AI Course

Capstone Project
Final Report

Together for Tomorrow!
Enabling People
Education for Future Generations

NLP CyberGuard: Natural Language Processing for Cyberbullying Detection.

09/12/2025

NLP CyberGuard

Antonio Moreno Rodriguez
Cristina Galán Berenguel

Content

1. Introduction

1.1. Background Information

The core focus of this project is the development of an Artificial Intelligence (AI) model specifically engineered for the detection and classification of cyberbullying within social media interactions. Our primary objective is to construct a robust and accurate system capable of distinguishing harmful content from neutral discourse in textual data, leveraging Natural Language Processing (NLP) techniques.

To facilitate this, we curated a comprehensive dataset of approximately 97,000 data points by merging two distinct datasets sourced from Kaggle. This integration created a rich number of diverse examples and detailed labels, ensuring a solid, broad foundation for the subsequent training and rigorous validation of our predictive algorithms.

The resulting classification model aims to provide a scalable solution for online safety, featuring potential adaptability to various other social media platforms. By automating content moderation, this approach offers critical support for early detection of abusive behavior, thereby contributing to the maintenance of safer and more inclusive digital environments for all users.

1.2. Motivation and Objective

The escalating prevalence of toxic behavior in online environments presents a significant societal challenge, directly impacting user mental health and overall digital well-being. Our project was motivated by the need to develop a system that moves beyond basic keyword filtering to achieve a genuine contextual understanding of harassment, ensuring more effective and nuanced user protection.

Cyberbullying Detection was selected due to its considerable social impact and the sophisticated technical challenges it poses within the field of NLP. We strategically framed the core problem as a binary classification task (Harmful vs. Neutral) to maximize the model's precision in identifying toxic content, regardless of the specific nature of the abuse (e.g., related to religion, gender, or age).

Recognizing the inherent limitations of relying on a single, potentially unbalanced source, we elected to curate a unified, class-balanced corpus by integrating toxic samples with a large volume of neutral texts. This methodological decision ensures the model is equally proficient in identifying safe interactions as it is in detecting harmful ones.

The primary objective is twofold: to execute a thorough comparative analysis of various Artificial Intelligence models (as detailed in the subsequent sections) and, subsequently, to implement a fully functional, demonstrable application. This tool allows users to input text and receive a clear, real-time prediction as to whether the message constitutes cyberbullying.

1.3. Members and Role Assignment

The project team, consisting of Antonio, Cristina, and Daniel, formed during the Artificial Intelligence EOI course, where we acquired and practiced foundational skills in machine learning and neural networks.

We intentionally opted not to establish predefined, siloed roles. Instead, we adopted a fully collaborative, peer-to-peer environment. All members actively participated in every stage of the project lifecycle, from data acquisition and preprocessing to model training, experimentation, and final documentation. This comprehensive approach allowed every team member to gain hands-on experience with the entirety of the NLP development pipeline.

1.4. Schedule and Milestones

The project officially launched on November 18th, initiating the planning phase using a bottom-up approach. While the curriculum was structured around 20 hours of supervised tutoring sessions, the total project effort extended significantly beyond this timeframe. The team invested extensive hours in independent work and coding between these guided sessions.The initial time allocation was as follows:

- **Session 1**: **Project Initiation.** Defined the project scope, executed dataset acquisition, and performed comprehensive Exploratory Data Analysis (EDA).
- **Session 2**: **Data Preprocessing and Curation.** Focused on merging the datasets to achieve class balance, detailed text cleaning (using regular expressions and handling emojis), and preparing stratified data splits for training and testing.
- **Session 3: Model Development and Training**. Dedicated to the implementation and benchmarking of eleven distinct algorithms spanning Classical Machine Learning, Deep Learning, and Transformer families. This phase focused on fine-tuning advanced architectures (e.g., RoBERTa, BERTweet) and evaluating them against robust baselines to ensure a comprehensive performance analysis.
- **Session 4**: **Integration and Documentation.** Completed the final application integration (building the main menu) and performed the final analysis of performance metrics and formal documentation.

Anticipating that data balancing and cleaning might require significant effort, this schedule was deliberately designed to serve as a flexible framework rather than a rigid, fixed timeline.

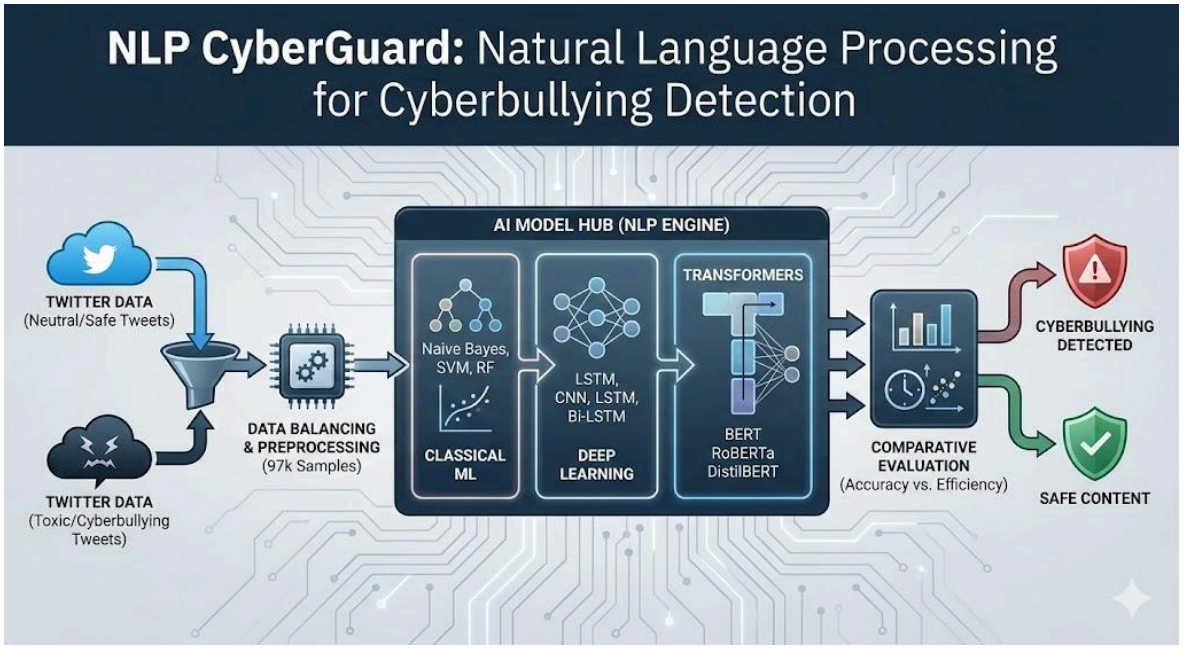

Figure 1. Summary workflow of the project.

2. Project Execution

2.1 Data Acquisition

To develop a robust model capable of detecting online harassment, the project relied on publicly available datasets containing real user-generated content from Twitter. Data sources were selected from the Kaggle platform due to their accessibility, large scale, and relevance to the field of Natural Language Processing (NLP).

The primary dataset, "Cyberbullying Classification" , provided the initial foundation. It includes thousands of posts categorized into specific forms of harassment (gender-based, religion-based, age-based, ethnicity-based) alongside a non-cyberbullying class. While this offered a valuable basis for identifying abusive linguistic patterns, initial inspection revealed critical limitations: the dataset was highly imbalanced (approximately 84% bullying vs. 16% non-bullying) and contained significant noise, including non-textual elements (emojis, URLs) and mixed languages.

To address the class imbalance and improve model generalization, we incorporated a second corpus, "Sentiment140". Specifically, we extracted 50,000 samples annotated with positive sentiment to serve as a control set of non-offensive, neutral communication. The integration of both datasets resulted in a unified, balanced collection of approximately 97,000 samples. This merger established a representative ground truth for the subsequent preprocessing and training stages.

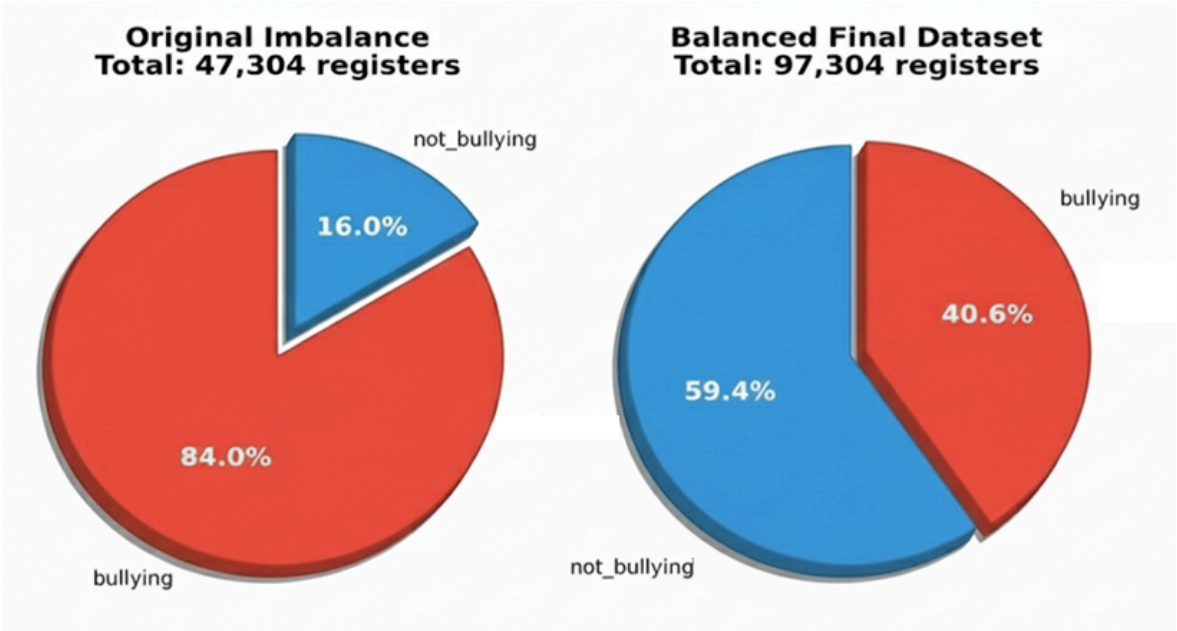

Figure 2. Data distribution before and after integrating the Sentiment140 corpus.

2.2. Training Methodology

To ensure a comprehensive evaluation, the project adopted a multi-tiered comparative methodology. We implemented a diverse range of eleven algorithms categorized into three levels of architectural complexity. This stratified approach allowed us to analyze the trade-off between computational efficiency and predictive power.

- **Classical Machine Learning:** We established performance baselines using traditional algorithms including Naive Bayes, Logistic Regression, SVM, Random Forest, XGBoost, and LightGBM. These models were selected for their interpretability and speed.
- **Deep Learning**: To capture sequential dependencies and local patterns within the text, we utilized neural network architectures: LSTM, Bi-LSTM, and CNNs.
- **Transformer-Based Models:** Representing the state-of-the-art in NLP, we fine-tuned pre-trained Large Language Models (LLMs), specifically DistilBERT, RoBERTa, and BERTweet, to leverage their superior contextual understanding.

The training strategy focused on maximizing the F1-Score while monitoring computational cost. The system was designed to execute these models either individually for granular analysis or in a batch process for holistic benchmarking.

2.3. Workflow

The project's execution workflow is orchestrated through a central control script (main.py) that manages the pipeline from data input to result visualization.

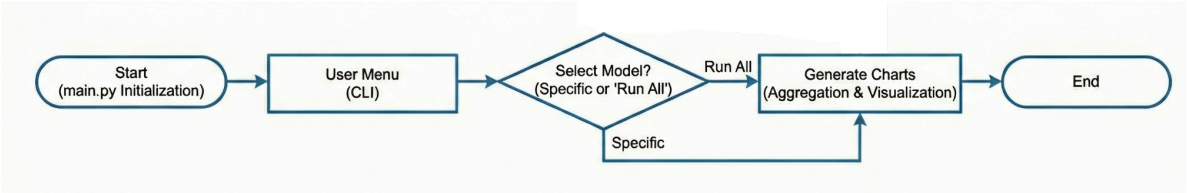

Figure 3. Pipeline of the control logic from user input to the final generation and visualizations.

The process follows a structured sequence:

1. **Initialization:** The environment is prepared by suppressing unnecessary framework logs (TensorFlow/oneDNN) to ensure clean output.
2. **Model Selection:** A command-line interface (CLI) presents the user with a menu to select specific models or trigger a "Run All" comparative mode.
3. **Execution Pipeline:**
    - If a specific model is selected, the system calls the corresponding pipeline function (e.g., run_distilbert or run_classical), processes the dataset, and saves the specific results.
    - In comparative mode, the system iterates through all 12 defined tasks. It includes a fault-tolerance mechanism that checks for existing results, allowing

the process to resume from the last successful checkpoint in case of interruption.

4. **Result Aggregation and Visualization:** Upon completion, the system aggregates metrics into a CSV file (resultados_comparativos.csv) and automatically generates bar charts comparing Accuracy and Execution Time, storing them in a dedicated output directory.

2.4. System Diagram

The system architecture follows modular design principles, decoupling the main execution logic from the specific implementations of each model family. This separation of concerns facilitates maintenance and scalability.

- **Input Layer:** Handles the ingestion of the processed dataset(tweets_trad.csv).
- **Orchestration Layer** (main.py): Acts as the system controller. It is responsible for handling user input, managing the execution flow, and error handling.
- **Processing Layer:**Contains the isolated logic for model training and inference, divided into specialized modules:
  - *Transformers Module:* Handles RoBERTa and BERTweet.
  - *DistilBERT Module:* Dedicated pipeline for the distilled architecture.
  - *Deep Learning Module:* Manages PyTorch implementations (LSTM, Bi-LSTM, CNN).
  - *Classical Module:* Contains Scikit-learn and Boosting implementations.
- **Output Layer:** Generates the final artifacts, including the comparative CSV report and performance graphs.

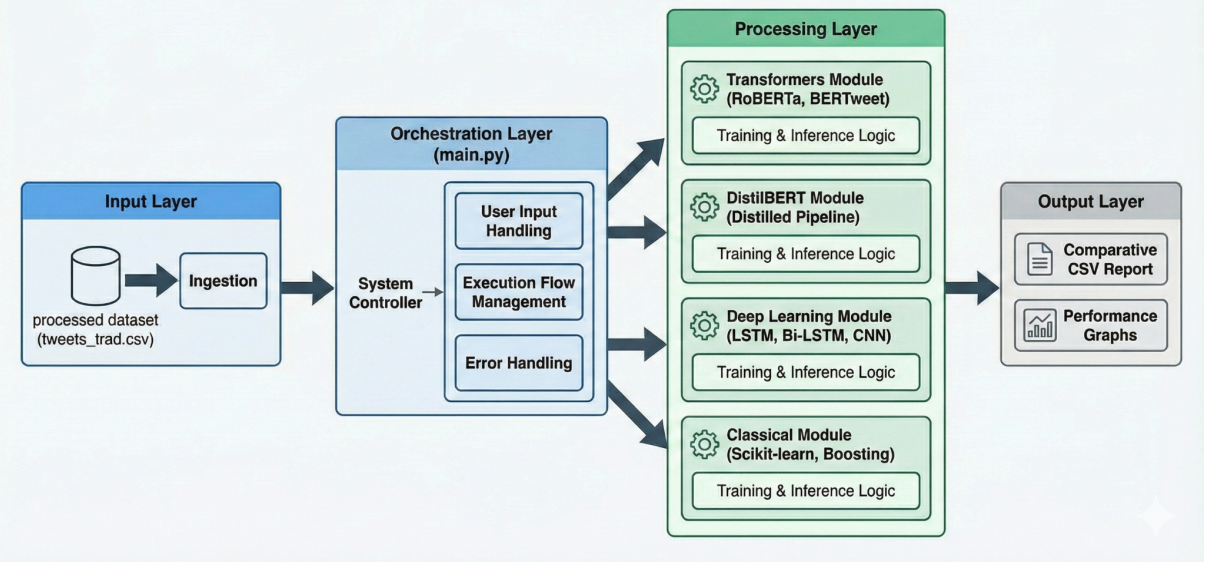

Figure 4. High-Level System Architecture. Data flow and control logic of the different layers on the project.

3. Results

3.1. Data Preprocessing

Prior to model training, extensive preprocessing was conducted to ensure textual consistency, eliminate noise, and standardize the linguistic characteristics of the combined corpus. Since the data originated from social media, it contained substantial variability in syntax, abbreviations, and informal expressions. The following steps were applied systematically to prepare the dataset for analysis:

- **Text Normalization and Noise Removal:** All messages were converted to lowercase to prevent redundant token distinctions. Irrelevant metadata such as URLs, user mentions, and hashtags were removed, alongside numerical values and special punctuation, to isolate purely linguistic features.
- **Semantic Handling:** To preserve sentiment, emojis were converted into their textual descriptors (e.g., 😠 → "angry face"), and English contractions were expanded (e.g., "don't" → "do not")**.**
- **Translation & Standardization:** Non-English entries were automatically translated using MarianMT, ensuring linguistic homogeneity across the dataset. Finally, NLTK-based lemmatization reduced words to their base forms (e.g., "running" → "run"), and non-informative stopwords were filtered out.
- **Feature Representation:** To convert text into machine-readable formats, two distinct approaches were employed:
    - **TF-IDF Vectorization:** Used for Classical models, assigning weights based on word distinctiveness. Vocabulary was capped at 10,000 features for linear models and 5,000 for tree-based ensembles to optimize dimensionality.
    - **Token Embeddings:** Deep Learning and Transformer models utilized dense vector embeddings. Specifically, Transformers leveraged pre-trained tokenizers (Hugging Face) to encode bidirectional contextual relationships.
- **Dataset Label Encoding:**
    - text — the preprocessed tweet content, and
    - label — a binary indicator, where 1 denotes cyberbullying and 0 denotes non-bullying.
- **Balancing and Validation:**

The final dataset consisted of approximately 97,000 balanced samples (40% cyberbullying, 60% non-cyberbullying), free of duplicates and noise, providing a reliable foundation for training.

3.2. Exploratory Data Analysis (EDA)

Exploratory Data Analysis is a critical step where we analyze the dataset to uncover patterns, correlations and potential issues. This phase helps us gain a deeper understanding of the data's structure and quality guiding subsequent preprocessing and modeling

decisions. Through visualizations and summary statistics, we identify trends, detect outliers and ensure the dataset aligns with the project's goals.

Before modeling, we analyzed the dataset to verify the distinguishability of classes. A key qualitative assessment involved generating **comparative Word Clouds.**

Comparative Word Clouds - Bullying vs. Non-Bullying

Figure 5. Comparative Word Clouds illustrating the semantic difference between toxic and neutral tweets.

As observed in *Figure 5*, the distinction is evident. The "Cyberbullying" cloud is dominated by aggressive terminology, profanity, and derogatory slurs targeting specific demographics. In stark contrast, the "Non-Cyberbullying" cloud features neutral, conversational terms such as "school," "day," "love," and "time." This semantic separation confirms that the dataset contains strong linguistic signals, validating its suitability for supervised learning models.

3.2.1 Data Partitioning

The final dataset, consisting of approximately 97,000 balanced samples (40% cyberbullying, 60% non-cyberbullying), was split into training (80%) and testing (20%) subsets using stratified sampling to preserve class distribution, preventing bias during evaluation.

Although the primary objective of this project was to develop a binary classification model (determining the presence or absence of cyberbullying), an analysis of the original dataset reveals a granular taxonomy of harassment types. As illustrated in the top figures, the 'Bullying' class is composed of five distinct sub-categories: Religion, Age, Gender, Ethnicity, and Other Cyberbullying.

The data exhibits a remarkably uniform distribution across these categories, with each type representing approximately 20% of the total bullying instances (ranging from 7,823 to 7,998 registers per category). This balance is statistically significant; unlike the overall imbalance between bullying and non-bullying content observed earlier, the internal composition of the toxic data is evenly distributed. This uniformity suggests that while the current system treats all toxicity as a single class, the dataset is perfectly primed for future scalability into multi-class classification, where specific types of discrimination could be individually identified without facing class imbalance issues.

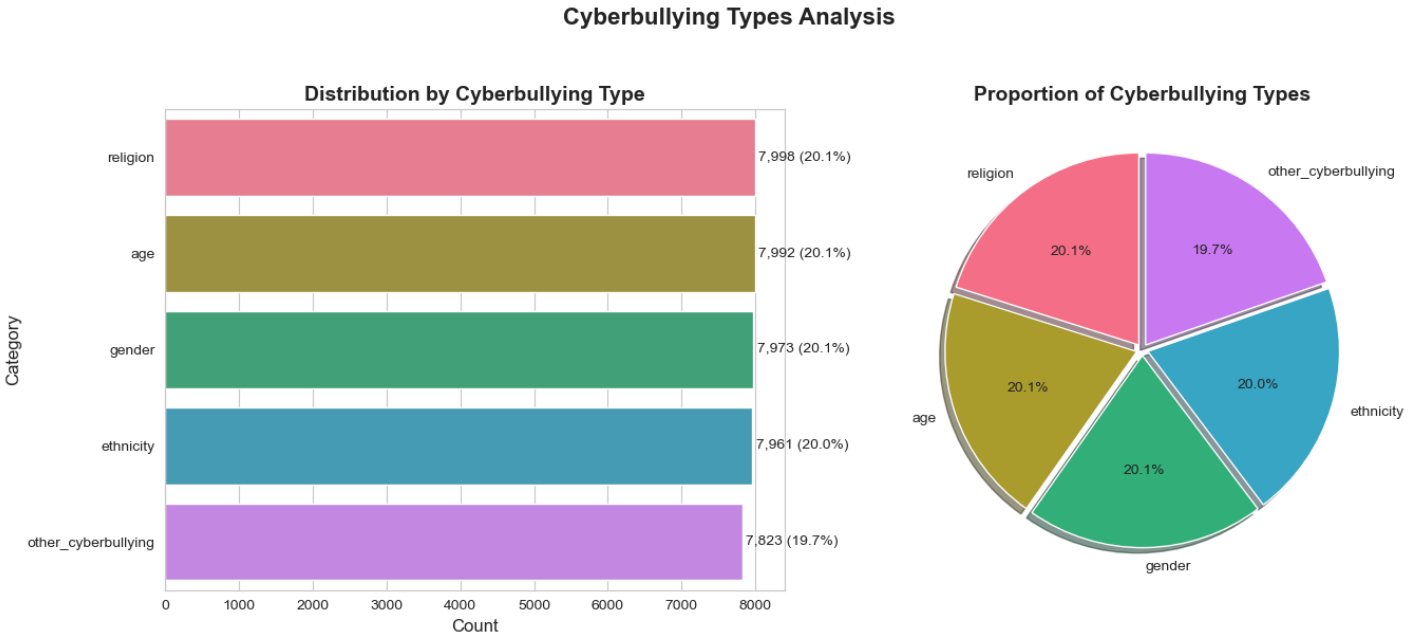

Figure 6. Distribution by cyberbullying type of the original dataset. The graphs shows the significant balance of the data.

3.3. Modeling

We executed a comprehensive benchmark of 12 algorithms, ranging from classical Machine Learning to state-of-the-art Transformers. The primary goal was to identify the model that offers the best balance between predictive power and computational efficiency.

3.3.1 Model Selection

A total of eleven classification models were implemented and compared, grouped into three methodological families. The goal was to assess their performance in classifying the distinct types of cyberbullying.

1. Classical Machine Learning Models

- **Naive Bayes (MultinomialNB):** This is a **fast probabilistic model** suitable for text classification. It operates based on **word frequencies (TF-IDF)** and applies the Bayes theorem under the assumption that features (words) are independent. It is often used as a baseline due to its simplicity and efficiency.
- **Logistic Regression:** A **linear model** that uses the sigmoid function to estimate the probability of a data point belonging to a certain class. It is robust for binary tasks and optimizes decision boundaries through gradient descent, making it effective when **linear separation** of features is possible in the high-dimensional TF-IDF space.
- **Support Vector Machine (SVM):** A **powerful linear classifier** designed to maximize the margin between classes in the high-dimensional TF-IDF space. SVM is highly effective in text classification, especially when dealing with a large number of features, by using the **kernel trick** to implicitly map inputs into higher-dimensional feature spaces.
- **Random Forest:** An **ensemble of decision trees** that uses **majority voting** to produce the final classification. By aggregating the predictions of multiple diverse trees, it significantly reduces the risk of **overfitting** compared to single-tree models, enhancing generalization.
- **XGBoost (Extreme Gradient Boosting) and LightGBM (Light Gradient Boosting Machine):** These are advanced **Gradient Boosting algorithms**. They iteratively optimize classification performance by **correcting the errors** of previous weak prediction models (typically decision trees). LightGBM is generally faster and uses less memory, while both are known for high accuracy and robust handling of non-linear relationships.

2. Deep Learning Models

- **LSTM (Long Short-Term Memory):** A type of **Recurrent Neural Network (RNN)**, this is a **sequential model** specifically designed to overcome the vanishing gradient problem. It excels at capturing **long-range temporal dependencies and context** across words, making it effective for understanding how early words in a post influence the meaning of later words.
- **Bi-LSTM (Bidirectional LSTM):** This model enhances the standard LSTM by reading the text sequence **in both directions (forward and backward)**. This provides a richer and more complete **contextual comprehension** of phrases, negations, and dependencies, crucial for nuanced sentiment and intent analysis like identifying sarcasm or implied threats.
- **CNN (Convolutional Neural Network):** In text tasks, CNNs use **convolutional filters** applied to word embeddings to extract **local linguistic patterns** (analogous to n-grams). This approach is highly effective at identifying key phrases, sequences, or specific word combinations that are strong indicators of a cyberbullying type.

3. Transformer-Based Models

- **DistilBERT:** A **distilled version of BERT** (reducing the number of layers), this model is optimized for **efficiency and speed** while maintaining a high degree of the original

model's contextual understanding. It provides a faster inference time with minimal performance degradation, ideal for resource-constrained environments.

- **RoBERTa (Robustly optimized BERT approach):** This variant of BERT features an **improved pretraining strategy** on a significantly larger and more diverse corpus. It uses a dynamic masking approach and eliminates the Next Sentence Prediction (NSP) task, resulting in **better general-purpose language understanding** and superior performance on downstream tasks.
- **BERTweet:** A BERT-based model **pre-trained specifically on 850 million tweets**. Its specialized training data allows it to offer a much better understanding and handling of **Twitter-specific linguistic features** such as hashtags, user mentions, emojis, and highly informal language, which are critical components of social media cyberbullying detection.

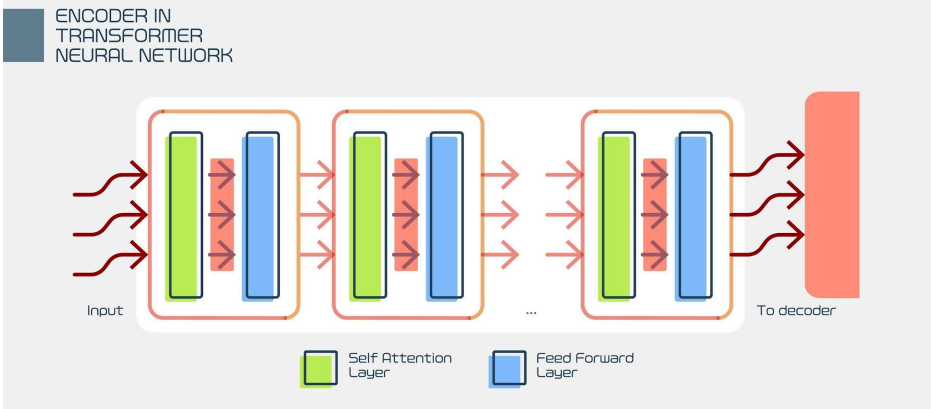

Figure 7. Internal Structure of the Transformer Encoder. The figure details the sequential data flow through the encoding layers. This architecture enables the advanced feature extraction and context understanding required for the fine-tuning of the BERT-based models.

1. **Classical Machine Learning:** We established baselines using Naive Bayes (probabilistic), Logistic Regression and SVM (linear classifiers), and ensemble methods like Random Forest, XGBoost, and LightGBM. These were selected for their interpretability and efficiency.
1. **Deep Learning:** We utilized LSTM and Bi-LSTM to capture sequential dependencies, and CNNs to extract local linguistic patterns (n-grams).
2. **Transformers:** For state-of-the-art context understanding, we fine-tuned DistilBERT, RoBERTa, and BERTweet (pre-trained specifically on 850M tweets).

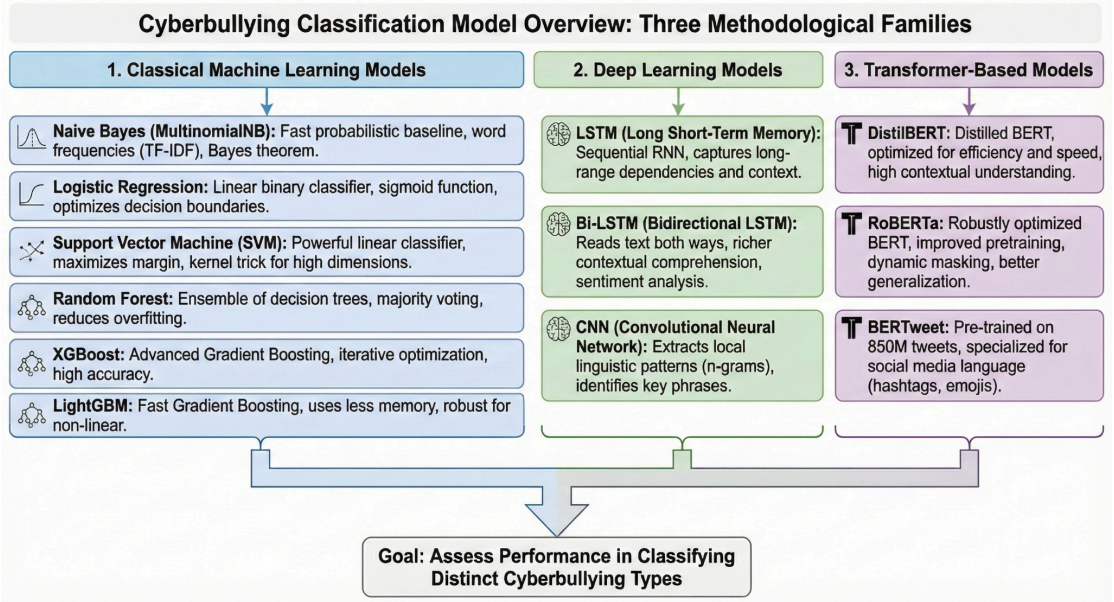

Figure 8. Comparative Modeling Strategy. The twelve selected algorithms were grouped into three distinct architectural categories (Classical, Deep Learning, and Transformers) to execute a rigorous, multi-tiered performance benchmark against the cyberbullying detection task.

Each model was trained independently using the same training/testing partitions and evaluated using identical performance metrics to ensure a fair and methodologically rigorous comparison.

3.3.2. Training Environment

Training was executed in a standardized **Python (v3.10)** environment. To handle the computational demands of Deep Learning and Transformer models, training was accelerated using an **NVIDIA GeForce RTX 3060 GPU** via the **CUDA backend.** This hardware acceleration was critical for feasibility, while classical models were efficiently executed on the CPU.

This choice of hardware was critical, as it significantly reduced the overall training times for the complex neural architectures and ensured the feasibility of iterative hyperparameter tuning, which is often computationally prohibitive without dedicated GPU resources.

3.3.3 Evaluation Metrics

Models were evaluated using standard metrics (Accuracy, Precision, Recall, F1-Score) and execution time. The summary results are presented below:

Table 1. Performance metrics sorted by Accuracy in different algorithms.

Modelo	Accuracy	Precision	Recall	F1-Score	AUC-ROC	Tiempo (s)

DistilBERT	0.929	0.925	0.898	0.911	0.983	94.622
RoBERTa	0.915	0.904	0.885	0.894	0.968	89.455
BERTweet	0.932	0.905	0.930	0.918	0.985	97.205
LSTM	0.915	0.918	0.868	0.892	0.977	2.778
Bi-LSTM	0.916	0.896	0.898	0.897	0.977	3.292
CNN	0.910	0.899	0.877	0.888	0.972	2.210
Naive Bayes	0.891	0.876	0.851	0.863	0.955	1.661
Logistic Regression	0.906	0.927	0.833	0.878	0.963	1.652
SVM	0.907	0.928	0.836	0.879	0.955	106.515
Random Forest	0.867	0.987	0.682	0.807	0.937	1.808
XGBoost	0.904	0.944	0.811	0.873	0.953	1.700
LightGBM	0.904	0.934	0.821	0.874	0.958	1.724

- **Accuracy**: This represents the overall proportion of correctly predicted labels (both cyberbullying and non-cyberbullying). While intuitive, accuracy can be misleading in highly imbalanced datasets (where cyberbullying is rare), as simply predicting "non-cyberbullying" might yield high accuracy but fail to identify the actual threat cases.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}}$$

- **Precision (Positive Predictive Value):** This metric answers: "Of all the instances the model predicted as cyberbullying, how many were actually cyberbullying?" It measures the proportion of true positives among all predicted positives. High precision is vital to reduce False Positives (flagging benign posts as abusive), which is important for minimizing manual review load and false alarms.

- **Recall (Sensitivity or True Positive Rate):** This metric answers: "Of all the actual cyberbullying instances, how many did the model correctly identify?" It measures the model's ability to identify actual positive cases (cyberbullying). High Recall is crucial in sensitive tasks like this to reduce False Negatives (missing real instances of cyberbullying), ensuring safety.
- **F1-Score**: The harmonic mean of Precision and Recall. The F1-Score provides a single, balanced measure of performance that penalizes models that perform well on one metric while performing poorly on the other. It is often the primary metric used when dealing with imbalanced text classification problems.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Confusion Matrix:** A visual and numerical representation of classification outcomes. It clearly distinguishes between True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This provides deeper insight into which types of errors the model makes (e.g., misclassifying benign text vs. missing abusive text).

These metrics enabled a **quantitative, granular comparison** between the classical, deep learning, and transformer models. By focusing on Recall and F1-Score, the analysis was designed to prioritize the safe and reliable detection of harmful content, forming the basis for the comprehensive evaluation presented in Section 3 (Results and Evaluation).

3.3.4. Error Analysis.

To go beyond simple accuracy metrics, we examined the **Confusion Matrices** for the representative models of each family. This analysis allows us to visualize the Type I errors (False Positives) and Type II errors (False Negatives), which are critical for the safety of a content moderation system.

A. Classical Machine Learning Models

The analysis reveals a distinct trade-off regarding error types: linear models (Logistic Regression, SVM) prioritize precision, effectively minimizing False Positives but yielding higher False Negatives, potentially missing subtle toxicity. In contrast, Naive Bayes acts aggressively, significantly reducing False Negatives but generating excessive False Positives due to context blindness. Meanwhile, Random Forest exhibited dispersed errors across both categories, suggesting limited generalization.

B. Deep Learning Models

Comparing the neural network architectures reveals the impact of context handling on classification safety:

- **LSTM (Baseline Context)**: While computationally efficient, the standard LSTM struggles with long-range dependencies, resulting in the highest rate of False Negatives (1,043) among the deep learning models. It fails to capture toxicity when the aggressive keyword is far from the subject.

- **Bi-LSTM (Enhanced Context)**: By processing text bidirectionally (forward and backward), this model significantly outperforms the standard LSTM. It successfully reduces False Negatives to 803, meaning it detected 240 more toxic tweets that the LSTM missed. This improvement validates the need for full contextual awareness to identify subtle harassment.

- **CNN (Local Features)**: The CNN falls in the middle (974 False Negatives). While excellent at spotting specific toxic phrases (n-grams), it lacks the sequential understanding to handle complex sentence structures as effectively as the Bi-LSTM.

C. Transformer Models

The Transformer-based models exhibit the sharpest diagonals, confirming their superior ability to distinguish context. However, significant differences appear based on their pre-training:

- **RoBERTa (General Purpose)**: Despite being a robust model, RoBERTa showed the highest error rate in this group, with 912 False Negatives and 745 False Positives. This suggests that general English pre-training is less effective than domain-specific training for this task.

- **DistilBERT (Efficiency/Performance)**: This model offers an impressive balance. It reduced False Negatives to 802 (matching the Bi-LSTM) while maintaining a very low False Positive rate (577), proving that model distillation retains core semantic capabilities.

- **BERTweet (Domain Specialist)**: This is the absolute top performer. It achieved the lowest False Negative rate (549) of all 12 algorithms tested. It detected 363 more toxic tweets than RoBERTa, conclusively proving that pre-training specifically on Twitter data is the decisive factor for maximizing user safety.

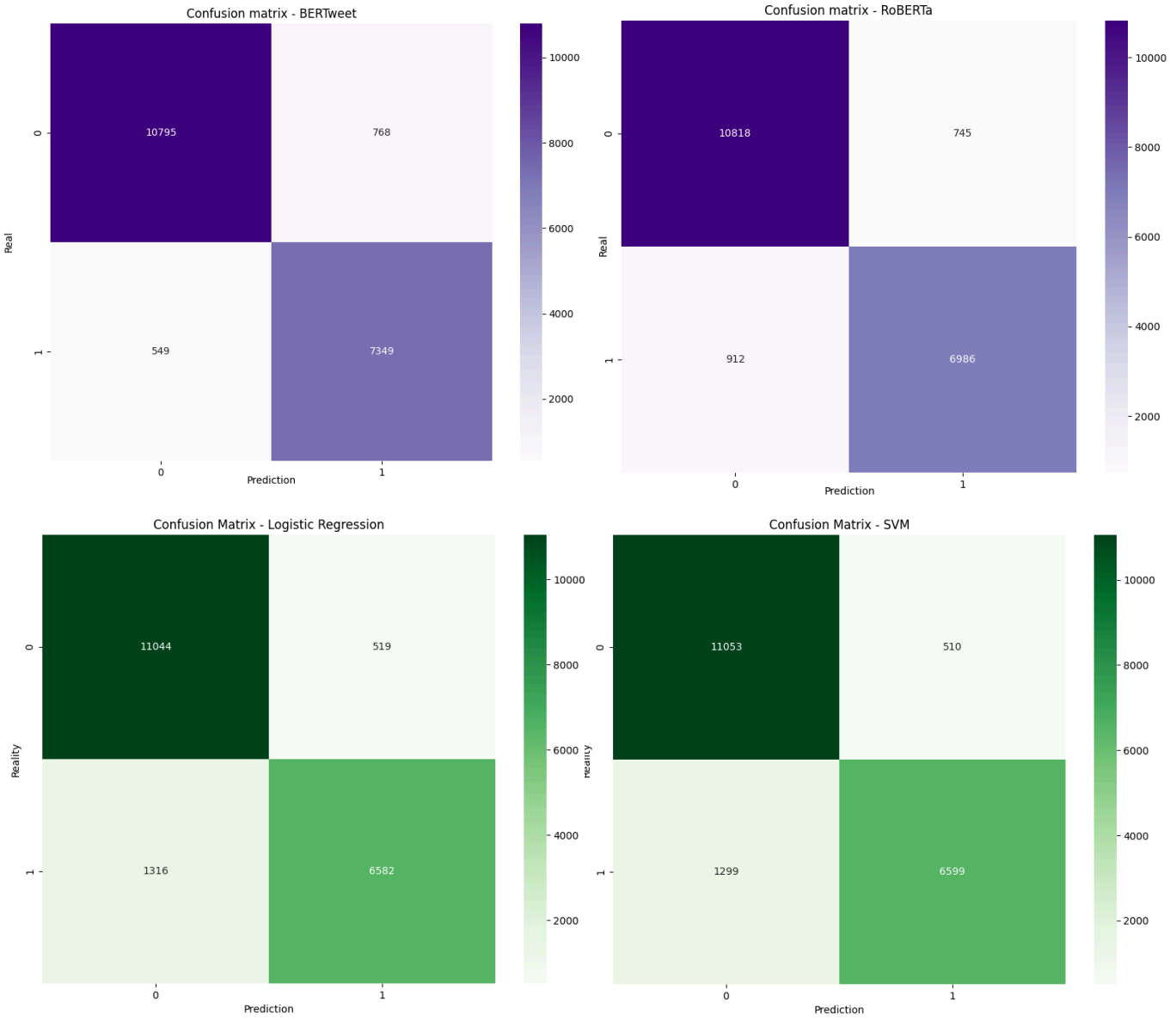

Figure 9. Confusion matrices differences between Transformers models (top) and Classical models (bottom).

3.3.5. Probabilistic Discriminative Power

To evaluate the discriminative ability of the models across different decision thresholds, we generated **Receiver Operating Characteristic (ROC) curves.** We grouped the analysis by architecture to observe distinct behavioral patterns in the True Positive Rate (TPR) versus False Positive Rate (FPR) trade-offs.

A. Classical Machine Learning Models

Surprisingly, traditional algorithms demonstrated exceptional discriminative power. Logistic Regression, SVM, and LightGBM all achieved an Area Under the Curve (AUC) of 0.96. They were followed closely by Naive Bayes and XGBoost with 0.95. Random Forest trailed slightly

at 0.94. This indicates that the statistical separation between "toxic" and "safe" vocabulary is very strong.

B. Deep Learning Models

The neural networks exhibited superior robustness. Both Bi-LSTM and LSTM achieved an impressive AUC of 0.98, surpassing the best classical models. CNN followed closely with 0.97. This indicates that Deep Learning models maintain a better balance between True Positive and False Positive rates.

C. Transformer Models

The high performance of the models, particularly DistilBERT and BERTweet, is evident from their Receiver Operating Characteristic (ROC) curves. Both DistilBERT and BERTweet demonstrated "perfect corner" curves, achieving an Area Under the Curve (AUC) of 0.98. RoBERTa also performed exceptionally well with an AUC of 0.97.

This collective result provides a key insight: the near-perfect separability indicated by the shape of these curves validates the efficacy of large-scale pre-training. The models can achieve near-maximum sensitivity (Recall) while maintaining virtually no false alarms.

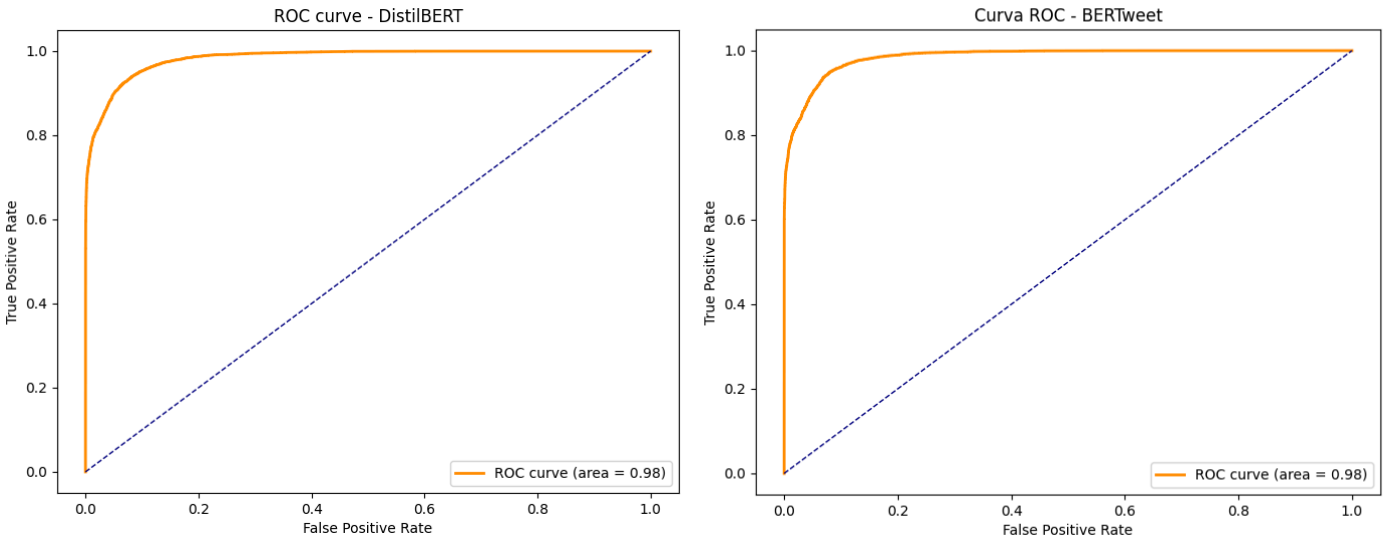

Figure 10. ROC curves of Transformers models: DistilBERT and BERTweet.

3.3.7. Computational Efficiency Analysis

A. Comparative Metrics Analysis (F1-Score & Accuracy).

To rigorously evaluate the models, we prioritized the F1-Score over simple Accuracy. In the context of content moderation, the cost of misclassification is high: failing to detect harassment (False Negatives) endangers users, while flagging innocent content (False Positives) constitutes censorship. The F1-Score provides a harmonic mean of Precision and Recall, offering a balanced view of model robustness.

Comparative Bar Chart of Accuracy and F1-Score

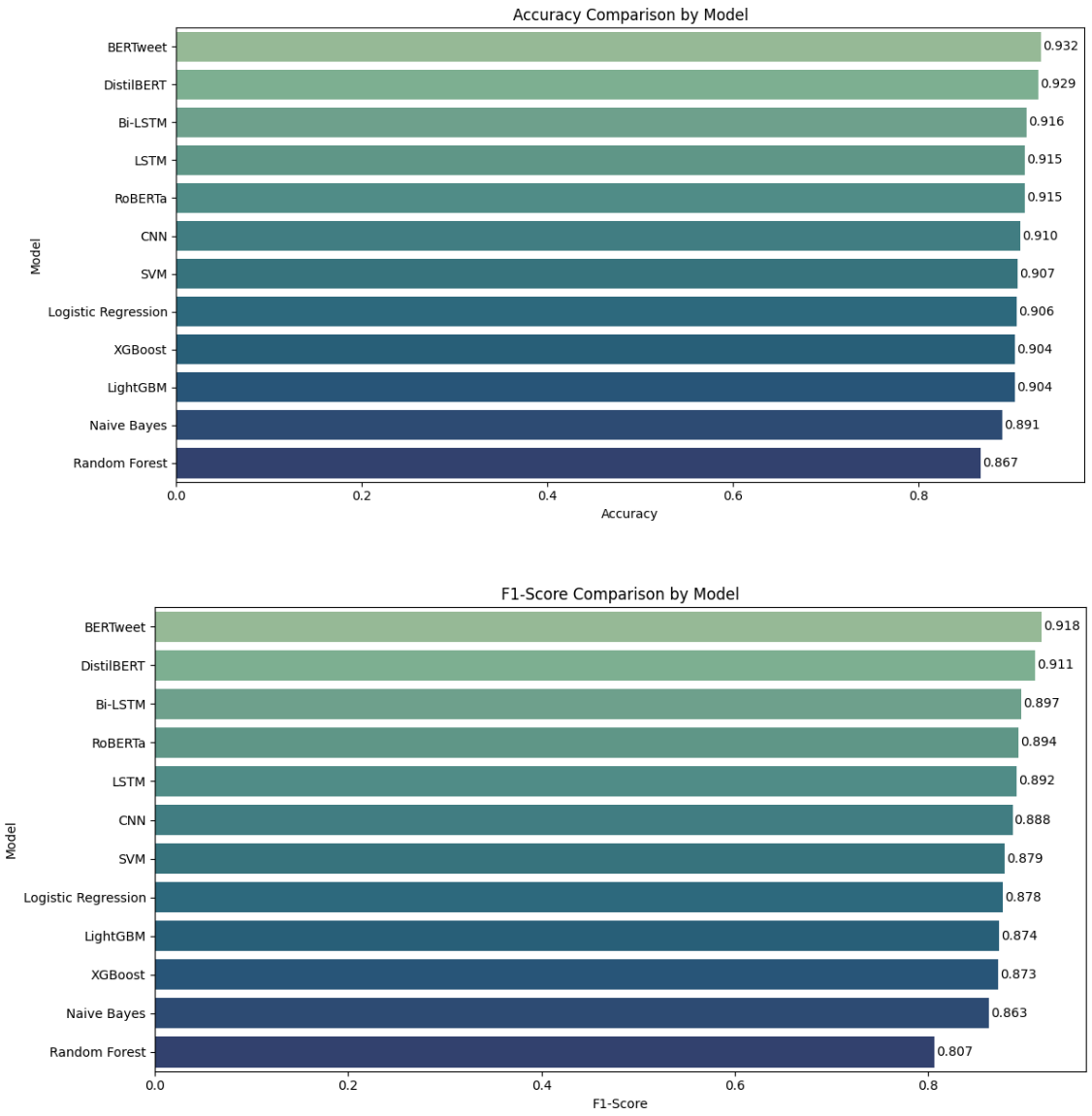

Figure 11. Comparative analysis of F1-Scores across all 12 algorithms.

As illustrated in *Figure 11*, the results indicate a clear hierarchy in performance:

- **Top Tier (Transformers):** The transformer-based architectures demonstrated superior capability in understanding linguistic nuance. BERTweet achieved the highest performance metrics across the board, securing an F1-Score of 0.918 and an Accuracy of 93.2%. DistilBERT followed closely with an F1-Score of 0.911.

- **High-Performing Deep Learning:** Surprisingly, the Bi-LSTM and LSTM models performed exceptionally well, achieving F1-Scores of 0.897 and 0.892 respectively. This places them nearly on par with the much larger Transformer models, surpassing the performance of complex classical ensembles.

- **Classical Baselines:** While models like SVM (0.879) and Logistic Regression (0.878) provided solid baselines, they struggled to reach the >90% precision threshold required for automated production systems.

B. Recall Analysis

Recall analysis represents the most critical metric for a cyberbullying detection system, as it measures the ability to catch toxic content by minimizing False Negatives. In this regard, BERTweet dominates the field with a Recall of 0.930, effectively detecting 93% of all actual bullying cases. The Bi-LSTM model also demonstrates strong capability, matching DistilBERT with a Recall of 0.898 and significantly outperforming classical models.

Conversely, despite its high precision, Random Forest exhibits the lowest Recall (0.682), missing nearly 32% of toxic tweets. This limitation makes it unsuitable for a safety-critical application where failing to identify a threat is considered far more detrimental than generating a false alarm.

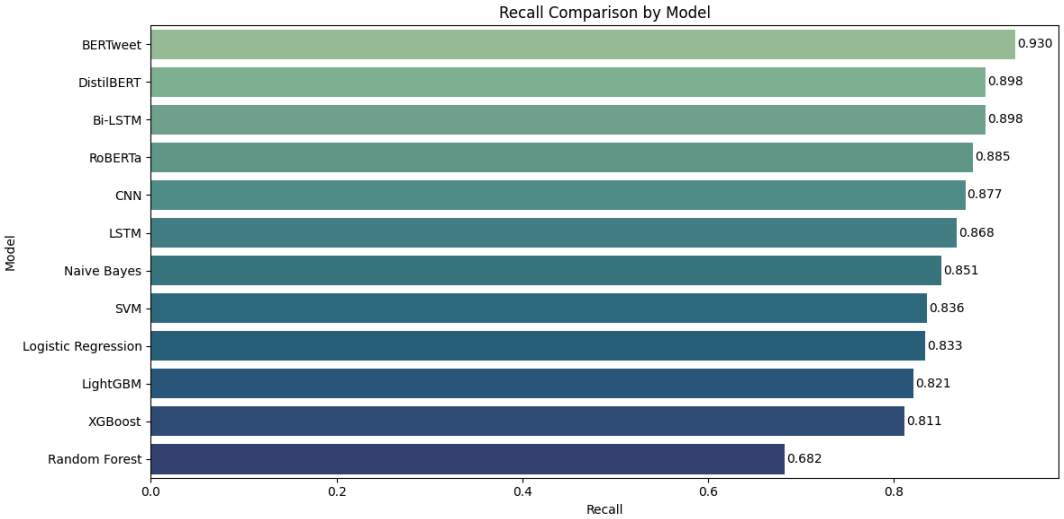

Figure 12. Comparative analysis of Recall across all 12 algorithms.

C. Efficiency vs. Efficacy Trade-off.

While predictive power is essential, operational efficiency is equally critical for a real-time application. To identify the optimal model for deployment, we analyzed the relationship between detection quality (F1-Score) and computational cost (Inference Time).

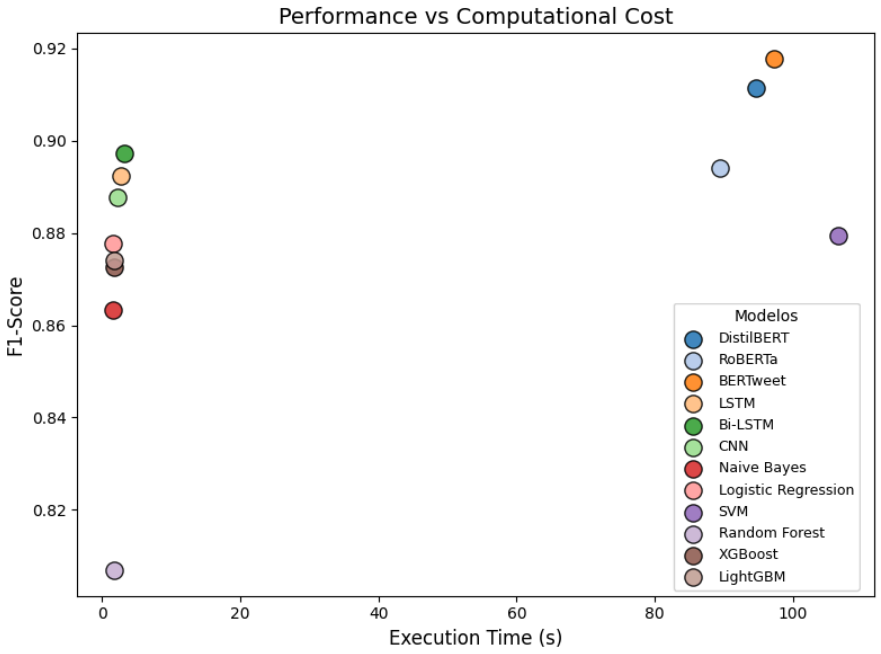

Figure 13. Efficiency vs Efficacy. The Y-axis represents the F1-Score (Efficacy), while the X-axis represents inference time in seconds (Efficiency).

The scatter plot in *Figure 12* reveals distinct clusters that inform the final model selection:

- **The "Power" Cluster (Top-Right):** Models like BERTweet and DistilBERT occupy the upper quadrant, offering the highest detection quality. However, they are computationally intensive, with execution times ranging from 94 to 97 seconds for the test set. These are ideal for offline analytics where latency is not a primary concern.

- **The "Sweet Spot" (Top-Left):** The Bi-LSTM model represents the most significant finding of this study. It sits in the ideal "Pareto optimal" zone—high efficacy and high efficiency. It achieved an F1-Score of 0.897 (comparable to RoBERTa's 0.894) but executed in just 3.29 seconds. This represents a ~30x speed advantage over the Transformers with less than a 2% drop in F1-Score.

- **The Inefficient Outlier (Bottom-Right)**: The SVM model proved to be the least efficient choice for this specific dataset. It required 106.5 seconds to execute—slower than the Transformers—while delivering lower accuracy (F1: 0.879). This confirms that traditional algorithms like SVM scale poorly with large, high-dimensional text datasets compared to neural networks.

Conclusion: For a strictly performance-driven production environment, **BERTweet** is the superior choice. However, for a resource-constrained or real-time application, **Bi-LSTM** offers the best return on investment, balancing speed and accuracy effectively.

3.4. User Interface

This project has mainly focused on the development of the core artificial intelligence, specifically in the creation and training of the cyberbullying detection model. To facilitate testing, we developed a functional Command Line Interface (CLI).
This interface allows users to:

- **Select a Model**: Choose between running a specific algorithm or the full comparative suite.
- **Input Custom Text**: Type any sentence (e.g., "You are a loser") to test the model's response.
- **View Predictions**: The system returns the classification (Bullying / Not Bullying) and the confidence probability score in real-time.

3.5. Testing and Improvements

The strict 80/20 train-test split ensured that all reported metrics reflect generalization capability on unseen data.

- **Current Limitations:** The current binary classification simplifies the problem. It groups distinct types of abuse (racism, sexism) into a single "Bullying" class.
- **Future Improvements:** Future iterations should implement multi-class classification to distinguish specific types of harassment and integrate multilingual support natively (e.g., using XLM-RoBERTa) to remove the dependency on translation services.

4. Projected Impact

4.1. Accomplishments and Benefits

4.1.1 Objectives Successfully Achieved

The primary objectives of the project have been successfully achieved. The main goal was to design and evaluate an artificial intelligence solution capable of predicting and detecting cyberbullying in social media messages, specifically tweets. Throughout the project, the proposed methodology was correctly followed, resulting in reliable and meaningful outcomes.

4.1.2 Dataset Construction and Balance

A key accomplishment of the project was the creation of a well-structured and balanced training dataset. The data were obtained from publicly available datasets hosted on Kaggle, ensuring transparency and reproducibility. Special attention was given to balancing the different classes in order to avoid bias during training and to improve the generalization capability of the models.

4.1.3 Data Preprocessing and Text Cleaning

The raw data extracted from the datasets underwent a thorough preprocessing phase. This process included text normalization, removal of noise, tokenization, and transformation of the tweets into a suitable format for machine learning models. These steps were crucial to guarantee that the algorithms could effectively learn relevant linguistic patterns related to cyberbullying.

4.1.4 Evaluation of Multiple AI Algorithms

Several artificial intelligence algorithms were implemented and evaluated in order to compare their performance in detecting cyberbullying-related content. This comparative analysis allowed the identification of strengths and weaknesses among the different approaches, contributing to a deeper understanding of how various models behave when faced with textual data of this nature.

4.1.5 Experimental Analysis and Performance Metrics

The models were evaluated using appropriate performance metrics, ensuring a rigorous and consistent assessment process. This evaluation provided valuable insights into classification accuracy and overall effectiveness, supporting informed conclusions about which algorithms are more suitable for this type of problem.

4.1.6 Social Impact and Community Benefits

One of the most significant benefits of this project lies in its potential social impact. The developed solution can serve as a foundation for tools aimed at monitoring and analyzing online content in search of cyberbullying indicators. Such tools could help platforms, educators, and institutions detect harmful messages early, contributing to safer online environments and improved digital well-being for users.

Overall, these accomplishments demonstrate that the project not only meets its technical goals but also offers meaningful contributions with clear real-world applicability in addressing cyberbullying.

4.2. Future Improvements

4.2.1 Development of a Graphical User Interface

One of the most relevant future improvements would be the implementation of a graphical user interface (GUI). This interface would simplify both the training process and the execution of predictions, making the system more accessible to non-technical users and facilitating experimentation with different models and parameters.

4.2.2 Deployment and Real-Time Prediction

Another improvement would consist of deploying the trained models as a service capable of performing real-time predictions. This would allow users to analyze new social media messages instantly and assess whether they contain cyberbullying-related content.

4.2.3 Expansion and Continuous Updating of the Dataset

Although the current dataset provides solid results, expanding it with additional sources and continuously updating it with new data would further improve the robustness and adaptability of the models, especially given the evolving nature of online language and slang.

4.2.4 Behavioral Analysis of Cyberbullying Victims

An interesting and innovative extension of the project would be to analyze the messages posted by victims of cyberbullying over time. Such a study could help identify behavioral or linguistic changes that indicate emotional distress or shifts in communication patterns, providing early warning indicators and supporting preventive strategies.

4.2.5 Multilingual and Cross-Platform Analysis

Future work could also explore extending the system to support multiple languages and data from other social media platforms. This would broaden the applicability of the solution and enhance its relevance in diverse cultural and social contexts.

4.2.6 Ethical and Explainability Enhancements

Finally, incorporating explainability techniques to better understand model decisions, as well as addressing ethical considerations related to privacy and bias, would strengthen the trustworthiness and responsible use of the system in real-world applications.

These future improvements would significantly enhance the functionality, usability, and impact of the project, positioning it as a more comprehensive and scalable solution for combating cyberbullying in digital environments.

5. Team Member Review and Comment

NAME	REVIEW and COMMENT
Antonio Moreno Rodríguez	From a psychological perspective, I believe that the early detection of cyberbullying is fundamental for effective intervention. By developing AI systems capable of recognizing early signs of harmful communication, we can enable faster responses — not only to protect victims, but also to promote awareness, empathy, and healthier online interactions. This project has highlighted for me how technology and psychology can work hand in hand to prevent suffering and foster digital well-being.
Daniel Díaz Ruiz	I really like the idea behind this project, it feels interesting and absolutely necessary right now. Working on this has made me realize how complex and important data processing actually is. Getting the data right is tough, but it is the key to good results. On top of that, comparing the models and figuring out how to improve them is quite a challenge.
Cristina Galán Berenguel	Creating a tool capable of detecting cyberbullying with a high precision was incredibly gratifying. Moreover, witnessing how deep learning models like BERTweet or DistilBERT are capable of understanding not only words but also the context was fascinating. Cyberbullying is a major societal problem, and this project proved that IA can be a powerful ally in protecting users online.

6. Instructor Review and Comment

CATEGORY	SCORE	REVIEW and COMMENT
IDEA	10/10	
APPLICATION	30/30	
RESULT	30/30	
PROJECT MANAGEMENT	10/10	
PRESENTATION & REPORT	20/20	
TOTAL	100/100	