When I set out to build **PeerSupportBot**, my goal wasn’t simply to “detect toxicity” in a generic sense it was to **understand the nuances of real conversations in peer support spaces**. Off-the-shelf models like DistilBERT or BERTweet are strong, but they’re broad. They can tell me *something is toxic or sarcastic*, but not with the **reliability, sensitivity, and context-awareness** that a supportive community requires.

I wanted my bot be context-aware and based on them make final calls whether to just log a sensitive/toxic message or redact and push warnings to create a safe environment.

That’s why I chose to **fine-tune two specialized models**:

1. **Toxicity detection (DistilBERT + LoRA)** on the Jigsaw dataset, so the bot could *reliably identify harassment, slurs, and threats*. Fine-tuning gave me sharper thresholds and better calibration compared to a general classifier.

2. **Sarcasm detection (BERTweet)**, because sarcasm is *the Achilles’ heel of moderation*. A message like *“oh sure, you’re a genius 🙄”* looks harmless to a generic toxicity model, but in context it’s cutting. By specializing on sarcasm, I reduced false positives and made the bot **less brittle in playful, informal environments**.

Task Specialization & Reliability

- **Task Specialization:**
  I deliberately narrowed each model’s responsibility: one model purely for toxicity dimensions, another purely for sarcasm. This modular design makes each model easier to evaluate and lets me combine their strengths in the policy layer.

- **Improved Reliability:**
  Fine-tuning on well-curated datasets ensured the models weren’t just “passable” at moderation but **trustworthy in high-stakes contexts** whether that meant catching crisis language (“I want to kill myself”) or extreme threats. Reliability here isn’t about perfection; it’s about being **predictably cautious**, always leaning toward user safety.

- **Adapted Style:**
  The communities I’m targeting use casual, slang-heavy, often ironic language. Generic models misfire in these settings, either over-moderating harmless banter or under-moderating subtle harassment. Fine-tuning allowed me to **adapt the models to the tone, style, and lived realities** of these conversations.

Human Impact

For me, the fine-tuning choice wasn’t only about model metrics it was about **human outcomes**:

- Reducing frustration for well-meaning users by lowering false positives.
- Responding faster and more confidently to harassment or crisis language.
- Creating moderation that feels **empathetic, not robotic** because the bot “gets” the difference between *“this exam is stupid”* and *“you are stupid.”*

In short: I chose to fine-tune because my goal wasn’t just classification accuracy it was to **earn trust** in sensitive spaces, where missing a crisis or wrongly flagging a message can both do real harm. Task specialization, reliability, and adaptation to community style are what make PeerSupportBot more than “just another moderation bot.”