Cyberbullying Detection

Mattia Segreto

1 Introduction

In last years, the phenomenon of cyberbullying increases significantly; this is likely due to the social network global spread and the growing accessibility of communication tools. To recognize and counter harmful behavior is an urgent challenge in order for promoting a safer digital environment.

The project aim is to propose a detection system for cyberbullying attacks based on the analysis of tweets. Following the detection, a second purpose has been identified: to provide an explanation of how the system reached such a conclusion.

2 Methodology

This chapter presents the methodological framework adopted for the development of the cyberbullying detection system. The approach follows a structured pipeline that includes data preprocessing, feature extraction, supervised learning techniques, and model evaluation. Particular attention is given to the classification process, which is designed in multiple stages to enhance accuracy and interpretability. Furthermore, explainability methods are integrated to improve transparency and align the system with current ethical and regulatory standards.

2.0.1 Data Visualization

Tag clouds were generated for each of the six dataset classes in order to provide a visualization that provides a qualitative overview of the most frequent words associated with each label and offering also an initial insights of the characterizing language patterns among different categories.

Figure 1: Tag clouds for each class.

The tweet lengths distribution across classes was also take in consideration through a boxplot, in order to assess possible differences in verbosity among classes. No significant patterns were identified from this analysis. Consequently, no features related to tweet length were created and included in the final model.

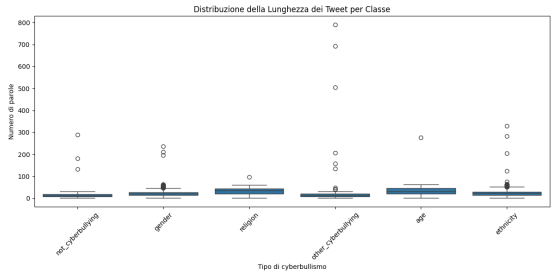


Figure 2: Distribution of tweet lengths per class.

2.0.2 Data Splitting and Stratification

The dataset was split into training and test sets using an 80/20 ratio. Stratified sampling was applied based on the multiclass label in order to preserve the original class distribution in both subsets. A fixed random seed was used to ensure reproducibility of the split.

2.1 Data Collection and Preprocessing

2.1.1 Dataset Description

The following dataset description is based on the dataset introduced in the paper *SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection*, which was specifically created to support fine-grained classification of cyberbullying on Twitter. This dataset contains more than 47000 tweets labelled according to the class of cyberbullying:

- Age;

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

The data has been balanced in order to contain about 8000 of each class.It has two columns that are

- tweet_text: the raw content of the tweet (string format).
- cyberbullying_type: the corresponding label, which can be either not_cyberbullying or one of several cyberbullying types.

The dataset used in the SOSNet model was initially built by merging six public datasets focused on cyberbullying on Twitter, such as those by Waseem, Davidson, and Chatzakou. Since many of these datasets contained only tweet IDs, the authors retrieved the corresponding tweet texts using the Twitter API, though only a portion could be recovered due to tweet deletions over time. The resulting dataset was highly *imbalanced*, with a strong dominance of the *gender* and *other* classes (about 85% of the total), while the *age* and *ethnicity* classes were severely underrepresented (less than 2.5%). To address this issue, the authors introduced a *Dynamic Query Expansion (DQE)* [1]; an iterative process that allows the authors to naturally and effectively balance the dataset.

2.1.2 Text Cleaning

In order to reduce noise and standardize the input for the feature extraction step, a set of preprocessing techniques was applied to the raw tweets. These steps are especially important in the context of Twitter data, which is typically informal, highly variable, and often includes non-linguistic symbols. The cleaning procedure includes:

- **Lowercasing:** All text is converted to lowercase to ensure that words like `Hate` and `hate` are treated as the same token. This helps reduce the vocabulary size without losing semantic meaning.
- **Removal of URLs, mentions, and punctuation:** Tweets often contain hyperlinks, user mentions (e.g., `@username`), and various punctuation marks that do not contribute meaningfully to cyberbullying detection. Removing them simplifies the input without affecting classification performance.

[1] DQE is a semi-supervised strategy that does not generate synthetic tweets, but instead retrieves real tweets from Twitter. It starts from manually selected *seed queries* (e.g., "middle school" for the *Age* class). These queries are used to collect new, real tweets through the `GetOldTweets3` library. The process is repeated iteratively, updating the keywords each time, until a sufficiently rich and balanced dataset is obtained.

- **Stopword removal:** Common words such as `the`, `is`, and `and` are removed to focus on the most informative terms in each tweet. This is particularly helpful for short texts like tweets, where maximizing the signal-to-noise ratio is essential.
- **Stemming:** Words are reduced to their root form (e.g., `harassing` → `harass`), which helps group different inflected forms of the same word under a single representation. This reduces sparsity and improves the effectiveness of models relying on token frequency or similarity. Lemmatization was not used because, given the unstructured nature of the text, it would have been less efficient: lemmatization relies on correct grammatical structure and part-of-speech tagging, which are often unreliable in noisy or informal text.

These choices make the text cleaner, more compact, and more suitable for feature extraction methods, while preserving the linguistic signals needed for cyberbullying detection.

Binary Label Creation: To support the two-stage classification pipeline, the original multiclass label has been accompanied by a binary label distinguishing between cyberbullying and non-cyberbullying content. In this transformation, all tweets originally labeled as `not_cyberbullying` were grouped under the class `0` (non-cyberbullying), while all remaining tweets were grouped under the class `1` (cyberbullying). This binary representation is particularly useful as a preliminary filter in a cascaded classification system, where the first stage detects whether a tweet is abusive at all, and only the second stage performs fine-grained categorization among the different types of cyberbullying. Such a design reduces the complexity of the fine-grained classifier and helps focus its training on content that is more likely to be harmful.

2.2 Feature Extraction

In this project, three different text vectorization methods were employed to extract meaningful features from the tweets: Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word2Vec(W2V-1) embeddings. These techniques allow the conversion of raw textual data into numerical representations suitable for downstream classification tasks.

BoW was used to construct a frequency matrix that reflects how often each word appears across the dataset. While effective in capturing general word usage, this method tends to overrepresent common terms such as *like*, *people*, or *know*, which may carry limited discriminative power for identifying cyberbullying content.

To mitigate this, the TF-IDF approach was also applied. It enhances the BoW representation by down-weighting ubiquitous terms and up-weighting words that are more unique to specific tweets. This helps focus the model on more informative and discriminative tokens, particularly useful in a dataset where repeated expressions of emotion or aggression are relevant.

Additionally, a Word2Vec model was trained to capture semantic and syntactic relationships between words. The model was trained on a large, unlabeled corpus composed of two datasets related to cyberbullying and offensive content, totaling approximately 120,000 preprocessed sentences. The Skip-Gram architecture (`sg=1`) was used, with a `vector_size` of 200 and a `window` of 7. Different `min_count` values were evaluated: setting it to 2 excluded 50,868 words out of 69,392 (73.3%), retaining 18,524 tokens. A lower threshold (1) excluded 62.4%, while values of 5 or 10 would have eliminated over 83% and 88% of the vocabulary, respectively.

This multi-faceted feature extraction setup provided several distinct vector representations of the text, allowing models to be trained and compared across different input spaces.

2.3 Classification Overview

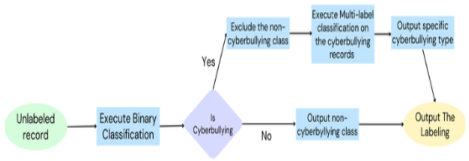


Figure 3: Two-stage classification pipeline.

Source: Self-Training for Cyberbully Detection: Achieving High Accuracy with a Balanced Multi-Class Dataset, available at `https://uregina.ca/~nss373/papers/cyberbully-detection.pdf`

The two-stage classification improves the system's interpretability: by splitting the process into detection and specific categorization, it becomes easier to identify where an error occurs and intervene accordingly, with greater precision in tuning and feature analysis.

Moreover, this structure mirrors human reasoning, which typically involves first assessing whether content is problematic and only then analyzing its severity or nature. As a result, errors can also be interpreted on two different levels of severity: failing to detect harmful content is clearly more critical than misclassifying the specific type of cyberbullying. This is especially true since some offensive messages may not fit neatly into a single label, and therefore an error at the multiclass stage does not necessarily represent a serious failure. This approach enables a more realistic understanding of the model's behavior and a more informed management of its performance in practical applications. For example, consider the following case: the ground truth label for the tweet "my 2282 says that you can replace the testimony of one man with that of two women" was set as religion, whereas the model predicted gender—and arguably, it was correct to do so.[2] This suggests that the reported results may underes-

[2]The tweet likely refers to verse 2:282 of the Quran, which discusses financial contracts

timate the models' actual real-world performance, as some supposed "errors" stem from questionable ground truth labels rather than true misclassifications.

2.4 Binary Classification Stage

The binary classification stage aims to distinguish between tweets containing cyberbullying and those that are harmless. Each tweet is labeled as either `cyberbullying` or `not_cyberbullying`, framing the task as a supervised binary classification problem.

In the pipeline, a balancing step using SMOTE (Synthetic Minority Oversampling Technique) was also included prior to classifier training. The use of SMOTE was motivated by the class imbalance, with tweets labeled as "not_cyberbullying" being significantly underrepresented compared to "cyberbullying" tweets. In the text mining context, SMOTE was applied after textual transformation via vectorization, making it possible to generate synthetic interpolation in the feature space. This approach allowed for the creation of realistic artificial samples of the minority class, improving the model's ability to learn more balanced representations and reducing the risk of bias toward the majority class. The choice of SMOTE over other balancing methods such as random oversampling or undersampling was based on both empirical and methodological considerations. Unlike random oversampling, which merely replicates existing examples and increases the risk of overfitting, SMOTE synthesizes new samples by interpolating between neighboring instances of the minority class in the feature space. This introduces greater variability and supports better model generalization. In this specific case, balancing would have required increasing the number of "not_cyberbullying" tweets by approximately five times; simple oversampling would have led to excessive repetition of the same texts, reducing dataset diversity and potentially impairing learning. Conversely, applying undersampling would have required removing a substantial portion of "cyberbullying" tweets, resulting in a significant loss of information—particularly problematic in a domain where linguistic and contextual nuances play a crucial role in classification. Alternative text-specific data augmentation techniques, such as *back translation* and *synonym replacement*, were also considered. However, these techniques don't take effective benefits in this context due to the often irregular syntax, informal language, and frequent use of abbreviations or slang typical of tweets. Moreover, synonym replacement in a highly connoted domain with idiomatic expressions may introduce noise rather than enriching the dataset. Therefore, all this considerations lead the choice to SMOTE.

Nested cross-validation was performed for each of the nine model–vectorizer combination pipelines to obtain unbiased performance estimates and tune hyperparameters.The classifiers that has been used are:Random Forest, Logistic Regression, and Linear SVM. Model evaluation employed a nested stratified k-fold framework: an outer loop estimated generalization performance, while

and stipulates that, if two male witnesses are not available, one man and two women may testify. Although the rule originates from a religious text, the content of the tweet explicitly highlights a gender-related issue, leading the model to a plausible alternative classification.

an inner loop conducted hyperparameter tuning. From each inner fold, the optimal parameter set was recorded, and upon completion of all folds the final hyperparameters were chosen as the most frequent combination ("mode") across folds. That configuration was then used to retrain the pipeline on the entire training set. Optimization targeted the macro-F1 score to balance precision and recall, thus addressing both false positives and false negatives. Model selection involved pairwise comparisons among all pipelines, with a "win" noted when one model outperformed another. To ensure robustness and ample sample size, all comparisons were based on a repeated stratified k-fold cross-validation scheme. The normality of per-fold F1-macro differences was assessed using the Shapiro–Wilk test; if normality held, a paired t-test was applied, otherwise the Wilcoxon signed-rank test was used.

2.5 Multiclass Classification Stage

The goal of the multiclass classification stage is to assign a specific type of cyberbullying to each tweet previously identified as harmful. The hyperparameter-tuning procedure was identical to that used in the binary stage(without rebalancing step): each of the three classifiers (Random Forest, Logistic Regression, and Linear SVM) was paired with each of the three vectorization methods, yielding nine model configurations. Nested stratified k-fold cross-validation was employed to optimize hyperparameters;this time targeting macro-f1 and, once tuning was complete, each pipeline was retrained on the full training set. Model selection again relied on pairwise comparisons of per-fold scores (wins counted per fold) with significance assessed via Shapiro–Wilk for normality and paired t-tests or Wilcoxon signed-rank tests as appropriate.

2.6 Explainability Module

To improve model transparency and interpretability, an explainability module was integrated into the classification pipeline. This component plays a crucial role in enhancing trust and accountability in machine learning applications, especially in sensitive tasks such as cyberbullying detection. Both local and global explanability were provided.

Global Explainability. Pattern mining techniques were applied to study the most frequent and informative word associations within the dataset. Specifically, the most frequently occurring words were analyzed, and maximal and closed itemsets meeting a predefined support threshold were extracted. Rather than focusing on association rules, which are better suited to modeling directional relationships between items, maximal and closed itemsets has been chosen. This choice was motivated by the nature of the data: tweets are extremely short and informal, making it difficult to establish strong directional associations between words. Maximal and closed itemsets allowed us to capture robust co-occurrence patterns without introducing noise or spurious dependencies, providing a clearer and more stable view of the most informative word groupings associated with

different cyberbullying categories. In parallel, for the multiclass classifier, the *feature_importance* attribute of the Random Forest model (which showed better performance) was used to rank features by their global relevance. For the binary classifier, the `.coef_` attribute of the Logistic Regression model (which also showed better performance) was used for the same purpose, offering insight into the textual elements that most influenced the model during training. This dual approach enables a comprehensive form of global explainability: on one hand, it provides an understanding of the model's behavior through feature importance; on the other, it offers an intrinsic explanation of the phenomenon itself by uncovering the underlying word patterns through pattern mining.

Local Explainability. For multiclass instance-level interpretation, the TreeInterpreter tool was employed , while, For binary instance-level interpretation, the SHAP library was employed. These methods decompose the output of the models into individual feature contributions, making it possible to understand which specific words influenced the classification of a given tweet. Thanks to the use of TF-IDF and BoW vectorizations (that are the vectorizations used rispectly by best performing multiclass and binary classifiers), each explanation could be interpreted in terms of the positive or negative contribution of each word. Interestingly, the model also relies on the absence of certain words: even terms not present in the tweet can carry weight by shifting the decision boundary. This aspect highlights the nuanced reasoning adopted by the classifier and provides users with a deeper understanding of its behavior.

3 Experimental Results

This section presents the results obtained at each stage of the proposed pipeline, following the methodological steps described previously. For each stage, the evaluation metrics used, the performance of the tested models, and a short discussion of the results were reported.

3.1 Binary Classification Results

Experimental Setup and Grid Search. As described in Section 2.4, three classifiers Random Forest, Logistic Regression, and Linear SVM were evaluated in combination with three vectorization methods: Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and custom-trained Word2Vec embedding (W2V-1). This led to a total of 9 distinct model configurations. The grid of hyperparameters used during tuning was the following:

- **Logistic Regression:**
  - `C:` {0.01, 0.1, 1}
  - `class_weight:` "balanced"
- **Linear SVM:**

  - `C`: {0.01, 1, 10}

- **Random Forest:**
  - `n_estimators`: {100, 500, 1000}
  - `max_depth`: {None, 20}
  - `class_weight`: "balanced"
  - `random_state`: 42

Model evaluation Table 1 reports the optimal hyperparameters that were selected via nested cross-validation for each vectorizer–classifier pipeline. For each configuration, it's been showed the mode of the best parameters found across the ten outer folds.

Configuration	Chosen Parameters
BoW + LogisticRegression	{'C': 0.1, 'class_weight': 'balanced'}
BoW + RandomForest	{'class_weight': 'balanced', 'max_depth': 20, 'n_estimators': 1000, 'random_state': 42}
BoW + LinearSVM	{'C': 0.01}
TF-IDF + LogisticRegression	{'C': 1, 'class_weight': 'balanced'}
TF-IDF + RandomForest	{'class_weight': 'balanced', 'max_depth': 20, 'n_estimators': 1000, 'random_state': 42}
TF-IDF + LinearSVM	{'C': 1}
W2V-1 + LogisticRegression	{'C': 0.1, 'class_weight': 'balanced'}
W2V-1 + RandomForest	{'class_weight': 'balanced', 'max_depth': 20, 'n_estimators': 500, 'random_state': 42}
W2V-1 + LinearSVM	{'C': 0.01}

Table 1: Chosen parameters for each configuration (mode over 10 folds)

Table 2 summarizes the key evaluation metrics: mean accuracy, precision, recall F1 score and F1_macro score along with their standard deviations across the outer folds. This allows you to compare not only the average performance but also the stability of each model.

Vectorizer	Classifier	Mean f1_macro	Mean Acc.	Mean Prec.	Mean Rec.	Mean F1	Std f1_macro	Std Acc.	Std Prec.	Std Rec.	Std F1
TF-IDF	LogisticRegression	0.726069	0.804865	0.954029	0.804667	0.872959	0.006945	0.007003	0.004034	0.009846	0.005238
BoW	LogisticRegression	0.725971	0.813960	0.941532	0.828254	0.881224	0.006746	0.005840	0.004872	0.008769	0.004247
TF-IDF	LinearSVM	0.723712	0.800409	0.956956	0.796364	0.869248	0.007846	0.008319	0.003659	0.011461	0.006295
BoW	LinearSVM	0.723284	0.808796	0.944441	0.818757	0.877098	0.006590	0.005756	0.003389	0.007120	0.004099
BoW	RandomForest	0.717003	0.782376	0.975631	0.757807	0.853000	0.007483	0.007750	0.002173	0.009383	0.005980
TF-IDF	RandomForest	0.716418	0.788064	0.968466	0.772275	0.858258	0.008257	0.011832	0.023774	0.036043	0.011849
W2V-1	LogisticRegression	0.688008	0.771394	0.942390	0.772966	0.849284	0.005488	0.005804	0.003036	0.007766	0.004479
W2V-1	LinearSVM	0.685688	0.767620	0.943780	0.766865	0.846139	0.006355	0.006880	0.002972	0.008999	0.005342
W2V-1	RandomForest	0.677147	0.810945	0.897570	0.872755	0.884980	0.007977	0.005702	0.002755	0.005850	0.003708

Table 2: Nested CV Results: mean metrics and standard deviations

Model selection In Table 3 has been presented the pairwise victory matrix across all model–vectorizer combinations. A value of 1 indicates that the model in the row significantly outperformed the model in the column in head-to-head statistical tests, while a value of -1 indicates that the model underperformed the model in column; whereas 0 denotes no statistically significant win.

	LR + TF-IDF	LR + BoW	LSVM + TF-IDF	LSVM + BoW	RF + BoW	RF + TF-IDF	LR + W2V-1	LSVM + W2V-1	RF + W2V-1
LR + TF-IDF	0	0	1	1	1	1	1	1	1
LR + BoW	0	0	1	1	1	1	1	1	1
LSVM + TF-IDF	-1	-1	0	0	1	1	1	1	1
LSVM + BoW	-1	-1	0	0	1	1	1	1	1
RF + BoW	-1	-1	-1	-1	0	0	1	1	1
RF + TF-IDF	-1	-1	-1	-1	0	0	1	1	1
LR + W2V-1	-1	-1	-1	-1	-1	-1	0	1	1
LSVM + W2V-1	-1	-1	-1	-1	-1	-1	-1	0	1
RF + W2V-1	-1	-1	-1	-1	-1	-1	-1	-1	0

Table 3: Pairwise victory matrix (1 = win, 0 = tie, -1 = loss)

Table 4 summarizes the total number of head-to-head victories for each pipeline, giving an at-a-glance ranking of overall comparative performance. Higher counts reflect more frequent wins against competing configurations.

Model	**Victories**
LogisticRegression + TF-IDF	7
LogisticRegression + BoW	7
LinearSVM + TF-IDF	3
LinearSVM + BoW	3
RandomForest + BoW	-1
RandomForest + TF-IDF	-1
LogisticRegression + W2V-1	-4
LinearSVM + W2V-1	-6
RandomForest + W2V-1	-8

Table 4: Number of head-to-head victories per model

Focusing on the two top-ranked models, Table 5 details the results of the pairwise statistical tests used to establish significance. For each winner–opponent pair has been reported the test type, test statistic, p-value, and the final outcome (win/no win).

Winner	Opponent	Test	Stat.	p-value	Outcome
LogisticRegression + TF-IDF					
	LogisticRegression + BoW	t-test	-0.4003	0.6918	no win
	LinearSVM + TF-IDF	t-test	3.8696	0.0006	win
	LinearSVM + BoW	t-test	2.2984	0.0289	win
	RandomForest + BoW	t-test	10.0545	0.0000	win
	RandomForest + TF-IDF	t-test	8.7312	0.0000	win
	LogisticRegression + W2V-1	t-test	30.8589	0.0000	win
	LinearSVM + W2V-1	t-test	30.7596	0.0000	win
	RandomForest + W2V-1	t-test	27.8691	0.0000	win
LogisticRegression + BoW					
	LogisticRegression + TF-IDF	t-test	0.4003	0.6918	no win
	LinearSVM + TF-IDF	t-test	2.9451	0.0063	win
	LinearSVM + BoW	t-test	7.6112	0.0000	win
	RandomForest + BoW	t-test	10.1741	0.0000	win
	RandomForest + TF-IDF	Wilcoxon	20.0000	0.0000	win
	LogisticRegression + W2V-1	t-test	28.9065	0.0000	win
	LinearSVM + W2V-1	t-test	29.9886	0.0000	win
	RandomForest + W2V-1	t-test	26.0417	0.0000	win

Table 5: Head-to-head statistical tests for the top two models

Logistic Regression combined with TF-IDF and with Bag of Words (BoW) yielded similar performance in terms of macro-averaged F1 score, with no statistically significant differences. Since the primary metric did not provide a clear preference, secondary metrics were evaluated using a stratified repeated k-fold cross-validation to support the final model selection. Logistic Regression with BoW achieved consistently better results in accuracy, recall, and binary F1-score, with statistically significant differences compared to the TF-IDF counterpart. Although TF-IDF showed a modest advantage in precision, this difference was not sufficient to outweigh the gains in recall and F1. Given the nature of the task(detecting cyberbullying) recall and F1-score are especially critical, as missing abusive content (false negatives) is more detrimental than flagging benign content. Based on these considerations, the Logistic Regression + BoW model was selected as the binary classifier within the binary-plus-multiclass pipeline.

3.2 Multiclass Classification Results

Experimental Setup and Grid Search. As described in Section 2.4, three classifiers Random Forest, Logistic Regression, and Linear SVM were evaluated in combination with three vectorization methods: Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and custom-trained Word2Vec embedding (W2V-1). This led to a total of 9 distinct model configurations. The grid of hyperparameters used during tuning was the following:

- **Logistic Regression:**
  - `C`: {0.01, 0.1, 1}

- **Linear SVM:**
  - `C`: {0.01, 1, 10}
- **Random Forest:**
  - `n_estimators`: {100, 500, 1000}
  - `max_depth`: {None, 20}
  - `random_state`: 42

Model evaluation Table 6 reports the optimal hyperparameters that were selected via nested cross-validation for each vectorizer–classifier pipeline in the multiclass setting. For each configuration, we show the mode of the best parameters found across the ten outer folds.

Configuration	Chosen Parameters
BoW + LogisticRegression	`'model__C': 1`
BoW + RandomForest	`'model__max_depth': None, 'model__n_estimators': 500, 'model__random_state': 42`
BoW + LinearSVM	`'model__C': 1`
TF-IDF + LogisticRegression	`'model__C': 1`
TF-IDF + RandomForest	`'model__max_depth': None, 'model__n_estimators': 1000, 'model__random_state': 42`
TF-IDF + LinearSVM	`'model__C': 1`
W2V-1 + LogisticRegression	`'model__C': 1`
W2V-1 + RandomForest	`'model__max_depth': None, 'model__n_estimators': 1000, 'model__random_state': 42`
W2V-1 + LinearSVM	`'model__C': 1`

Table 6: Chosen parameters for each configuration (mode over 10 folds) – Multiclass

Table 7 then summarizes the key evaluation metrics—mean accuracy, precision, recall and F1-macro—along with their standard deviations across the outer folds. This allows for a comprehensive comparison of both the average performance and the consistency of each model.

Vectorizer	Classifier	Mean F1	Mean Acc.	Mean Prec.	Mean Rec.	Std F1	Std Acc.	Std Prec.	Std Rec.
TF-IDF	RandomForest	0.932648	0.932258	0.934340	0.932075	0.004448	0.004524	0.004380	0.004552
BoW	LogisticRegression	0.928662	0.928075	0.931726	0.927980	0.004978	0.005079	0.004646	0.005085
BoW	LinearSVM	0.928478	0.927855	0.932123	0.927780	0.004613	0.004716	0.004409	0.004734
BoW	RandomForest	0.928417	0.928012	0.929886	0.927791	0.003973	0.004083	0.003739	0.004096
TF-IDF	LogisticRegression	0.927957	0.927383	0.930868	0.927272	0.005030	0.005110	0.004667	0.005122
TF-IDF	LinearSVM	0.927945	0.927477	0.930539	0.927351	0.004989	0.005102	0.004760	0.005126
W2V-1	RandomForest	0.892812	0.892317	0.899613	0.892214	0.003530	0.003379	0.003316	0.003390
W2V-1	LinearSVM	0.878282	0.879359	0.878700	0.878843	0.006765	0.006682	0.006861	0.006729
W2V-1	LogisticRegression	0.878051	0.878416	0.878964	0.877938	0.006858	0.006894	0.006865	0.006931

Table 7: Nested CV Results: mean metrics followed by standard deviations – Multiclass

Model selection Table 8 shows the pairwise victory matrix across all model–vectorizer combinations. A value of 1 indicates that the model in the row significantly outperformed the model in the column according to a statistical test, while -1 denotes an underperforming and 0 indicates that there is no statistical difference.

	RF + TF-IDF	LR + BoW	RF + BoW	LSVM + BoW	LSVM + TF-IDF	LR + TF-IDF	RF + W2V-1	LSVM + W2V-1	LR + W2V-1
RF + TF-IDF	0	1	1	1	1	1	1	1	1
LR + BoW	-1	0	0	0	0	1	1	1	1
RF + BoW	-1	0	0	0	0	0	1	1	1
LSVM + BoW	-1	0	0	0	0	0	1	1	1
LSVM + TF-IDF	-1	-1	0	0	0	0	1	1	1
LR + TF-IDF	-1	-1	0	0	0	0	1	1	1
RF + W2V-1	-1	-1	-1	-1	-1	-1	0	1	1
LSVM + W2V-1	-1	-1	-1	-1	-1	-1	-1	0	1
LR + W2V-1	-1	-1	-1	-1	-1	-1	-1	-1	0

Table 8: Pairwise victory matrix (1 = win, 0 = draw, -1 = defeat)

Table 9 reports the total number of head-to-head victories per pipeline. A higher number indicates stronger overall performance in the statistical comparison.

Model	**Victories**
RandomForest + TF-IDF	8
LogisticRegression + BoW	4
RandomForest + BoW	2
LinearSVM + BoW	2
LinearSVM + TF-IDF	1
LogisticRegression + TF-IDF	1
RandomForest + W2V-1	-4
LinearSVM + W2V-1	-6
LogisticRegression + W2V-1	-8

Table 9: Number of head-to-head victories per model

Table 10 lists the statistical test results for the top-ranked pipeline. For each comparison, the test type, statistic, p-value, and outcome (win or no win) are reported.

Winner	**Opponent**	**Test**	**Stat.**	**p-value**	**Outcome**
RandomForest + TF-IDF					
	LogisticRegression + BoW	t-test	6.4230	0.0000	win
	LinearSVM + BoW	t-test	7.2134	0.0000	win
	RandomForest + BoW	t-test	9.6959	0.0000	win
	LogisticRegression + TF-IDF	t-test	7.6588	0.0000	win
	LinearSVM + TF-IDF	t-test	7.9490	0.0000	win
	RandomForest + W2V-1	t-test	46.3386	0.0000	win
	LinearSVM + W2V-1	t-test	55.1314	0.0000	win
	LogisticRegression + W2V-1	t-test	51.0813	0.0000	win

Table 10: Head-to-head statistical tests for RandomForest + TF-IDF

RandomForest + TF-IDF emerged as the pipeline with statistically superior performance so it represents the final choice as the multiclasse classifier to feed into the overall binary-plus-multiclass pipeline. Random Forest offers native

global interpretability through its built-in feature-importance scores and seamless local explanations via TreeInterpreter, while TF-IDF delivers a normalized feature space where each term's weight can be assessed directly.

Further Observation From the confusion matrices(it's been taken the confusion matrix due its easy interpretability), it's possible observe that most misclassifications occur between the `gender` and `other_cyberbullying` classes. This is likely due to the semantic overlap between expressions of gender-based harassment and more generic forms of offensive language. Notably, this pattern of confusion is consistent with observations reported in previous work[3], where misclassifications between gender and other_cyberbullying categories accounted for the majority of classification errors. Nevertheless, class-wise performance remains robust, and no class shows signs of significant bias or neglect.

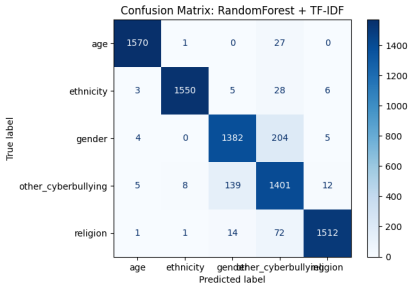


Figure 4: Multiclass confusion matrix.

3.3 Pipeline Evaluation

In this final evaluation, the full classification pipeline was tested in its complete form. The system first applies a binary classifier to distinguish between cyberbullying and non-cyberbullying tweets. The tweets identified as cyberbullying (true positives) are then passed to a second classifier, which assigns a specific type of cyberbullying.

This two-stage structure reflects a realistic moderation workflow, where content is first flagged as problematic and then further analyzed for nature.

[3]Wang, J., Fu, K., & Lu, C.-T. (2020). SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection. In their analysis of the confusion matrix, the authors found that gender and other were the most confused classes, accounting for the majority of the overall error.

3.3.1 Binary Stage Performance.

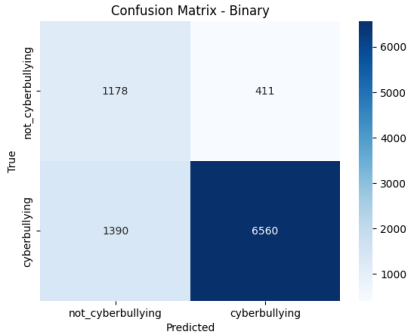


Figure 5: Confusion matrix for binary stage.

3.3.2 Multiclass Stage Performance (on true positives).

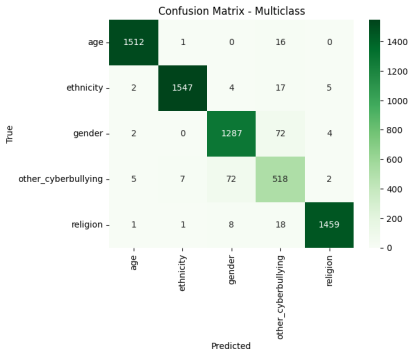


Figure 6: Confusion matrix for multiclass stage (on binary true positives).

Overall Pipeline Performance. By combining the two stages, the overall pipeline correctly classified 7501 tweets out of 9539, resulting in an **end-to-end accuracy of** 0.7864. This metric reflects the proportion of tweets that were both correctly identified as cyberbullying/non-cyberbullying, and, when applicable, correctly categorized into one of the fine-grained classes.

Further Consideration A notable aspect of the evaluation is the improvement in classification accuracy from the multiclass stage alone (0.93) to the same model when used within the pipeline (0.96), despite being applied to the same original test set. This raises an important question: why does the accuracy improve if the dataset is unchanged?

The answer can be found by analyzing the confusion matrix of the multiclass classifier within the pipeline. The `other_cyberbullying` category, which

is the most generic and typically the hardest to classify, appears significantly underrepresented compared to the other labels. However, the original dataset is balanced across classes; suggesting that this discrepancy is not due to class imbalance in the input data.

The cause lies in the binary classifier, which struggles to identify `other_cyberbullying` tweets as harmful. As a result, many examples from this category are incorrectly filtered out in the first stage and never reach the multiclass classifier. This leads to a lower number of `other_cyberbullying` examples in the second stage inflating its accuracy.

This hypothesis is supported by the label distribution shown in the final classification report: only 604 instances of `other_cyberbullying` are processed at the second stage, compared to an average of approximately 1500 for the other categories. Additionally, performance metrics for this class are noticeably lower than for the others, confirming that it remains the most challenging category for the pipeline to handle. So having less occurrences of most problematic labels we can observe an "improvement" of accuracy.

Label	**Precision**	**Recall**	**F1-score**	**Support**
age	0.99	0.99	0.99	1529
ethnicity	0.99	0.98	0.99	1575
gender	0.94	0.94	0.94	1365
other_cyberbullying	0.81	0.86	0.83	604
religion	0.99	0.98	0.99	1487

Table 11: Per-class metrics for the multiclass classifier (pipeline stage).

3.4 Explainability Insights

To support transparency and user trust, the system integrates both global and local explainability modules.

Global Insights. To gain a high-level understanding of the model's behavior, two complementary techniques were used.

Pattern mining was applied to identify the most frequent and informative terms associated with each cyberbullying class. These word associations provide an intuitive view of how language patterns differ across abuse categories. To better understand the structure of these patterns, the distribution of closed and maximal itemsets across the classes was analyzed.

Class	Number of closed itemsets	Number of maximal itemsets
Multiclass Target		
age	147	25
ethnicity	131	25
religion	63	30
gender	62	20
other_cyberbullying	8	8
Binary Target		
cyberbullying	51	23
not_cyberbullying	5	5

Table 12: Distribution of closed and maximal itemsets across multiclass and binary targets

The analysis shows that `age` and `ethnicity` are the categories with the largest number of closed itemsets (147 and 131 respectively), indicating a greater lexical richness and variability in the way users engage in bullying related to these aspects. In contrast, the `other cyberbullying` class exhibits very few patterns (only 8 closed and maximal itemsets), suggesting a lower consistency or a higher heterogeneity in the associated language. Focusing on maximal itemsets, the most representative and non-redundant patterns; `religion` emerges with the highest number (30), followed closely by `age` (25), `ethnicity` (25), and `gender` (20). This indicates that although `age` and `ethnicity` present more overall patterns, bullying related to `religion` tends to form more compact and specific linguistic structures. These findings confirm and reinforce the observations made during the multiclass classification phase, where `other cyberbullying` emerged as the most difficult class to identify, and `gender` also showed a certain degree of difficulty (albeit to a lesser extent). Conversely, classes such as `age`, `ethnicity`, and `religion` appeared more clearly separable, both from a predictive modeling perspective and from the analysis of their underlying language patterns.

When shifting the focus to the binary setting, where the goal is to distinguish between cyberbullying and non-cyberbullying texts, a different pattern emerges. The cyberbullying class yields 51 closed itemsets and 23 maximal ones, indicating a high compression rate and limited redundancy. More strikingly, the not_cyberbullying class produces only 5 closed itemsets, all of which are also maximal. This perfect overlap highlights an extremely sparse and flat pattern structure, likely due to the lexical heterogeneity and general nature of non-abusive content. In such cases, the absence of recurring co-occurrences prevents the emergence of structured itemsets, making this class inherently less amenable to pattern-based discrimination. As a result, most meaningful structures arise only from the cyberbullying class, where offensive or harmful expressions are more likely to cluster in consistent and repeated forms. This reinforces the idea that pattern mining approaches, particularly those targeting redundancy-aware or closed/maximal itemsets, are more effective when applied to abusive content, while neutral or generic language tends to resist such compression.

In a second instance, for multiclass classifier, the `feature_importances_` attribute of the Random Forest classifiers trained on TF-IDF features was analyzed. The top 50 most important terms are reported in Figure 7, showing

which words had the greatest influence on the model's overall decision-making.

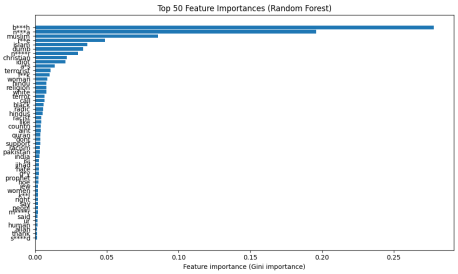


Figure 7: Top 50 most important features based on Random Forest `feature_importances_`.

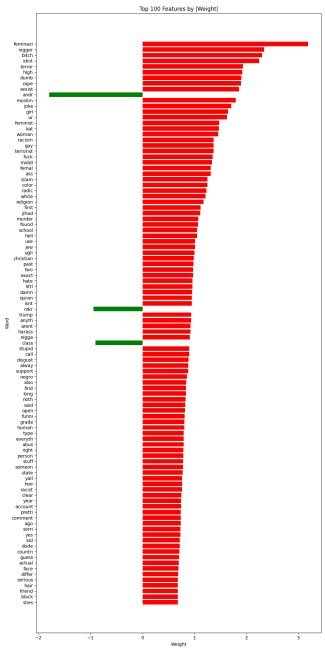


Figure 8: Top 100 most important features based on Logistic Regression `coef_` absolute values.

While, for the binary classifier, a ranking of coefficients was employed to identify the most influential features, this method provides not only interpretability but also a transparent view of the model's decision-making process.

Notably, among the top 100 features ranked by absolute coefficient magnitude, 97 are associated with the `cyberbullying` class. This further emphasizes the model's strong lexical alignment with harmful content. However, this skewed distribution is also due to the nature of `not_cyberbullying` examples, which are often recognized by the absence of aggressive terms rather than the presence of clearly indicative ones. This behavior closely reflects human judgment, which tends to flag negative content based on explicit linguistic cues and tolerate neutral or vague messages by default. This analysis further confirms the observations drawn from the frequent pattern mining. The top-ranked features by absolute coefficient magnitude are overwhelmingly aligned with the `cyberbullying` class, reinforcing the hypothesis that the model is not attempting to evenly partition the input space between the two classes. Instead, it concentrates on detecting the dense and lexically marked nature of class 1. This is consistent with the earlier finding that `cyberbullying` yielded 23 maximal itemsets, while `not_cyberbullying` generated only 5, pointing to a flatter and sparser pattern distribution.

In logistic regression, this asymmetry manifests through a clear imbalance in coefficient strength: most high-weight features support the positive class, while the negative class is characterized by numerous low-impact or weakly discriminative features. This confirms that the model captures a selective pattern

space, rich, compact, and expressive for class 1, while treating class 0 more as a heterogeneous residual category. Although the absolute value of the coefficients provides a useful proxy for importance, it is necessary to acknowledge that some features may interact or balance each other out, potentially diminishing or amplifying their individual impact. For instance, a term with a large positive weight may be counteracted by another correlated term with a negative weight, leading to more subtle decision boundaries than the magnitude alone suggests. Nonetheless, the general trend observed, namely the dominance of strongly weighted features aligned with `cyberbullying`, remains robust and interpretable, particularly when considered together with the results from pattern mining.

Local Multiclass Interpretability. To explain individual predictions produced by the multiclass classifier, the TreeInterpreter tool was employed. This method breaks down the decision of a tree-based model into the contributions of each input feature, allowing for a detailed analysis of how specific terms influenced the classification outcome.

One particularly interesting insight emerging from this analysis concerns the role of feature absence. In addition to quantifying the positive or negative impact of words that appear in the input text, TreeInterpreter also highlights how the absence of certain terms can affect the prediction. In the context of TF-IDF representations, this means that words which typically characterize other classes — but are missing from the current input — can decrease the likelihood of those classes being predicted, effectively reinforcing the assigned label.

For example, consider the tweet:

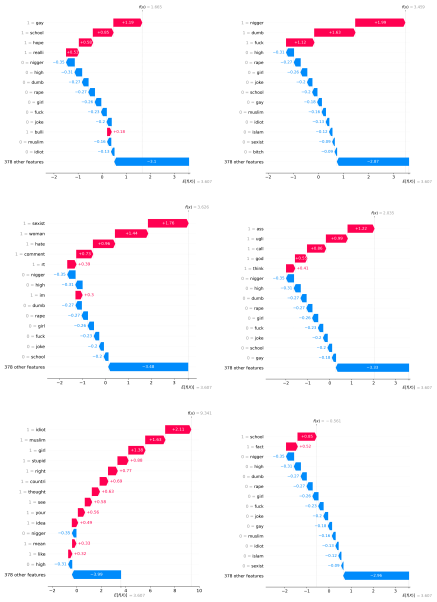
```
"I don't respect your religion, it's all nonsense."
```

TreeInterpreter reveals that terms such as `"religion"`, `"respect"`, and `"nonsense"` made strongly positive contributions toward the prediction of the `religion` cyberbullying class. These words are often associated with dismissive or mocking discourse targeting religious identity or belief systems, which are strong indicators for this category.

At the same time, the absence of features typically linked to other classes, such as gendered terms, racial or ethnic identifiers, negatively impacted their respective class probabilities. This absence, combined with the presence of highly indicative features for the `religion` class, increased the model's overall confidence in its prediction.

This kind of interpretability not only clarifies why a particular label was assigned, but also uncovers subtle linguistic dynamics that may not be immediately apparent to a human reviewer.

Local Binary Interpretability. To explain individual predictions produced by the logistic regression classifier, local explanations were generated using the SHAP framework.



SHAP (SHapley Additive exPlanations) assigns a contribution score to each feature in a specific prediction, indicating how much each term increased or decreased the model's confidence in the assigned label. Positive SHAP values indicate that a term pushes the prediction toward the cyberbullying class, while negative values support the not cyberbullying outcome. This approach enables a detailed, case-by-case understanding of the model's behavior, highlighting the specific words that most influenced each classification. By visualizing these contributions, one can intuitively grasp not only the final prediction, but also the reasoning behind it, aligning the model's behavior with human expectations and making its decisions more interpretable and trustworthy.

Figure 9: Local explanations of predictions using SHAP on a logistic regression classifier with TF-IDF features.

4 Graphical User Interface

The graphical user interface (GUI) of the *Cyberbullying Detection* system is designed to be user-friendly while providing interpretable results. It is implemented using `Tkinter` and consists of two main views: the classification interface and the explanation interface.

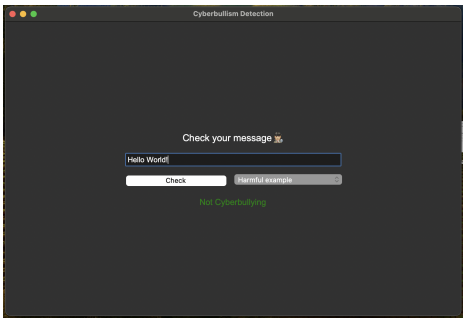


Figure 10: Main interface: the message "Hello World!" is correctly classified as safe.

In the main window (Figure 10), the user is prompted to enter a message into a text field. Upon clicking the `Check` button, the system analyzes the content using a binary classifier trained to distinguish cyberbullying from harmless messages. If the message is safe, the interface displays a green label saying *Not Cyberbullying*. Otherwise, it shows a red warning along with the specific type of cyberbullying detected (e.g., *ethnicity*, *sexual*, etc.), as illustrated in Figure 11.

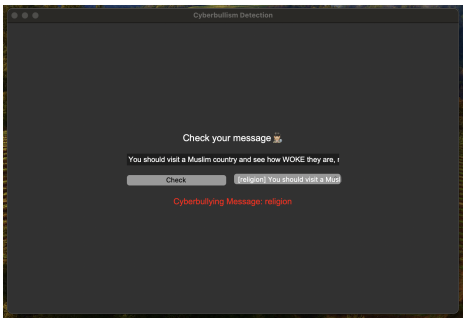


Figure 11: Example of a detected cyberbullying message classified under the 'religion' category.

To enhance interpretability, the application includes an additional explanation panel (Figure 12) that provides insights into the model's decision process. This window is divided into three sections:

- **Left panel – TreeInterpreter analysis:** displays the top 50 textual features (word stems) that contributed to the classification. Each row includes the word, its contribution score (positive or negative), and its TF-IDF value. This breakdown helps understand which terms were most influential in the prediction.
- **Top-right panel – Class word distribution:** a bar chart shows the 25 most frequent words associated with the predicted cyberbullying category.

This gives a visual representation of common vocabulary within the class.

- **Bottom-right panel – Closed/maximal Itemsets:** this section lists the most relevant itemsets mined from the dataset for the predicted class. These itemsets highlight frequent co-occurring patterns of words that characterize each type of cyberbullying. In addition to the itemsets is specified the type, closed or maximal, and the relative support of them.

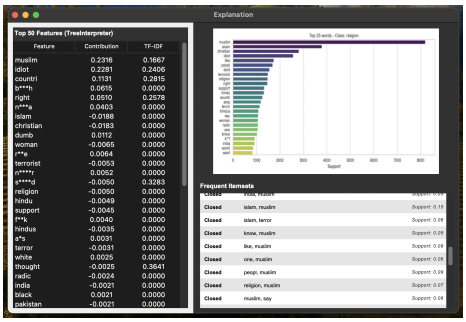


Figure 12: Explanation interface: feature contributions (left), top words by support (top-right), and association rules (bottom-right).