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.

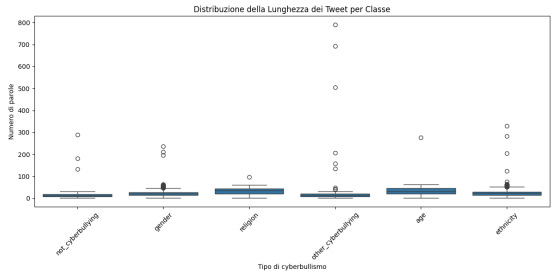


Figure 2: Distribution of tweet lengths per class.

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), identifies the most representative keywords via TF-IDF, and then constructs combined search queries. 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 a 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 — that is, those classified as `age`, `ethnicity`, `gender`, `religion`, or `other` — 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, two Word2Vec models were 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

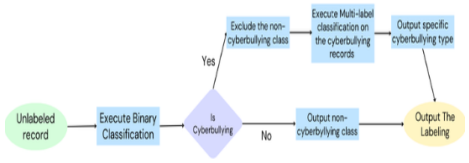


Figure 3: two stage classification pipeline[2]

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,

[2]this picture originally comes from the paper: *Self-Training for Cyberbully Detection: Achieving High Accuracy with a Balanced Multi-Class Datase*

it was correct to do so.[3] This suggests that the reported results may underestimate 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.

At this level, the dataset is affected by class imbalance, with a significantly higher number of tweets labeled as `cyberbullying`. To address this issue, synonym replacement was applied to the minority class, i.e., the `not_cyberbullying` category. This data augmentation technique generates paraphrased versions of existing samples, increasing the diversity and representativeness of the harmless class and improving the model's ability to distinguish harmful content from harmless content. Alternative approaches were also considered based on the literature. Techniques such as SMOTE, back-translation, and undersampling were evaluated. However, they were not well-suited to our specific case: SMOTE, designed for continuous numerical features, produces incoherent examples when applied to text data; back-translation, although effective on structured and longer texts, often distorts short, informal tweets; undersampling waste a large portion of data that cpould provide a more rapresentative picture of cyberbllying. For these reasons, synonym replacement was selected as the most appropriate method, allowing controlled variation of the minority class while preserving the concise and meaningful structure typical of tweets.Various data augmentation strategies were tested to achieve the best possible rebalancing of the dataset. The most effective approach, which allowed the models to generalize better, was to augment both the minority and majority classes. This prevented the introduction of linguistic biases related to the specific words used during augmentation by synonym replacment like some words used only in the augmented sample. After augmentation, an undersampling of both classes was performed to equalize the number of instances, avoiding an excessive number of overly similar samples that could introduce biases into the trained models. This strategy proved to offer the best trade-off between balancing the dataset and minimizing distortion.

Hyperparameter optimization was performed through grid search for each of the selected classifiers: Random Forest, Logistic Regression, and Linear SVM. These models were chosen based on findings from the literature, which suggest they are suitable for the cyberbullying detection task. The search was conducted using 10-fold cross-validation on the training set, with the `F1` score as

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

the evaluation metric. This choice balances the need to minimize false negatives while also reducing false positives, aiming to limit model bias.

Each classifier was tuned and evaluated using all three vectorization methods, resulting in a total of 9 models. All models were initially evaluated via k-fold cross-validation to identify the two best-performing combinations.

These top two models were then assessed on the test set using multiple evaluation metrics: F1 score, precision, recall, accuracy (including balanced accuracy), ROC-AUC, PR-AUC, and a detailed analysis of the contingency table. Statistical significance between their performances was verified using the McNemar test.

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 dataset in this phase is already balanced across classes, as this is its original purpose according to the reference paper.

The hyperparameter tuning approach follows the same procedure adopted in the binary classification stage. Each of the three classifiers (Random Forest, Logistic Regression, and Linear SVM) was trained using all three vectorization methods, resulting in a total of 9 models. A 10-fold cross-validation was performed on the training set, using `accuracy` as the scoring metric.

Following this initial evaluation, a second cross-validation step was carried out to select the two best-performing models. These were then assessed on the test set using a broader set of evaluation metrics: balanced accuracy, macro-averaged accuracy, macro F1-score, macro precision, macro recall, and contingency table. The McNemar test was also employed to verify whether performance differences between the models were statistically significant.

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. Notably, since the best-performing model was a Random Forest trained on TF-IDF features, both global and local interpretability techniques could be effectively applied. Random Forest inherently supports feature importance analysis, while its tree-based structure allows for instance-level decomposition of predictions—making it an ideal candidate for explainability.

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, we chose to work with maximal and closed itemsets. 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, the `feature_importance_` attribute of the Random Forest model was used to rank features by their global relevance, 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 instance-level interpretation, the TreeInterpreter tool was employed. This method decomposes the output of the Random Forest model 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 vectorization, 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 training set was augmented using synonym replacement to address class imbalance, and each model was optimized through grid search with 10-fold cross-validation. The `F1-score` was used as the selection metric to prioritize a balance

between precision and recall.

The grid of hyperparameters used during tuning was the following:

- **Logistic Regression:**
  - `C`: {0.01, 1, 10}
  - `class_weight`: "balanced"
- **Linear SVM:**
  - `C`: {0.01, 0.1, 1, 10}
- **Random Forest:**
  - `n_estimators`: {100, 200, 500, 1000}
  - `max_depth`: {None, 10, 20}
  - `class_weight`: "balanced"
  - `random_state`: 42

Model Selection. Table 1 reports the performance of each of the 9 models evaluated via cross-validation. The best performing configuration was **Random Forest with BoW**, followed closely by **Random Forest with TF-IDF**, both of which were selected for final testing.

Model	Vectorizer	Accuracy	Precision_1	Recall_1	F1_1	Precision_0	Recall_0	F1_0
RandomForest	**BoW**	**0.882600**	**0.943363**	**0.81410**	**0.873949**	**0.836544**	**0.95110**	**0.890135**
RandomForest	**TF-IDF**	0.879675	0.942437	0.80875	0.870462	0.832564	0.95060	0.887659
RandomForest	W2V-1	0.846425	0.894772	0.78520	0.836401	0.808653	0.90765	0.855289
LogisticRegression	BoW	0.840075	0.897304	0.76810	0.827658	0.797318	0.91205	0.850815
LinearSVM	BoW	0.840850	0.909955	0.75660	0.826194	0.791730	0.92510	0.853220
LogisticRegression	TF-IDF	0.833600	0.871115	0.78315	0.824760	0.803047	0.88405	0.841581
LinearSVM	TF-IDF	0.834100	0.879163	0.77475	0.823631	0.798672	0.89345	0.843387
LogisticRegression	W2V-1	0.764425	0.756285	0.78030	0.768064	0.773224	0.74855	0.760642
LinearSVM	W2V-1	0.768750	0.770624	0.76535	0.767926	0.767040	0.77215	0.769537

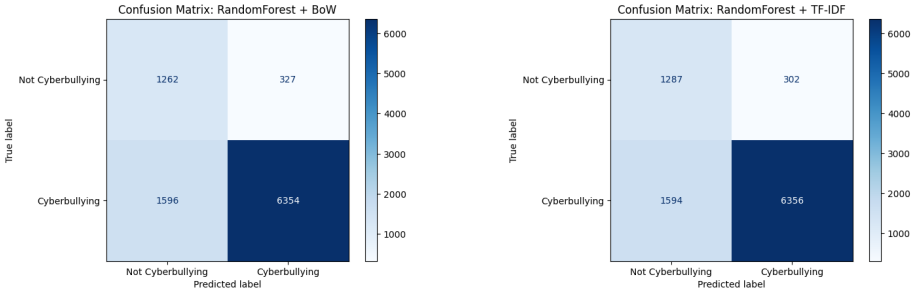
Table 1: Cross-validation results for binary classification. Best values in each column are bolded.

The two selected models (Random Forest combined with BoW and with TF-IDF) achieved the highest `F1-score` among all configurations. In addition, they consistently ranked among the top positions in terms of accuracy, precision, and recall. This overall strong performance across multiple metrics justified their selection for final evaluation on the test set.

Test Set Performance. The two selected models: **Random Forest + BoW** and **Random Forest + TF-IDF** were evaluated on the test set. Table 2 summarizes the key performance metrics, while Figures 4a and 4b show the corresponding confusion matrices.

Metric	Random Forest + BoW	Random Forest + TF-IDF
Accuracy	0.7984	**0.8012**
Balanced Accuracy	0.7967	**0.8047**
Precision	0.9511	**0.9546**
Recall (Sensitivity)	0.7992	**0.7995**
Specificity	0.7942	**0.8099**
F1 Score	0.8606	**0.8702**
ROC-AUC	0.8804	**0.8865**
PR-AUC	0.9755	**0.9771**

Table 2: Test set performance of the top two models. Best values in bold.



(a) Random Forest + BoW

(b) Random Forest + TF-IDF

Figure 4: Confusion matrices of the two selected models on the test set.

Although the McNemar test did not highlight a statistically significant difference between the two models ($p = 0.2289$), **Random Forest + TF-IDF** was selected because it exhibited better and balanced performance across multiple key metrics, particularly on the imbalanced test set. Specifically, despite only slight differences, the TF-IDF-based model achieved slightly higher F1-score and Balanced Accuracy, indicating greater reliability in cyberbullying detection, while maintaining appropriate caution, as also evidenced by its higher specificity, meaning a lower rate of false positives.

3.2 Multiclass Classification Results

Model Selection. As described in the methodology, a total of 9 models were trained and evaluated using 10-fold cross-validation, combining three classifiers (Random Forest, Logistic Regression, Linear SVM) with three vectorization methods (BoW, TF-IDF, W2V-1). The main selection metric was `accuracy`, while `precision`, `recall`, and `F1-score` were also considered to ensure robustness across evaluation perspectives. In addition, the standard deviation of accuracy was computed across the folds to assess model stability.

The grid search explored the following hyperparameter configurations:

- **Logistic Regression:**
  - `C`: {0.01, 1, 10}
- **Linear SVM:**
  - `C`: {0.01, 0.1, 1, 10}
- **Random Forest:**
  - `n_estimators`: {100, 200, 500, 1000}
  - `max_depth`: {None, 10, 20}
  - `random_state`: 42

Table 3 reports the average cross-validation results for all configurations. The two best-performing models were **Random Forest + TF-IDF** and **Logistic Regression + BoW**, which achieved the highest accuracy and also ranked among the top three models across all other metrics. These two configurations were selected for final testing.

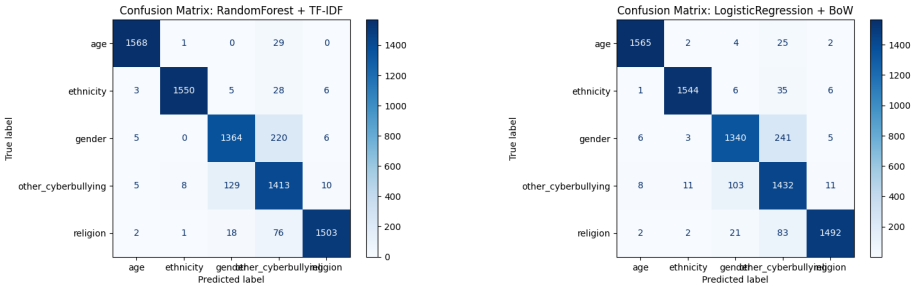
Model	Vectorizer	Accuracy	Precision	Recall	F1	Accuracy Std
RandomForest	**TF-IDF**	**0.931818**	**0.933303**	**0.931313**	**0.931733**	**0.004533**
LogisticRegression	**BoW**	0.928463	0.931737	0.928180	0.928653	0.005186
LinearSVM	BoW	0.927640	0.931363	0.927390	0.927826	0.005220
LogisticRegression	TF-IDF	0.927608	0.930571	0.927274	0.927727	0.005960
LinearSVM	TF-IDF	0.926912	0.929605	0.926560	0.926956	0.005216
RandomForest	BoW	0.926722	0.927857	0.926100	0.926560	0.005633
RandomForest	W2V-1	0.891776	0.897811	0.891407	0.891766	0.006047
LinearSVM	W2V-1	0.878450	0.877053	0.877139	0.876625	0.004443
LogisticRegression	W2V-1	0.877785	0.877308	0.876570	0.876603	0.004445

Table 3: Cross-validation results for multiclass classification. Best value in each column is bolded.

Test Set Performance. The two best-performing configurations from the cross-validation stage are **Random Forest + TF-IDF** and **Logistic Regression + BoW**. Them were evaluated on the test set. Their results are summarized in Table 4, while Figures 5a and 5b show the corresponding confusion matrices.

Metric	Random Forest + TF-IDF	Logistic Regression + BoW
Accuracy	**0.9306**	0.9274
Balanced Accuracy	**0.9305**	0.9274
Macro Precision	**0.9339**	0.9319
Macro Recall	**0.9305**	0.9274
Macro F1 Score	**0.9313**	0.9283

Table 4: Test set performance for multiclass classification.



(a) Random Forest + TF-IDF (b) Logistic Regression + BoW

Figure 5: Confusion matrices for the two selected models on the multiclass test set.

A McNemar test was also conducted to compare the predictions of the two models. The test yielded a statistic of 123.0 with a p-value of 0.1447, indicating that the performance difference is not statistically significant.

For this reason, the final model selection could not rely exclusively on test accuracy or classification metrics. Instead, the choice of the **Random Forest + TF-IDF** configuration was driven primarily by considerations of model transparency and explainability, which were defined as core requirements of the system from the early design phase.

In particular, Random Forest provides native support for both global and local interpretability: it exposes feature importance scores through the `feature_importances_` attribute, and it is fully compatible with `TreeInterpreter` for per-instance explanation. The TF-IDF vectorization further enhances this compatibility, producing a normalized and meaningful feature space where the influence of each word can be assessed individually.

In contrast, while Logistic Regression offers global interpretability through linear coefficients (`coef_`), these are only reliable when features are standardized that is a condition not guaranteed with BoW. Moreover, local interpretability for Logistic Regression typically requires model-agnostic tools like LIME or SHAP. Finally, as a linear model, Logistic Regression lacks the capacity to capture non-linear feature interactions, limiting its expressiveness in more complex linguistic scenarios.

For these reasons, the Random Forest + TF-IDF model was selected as the final classifier for the multiclass stage, as it best aligned with both the interpretability goals and the architectural consistency of the entire pipeline.

Observations. Both models performed extremely well across all metrics, with overall accuracy and macro-averaged scores above 0.92.

From the confusion matrices, we 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 con-

sistent with observations reported in previous work[4], where misclassifications between gender and other categories accounted for the majority of classification errors. Nevertheless, class-wise performance remains robust, and no class shows signs of significant bias or neglect.

The results confirm that the combination of TF-IDF features with a tree-based classifier provides a reliable and interpretable solution for fine-grained cyberbullying classification.

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 severity or nature. Metrics were computed separately for each stage, and also globally to assess the end-to-end system.

Binary Stage Performance.

- **Accuracy:** 0.80
- **Precision (cyberbullying):** 0.95
- **Recall (cyberbullying):** 0.80
- **F1-score (cyberbullying):** 0.87

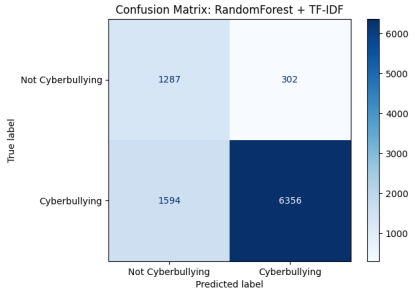


Figure 6: Confusion matrix for binary stage.

[4]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.

Multiclass Stage Performance (on true positives).

- **Accuracy:** 0.97
- **Macro Precision:** 0.94
- **Macro Recall:** 0.94
- **Macro F1-score:** 0.94
- **Weighted F1-score:** 0.97

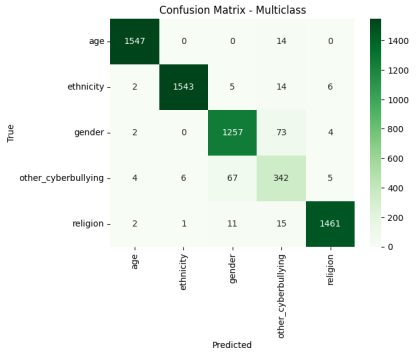


Figure 7: Confusion matrix for multiclass stage (on binary true positives).

Overall Pipeline Performance. By combining the two stages, the overall pipeline correctly classified 7437 tweets out of 9539, resulting in an **end-to-end accuracy of** 0.7796. 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.

Observation. 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 424 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.

Label	**Precision**	**Recall**	**F1-score**	**Support**
age	0.99	0.99	0.99	1561
ethnicity	1.00	0.98	0.99	1570
gender	0.94	0.94	0.94	1336
other_cyberbullying	0.75	0.81	0.78	424
religion	0.99	0.98	0.99	1490

Table 5: 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. These provide insight into the reasoning of the classifier, helping developers and stakeholders understand what features drive predictions and how individual decisions are made.

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
Age	147	25
Ethnicity	131	25
Religion	63	30
Gender	62	20
Other cyberbullying	8	8

Table 6: Distribution of closed and maximal itemsets across classes

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.

In a second instance , the `feature_importances_` attribute of the Random Forest classifier trained on TF-IDF features was analyzed. The top 50 most important terms are reported in Figure 8, showing which words had the greatest influence on the model's overall decision-making.

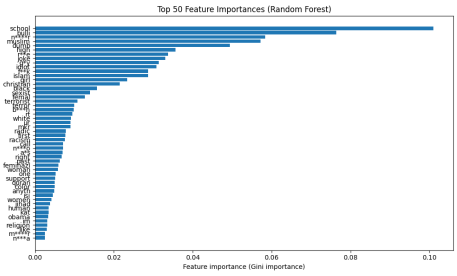


Figure 8: Top 50 most important features based on Random Forest `feature_importances_`.

Local 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:

```
"you're just a stupid old man, no one cares about boomers
like you."
```

TreeInterpreter reveals that words such as `"old"`, `"man"`, and `"boomers"` made strongly positive contributions toward the prediction of the `age` cyberbullying class. At the same time, the absence of terms typically linked to other categories — such as `"religion"`, `"race"`, or `"women"` — contributed negatively to their respective classes, indirectly increasing the confidence in the `age` label.

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.

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.

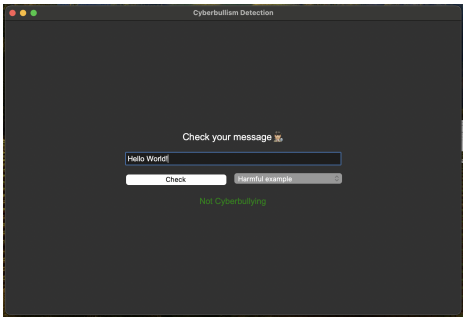


Figure 9: Main interface: the message "Hello World!" is correctly classified as safe.

In the main window (Figure 9), 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 10.

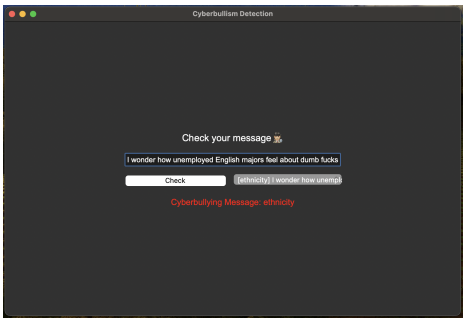


Figure 10: Example of a detected cyberbullying message classified under the "ethnicity" category.

To enhance interpretability, the application includes an additional explanation panel (Figure 11) 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.

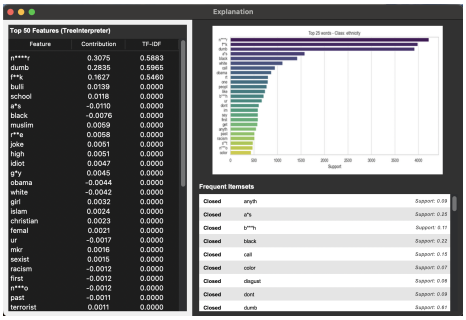


Figure 11: Explanation interface: feature contributions (left), top words by support (top-right), and association rules (bottom-right).