**📝 Technical Report**

**Project Title: AI Safety POC – Multi-Model Moderation System**

**1. Introduction**

**With the rise of online platforms, user-generated content moderation is critical to ensure safety, prevent harm, and maintain community guidelines. This project builds a Proof of Concept (POC) for an AI Safety Chat Moderation System.**

**The system integrates four AI models trained on publicly available datasets to handle abuse detection, crisis intervention, emotional escalation, and age-appropriate content filtering. It combines these models into a single pipeline that evaluates messages in real time and outputs safety decisions.**

**2. Objectives**

* **Detect and flag abuse/toxic content.**
* **Recognize emotional escalation in dialogues.**
* **Identify crisis situations (suicidal ideation, self-harm).**
* **Ensure age-appropriate safety filtering.**
* **Build a chat pipeline that integrates all four models.**
* **Provide a demo chatbot interface (CLI & Streamlit).**

**3. Datasets Used**

**3.1 Abuse Detection (Toxic Content)**

* **Dataset: Jigsaw Toxic Comment Classification (Kaggle).**
* **Labels: toxic, severe\_toxic, obscene, threat, insult, identity\_hate.**
* **Size: 150k+ comments.**
* **Output: Multi-label classification.**

**3.2 Escalation Recognition (Emotion Detection)**

* **Dataset: DailyDialog + EmpatheticDialogues.**
* **Labels: Neutral, Anger, Disgust, Fear, Joy, Sadness, Surprise.**
* **Output: Multi-class classification.**

**3.3 Crisis Intervention**

* **Dataset: Suicide Detection Dataset (Reddit posts).**
* **Labels: sui / non-sui.**
* **Output: Binary classification.**

**3.4 Content Filtering (Age-Appropriate Safety)**

* **Dataset: Jigsaw dataset reused for Safe/Unsafe classification.**
* **Labels: Safe, Unsafe.**
* **Output: Binary classification (mapped for kid/teen/adult profiles).**

**4. Preprocessing**

* **Lowercasing text.**
* **Removing special characters & punctuations.**
* **Handling CSV-specific formatting issues (e.g., "[0 0 1]" → [0,0,1]).**
* **Splitting into train, validation, and test sets.**
* **TF-IDF vectorization (ngram\_range=(1,2), max\_features=5000–10000).**

**5. Model Training**

**Each module was trained separately using Scikit-learn Pipelines:**

**5.1 Abuse Detection**

* **Pipeline: TF-IDF → OneVsRestClassifier(Logistic Regression).**
* **Metric: Multi-label F1-score.**
* **Output: models/jigsaw\_pipeline.pkl.**

**5.2 Escalation Recognition**

* **Pipeline: TF-IDF → Logistic Regression (OvR).**
* **Classes: 7 emotion categories.**
* **Output: models/dailydialog\_emotion\_model.pkl.**

**5.3 Crisis Intervention**

* **Pipeline: TF-IDF → Logistic Regression (balanced weights).**
* **Classes: sui / non-sui.**
* **Output: models/crisis\_model.pkl.**

**5.4 Content Filtering**

* **Pipeline: TF-IDF → Logistic Regression.**
* **Classes: Safe / Unsafe.**
* **Output: models/content\_filter\_model.pkl.**

**All trained models and vectorizers are stored in ai\_safety\_poc/models/.**

**6. Integrated Chat Pipeline**

**A unified pipeline (src/inference/chat\_pipeline.py) integrates all models:**

**Workflow:**

1. **User sends a message.**
2. **Message is processed by all four models.**
3. **Results are combined in the Decision Engine:**
   * **Crisis detected → 🚨 Escalate to moderator.**
   * **Abuse detected → ⚠️ Warn/Block.**
   * **Unsafe content for age group → ❌ Block.**
   * **Otherwise → ✅ Allow.**

**Example Output:**

**User >> You are an idiot**

**--- Moderation Results ---**

**Abuse: {'flagged': True, 'categories': ['toxic', 'insult']}**

**Emotion: Neutral**

**Crisis: Not Crisis**

**Content Filter: {'status': 'Unsafe', 'decision': '❌ Blocked (Kids)'}**

**Final Action: ⚠️ Abuse detected → Warn/Block**

**7. Interfaces**

**7.1 CLI Interface**

* **Run python -m src.inference.chat\_pipeline.**
* **Interactive Q&A with real-time moderation outputs.**

**7.2 Chatbot (Streamlit Prototype)**

* **Run streamlit run src/inference/chatbot\_app.py.**
* **Select user profile (kid/teen/adult).**
* **Type messages and see moderation decisions.**
* **(Note: Pipeline faced NotFittedError due to sklearn version mismatch → fix requires retraining with consistent versions).**

**8. Pipeline (Text)**

1. **User Input → Preprocessing**
2. **TF-IDF Vectorization**
3. **Model Inference**
   * **Abuse Detection**
   * **Escalation Recognition**
   * **Crisis Detection**
   * **Content Filter**
4. **Decision Engine → Final Action**
5. **Output Response**

**9. High-Level Architecture (Text)**

* **Data Layer → Raw datasets (data/raw/...).**
* **Model Layer → Trained scikit-learn pipelines (models/\*.pkl).**
* **Application Layer → Training & inference scripts.**
* **Decision Layer → Combines predictions into policy decisions.**
* **Interface Layer → CLI (chat\_pipeline.py), Streamlit demo (chatbot\_app.py).**

**10. Results Summary**

* **Abuse Detection: F1 ≈ 0.67 (multi-label).**
* **Escalation Recognition: Macro F1 ≈ 0.23 (imbalanced dataset).**
* **Crisis Intervention: Accuracy ≈ 0.9.**
* **Content Filtering: Accuracy ≈ 0.92 (Safe vs Unsafe).**

**11. Limitations**

* **Imbalanced datasets reduced recall on minority classes (anger, threat, etc.).**
* **Streamlit chatbot had sklearn version mismatch issues (fixed by retraining under consistent versions).**
* **Emotion recognition is challenging → needs larger datasets.**

**12. Future Improvements**

* **Replace Logistic Regression with transformer-based models (BERT, RoBERTa).**
* **Use data augmentation to balance underrepresented classes.**
* **Deploy as a REST API for real-world integration.**
* **Add a real chatbot front-end (web or mobile).**
* **Implement continuous learning with feedback loop.**

**13. Conclusion**

**This POC successfully demonstrates an AI-driven safety moderation system that integrates four independent models (Abuse, Escalation, Crisis, Content Filtering) into a unified pipeline.**

**It can flag abuse, detect crises, monitor emotional escalation, and enforce age-based safety policies. With improvements (transformer models, deployment, richer UI), this system can serve as a robust foundation for responsible AI moderation in social platforms.**