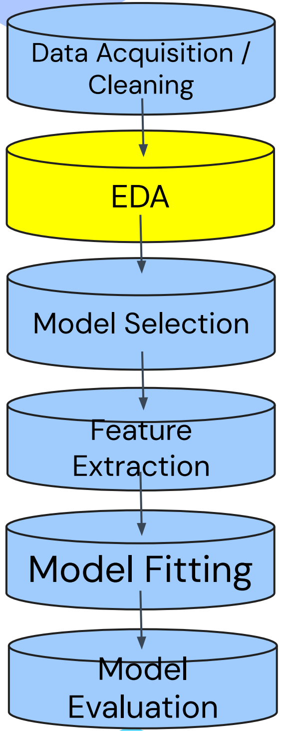
Imbalanced data

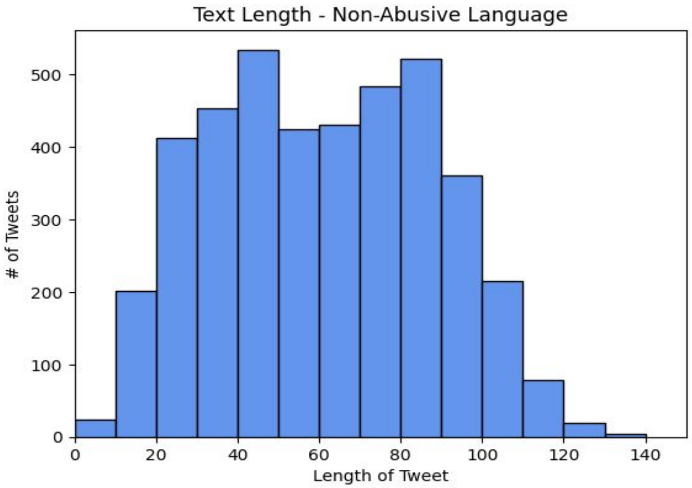


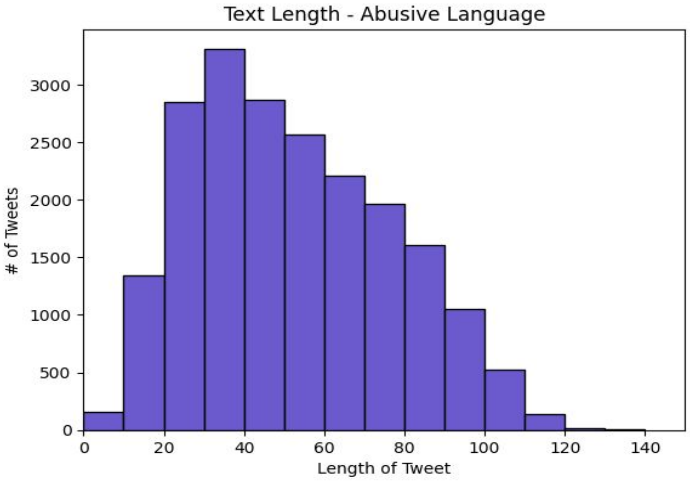




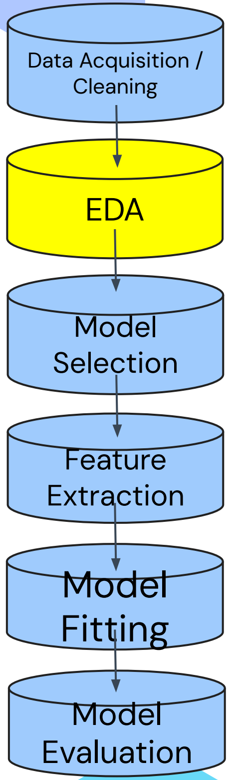
Average Tweet Length

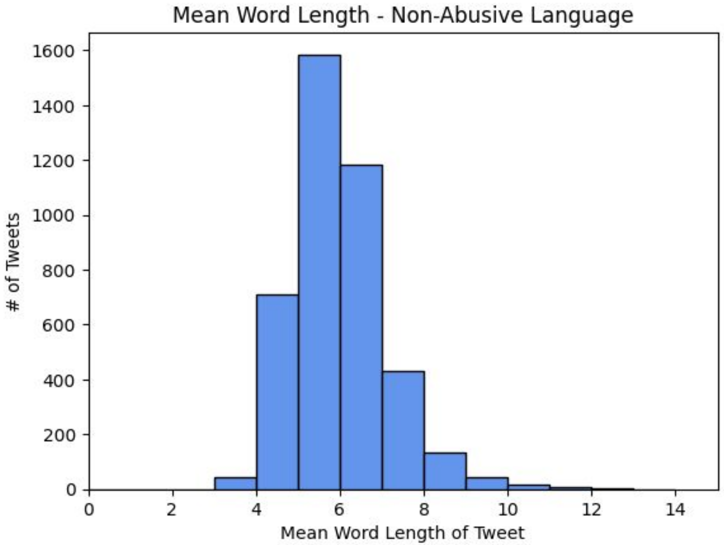


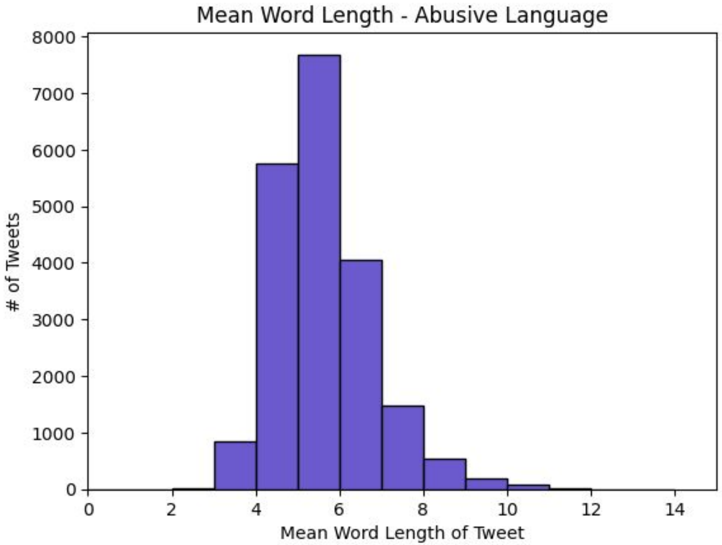




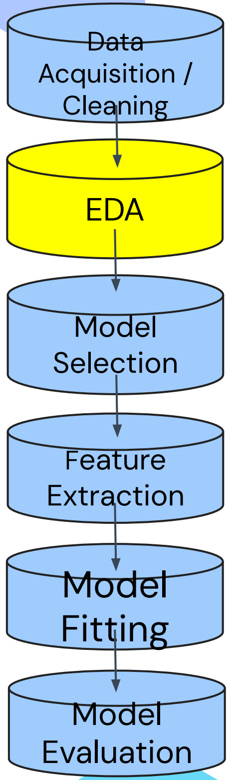
Average Word Length

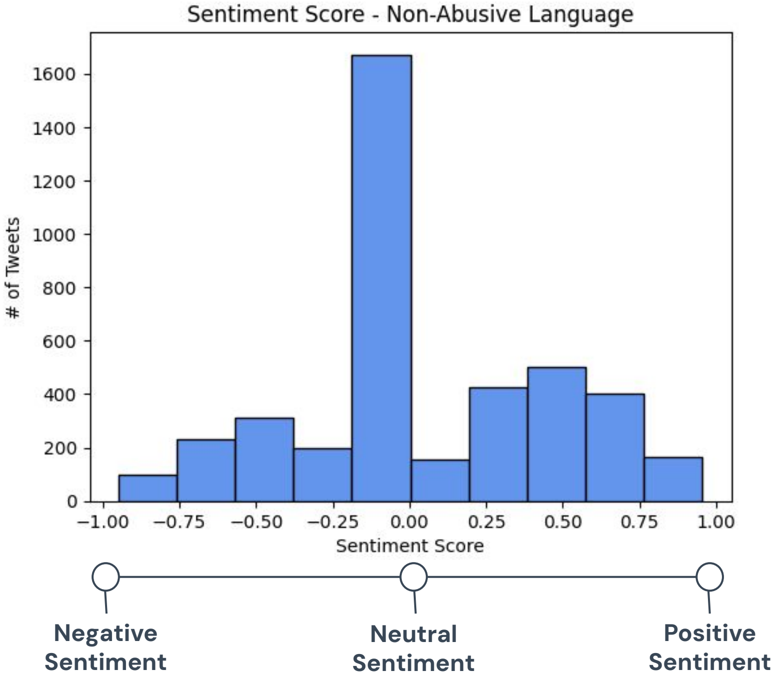


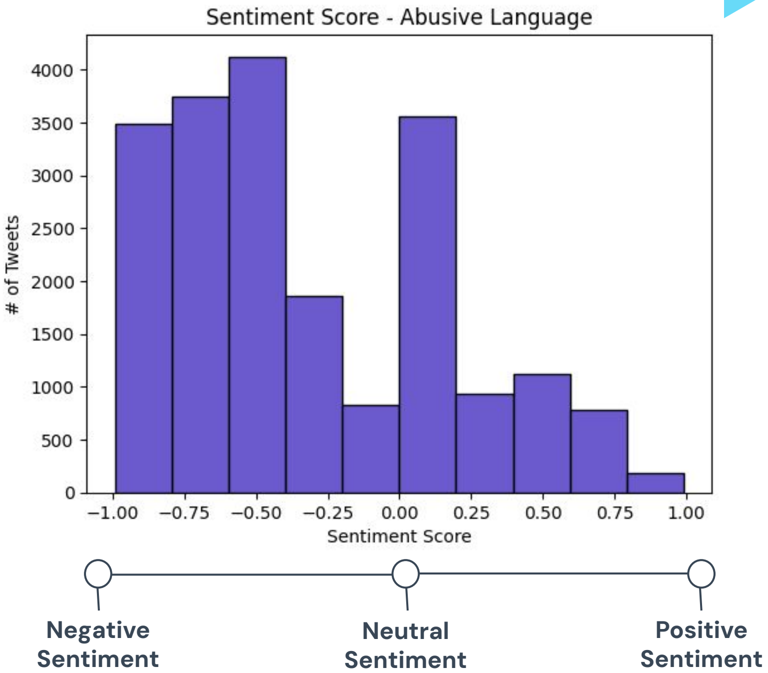




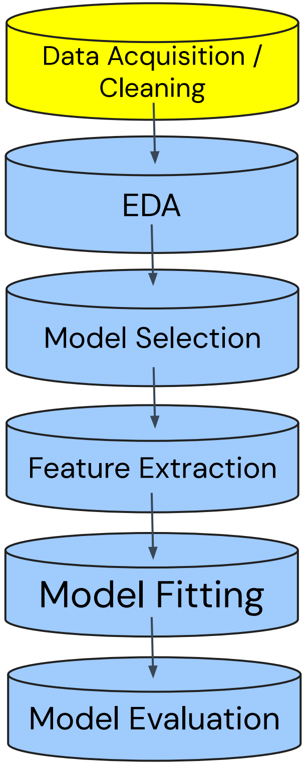
Sentiment Score







Additional Datasets

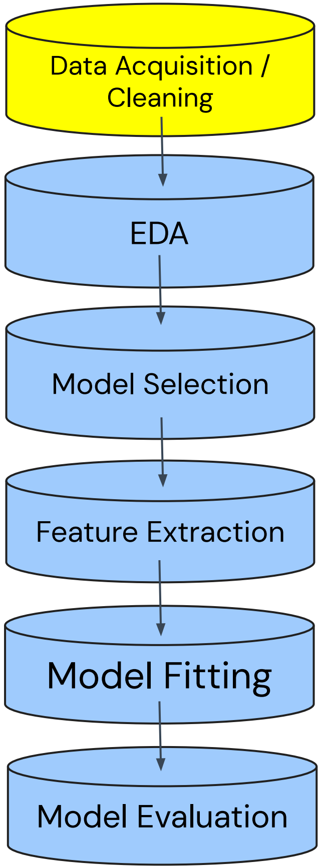


- To combat these issues, we created a new dataset by combining our original dataset with two other datasets

YouTube Comments Dataset[2]	Reddit, Twitter, YouTube Dataset[3]

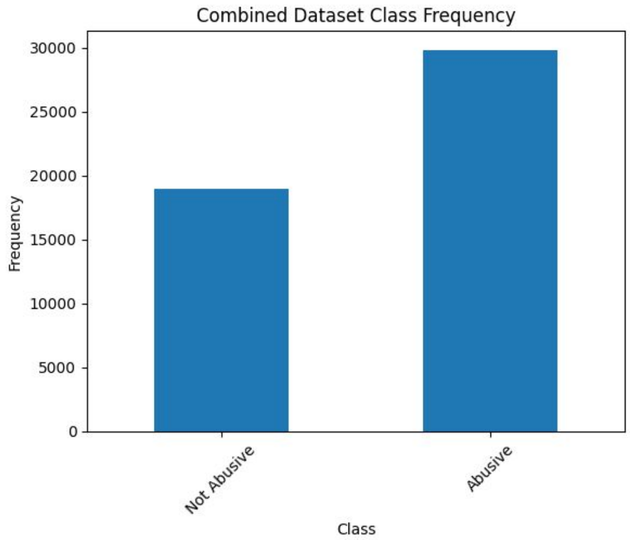
• From 29 manually selected YT videos[4], chosen for their large number of comments
• Random sample of over 160,000 comments
• Manually classified:
  ◦ 0: Not Abusive(2009)
  ◦ 1: Abusive (1970)
	• Continuous abusive-speech score, with a score of > 0.5 being classified as abusive
	• From r/all, random Twitter API, and the trending tab from the top 300 most populated US cities
	• Random sample of 20,000 posts/comments from 39,565 manually classified:
	  ◦ 0: Not Abusive(12755)
	  ◦ 1: Abusive (7245)

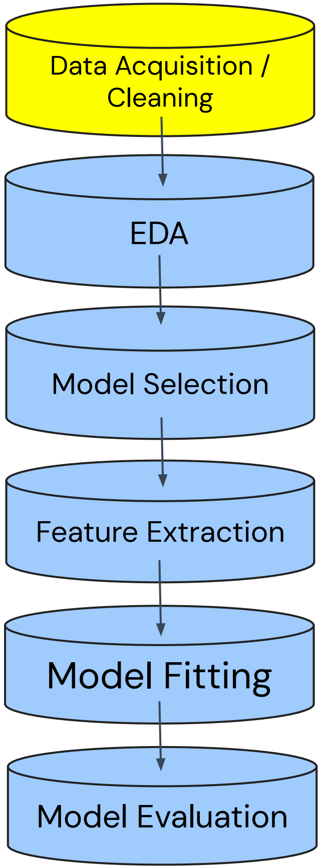
[2] https://github.com/Noman712/contextual-abusive-language-detection/tree/main/dataset/labelled
[3] https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech/viewer
[4] https://docs.google.com/spreadsheets/d/1wYtrNMxdv6OgkEXVk3lital6JE6dAhC3BsNZHMb1KmA/edit#gid=1358874100



Final Ensemble Dataset

- 48,762 posts from various social media platforms
  - 0: Non-Abusive (18927)
  - 1: Abusive (29835)



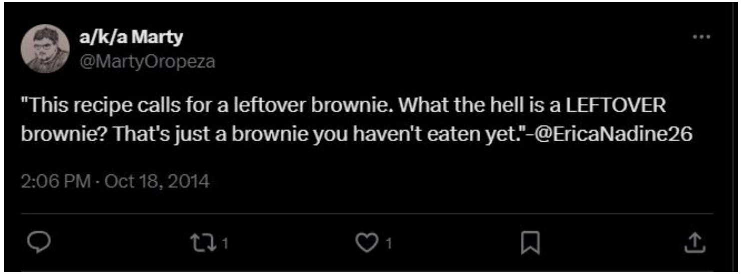


Data Cleaning

- Converted all text to lowercase
- Removed URLs/links
- Removed user handles (any contiguous string after an @ symbol, typically a username/ID)
- Removed stopwords ('the, 'is', 'and', etc..)
- Removed 'rt' (indicating a retweet)
- Removed extra spaces
- Removed special characters
- Converted contractions (isn't -> is not, etc)
- Removed any non-English words/characters
- Removed "nonsensical" or spam strings
- Tokenizing

Cleaning Example

Original tweet



"Cleaned" tweet

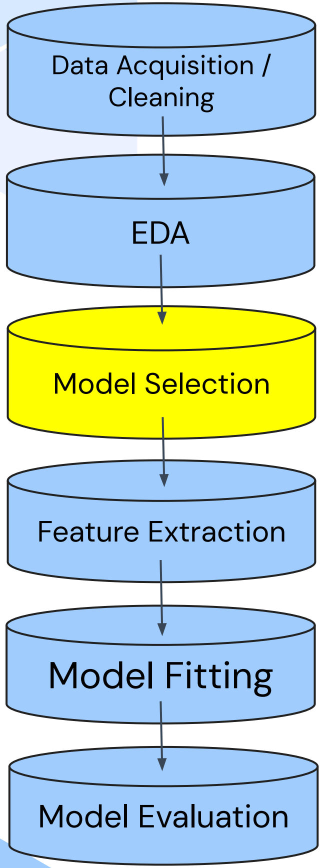
```
recipe calls leftover brownie hell leftover
brownie that's brownie eaten yet ericanadine
```

Original tweet

"Cleaned" tweet

```
worldseriesgame hunter pence annoying red sox
player shave fool take vyvanse yankees
```

Classification Models



01

TF-IDF + Linear SVM

02

TF-IDF + Naive Bayes

03

TF-IDF + KNN Clustering

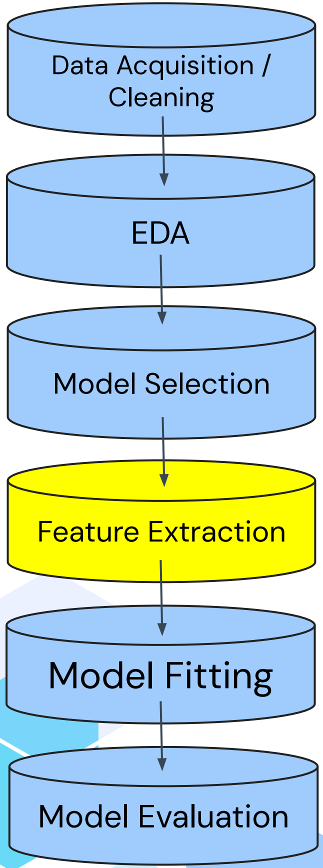
04

TF-IDF + Logistic Regression

05

Word2Vec + Neural Network (FFNN)

Word Vectorizing



TF-IDF

Numeric vectors created based on frequency of word in data, uses calculated TF-IDF scores

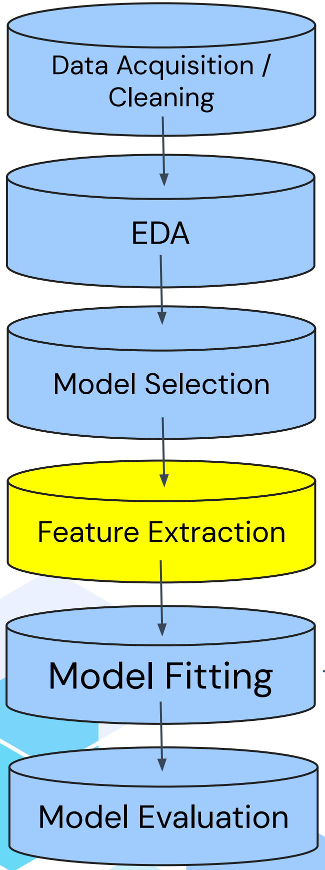
→ How we extract the most relevant terms from the posts

GloVe Vectors + Word2Vec

Loading Pre-trained GloVe Vectors, that have been saved in Word2Vec format

→ Provides pre-trained word embeddings

Parameter Tuning on the Validation Set



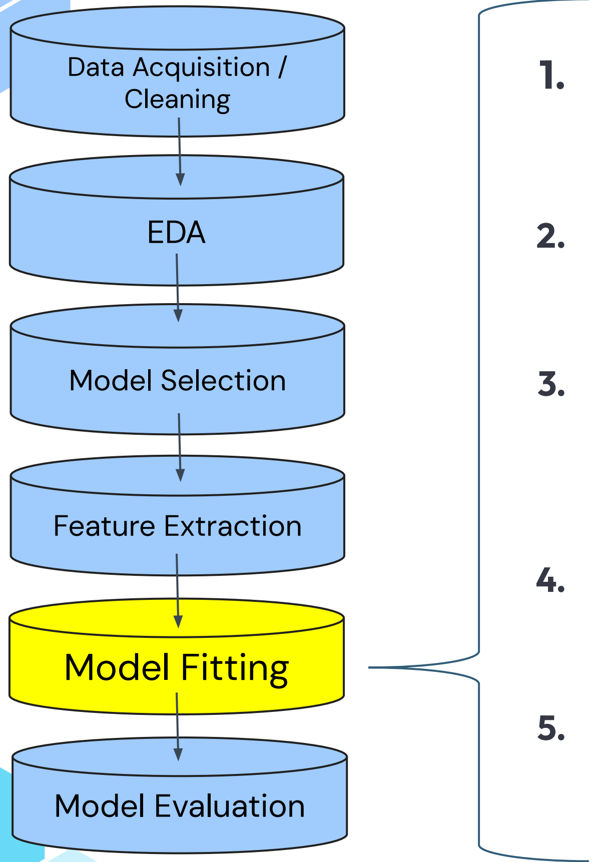
SVM: Kernel – Linear, C – 1

KNN: Neighbors – 10, Distance: Euclidean

Naive Bayes: Alpha – 0.1

Logistic Regression: C – 10S

Model Building Summary



1. **Data Preparation**
    a. Cleaning and tokenization of data
2. **Data Splitting**
    a. 70-10-20 training/validation/testing split
3. **Text Vectorization**
    a. Text from train/test data is vectorized using TF-IDF or pre-trained GloVe Vectors & Word2Vec
4. **Hyperparameter Tuning**
    a. Grid-searching on the validation set
5. **Model Training and Evaluation**
    a. Vectorized data then used to train our models and performance is compared to its respective test set

Evaluation on the Testing Set

Train: 34,132
Validation: 4,877
Test: 9,753

Chosen Metrics for Evaluation

F1-Score

Balances recall and precision to reflect completeness and accuracy of abusive speech detection

Recall

A measure of a model's ability to correctly identify all actual positive instances (E.g. abusive text) from a dataset.

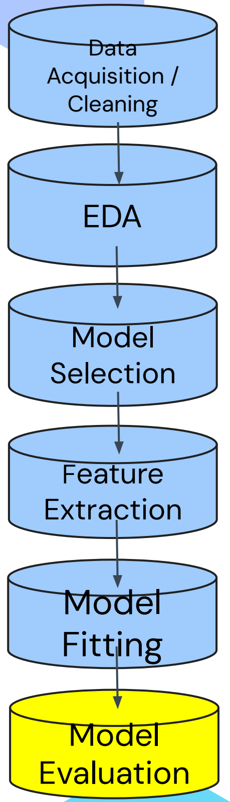
AUC

To measure overall classifier performance

Accuracy

Measure of the proportion of correctly classified instances out of all instances.

Test Evaluation Results



SVM

	precision	recall	f1-score	support
0	0.8393	0.8664	0.8526	3780.0000
1	0.9137	0.8950	0.9043	5973.0000
accuracy	0.8839	0.8839	0.8839	0.8839

Logistic Regression

	precision	recall	f1-score	support
0	0.8577	0.8468	0.8522	3780.0000
1	0.9038	0.9111	0.9075	5973.0000
accuracy	0.8862	0.8862	0.8862	0.8862

KNN

	precision	recall	f1-score	support
0	0.7426	0.7312	0.7369	3780.0000
1	0.8315	0.8396	0.8356	5973.0000
accuracy	0.7976	0.7976	0.7976	0.7976

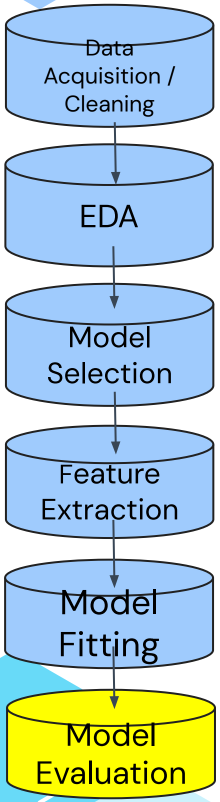
Naive Bayes

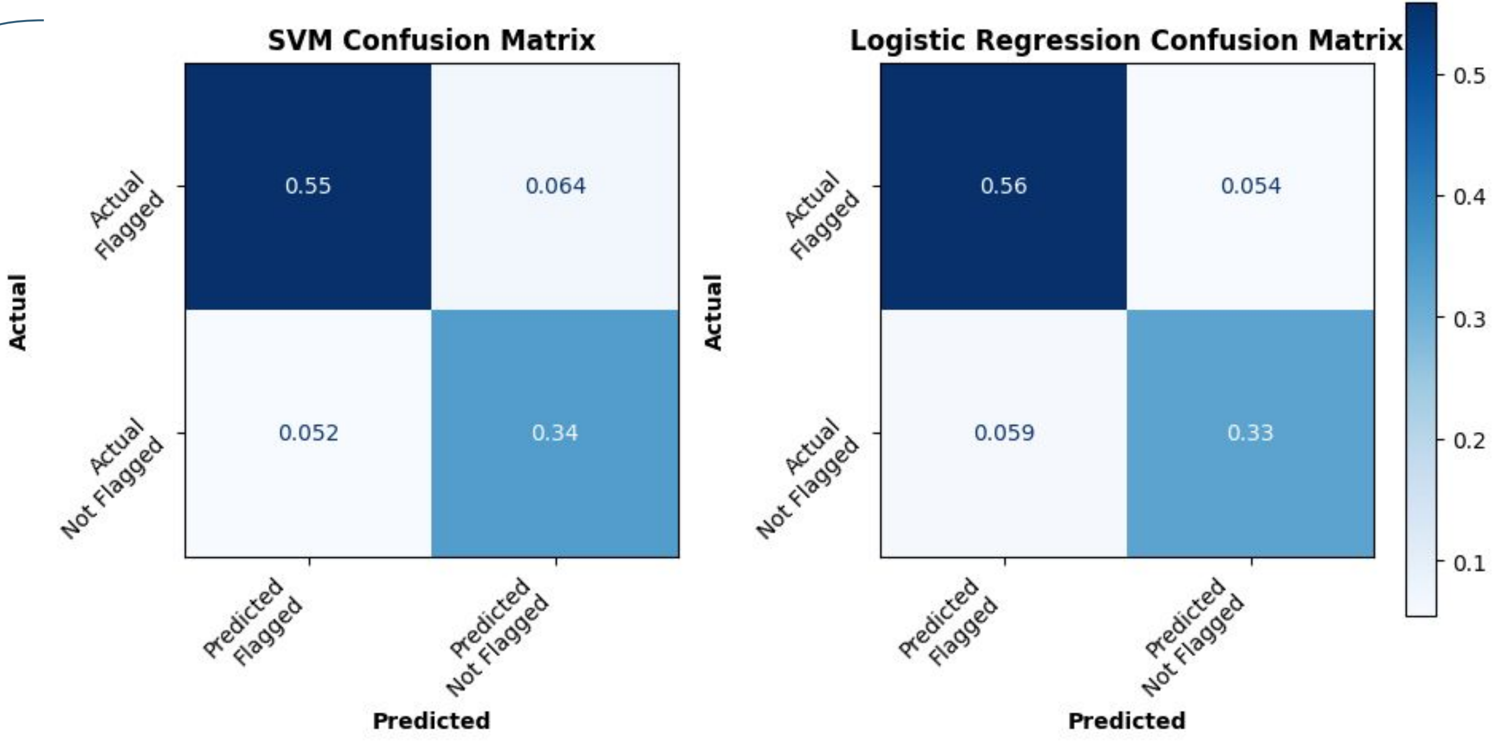
	precision	recall	f1-score	support
0	0.8269	0.7317	0.7764	3780.0000
1	0.8418	0.9031	0.8713	5973.0000
accuracy	0.8367	0.8367	0.8367	0.8367

Word2Vec Neural Network

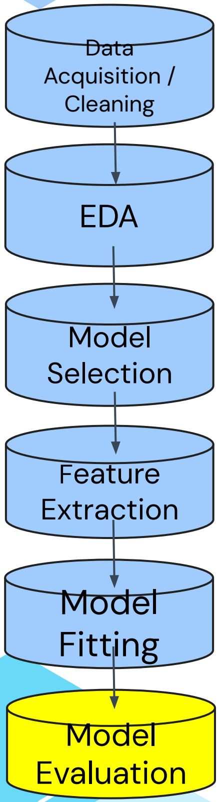
	precision	recall	f1-score	support
0	0.8518	0.2966	0.4400	3780.0000
1	0.6848	0.9674	0.8019	5973.0000
accuracy	0.7074	0.7074	0.7074	0.7074

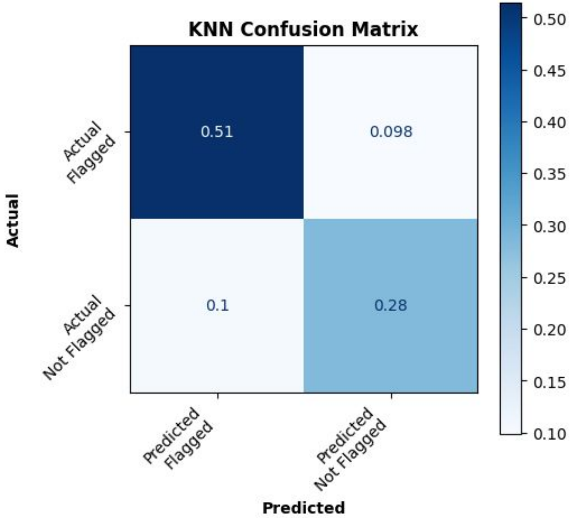
Confusion Matrices

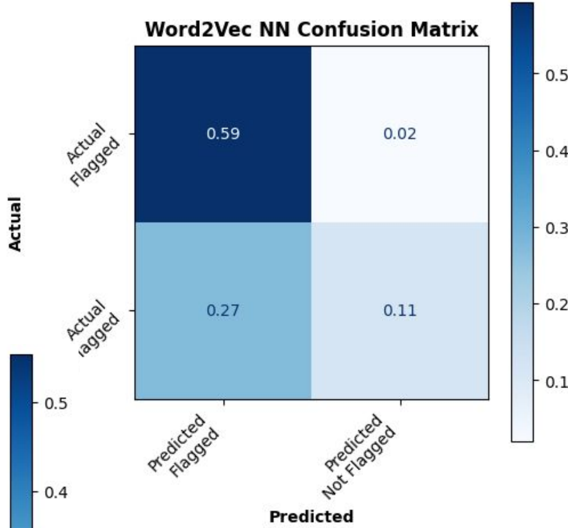


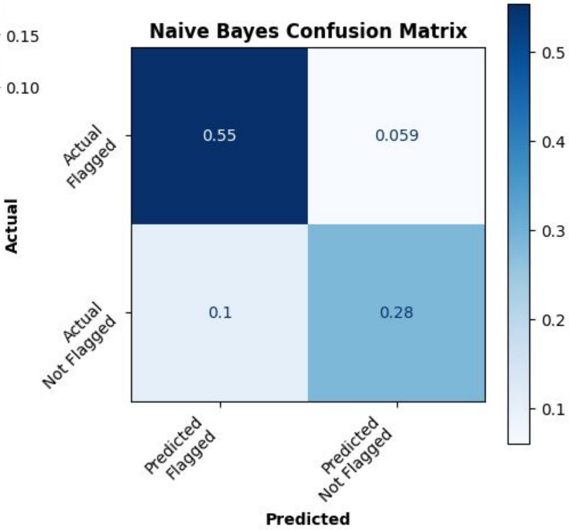


Confusion Matrices Continued

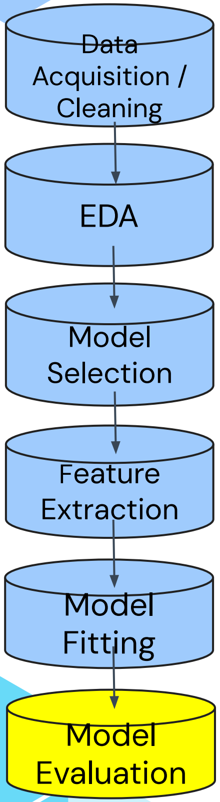


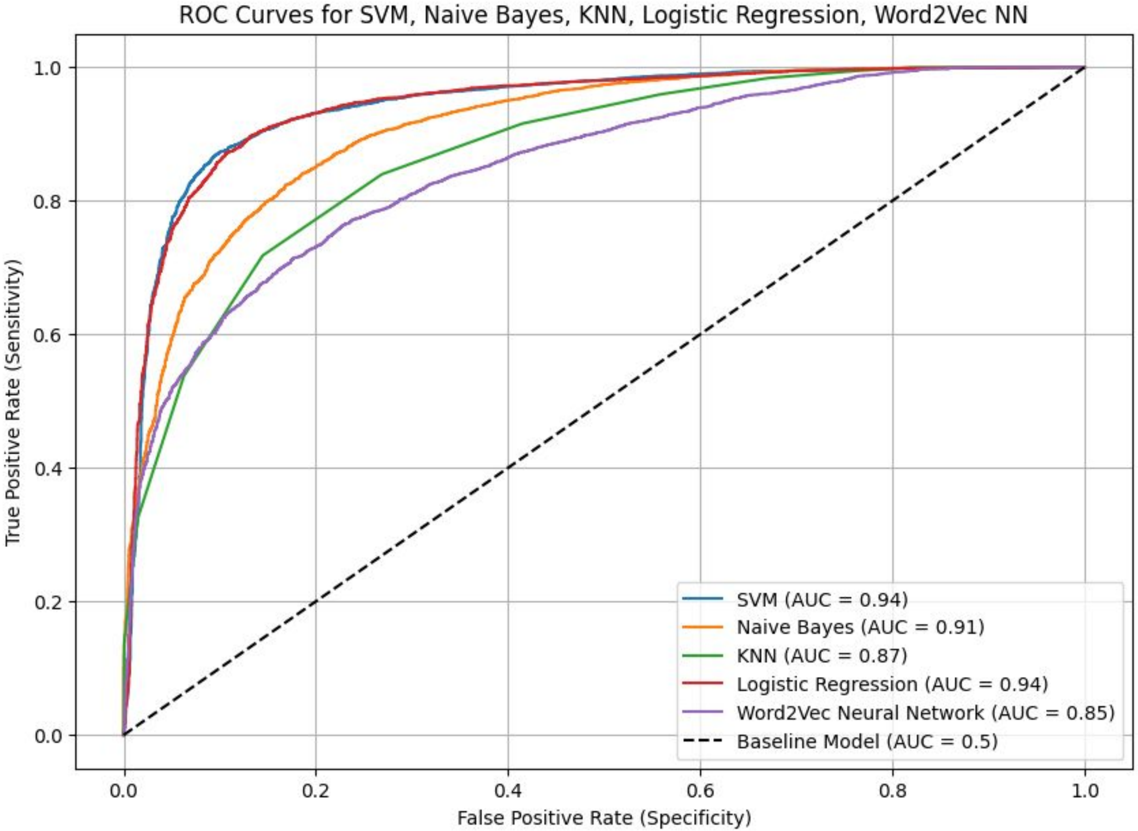




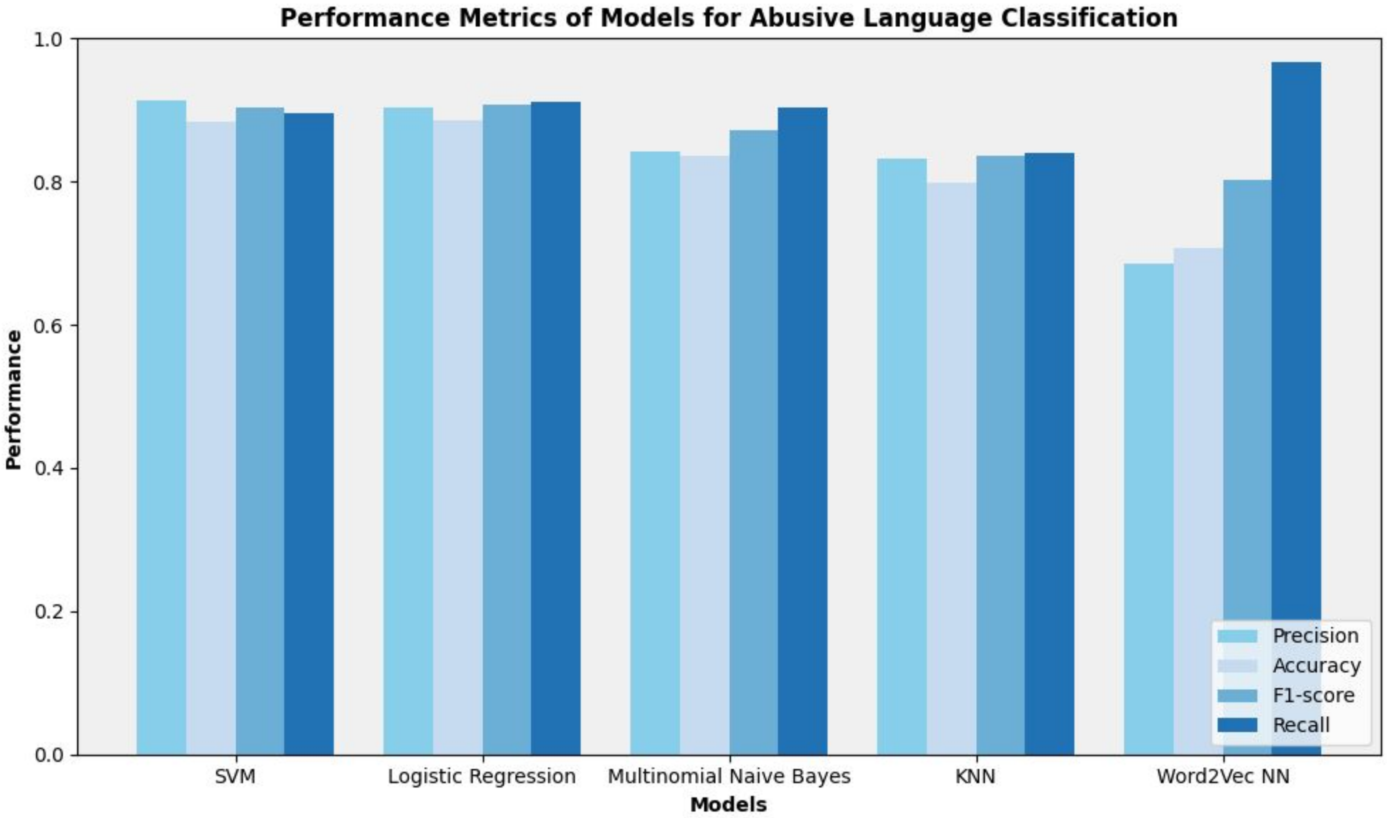


ROC Curves


Data Acquisition / Cleaning
EDA
Model Selection
Feature Extraction
Model Fitting
Model Evaluation


ROC Curves for SVM, Naive Bayes, KNN, Logistic Regression, Word2Vec NN
True Positive Rate (Sensitivity)
False Positive Rate (Specificity)
0.0
0.2
0.4
0.6
0.8
1.0
SVM (AUC = 0.94)
Naive Bayes (AUC = 0.91)
KNN (AUC = 0.87)
Logistic Regression (AUC = 0.94)
Word2Vec Neural Network (AUC = 0.85)
Baseline Model (AUC = 0.5)

Results


Performance Metrics of Models for Abusive Language Classification
Performance
1.0
0.8
0.6
0.4
0.2
0.0
SVM
Logistic Regression
Multinomial Naive Bayes
KNN
Word2Vec NN
Models
Precision
Accuracy
F1-score
Recall

Model for OOS Data: SVM

Support Vector Machine

- Utilized GridSearch (cv=5) for optimal parameters
  - C: 1
  - Kernel: 'Linear'

Parameter/Improving our SVM Model

1. Explored **Linear, Polynomial, RBF Kernels, degree, gamma**
   a. Linear SVM still achieved best performance
2. Explored **SMOTE resampling** to account for unbalanced data
   a. Improved f1-score & recall slightly, but precision decreased, was it significant?
      i. **T-test**: p-value = 0.42, which is <u>not</u> statistically significant at alpha = .05

SVM			
	precision	recall	f1-score
0	0.8393	0.8664	0.8526
1	0.9137	0.8950	0.9043
accuracy	0.8839	0.8839	0.8839

Model for OOS Data- Part 1: Word2Vec

What is Word2Vec?

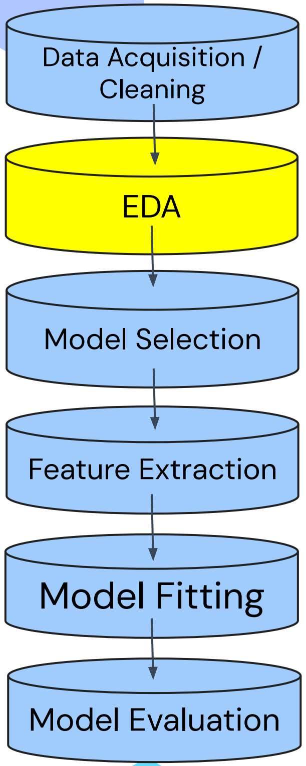
- **Word2Vec:** Converts words into a numeric space where semantic similarities are reflected by spatial proximity (generates **Word Embeddings**)
- **GloVe Model:** Pre-trained LLM trained on Twitter data (glove.twitter.27B.50d), providing an initial set of weights shaped by social media context.

Data Preparation:

- **Conversion:** GloVe vectors are transformed to Word2Vec format for compatibility.
- **Vectorization:** Tweets are converted into vectors by averaging Word2Vec vectors of constituent words.

These Word Embeddings generated from pre-trained Glove Vectors are then used to train the neural network (FFNN).

Word2Vec Word Embeddings FFNN Motivation



Model for OOS Data- Part 2: Neural Network (FFNN)

Fully Connected Feedforward Neural Network (FFNN) (aka Multilayer Perceptron)

- **Model Architecture:**
    - **Layers:** Input (vector size of 50) → Dense (128 units, ReLU) → Dropout (0.5) → Dense (64 units, ReLU) → Dropout (0.5) → Output (Sigmoid)
- **Optimizer:** Adam with learning rate adjustments.
- **Class Weights:** Applied to address class imbalance, enhancing focus on minority classes.
- **Training:**
    - Utilizes **early stopping** and a **learning rate scheduler**.
    - Trains on vectors derived from tweets, testing against a reserved set

Word2Vec Neural Network			
	precision	recall	f1-score
0	0.8518	0.2966	0.4400
1	0.6848	0.9674	0.8019
accuracy	0.7074	0.7074	0.7074

*Although doesn't achieve the best overall performance, high potential in NLP/text classification tasks for high recall on 1 and higher f-1 score by modifying class_weights and fine tuning

Conclusion

Best Performance:

1. **TF-IDF Vectorizer + SVM**

- Accuracy: 0.8839, Precision: 0.9137
  - **Recall:** 0.895
  - **F-1 score:** 0.9043
  - **AUC:** 0.94

2. **Second Choice:**

- TF-IDF Vectorizer + Logistic Regression

3. **Potential for Best Performance on Specific Metrics:**

- Word2Vec + FFNN

Potential Future Improvements

For a more robust model, train our model on data from more social media platforms like Twitter, Youtube, Instagram, Reddit, etc. to be able to flag content across several platforms

Further Model Exploration

- Fine tune the FFNN/Multilayer Perceptron
- Try utilizing Dimensionality Reduction (PCA) + T-SNE + Cosine Similarity as a different semantic similarity measure and calculation

Further Implementation

- Build, implement, and test with a Chrome extension or online user interface

Citations

Abusive language dataset - Twitter

https://huggingface.co/datasets/tdavidson/hate_speech_offensive

Associated paper

https://arxiv.org/abs/1703.04009

Abusive language dataset - YouTube comments

https://github.com/Noman712/contextual-abusive-language-detection/tree/main/dataset/labelled

Associated paper

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8507480/

Abusive language dataset (testing) - Twitter

https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech/viewer

Associated paper

https://arxiv.org/abs/2009.10277

Dealing with Imbalanced Data

https://machinelearningmastery.com/what-is-imbalanced-classification/