# Final Project Report

Ritisha Singh 2021089
Nikita Rajesh Verma 2021546
Jeremiah Malsawmkima Rokhum 2021533
Sanskar Ranjan 2021096

## 1. Abstract:

*According to a 2021 study by the National Sexual Assault Hotline, 20% of children have suffered sexual abuse online, which can result in post-traumatic stress disorder, depression, and anxiety. Because it can lead to addiction, mental health problems, strained relationships, and workplace disruptions, the proliferation of Not Safe For Work (NSFW) text data is a serious problem. The primary motivation for making an attempt to develop an NSFW content classifier is to make online environments safer so that users can interact in a more courteous and comfortable manner without being exposed to unwanted or objectionable stuff. To do this, we plan to develop four separate models that will be trained on a dataset that was taken from a variety of posts that were spread around social media sites like Reddit, etc. Naive Bayes, Logistic Regression, Support Vector Machines, and CNN are some of these models. Additionally, we thoroughly analysed each of them and compared the outcomes in the report. The objective is to create a classifier that successfully excludes NSFW content.*

## 2. Introduction

The internet hosts a vast amount of content, and it's essential to ensure a safe online environment. Reddit, being a popular social platform, contains a diverse range of content, including Not Safe For Work (NSFW) material. Identifying and filtering out NSFW content is crucial for maintaining a safe and appropriate online experience for users of all ages. This project aims to leverage machine learning techniques to develop an automated system capable of detecting NSFW content on Reddit.

## 3. Literature Survey:

1. **NSFW Text Identification:** This paper focuses on the identification of Not Safe For Work (NSFW) content on social media platforms, particularly Reddit and Twitter. The author experiments with various Transformer-based models, primarily DistilBERT, for NSFW content classification. A new NSFW Reddit dataset is created for training and evaluation purposes.

   The study compares different models' performance, finding that DistilBERT trained on a combination of post titles and self-text achieves the best results, with F1 scores ranging from 0.87 to 0.9. However, when evaluating on a Twitter dataset, performance drops due to nuances in content and label bias, indicating the need for domain-specific models.

   Additionally, the paper compares the author's models to OpenAI models for content moderation, showing that OpenAI's content filter model struggles with granularity, while their moderation model sets a high threshold for labeling NSFW content, resulting in low recall.

   In conclusion, the paper highlights the challenges of NSFW content detection on social media platforms and underscores the importance of domain-specific models to handle the nuances of each platform's content.

2. **Classification of Reddit posts: predicting "Not Safe For Work:** The paper "Classification of Reddit Posts using NLP Techniques" presents a study on the classification of Reddit posts as either "Safe For Work" (SFW) or "Not Safe For Work" (NSFW) using different classifiers and feature selection methods. The authors collected a dataset of 4 million Reddit post titles and used three benchmark classifiers: Naive Bayes, Bernoulli Naive Bayes, and Stochastic Gradient Descent. They also implemented bigram feature extraction, Tf-Idf weighing, $X^2$ scoring, and selection to improve the accuracy of their classifiers.

   The results of the study showed that the Bernoulli Naive Bayes classifier, which binarizes the occurrence of a token in a document, performed better than the other classifiers on this type of dataset. The authors obtained a score of 0.56 with a false negative rate of 40.7% in their most successful run. The F-score for the Bernoulli Naive Bayes classifier was 0.63, which was higher than the F-scores for the other classifiers. The authors also evaluated the classifiers' rate of miss-classifying NSFW posts as SFW and found that the Bernoulli Naive Bayes classifier had the lowest rate of miss-classification.

   The authors filtered out inappropriate words and created a word cloud of the top NSFW features. They ranked the features by their $X^2$ score and found the top 20 words whose occurrence in a post's title significantly increases its probability of linking to NSFW content. Interestingly, words such as "favorite", "friend", and "first" occur with the "F" truncated, which the authors discovered is a common format used by female users to indicate that the picture linked to the post is that of a female.

   The study demonstrates the effectiveness of using different classifiers and feature selection methods to predict NSFW content on Reddit. The authors' findings have important implications for content moderation and user safety on social media platforms. The methods and techniques used in the study can also be applied to other social media platforms to predict mature content.

## 4. Dataset:

### 4.1. Dataset Details:

The dataset used for our model comprises text comments, each paired with a toxicity label in the ”target” column. Our primary goal is to predict whether these comments contain toxic content or not. The ”target” column provides a numerical toxicity score, with the maximum value being 1.0, indicating a high level of toxicity, while the average toxicity value across all comments is approximately 0.103, suggesting a relatively low average toxicity level. In addition to the overall toxicity label, the dataset provides six subtype attributes: ”severe_toxicity,” ”obscene,” ”threat,” ”insult,” ”identity_attack,” and ”sexual_explicit,” offering insights into specific types of toxicity within the comments.

This comprehensive dataset, with 1,804,874 rows and 18 columns, equips us with a wide array of attributes to analyze and predict toxicity levels in comments, with subtype attributes offering insights into the specific types of toxicity present. But, the dataset on which we are training our model on is approximately 5% of the original dataset.

Using word cloud, which is visualization tool that displays words in a graphical manner, where the size of each word represents its frequency or importance in a given text or dataset, we describe the most prominent words under particular label to understand the comment text in our database.


Figure 1. Word Cloud where threat label is $> 0.5$.

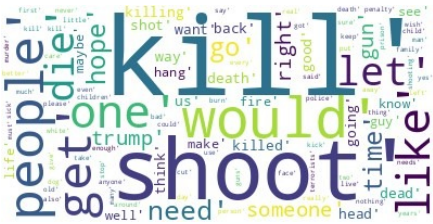


Figure 2. Word Cloud where severe toxicity is $> 0.5$.


Figure 3. Word Cloud where identity attack is $> 0.5$.

### 4.2. Data Pre-processing Techniques

There is a huge imbalance in the dataset values based on the class target. In the graph below, we can see the distribution of target in train set. Density is concentrated primarily around zero which means that our dataset has a huge amount of Non-NSFW data. This imbalance may lead to skewed results and therefore class imbalance mitigation was performed to make the amount of NSFW and non-NSFW data equal in the dataset. Since our database also has 1.8 million rows, we undersampled to 200,000 rows to make computation simpler.

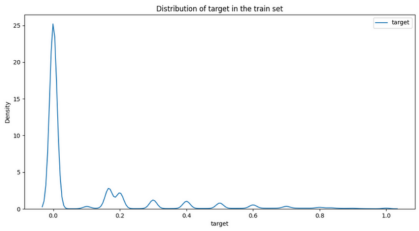


Figure 4. Target distribution of the original dataset

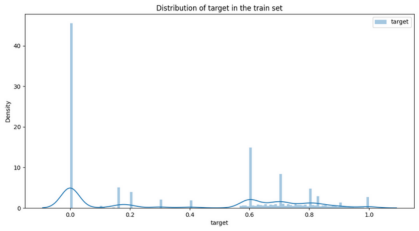


Figure 5. Target distribution of the balanced dataset

#### 4.2.1 Lower-casing:

Lower casing involves converting all text characters to lowercase. We applied this transformation to our input text to ensure consistency and facilitate downstream NLP tasks like tokenization and stemming.

#### 4.2.2 Removal of Special Characters & Contractions:

Special characters and contractions can add noise to the text, making it more difficult for the model to learn. Special characters include punctuation marks, symbols, and emojis. Contractions are words that have been shortened by removing some letters and replacing them with an apostrophe, such as ”can’t” instead of ”cannot”. We remove these to clarify the text and reduce noise, thereby boosting the model comprehension.

#### 4.2.3 Removal of stopwords:

To streamline the feature set and to focus on more meaningful content, we excluded common words like ”the”, ”is”, and ”of”, known as stop words, from our text data.

#### 4.2.4 Tokenization:

It is the process of splitting a text into individual tokens, such as words, numbers, and punctuation marks. To achieve this, we made use of word_tokenize from the nltk.tokenize submodule.

#### 4.2.5 Stemming:

Stemming is the process of reducing words to their root form. This is done by removing affixes, such as prefixes and suffixes. We performed stemming to improve the performance of the model by making it easier to identify similar words by using

the NLTK's PorterStemmer and word_tokenize functions. Example: "running", "ran", "runs" -¿ "run".

## 5. Methodology:

In the context of our NSFW content classification model, a fundamental requirement is to convert textual input into a format suitable for computational analysis. This conversion, known as vectorization or word embedding, is essential. It allows our model to process text and make predictions. By representing words as numerical values, we bridge the gap between language and machine learning. These numerical representations capture language nuances and relationships, enabling our model to accurately classify NSFW content and promote safer online spaces.

To achieve the requisite transformation of textual data into numerical values for computational purposes, we have opted for two prominent techniques: Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF).

### 5.1. Bag of Words (BoW):

This technique represents a document as an unordered collection of words, disregarding grammar and word order but retaining word frequency information. It constructs a vocabulary from the entire corpus of documents and creates a matrix where each row corresponds to a document, and each column represents a unique word in the vocabulary. The value in each cell of the matrix represents the count of how many times a particular word appears in the corresponding document. The formula for BoW can be defined as follows:

**Bow(w, d)= Number of times word w appears in document d**

### 5.2. Term Frequency-Inverse Document Frequency (TF-IDF):

TF-IDF is a more advanced vectorization technique that evaluates the importance of a word in a document relative to a corpus. It not only considers word frequency but also accounts for the uniqueness of a word across the entire dataset. TF-IDF is calculated by multiplying two components: Term Frequency (TF), which measures the frequency of a word in a document, and Inverse Document Frequency (IDF), which quantifies how unique or rare a word is across all documents. The formula for TF-IDF is as follows:

**Tf-idf(w, d)= Bow(w, d) * log(Total Number of Documents /(Number of documents in which word w appears))**

Where: TF(word) is the term frequency of the word in the document. IDF(word) is the inverse document frequency of the word across the entire corpus. It is calculated as the logarithm of the total number of documents divided by the number of documents containing the word.

### 5.3. Logistic Regression

Logistic regression is a statistical model and a widely used method in machine learning for binary classification and, in some cases, for multi-class classification problems. As for the regularisation parameter, C, we have considered 0.1 for BOW and 0.01 for TFIDF.

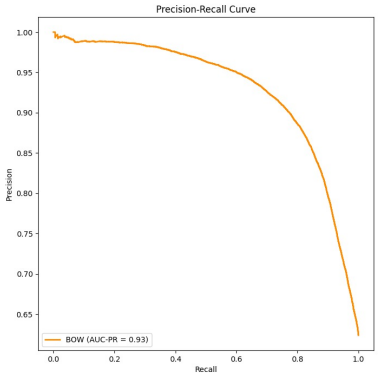


Figure 6. Precision Recall Curve for BOW

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.74 | 0.77 | 0.76 | 15042 |
| 1 | 0.86 | 0.84 | 0.85 | 24958 |
| accuracy | | | 0.81 | 40000 |
| macro avg | 0.80 | 0.81 | 0.80 | 40000 |
| weighted avg | 0.82 | 0.81 | 0.82 | 40000 |

Table 1. Using Logistic Regression' (BOW)

- Accuracy(BoW): 0.81485

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.75 | 0.80 | 0.77 | 15042 |
| 1 | 0.87 | 0.84 | 0.86 | 24958 |
| accuracy | | | 0.83 | 40000 |
| macro avg | 0.81 | 0.82 | 0.82 | 40000 |
| weighted avg | 0.83 | 0.83 | 0.83 | 40000 |

Table 2. Using Logistic Regression' (TF-IDF)

- Accuracy(TF-IDF): 0.8251

| | |
|---|---|
| 11678 | 3364 |
| 4017 | 20941 |

Table 3. Confusion Matrix for BOW feature

| | |
|---|---|
| 11911 | 3131 |
| 3868 | 21090 |

Table 4. Confusion Matrix for TF-IDF Feature

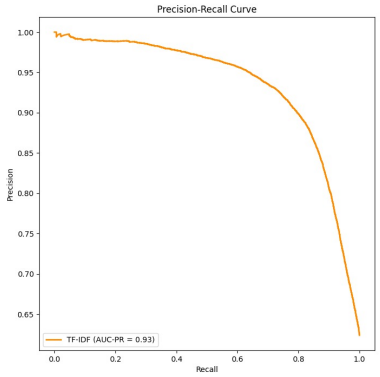


Figure 7. Precision Recall Curve for TF-IDF

### 5.4. Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm used for classification and text categorization tasks. It is based on Bayes' theorem, a fundamental principle in probability theory, and is particularly well-suited for problems involving the classification of data into multiple categories or classes.

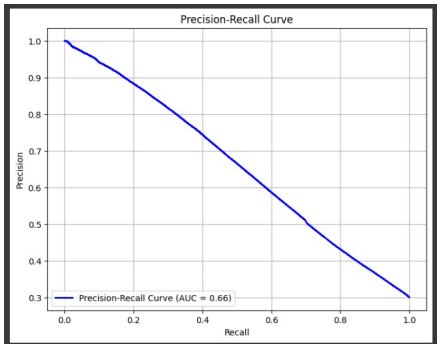


Figure 8. Precision Recall Curve for Naive Bayes

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.76 | 0.98 | 0.85 | 15042 |
| 1 | 0.83 | 0.28 | 0.42 | 24958 |
| accuracy | | | 0.77 | 40000 |
| macro avg | 0.80 | 0.63 | 0.64 | 40000 |
| weighted avg | 0.78 | 0.77 | 0.72 | 40000 |

Table 5. Using Naive Bayes'

### 5.5. Random Forest

Random Forest is an ensemble machine learning algorithm that builds multiple decision trees during training and merges them to improve accuracy and reduce overfitting. It combines the predictions of individual trees to enhance overall model performance. The hyperparameter used for n_estimators (i.e. the number of trees in a forest) is set as 100. The accuracy comes out to be 0.89.

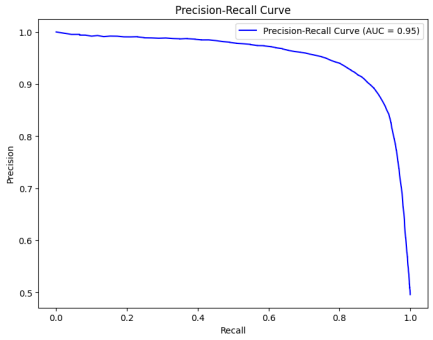


Figure 9. Precision Recall Curve for Random Forest

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.88 | 0.91 | 0.90 | 10078 |
| 1 | 0.91 | 0.87 | 0.89 | 9922 |
| accuracy | | | 0.89 | 20000 |
| macro avg | 0.89 | 0.89 | 0.89 | 20000 |
| weighted avg | 0.89 | 0.89 | 0.89 | 20000 |

Table 6. Using Random Forest'

### 5.6. Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that separates different classes in the feature space. The term "support vector" refers to the data points that are critical in determining the position of the hyperplane. We achieved a remarkable accuracy of 0.93675 on training SVM on our dataset.

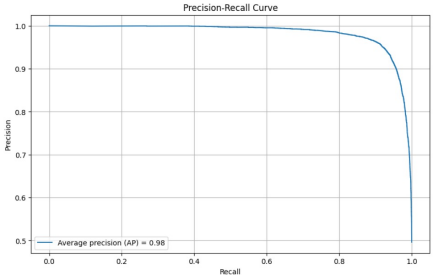


Figure 10. Precision Precall Curve for SVM

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.92 | 0.95 | 0.94 | 10078 |
| 1 | 0.95 | 0.92 | 0.94 | 9922 |
| accuracy | | | 0.94 | 20000 |
| macro avg | 0.94 | 0.94 | 0.94 | 20000 |
| weighted avg | 0.94 | 0.94 | 0.94 | 20000 |

Table 7. Using Support Vector Machine'

### 5.7. Convolutional Neural Network

Convolutional Neural Networks (CNNs) for binary classification use layers to automatically extract features from input data, enabling effective decision-making. Widely applied in tasks like image-based binary classification, CNNs learn patterns and relationships, enhancing the model's ability to distinguish between two classes. In our neural network, we used Lossy ReLU as the hidden activation function, Sigmoid for the last layer, and Binary

Crossentropy as the loss function. This combination is tailored for effective binary classification tasks. We achieved a remarkable score of 0.9273.

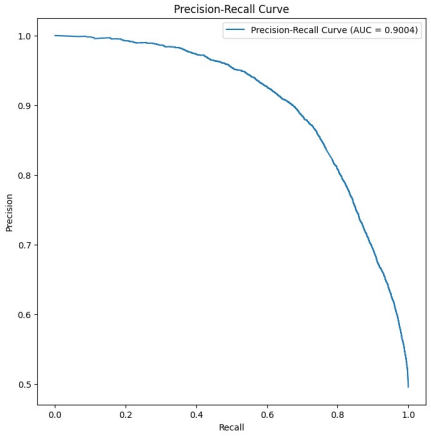


Figure 11. Precision Recall Curve for CNN

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.92 | 0.93 | 0.93 | 10078 |
| 1 | 0.93 | 0.92 | 0.93 | 9922 |
| accuracy | | | 0.93 | 20000 |
| macro avg | 0.93 | 0.93 | 0.93 | 20000 |
| weighted avg | 0.93 | 0.93 | 0.93 | 20000 |

Table 8. Using CNN'

## 6. Results and analysis

On Evaluating the five machine learning models that has been used, we have noticed the following:

- The SVM model demonstrated exceptional performance with an accuracy of 0.93675. The balanced precision, recall, and F1 scores for both classes indicate robust discrimination capabilities. The model's effectiveness in high-dimensional spaces and its ability to handle complex decision boundaries allows it to capture intricate patterns in NSFW text. Low number of false positives and negatives indicates reliability in classifying both NSFW and non-NSFW instances.
- The CNN model performed exceptionally well in the task of NSFW text identification, achieving an accuracy of 0.9263. Its effectiveness in capturing spatial hierarchies in sequential data, along with balanced precision, recall, and F1-scores for both classes, highlight its suitability for NLP tasks. The model's ability to learn hierarchical representations contributes to its strong performance in discriminating between NSFW and non-NSFW text.
- The Random Forests ensemble model demonstrated strong performance, achieving an accuracy of 0.89. The precision, recall, and F1-scores for both classes are balanced, thus depicting model's effectiveness in discriminating between NSFW and non-NSFW instances. The ensemble nature of Random Forests contributes to robustness and mitigates overfitting, making it a suitable choice for accurate discrimination between classes.
- Logistic Regression with a Bag-of-Words achieved a reasonable accuracy of 0.81485. The model has lower precision and recall for class 0 i.e. non-NSFW text thus showing slight imbalance in performance. The linear nature of logistic regression struggle to capture the complexity of NSFW text relationships.
- Logistic Regression with a TF-IDF achieved an accuracy of 0.8251, slightly outperforming its BOW counterpart. Similar to BOW, there is a slight imbalance in performance between the two classes in terms of precision. The TF-IDF representation has improved performance, but it stillstruggles in capturing nuanced relationships in NSFW text.
- The Naive Bayes model exhibited reasonable performance with an accuracy of 0.77. The low recall score for class 1 indicates challenges in correctly identifying NSFW instances. The naive assumption of independence between features limits its ability to capture complex relationships in NLP tasks.

| Models | Accuracies |
|---|---|
| Logistic Regression(BOW) | 0.81485 |
| Logistic Regression(TF-IDF) | 0.8251 |
| Naïve Bayes | 0.7724 |
| Random Forest | 0.89 |
| SVM | 0.93675 |
| CNN | 0.9273 |

Table 9. Accuracies

## 7. Conclusion

In conclusion, our NFSW detection ML project has been a challenging yet rewarding endeavor. Working within the constraints of limited RAM and CPU resources provided by Google Colab posed significant hurdles, but we persevered to achieve meaningful results.

One of the most critical challenges we faced was the stratification of our dataset. Due to the inherent bias towards non-toxic comments in the available data, our models tended to perform better in identifying non-toxic content. This bias created an imbalance that affected the overall accuracy of our ML models, making it challenging to achieve a balanced performance in detecting NFSW content.

Despite these challenges, our project has provided valuable insights into the complexities of NFSW detection in online content. We have learned the importance of robust dataset preparation, feature engineering, and the fine-tuning of ML algorithms to mitigate bias and improve the overall accuracy of the models. Additionally, working within the resource limitations of Google Colab has taught us to optimize our code and processes for efficiency.

Moving forward, it is clear that NFSW detection is a complex problem that requires ongoing research and development. We must continue to address bias in our data and explore more advanced techniques, including deep learning and transfer learning, to improve the accuracy of our models. Additionally, obtaining access to more computing resources will be essential for scaling our efforts and achieving even better results.

Furthermore, SVM and CNN serves as a solid starting point for researchers to work with when it comes to dealing with

NSFW text classification problems, as it provides a remarkable accuracy on the given data-set. In conclusion, this study demonstrates the potential of machine learning and data preprocessing in addressing NSFW content challenges and underscores the ongoing need for content moderation to enhance user safety on social media platforms.

## 8. References:


1. **Classifying-reddit-posts-with-natural-language-processing-and-machine-learning:**

   Britt. (2021, December 7). Classifying Reddit posts with natural language processing and machine learning. Medium. *classifying-reddit-posts-with-natural-language-processing-and-machine-learning-695f9a576ecb*.

2. **NSFW Text Identification:**
   Tsvetkov, A. (2022). NSFW text identification. ResearchGate. *364652449-NSFW-Text-Identification*.

3. **Classification of Reddit posts: predicting "Not Safe For Work:**
   University of British Columbia. (n.d.). *Title Classification of Reddit posts: predicting "Not Safe For Work" content.*.

4. **Jigsaw Unintended Bias in Toxicity Classification:**

   Dataset from Jigsaw Classification competition: *jigsaw-unintended-bias-in-toxicity-classification/data*.