**UNIVERSITY OF SOUTH FLORIDA**

DATA SCIENCE PROGRAMMING

**Project Report On**

**CYBER BULLYING TEXT DETECTION**

**TEAM MEMBERS**

Amena Shahid  
Anu Yadav Kambalapally  
Haritha Thiagu Kumar   
Priyanka Thakur

Youtube Link for Project Presentation: <https://youtu.be/7VCk_8qsgMA>

Table of Contents

[Topic Introduction & Motivation 2](#_Toc55911909)

[Problem statement & Business Question 2](#_Toc55911910)

[Data overview 2](#_Toc55911911)

[VISUALIZATIONS 3](#_Toc55911912)

[DATA SPLIT 3](#_Toc55911913)

[WORD CLOUD FOR BULLY AND NON-BULLY TEXT 4](#_Toc55911914)

[NUMERICAL FEATURES 4](#_Toc55911915)

[Data pre processing 5](#_Toc55911916)

[DATA CLEANING 5](#_Toc55911917)

[BAG OF WORDS 5](#_Toc55911918)

[TF IDF VECTORIZER 6](#_Toc55911919)

[COUNT VECTORIZER(CV) 6](#_Toc55911920)

[N GRAMS 6](#_Toc55911921)

[NUMERICAL FEATURE CREATION 6](#_Toc55911922)

[SMOTE TO HANDLE DATA SKEWNESS 7](#_Toc55911923)

[Data Modeling 7](#_Toc55911924)

[MODELS 7](#_Toc55911925)

[HYPERPARAMETER TUNING ON BEST MODELS 8](#_Toc55911926)

[LEARNING CURVE 8](#_Toc55911927)

[Model Evaluation 9](#_Toc55911928)

[EVALUATION METRICS 9](#_Toc55911929)

[RECALL & CONFUSION MATRIX ON TEST (VALIDATION) DATASET 9](#_Toc55911930)

[Results on Unseen Data 10](#_Toc55911931)

[PREPROCESSING 10](#_Toc55911932)

[UNSEEN DATA VISUALIZATION - DATA SPLIT 10](#_Toc55911933)

[RESULTS & OBSERVATIONS 10](#_Toc55911934)

[Insights 11](#_Toc55911935)

[FUTURE SCOPE & Recommendations 11](#_Toc55911936)

# Topic Introduction & Motivation

Cyberbullying is the use of electronic communication to bully a person, typically by sending messages of an intimidating or threatening nature. It causes significant emotional and psychological distress. 95% of teens are connected to social media and internet and reported Instagram as the top medium of cyber bullying! 64% of cyber bullied people report its effects on their ability to learn and feel safe. They face social, mental and behavioral problems as a result. In addition to low self-esteem, physical symptoms like stomach aches, headaches and suicidal thoughts are also a common side effect of being bullied.

The goal of the project is to identify the social media messages as bullying and non-bullying messages by using machine learning classification models.

# Problem statement & Business Question

Social networking sites provide a fertile medium for bullies, and teens and young adults who use these sites are vulnerable to attacks. Through machine learning, we can detect language patterns used by bullies and their victims and develop rules to automatically detect cyber bullying content. In a way giving a person a chance to reevaluate the choice of comment they’re posting.

It’s not always clear whether a comment comes under the category bullying and its difficult for the machine to exactly detect the bullying comments. The objective of the project is to classify and predict the comments as cyber bullying comments and normal comments, what factors/techniques can contribute to better classification & judging the cost of experience at which can we differentiate bully comments vs normal comments.

# Data overview

**Text

Description automatically generated**The dataset is derived from Kaggle (sampled instances below). It has 6595 rows & 3 cols.   
Insult: Specifies whether a comment is an insult or not (1 or 0)  
Date : The date when comment is posted(YYYY-MM-DD) – we won’t use date in our project  
Comment : The text of the comment

Graphical user interface, text

Description automatically generated

## VISUALIZATIONS

### DATA SPLIT

Chart, pie chart

Description automatically generatedChart, bar chart

Description automatically generated

As it is evident here, we have a skewed dataset and we will need to take care of it in our preprocessing so as to avoid any “illusions” in our evaluation matrices. An Imbalanced dataset is a challenge since it provides false high accuracy when in reality it just ignores one of the classes (the minority class) entirely.

### WORD CLOUD FOR BULLY AND NON-BULLY TEXT

A picture containing text

Description automatically generatedText

Description automatically generated

We created word cloud on cleaned data for most frequent words in bully text vs normal texts in order to get a better idea of the data. It can be seen clearly that bullies use abusive language and derogatory words as a medium of communication.

### NUMERICAL FEATURES

We also engineered a few numerical features like text length, punctuation count etc to get a better idea of the dataset.

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

As per the above visualizations, a text that’s meant to bully is usually shorter in length with fewer words and punctuations. Since these feature may be able to help with text classification, we will build models with and without them to compare the performance.

# Data pre processing

## DATA CLEANING

Since there’s a number of techniques that can be used for data cleaning, we added the following techniques one by one and noticed the impact on model performance before finalizing the techniques to use

1. removing spaces
2. removing URLs
3. replacing names in order to maintain data security
4. removing everything other than alphabets
5. Graphical user interface, table

   Description automatically generatedsplitting the sentence into words – we compared two approaches (split on space vs tokenize) for this and chose the split approach since that gave best results
6. Lemmatization – helps achieving the root forms of the derived words
7. Removing stop words - Words like the, in, at, that, which, and on model training time while providing minimal benefit
8. Table

   Description automatically generateddealing with punctuations – we compared the results with and without punctuations and ended up keeping punctuations in the dataset since it helped with overall classification

## BAG OF WORDS

There are two ways of creating bag of words. We used both the techniques on our models and noticed one worked better than the other for specific models. For Multinomial Naïve Bayes and XGBoost, TF IDF performed better whereas for the rest of the models, CV came out to be a better approach.

### TF IDF VECTORIZER

Term frequency–inverse document frequency is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. Down weights the frequently occurring words since they don’t contain enough info and upweights the less occurring words. Each word of a sentence is given weight.

### COUNT VECTORIZER(CV)

Table

Description automatically generatedAssociates index to each word in a vocabulary in a sparse matrix – each row has a length of the length equal to count of all unique words in data - each word as a result is given a flag of 1/0 . Since it creates a sparse vector, we changed it to the dense vector – increases the features.

### N GRAMS

Table

Description automatically generated**N**-**grams** are contiguous sequences of **n**-items in a sentence. We experimented with binary n grams and also n grams ranging from 2 to 4 and 2 to 6. Generally, 2 to 6 ranged n grams gave the best results.

## NUMERICAL FEATURE CREATION

As explained in the visualizations part, the numerical features were created to help provide extra information to aid the classification. However, we didn’t use them in the end since they didn’t improve the performance.

Table

Description automatically generated

## SMOTE TO HANDLE DATA SKEWNESS

Imbalanced classification involves developing predictive models on classification datasets that have a severe class imbalance. The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class. Since our models aim at identifying the minority class, new examples were synthesized from the existing examples to address imbalanced dataset. Using a balanced class definitely helped increase the recall/true position rate and detection of the minority class (bully texts).

Chart, pie chart

Description automatically generatedChart, pie chart

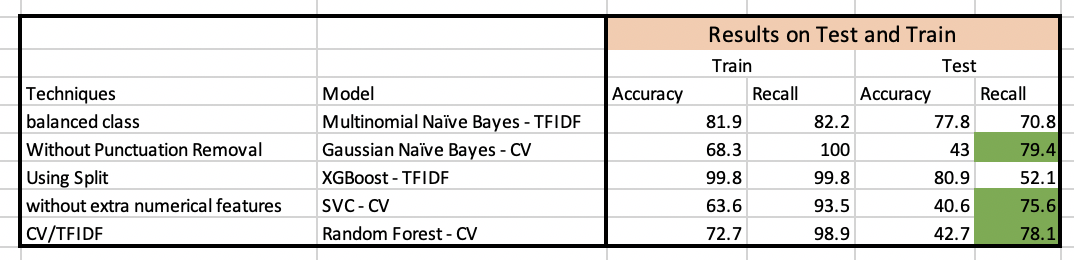
Description automatically generated

# Data Modeling

## MODELS

We opted for the following models for text classification

1. Naïve Bayes models- supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable
   1. Multi-nominal Naïve Bayes
   2. Gaussian Naïve Bayes
2. XGBoost – an ensemble method that specifically, implements the algorithm for decision tree boosting with an additional custom regularization term in the objective function
3. SVM Classifier – a supervised model that uses classification algorithms for two-group classification problems
4. Random Forest Classifier - an ensemble learning method for classification that operate by constructing a multitude of decision trees at training time



On the extreme left, there are techniques mentioned used in preprocessing and in front of each model, the Bag of Words technique is mentioned. Out of these models, we finalized the highlighted top 3 to proceed with and also tune them further.

## HYPERPARAMETER TUNING ON BEST MODELS

Table

Description automatically generatedResults after hyperparameter tuning are listed below. SVM and Random Forest improved after tuning.

## LEARNING CURVE

Chart, line chart

Description automatically generatedSince we had limited amount of data, we plotted the learning curves for our top 3 models to see whether increasing training samples will help improve performance and recall rate.

As per the learning curves, it can be seen the training and cross validation is converging and validation recall score continues to rise so additional training samples can help train the models better.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

# Model Evaluation

## EVALUATION METRICS

Since our project is aimed at classifying the bully text (insult=1) accurately and it doesn’t harm if it classifies a few non-bully texts as bully text, we used **Recall** (True positive rate) as our primary evaluation metric. We also evaluated the models based on confusion matrix for a clearer picture.

## RECALL & CONFUSION MATRIX ON TEST (VALIDATION) DATASET

Table

Description automatically generated

**Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated Gaussian SVM Random Forest**

\*Disclaimer: the results of Random Forest kept changing on each run, which is why the CM and Recall may not look aligned since both are from different runs here

# Results on Unseen Data

## PREPROCESSING

We preprocessed the unseen data using the same method as was used for training data (except using SMOTE in order to maintain integrity of the unseen test set) and created Bag of Words using Count Vectorizer given the final models we picked for our final evaluation.

## Chart, pie chart Description automatically generatedUNSEEN DATA VISUALIZATION - DATA SPLIT

There are a total of 2647 rows in our Unseen data and split of normal vs bully text is shown on the left

As it can be seen, our unseen data is also pretty imbalanced. However, the model has been trained to work well with skewed dataset as well.

## RESULTS & OBSERVATIONS

Chart

Description automatically generatedOut of our top 3 models, we went with **Gaussian Naïve Bayes** for testing the Unseen Data on. Reason being its recall rate on test (validation) dataset was least fluctuating in majority of the test runs we did with different random states and it wasn’t tuned (so least likely to end up overfitting the unseen data set). Also its best suited for generic classification amongst all.

A picture containing text

Description automatically generated

It can be seen from the results that the Gaussian Naïve Bayes model, the one that we didn’t tune to improve further actually did better on unseen data (recall 89.8) than it did on test(validation) data.

One reason for performing even better than the test (validation set) is that the split our model got for validation was probably much harder/complex than the one in the Unseen set. The model managed to classify all except 71 bully/insulting texts correctly.

# Insights

1. Integrating the use of EMOJIs (as part of punctuations) can help to improve the classification process on the basis that integrating punctuations did help
2. Using punctuations in the text actually help increase the accuracy
3. Go for models that are less tuned and generalize much better on multiple runs of varying splits than those which give almost perfect score on sample tests
4. Using information like mutual friends/friends while extracting tweets is expected to help the classifier classify better a friendly banter & a mean insult
5. Mean text length (and other numerical features) although seemed to help didn’t actually improve the performance of any models

# FUTURE SCOPE & Recommendations

As per our analysis, we were able to correctly predict approx. 90% insulting texts which is pretty good. However, this performance can be further improved by adding more features. The ability to detect insulting texts has numerous useful uses and can actually block users from insulting another person online by just putting a filter on the website/personal computer that blocks posting anything offensive. Future scope for the project:

1. Use RNN, LSTM classifier for prediction as they are more established with NLP tasks
2. Make predictions by fetching live tweets using Twitter API – this can be used to prevent offensive texts from being posted in the first place making social media a safer place
3. Use other social networking sites like You-tube, Facebook, Instagram to increase the database
4. Use of NLP augmentation to extrapolate the data in case there’s an issue of data availability
5. Build an application that can flag a text as an insult/bully/offensive before its posted so as to prevent anything offensive from going online. Giving someone a second chance to rethink what they’re about to post can really help him cool down and not type anything offensive online
6. Complete a test-run of the API so as to see whether frequent prompts on non-insulting texts impact the user experience way too negatively or whether it’s acceptable – adjust the recall threshold required accordingly
7. Identifying clusters of bullies using historical/sequential data over a period of time showing a pattern of bullying by using sequential models
8. Experimenting with social-media oriented cleaning techniques like spell check and correction, lemmatization with position tag to configure words better as nouns, verbs etc. and removal of highly common words