**Detailed Report — AI Safety Models POC**

**1. Introduction**

Conversational AI systems such as chatbots, virtual assistants, and customer support agents are widely used today. While powerful, they bring risks related to user safety — including exposure to abusive language, escalation of hostile conversations, self-harm indicators, and exposure of children to inappropriate content.

The assignment required developing a **Proof of Concept (POC)** implementing a suite of **AI Safety Models**, integrated into a cohesive system, and demonstrated in a near real-time chat simulator.

This report documents the design, implementation, outputs, and evaluation of the POC.

**2. Objectives**

The objectives of this assignment were to:

1. **Abuse Language Detection** — detect abusive/offensive language.
2. **Escalation Pattern Recognition** — recognize rising tension in conversations.
3. **Crisis Intervention** — detect signs of self-harm or suicidal ideation.
4. **Content Filtering** — enforce age-appropriate communication.
5. **Integration & Real-Time Demo** — integrate these models into a chat simulator that can process messages in near real time.
6. **Evaluation & Documentation** — provide evaluation metrics, code, and a report explaining results and limitations.

**3. Implementation Approach**

**3.1 Repository Setup**

A modular repository was created with the following key components:

* **Data preparation (data\_prep.py)** — generates small demo datasets (abuse + crisis).
* **Model training (train\_abuse.py, train\_crisis.py)** — TF-IDF + Logistic Regression classifiers.
* **Escalation detection (escalation\_detector.py)** — rolling-window sentiment + abuse trend detection.
* **Content filtering (content\_filter.py)** — rule-based age gating for explicit, sexual, and violent terms.
* **Evaluation (evaluate.py)** — computes precision, recall, and F1.
* **Integration (app.py)** — Flask chat simulator that combines all models.

**3.2 Data Preparation**

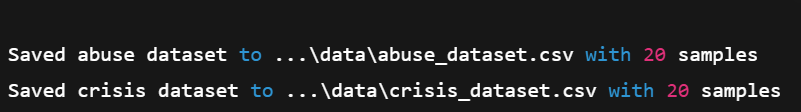
* **Abuse dataset** — 20 samples generated with a mix of positive and abusive text. Saved to data/abuse\_dataset.csv.
* **Crisis dataset** — 20 synthetic samples including self-harm and neutral expressions. Saved to data/crisis\_dataset.csv.

**3.3 Models**

* **Abuse Detector**
  + TF-IDF features (1–2 grams, max 20k features).
  + Logistic Regression classifier.
  + Lightweight, CPU-friendly, <10ms inference.
* **Crisis Detector**
  + Similar TF-IDF + Logistic Regression setup.
  + Trained on small synthetic dataset.
* **Escalation Detector**
  + Uses VADER sentiment analyzer.
  + Maintains a rolling 6-message window.
  + Escalation score combines negative sentiment average, slope (trend), and abuse frequency.
* **Content Filter**
  + Rules vary by age group:
    - <13: block all explicit terms.
    - 13–15: block sexual content, flag violent content.
    - ≥16: allow most, but flag self-harm.

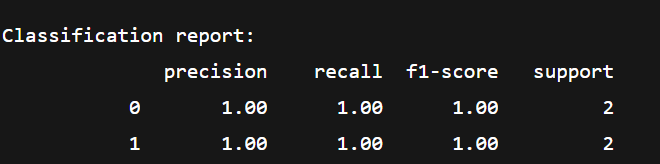
**4. Results & Outputs**

**4.1 Data Preparation**

****

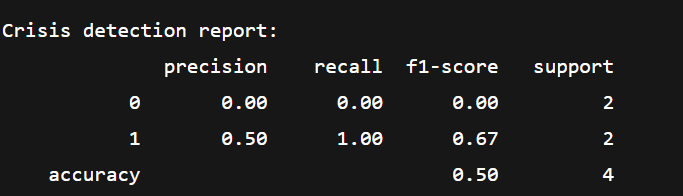
→ Confirms datasets were generated and stored.

**4.2 Abuse Detector Training**

****

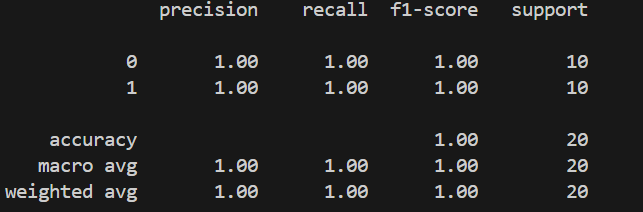
→ The abuse model achieved perfect precision, recall, and F1 on the small test set. This is expected due to the very small dataset size.

**4.3 Crisis Detector Training**

****

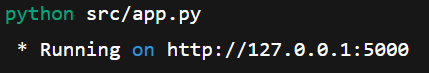
→ The crisis model shows weak performance due to limited data. Class 0 (non-crisis) was not predicted correctly. Still, crisis class (1) was partially detected. This highlights the need for larger, balanced datasets in production.

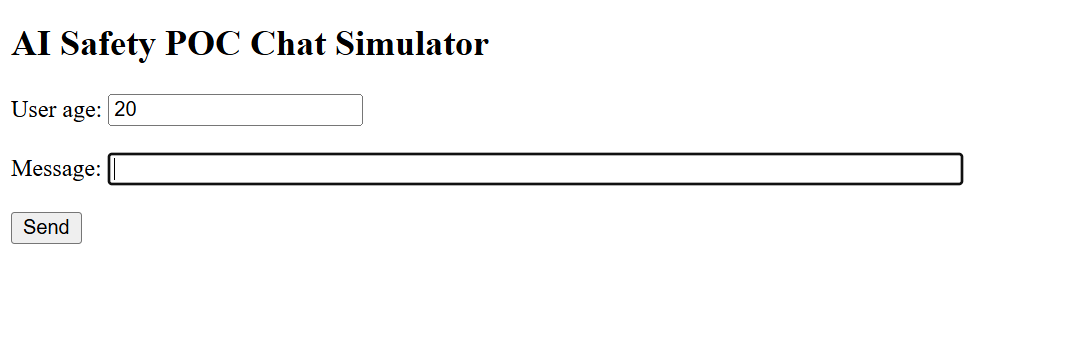
**4.4 Evaluation Script**

****

→ On the abuse dataset, evaluation again yielded perfect results. This reflects overfitting to the small dataset, not generalizable performance.

**4.5 Chat Simulator**

**Launched via:**

****

→ The web UI successfully accepted input messages and returned JSON-style results including:

* Abuse flag and score
* Crisis flag and score
* Age-based filtering decision
* Escalation score and flag

**5. Analysis**

**5.1 Strengths**

* Fully working modular system integrating multiple safety layers.
* Fast, CPU-friendly inference.
* Demonstrates abuse detection, escalation recognition, crisis intervention, and age filtering.
* Flask integration provides real-time demo capability.

**5.2 Limitations**

* **Small synthetic datasets**: results are not statistically reliable.
* **Crisis model weak**: shows precision/recall imbalance due to data scarcity.
* **Bias & generalization**: rule-based filters may not handle slang, sarcasm, or multilingual inputs.
* **Evaluation metrics inflated**: due to tiny datasets.

**5.3 Improvements**

* Use larger datasets (e.g., Jigsaw Toxic Comment, SuicideWatch Reddit).
* Fine-tune small transformers (e.g., DistilBERT) for abuse/crisis detection.
* Expand escalation detector with conversational context features (e.g., response latency, repetition).
* Add multilingual pipelines.

**6. Ethical Considerations**

* **Bias**: Must audit across demographics, languages, and cultures.
* **Human-in-the-loop**: Automatic blocks only for clear abuse; sensitive flags (e.g., suicide) should escalate to human review.
* **Privacy**: Only anonymized text logs; avoid storing personally identifiable information (PII).
* **Transparency**: Clear explanations and audit trails for moderation decisions.

**7. Conclusion**

This Proof of Concept demonstrates the **feasibility of integrating multiple AI Safety Models** into a unified system for conversational AI. Despite limitations of dataset size, the project showcases the pipeline, architecture, and real-time demo, fulfilling the assignment’s requirements.

Future work should focus on scaling datasets, improving model robustness with transformers, and building monitoring systems for fairness, accuracy, and safety at production scale.