MKSSS’s Cummins College of Engineering for Women,Pune
Department of Computer Engineering
Artificial Intelligence and Machine Learning Laboratory 23PCCE501L
TY B. Tech Semester-I 2025-26

CYBER BULLYING DETECTION IN SOCIAL MEDIA

UCE2023427- Shravani Joshi
UCE2023430- Janhavi Kakde
UCE2023448- Aditi Parekar

Problem Statement

- 46% adolescents face online harassment; 40% witness harmful interactions.
- Growth of social media → harder moderation at large scale.
- Rise of audio communication: 31% of interactions are voice-based (Meta, 2024); 7B+ WhatsApp voice notes/day.
- Audio cyberbullying is hard to detect due to accents, tone, sarcasm & background noise.
- Traditional text-only NLP models fail to catch multimodal abuse.
- Emojis used billions of times daily; certain emoji combinations act as hidden bullying cues, often missed by NLP
- Communication is also done through images containing text such as screenshots or memes of abusive messages.

Proposed Solution

Text-Based Detection

- Converts user comments into numerical features using text processing.
- Classifies the comment as “bullying” or “non-bullying.”
- Provides encryption message bullying detection.
- Uses a trained model that checks for harmful or abusive words in the text.

Emoji-Based Detection

- Checks if a comment contains emojis commonly used for bullying.
- Detects frequently occurring patterns of emojis associated with insults or mockery.
- Flags comments where emojis are used instead of words .
- Helps catch bullying even when users avoid text.

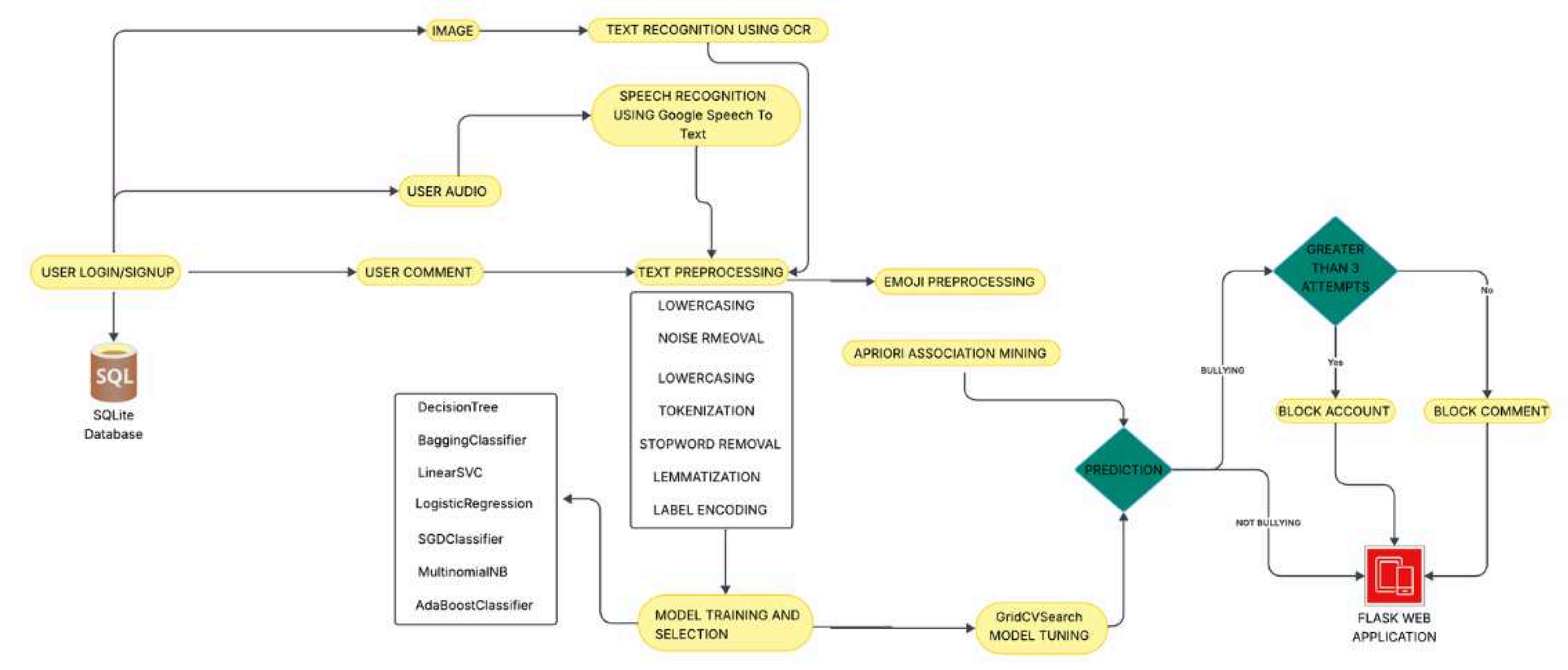
Audio-Based Detection

- Converts user voice messages into text using speech-to-text.
- Runs the converted text through the same bullying detection process as normal comments.
- Helps detect bullying hidden in spoken messages.
- Ensures that harmful speech cannot bypass moderation.

Image-Based Detection

- The uploaded image is processed using a text-recognition engine which automatically identifies characters, words, and sentences present within the visual content.
- Normalizes extracted text for consistent classification.
- Feeds the extracted text into the machine learning model.

System Architecture



- **User Login & Signup**
  - Users create accounts through a secure signup form where passwords are hashed before being saved, and during login, the system verifies the hash and creates a session for authenticated access.
  - This ensures only registered users can use the bullying detection features, maintaining privacy and controlled access.

System Architecture

- **SQLite Database**
  - A lightweight SQLite database stores user details such as username, and encrypted passwords in an organized table.
  - Flask connects to SQLite for every login/signup request, executes queries safely, and commits changes for reliable user management.

- **Emoji Processing**
  - Apriori identifies emoji combinations that commonly appear in bullying comments and learns them as frequent patterns. When a new comment contains these patterns, the system uses them as rules to more confidently detect bullying behavior.

- **Audio Processing**
  - Using the SpeechRecognition library with Google’s speech-to-text engine, the user’s audio is converted into text and merged with the typed comment for bullying detection.
  - SpeechRecognition is a flexible Python library used. It handles audio loading, noise adjustment, and transcription through simple functions such as record() and recognize_google().

Database

Comments
id
user_id
comment
prediction
created_at

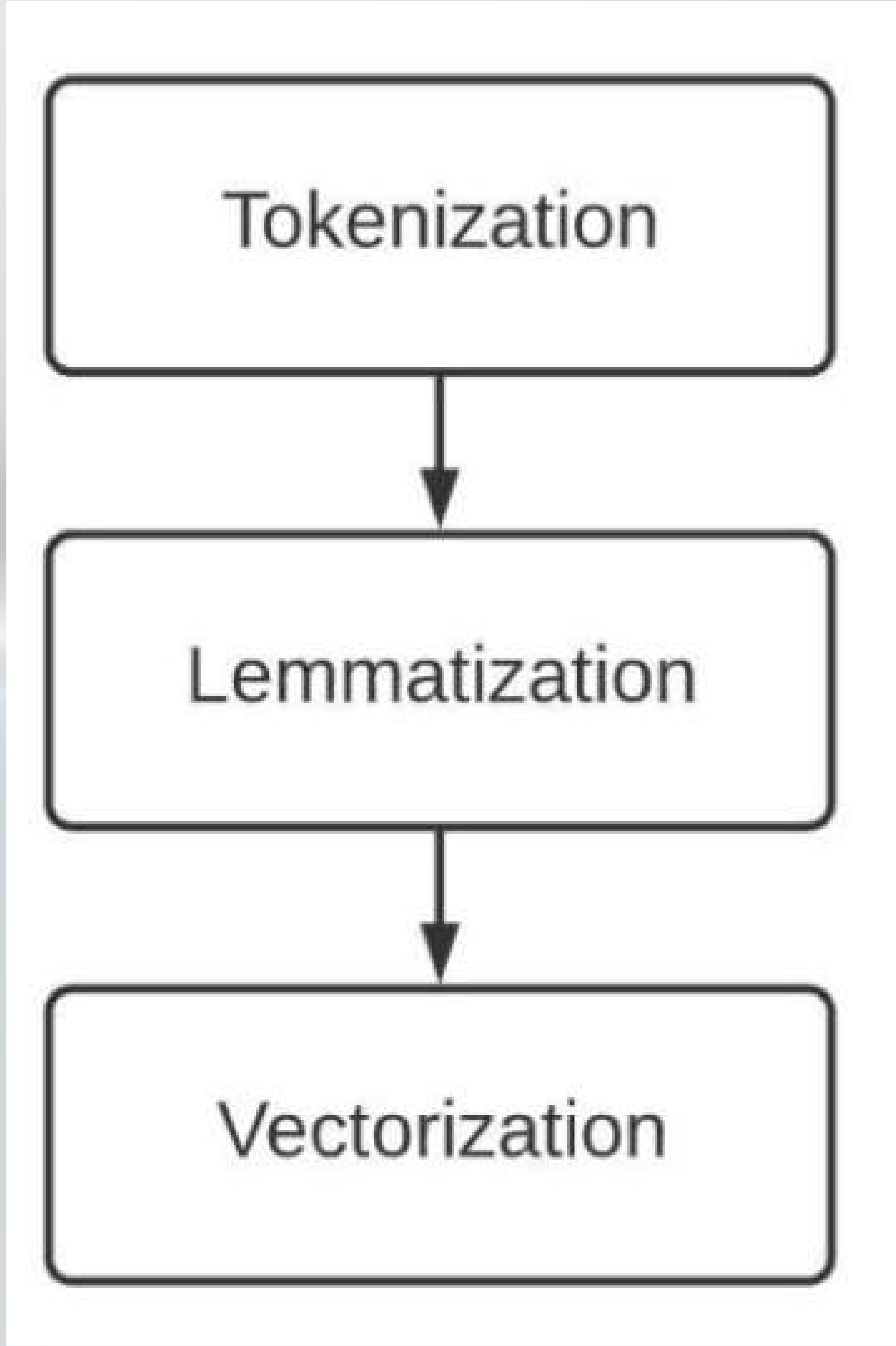
Users
id
username
password_hash
bullying_count
is_blocked

System Architecture

- **Image Processing**
  - When a user uploads an image, the module first applies Optical Character Recognition (OCR) to extract all visible text from the image.
  - The extracted text is then passed through the same bullying detection pipeline (machine-learning classifier + emoji analysis) used for normal typed comments.
- **Encrypted Message Classification**
  - Normalization is done through Leetspeak- A common technique used to replace letters with numbers or symbols
  - After normalizing symbols, the text may still be incomplete or intentionally misspelled. To fix that, we use SymSpell, a fast spell-correction library, trained here on a custom dictionary of abusive root words.
  - Non-alphabet characters from each word are removed then feed into SymSpell.
  - Combined pipeline of Leetspeak correction + SymSpell auto-correction allows to detect abusive language.
- **Languages supported**
  - The dataset initially contains both English and Hindi languages. Hence model is trained for Bullying detection in both languages.
  - To support multilingual cyberbullying detection, the system integrates an automatic translation module using Google Translate.
  - When a user submits Marathi text, the system first identifies the language and converts the input into English using the Google Translate API.

System Architecture

- **Text Preprocessing**
  - Text preprocessing cleans and standardizes typed and audio-converted comments by lowercasing, removing symbols, and eliminating stop-words. The text is then tokenized, lemmatized, and finally converted into numerical vectors so the ML model can accurately detect bullying.



- **Model Training, Selection & Tuning**
  - Multiple ML models—Decision Tree, Bagging, Linear SVC, Logistic Regression, SGD, Naïve Bayes, and AdaBoost—are trained on cleaned text to learn bullying patterns.
  - Their performance is compared using accuracy and F1-score, and the best model is chosen. **GridSearchCV** then fine-tunes its hyperparameters to improve accuracy and reduce overfitting before being saved for real-time prediction.

- **Prediction**
  - If the model detects bullying, the comment is blocked instantly, and the user receives a warning. After three bullying attempts, the system automatically blocks the user's account to prevent further harmful activity.

Algorithms Used

1)Linear SVC (Support Vector Classifier):

- Based on the Support Vector Machine (SVM) algorithm.
- It tries to find the best line (or hyperplane) that separates toxic and non-toxic comments in the high-dimensional TF-IDF feature space.
- The goal is to maximize the margin — the distance between the separating line and the nearest data points (called support vectors).
- Works really well with text data because it handles high-dimensional and sparse data effectively.

3)Decision Tree Classifier:

- Builds a tree-like structure of decisions.
- Each node checks for a condition and splits data accordingly until it reaches a decision (toxic or non-toxic).
- It's easy to interpret, but can overfit on text data if not controlled with pruning or depth limits.

2)Logistic Regression:

- Despite the name, it's a classification algorithm (not regression).
- It calculates the probability that a given input belongs to a particular class (toxic or non-toxic) using the sigmoid function.
- If the probability > 0.5 → toxic, else → non-toxic.
- Works well for binary text classification and gives interpretable coefficients for each word feature.

4)Naive Bayes Classifier:

- Based on Bayes' Theorem, which calculates the probability of a class given the features.
- "Naive" because it assumes all features (words) are independent of each other — which isn't strictly true, but it still works surprisingly well for text.
- Extremely fast and effective for text classification (especially Multinomial Naive Bayes).

Algorithms Used

5)AdaBoost Classifier (Adaptive Boosting):

- An ensemble method that combines multiple "weak" learners (often small decision trees) to create a strong classifier.
- It gives higher weight to misclassified examples in each round, forcing later models to focus on those harder cases.
- Helps improve accuracy and robustness.

6)Bagging Classifier (Bootstrap Aggregating):

- Another ensemble technique — it trains multiple versions of the same model on random subsets of data (sampled with replacement).
- Final prediction is based on majority voting among models.
- Reduces variance and overfitting.

7)SGD Classifier (Stochastic Gradient Descent):

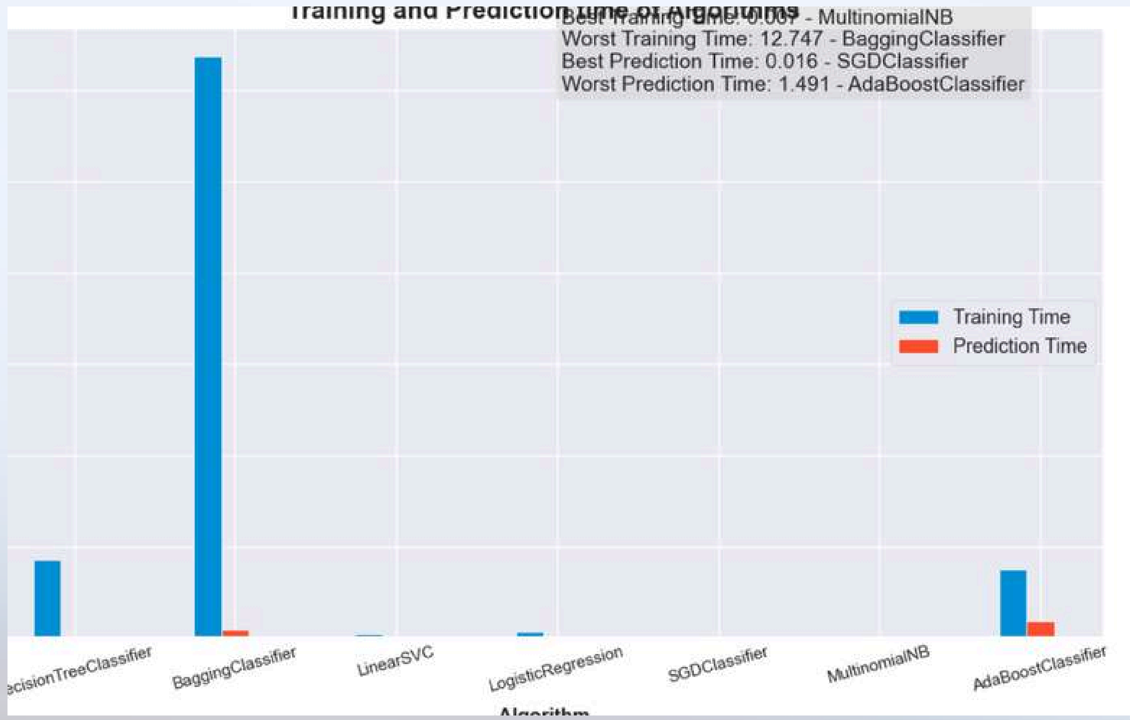
- Optimizes a loss function (like hinge loss or log loss) incrementally, one sample at a time.
- Extremely fast and memory-efficient, making it ideal for large text datasets.

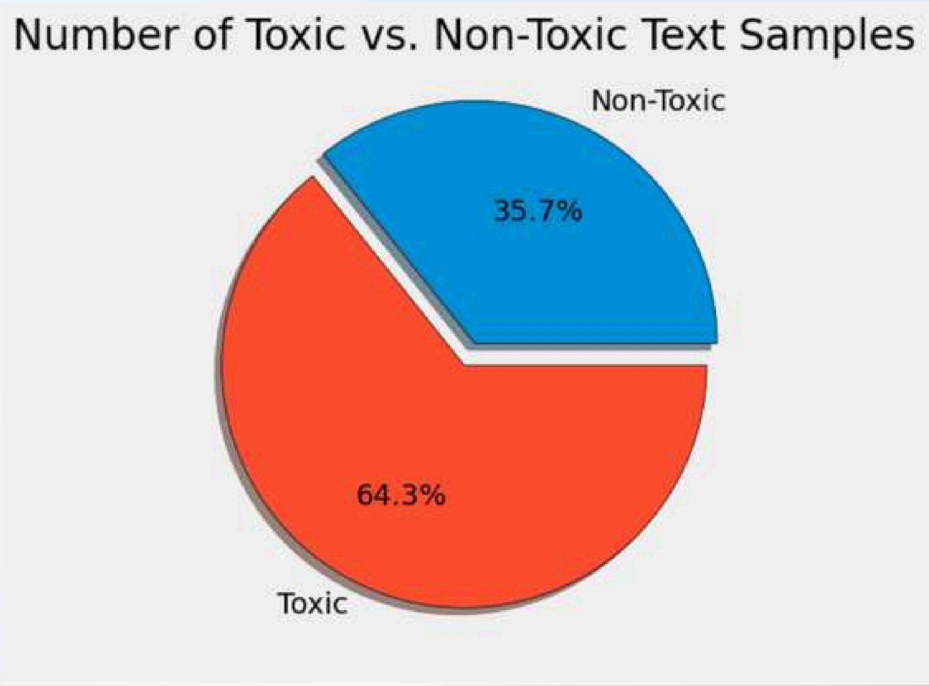
Model Performance

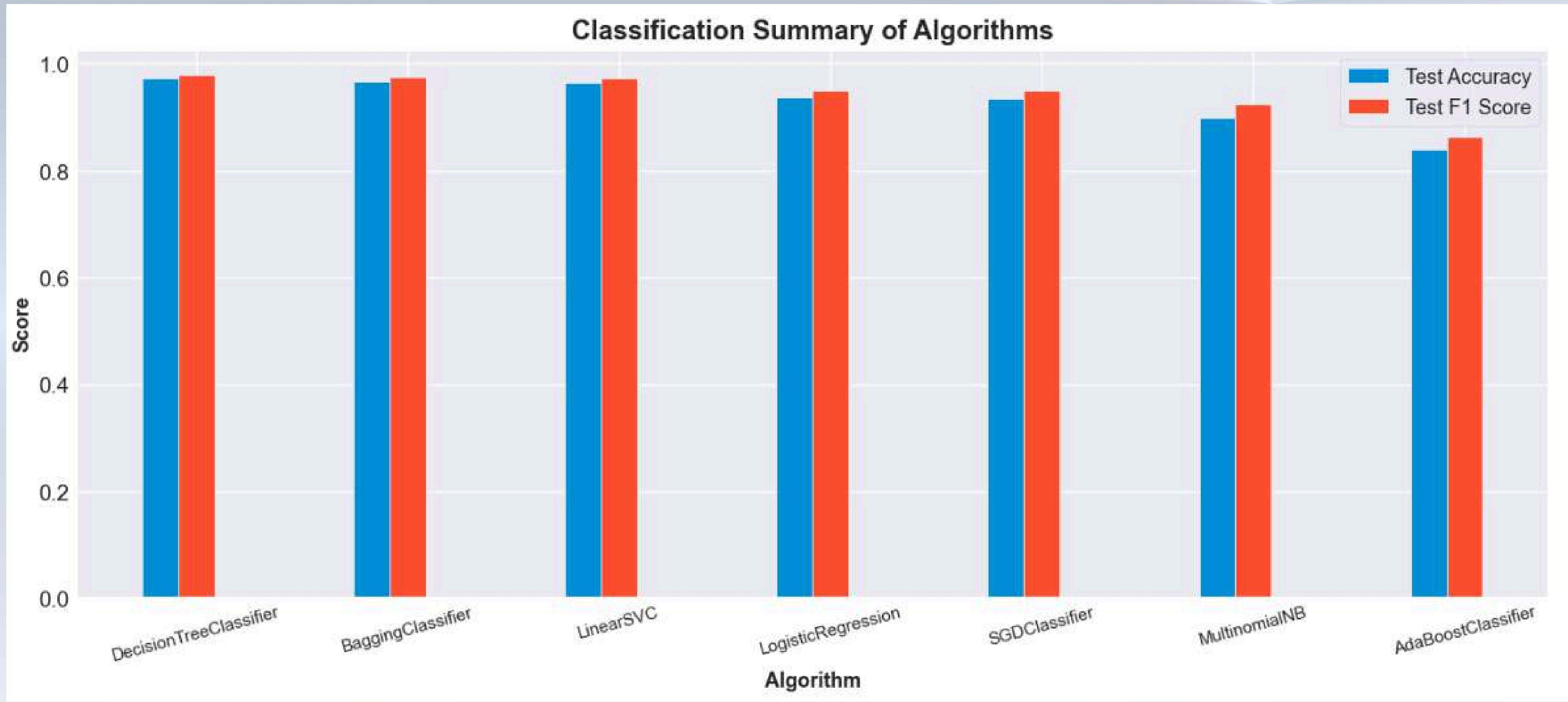
Algorithm	Training Time	Prediction Time	Accuracy (Test)	Accuracy (Train)	F1 Score (Test)	F1 Score (Train)	Precision (Test)	Precision (Train)	Recall (Test)	Recall (Train)
DecisionTreeClassifier	1.703846	0.015273	0.97438	0.996832	0.98033	0.997534	0.973126	0.996146	0.987641	0.998926
BaggingClassifier	12.747166	0.177979	0.967355	0.994868	0.974806	0.996004	0.972635	0.995336	0.976987	0.996672
LinearSVC	0.078223	0.000971	0.964738	0.989048	0.97269	0.991459	0.973937	0.992152	0.971447	0.990767
LogisticRegression	0.142333	0.00074	0.936364	0.961152	0.950514	0.969637	0.955632	0.972464	0.945451	0.966826
SGDClassifier	0.016153	0.000622	0.936088	0.957914	0.950054	0.966984	0.959974	0.973454	0.940337	0.960599
MultinomialNB	0.006839	0.002149	0.899174	0.927676	0.925428	0.945438	0.88659	0.916159	0.967824	0.976649
AdaBoostClassifier	1.490938	0.370727	0.840909	0.841955	0.863749	0.863402	0.967495	0.969063	0.780098	0.778517

- **Across all models tested, DecisionTreeClassifier achieved the highest test accuracy (97.43%) and the highest recall (98.76%), indicating strong sensitivity to bullying cues. However, as shown in the table, the LinearSVC exhibited the best balance of accuracy, F1-score, and prediction time, making it the optimal choice for real-time moderation.**

Results & Discussion

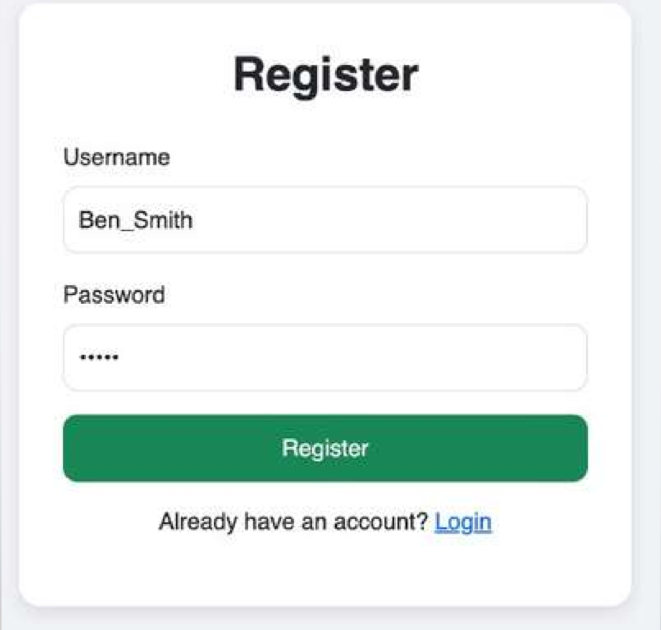


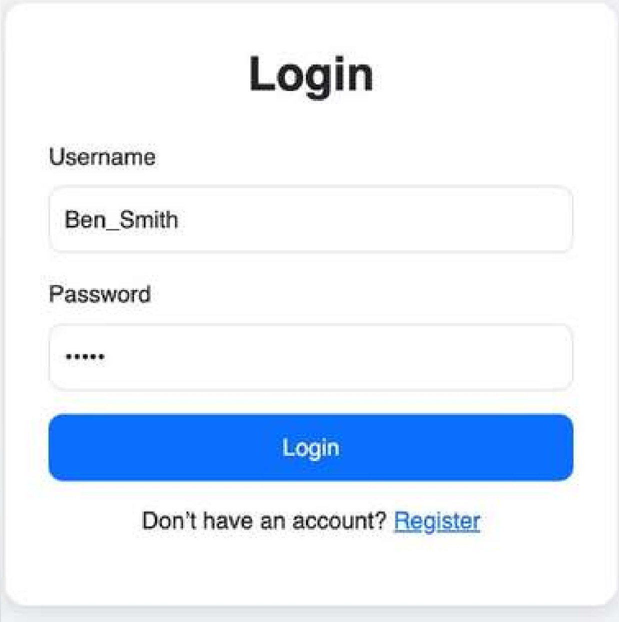


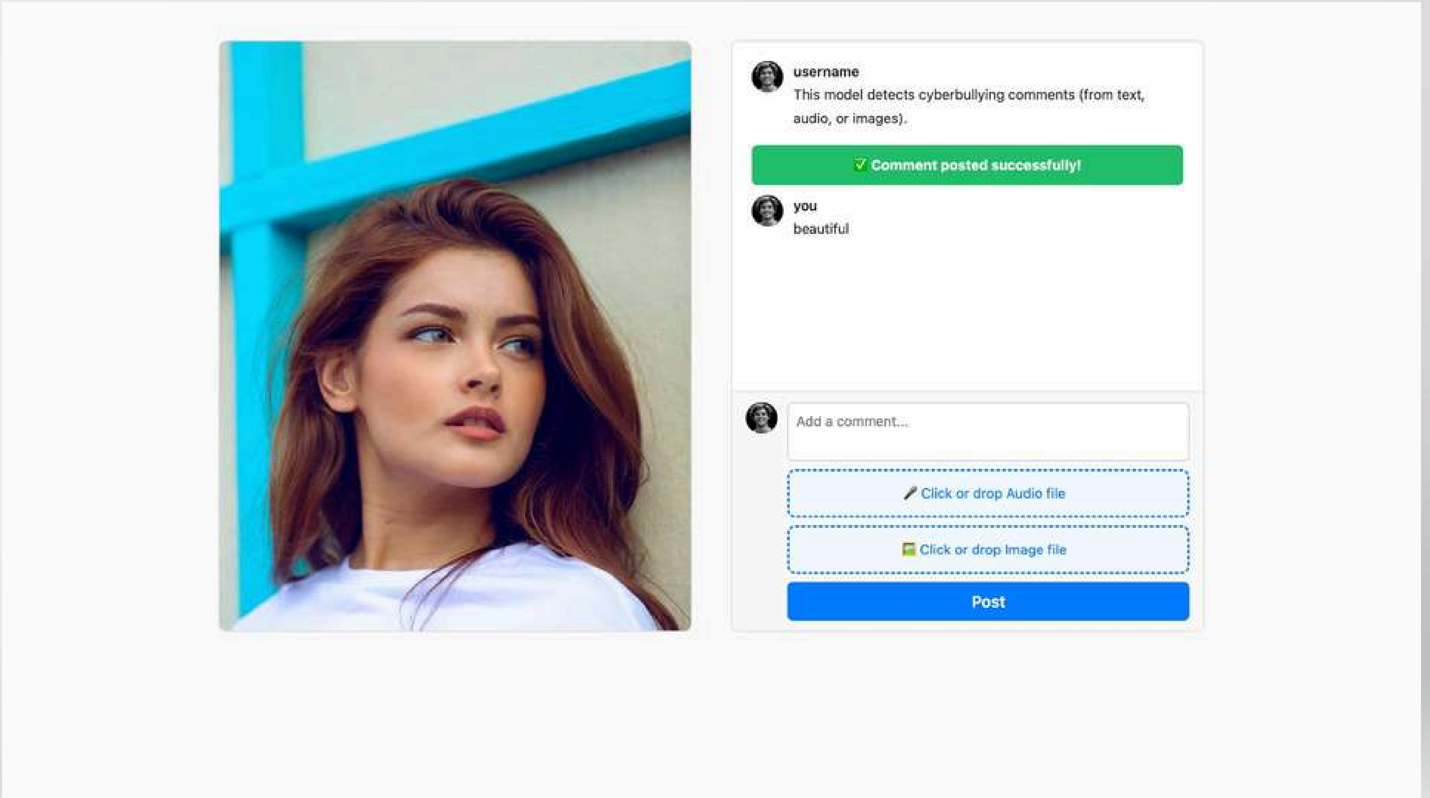


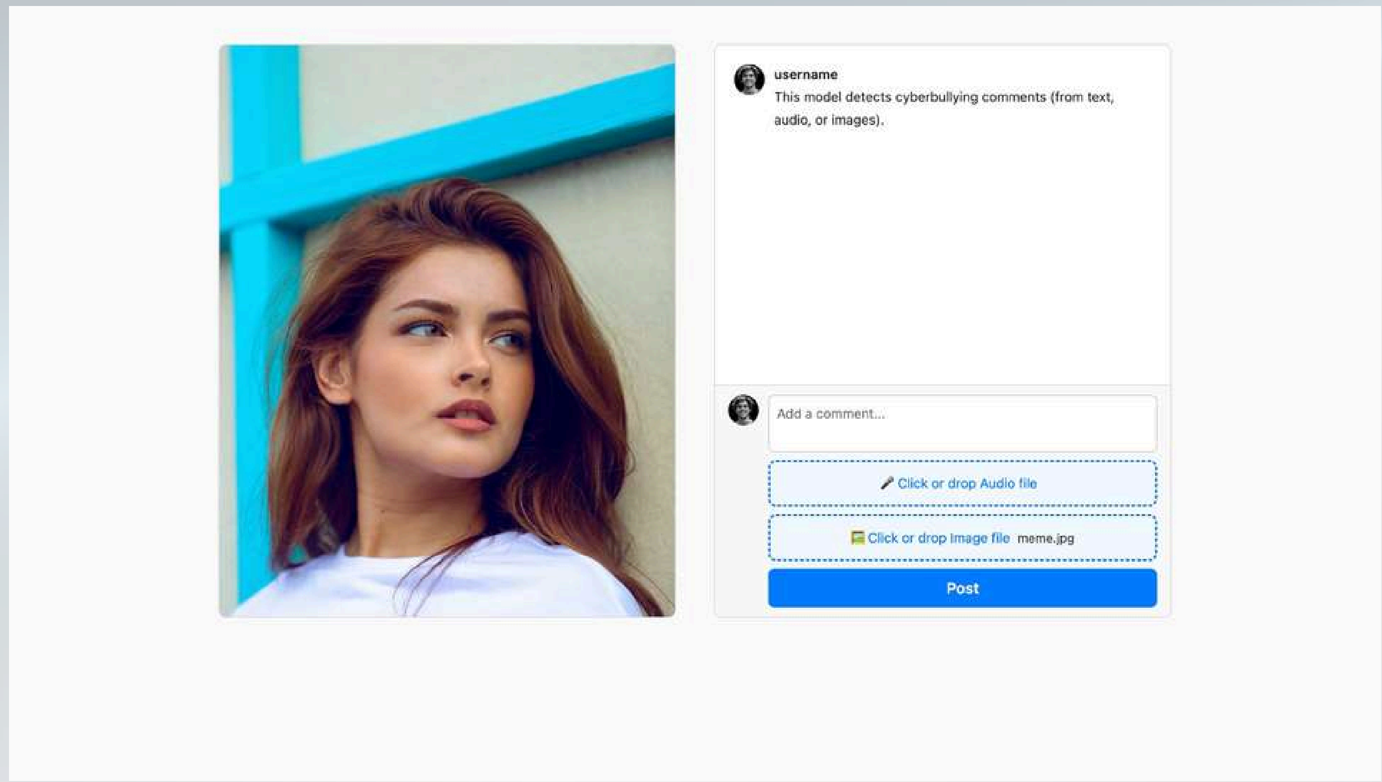
- The dataset is imbalanced with 64.3% toxic and 35.7% non-toxic text samples.
- Multiple machine learning models were evaluated by accuracy, F1 score, training time, and prediction time.
- **BaggingClassifier** shows high accuracy but requires significantly more training time, while **MultinomialNB** offers faster execution with slightly lower performance.
- **Linear SVC** was selected as the final model due to its high accuracy, strong F1 score, and efficient performance on text classification.
- The model was further **fine-tuned** using GridSearchCV to optimize hyperparameters and improve generalization.

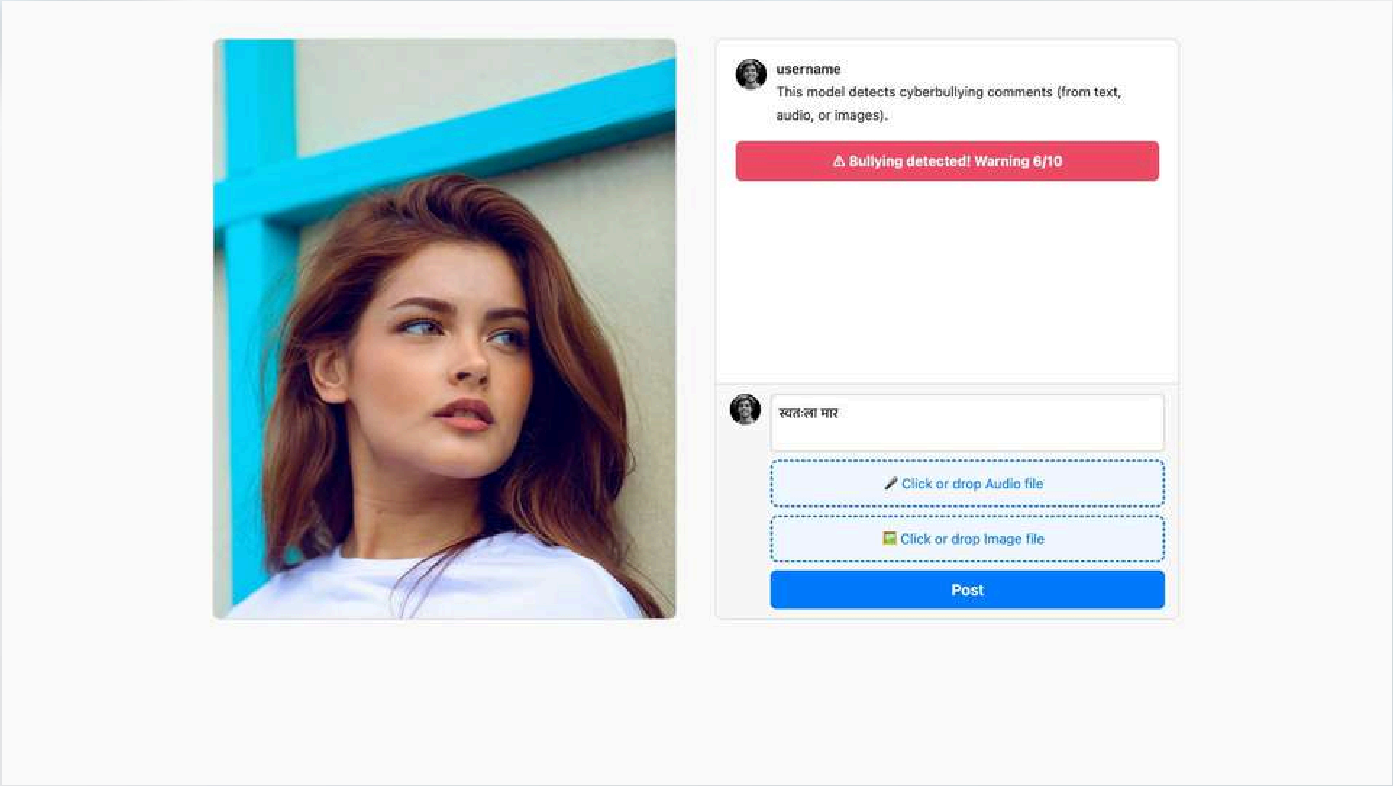
UI Screenshots

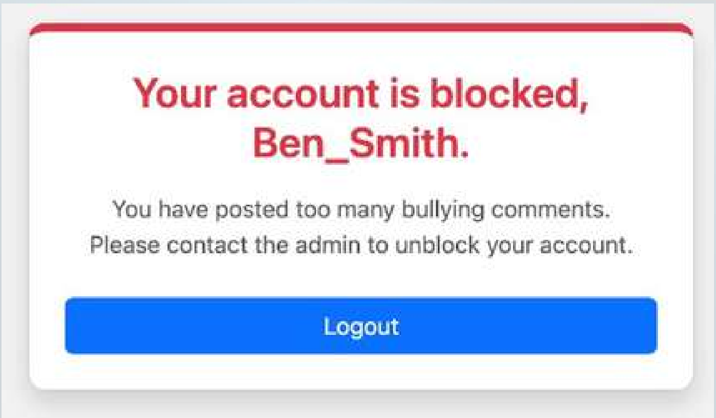












Conclusion

- This research introduces a cyberbullying detection system that analyzes text, audio, and emojis to provide more accurate and scalable content moderation on social media.

- Through evaluation of multiple machine learning models, Linear SVC emerged as the most effective classifier, delivering 96.47% accuracy with fast prediction times suitable for real-time use.

- The system's inclusion of speech-to-text transcription and emoji pattern mining (Apriori algorithm) helps identify subtle or non-verbal forms of bullying often missed by traditional text-only methods.

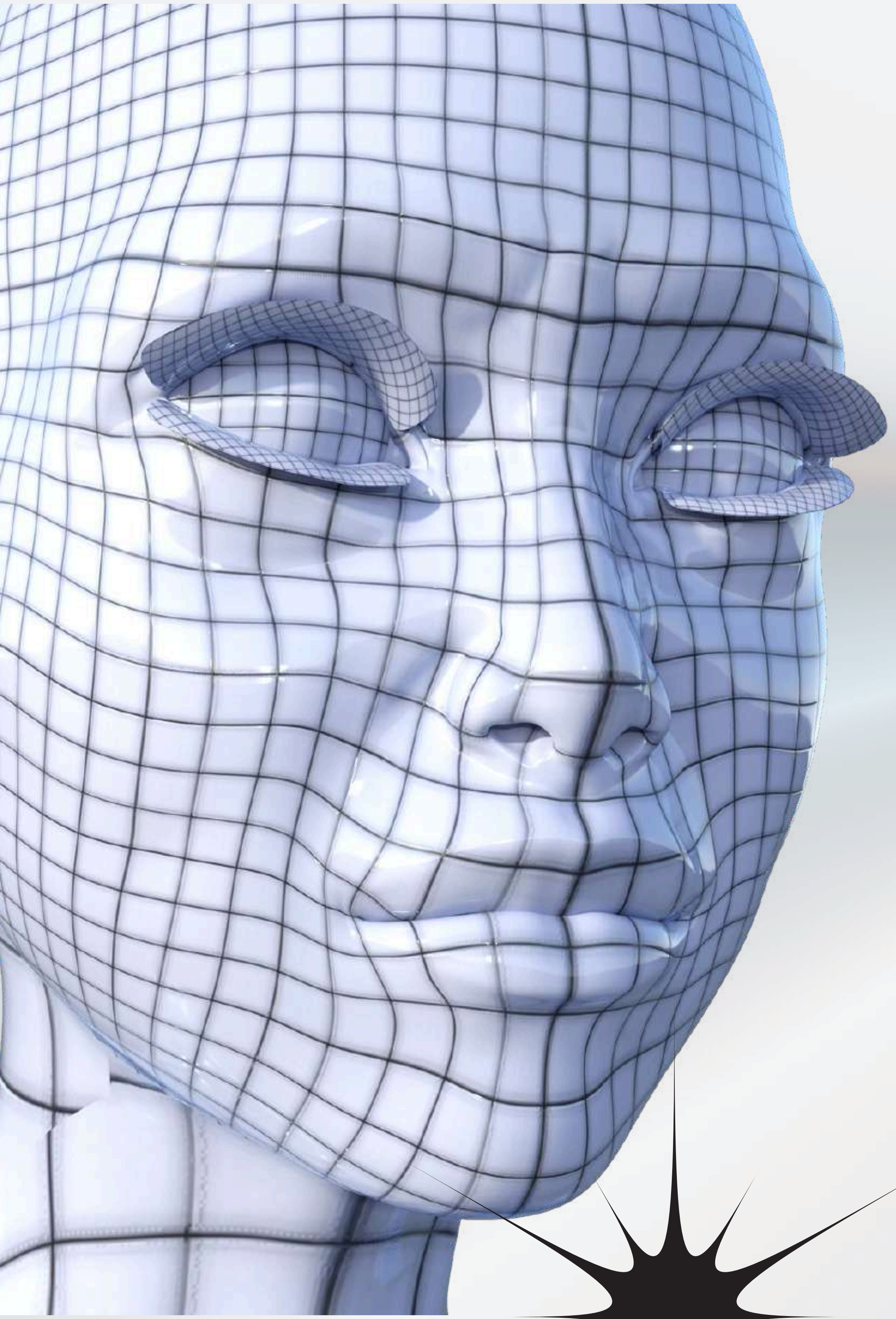
- An integrated account-blocking mechanism further reduces repeated abusive behavior.

- With support for English and Hindi, the system demonstrates strong applicability across diverse user groups.

- Overall, the study shows that a multi-modal and proactive design significantly improves cyberbullying detection and offers a practical solution for modern social platforms.

References

- G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)
- J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- K. Elissa, “Title of paper if known,” unpublished.
- R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
- Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
- M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
- K. Eves and J. Valasek, “Adaptive control for singularly perturbed systems examples,” Code Ocean, Aug. 2023. [Online]. Available: https://codeocean.com/capsule/4989235/tree
- D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, arXiv:1312.6114. [Online]. Available: https://arxiv.org/abs/1312.6114
- S. Liu, “Wi-Fi Energy Detection Testbed (12MTC),” 2023, gitHub repository. [Online]. Available: https://github.com/liustone99/Wi-Fi-Energy-Detection-Testbed-12MTC
- “Treatment episode data set: discharges (TEDS-D): concatenated, 2006 to 2009.” U.S. Department of Health and Human Services, Substance Abuse and Mental Health Services Administration, Office of Applied Studies, August, 2013, DOI:10.3886/ICPSR30122.v2



Thank You

Stay Safe, Stay Smart

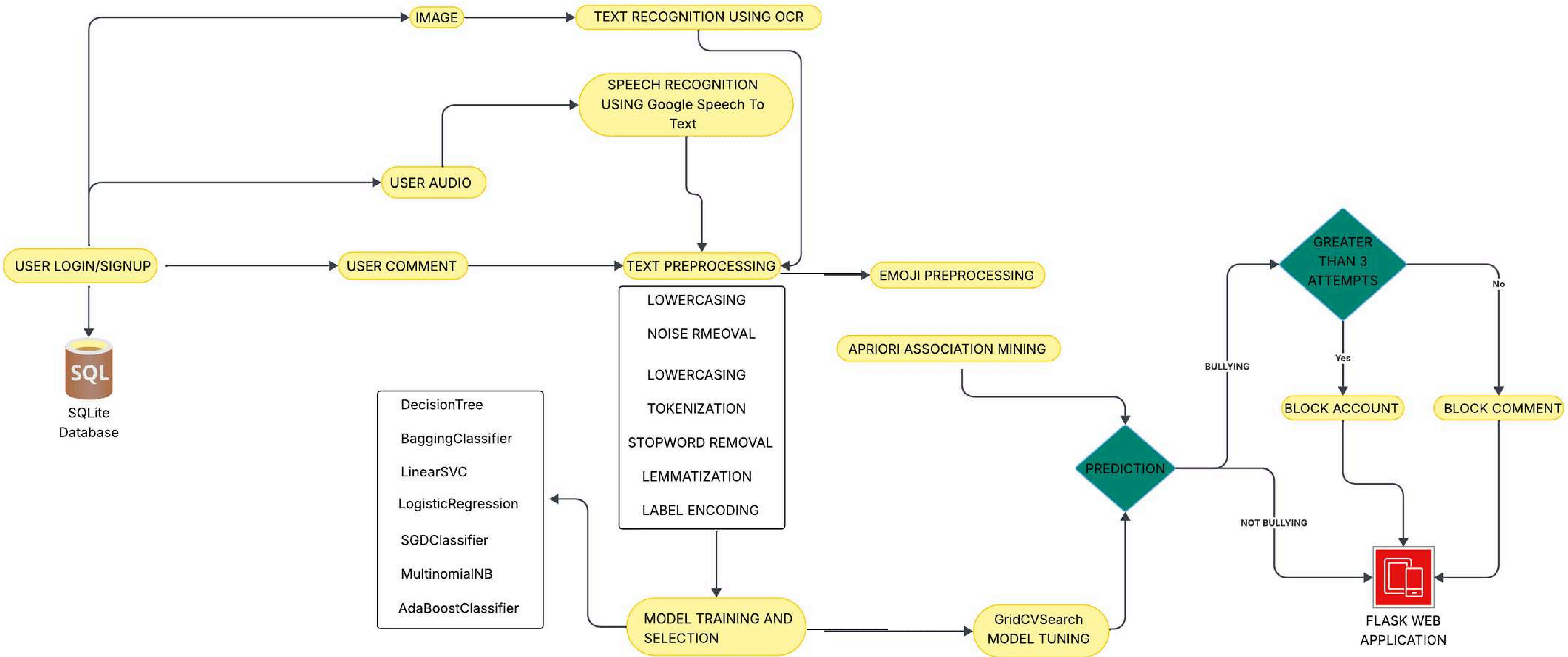


IMAGE
TEXT RECOGNITION USING OCR
SPEECH RECOGNITION USING Google Speech To Text
USER AUDIO
USER LOGIN/SIGNUP
USER COMMENT
TEXT PREPROCESSING
EMOJI PREPROCESSING
LOWERCASING
NOISE RMEOVAL
LOWERCASING
TOKENIZATION
STOPWORD REMOVAL
LEMMATIZATION
LABEL ENCODING
SQL
SQLite Database
DecisionTree
BaggingClassifier
LinearSVC
LogisticRegression
SGDClassifier
MultinomialNB
AdaBoostClassifier
MODEL TRAINING AND SELECTION
GridCVSearch MODEL TUNING
APRIORI ASSOCIATION MINING
PREDICTION
BULLYING
NOT BULLYING
GREATER THAN 3 ATTEMPTS
Yes
No
BLOCK ACCOUNT
BLOCK COMMENT
FLASK WEB APPLICATION