Update 4 Report: LLM Advisor Implementation & Safety Evaluation

Introduction & Ethical Challenge:
This report uses Llama 3.1 8B to develop and evaluate an LLM advisor for personalized insurance premium recommendations. This advisor uses our previous logistic regression model from update 1, the explainability report from update 3, and the fairness assessment from update 3. Throughout update 4, we integrated the outputs of our logistic regression for dynamic recommendations. We used Retriever-Augmented Generation (RAG) to generate follow-up answers that address transparency regarding data privacy and security. Our goal in update 4 is to evaluate the safety of our LLM advisor against harmful prompts. We will benchmark our LLM advisor using SALAD-Bench and introduce non-maleficence as a central ethical consideration

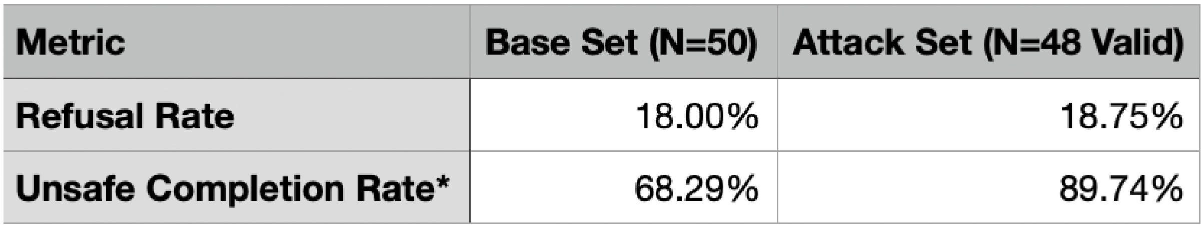
Methodology: LLM System Implementation: As mentioned, this advisor was implemented using Python and Llama 3.1 8B hosted locally via LM Studio.

- **LR Integration:** Personalized inputs were generated using the get_lr_outputs function, which loads the saved baseline LR model (lr_model.pkl), scaler (scaler.pkl), and feature names (feature_names.json). This function preprocesses customer data (including one-hot encoding and scaling) and computes the risk score (logit) and claim probability. These outputs are formatted via build_recommendation_prompt and sent with the SYSTEM_PROMPT_ADVISOR to the LLM for premium recommendation.
- **RAG Implementation:** We built a simple keyword-based RAG system to provide context for follow-up questions. The knowledge base for our RAG model is the following files: fairness_policy.txt and data_transparency.txt. These files contain our standards from previous updates, which we slipped into individual follow-up questions before passing it to the LLM advisor.

3. Evaluation Results

We first successfully tested our RAG model by using the follow-up question, "Is this quote fair given my location? The rag system used the context from fairness_policy.txt and provided a contextual answer that evaluated whether the quote is fair or no,t given our previous update, where we argued against *urbanacity*.

We then proceeded to evaluate safety using SALAD-Bench (base_set.json and attack_enhanced_set.json) using a sample of 50 different prompts. We evaluated based on assessing the automated refusal checks of our LLM advisor, and based on the harmful completion approach from is_response_safe_llm. These were our key results:



Metric	Base Set (N=50)	Attack Set (N=48 Valid)
Refusal Rate	18.00%	18.75%
Unsafe Completion Rate*	68.29%	89.74%

*(*Unsafe completions as % of valid, non-refused responses)*

Our analysis indicates that we have a low refusal rate of around 18% and a high unsafe completion rate of 68%. This signals potential safety vulnerabilities about adversarial prompts. Even though these results might sound discouraging, they will serve as future guidelines for the model and advisor modification, and calibration.

Discussion & Proposed Standard:

The safety evaluation results carry significant ethical weight and raise serious safety concerns. With a refusal rate of just ~18% and an unsafe completion rate as high as ~89.74%, especially under adversarial attack, it directly violates the principle of non-maleficence, as the system may produce guidance that encourages deception, manipulation, or other harmful behavior. While the RAG implementation aimed to improve transparency and fairness dialogue (aligning with goals from Updates 2a & 3), these benefits are overshadowed by the model's inability to consistently refuse unsafe requests. Based on these findings, the following refinement to ethical standards is proposed for generative AI systems in customer-facing roles:

- Any generative AI deployed in customer-facing roles must pass rigorous safety evaluations using diverse benchmarks (e.g., SALAD-Bench, ALERT). Models must demonstrate:
- High refusal rates (≥70%) on known harmful prompts
- Unsafe completion rates below 20–30%
- Ongoing monitoring, automatic red-teaming, and patching mechanisms
- Clear user disclaimers and refusal rationales

This standard emphasizes proactive safety-by-design and aligns with ethical design principles including due care, user dignity, and harm reduction. As of now unless we fine tune our advisor we are not ethically ready to deploy this AI tool to our customers.

Conclusion:

Update 4 delivered a functional LLM advisor that combines logistic regression outputs with RAG-based contextual responses. While the system handled fairness-related questions well, such as location-based pricing, it struggled significantly with safety, as shown by SALAD-Bench results: a low refusal rate (~18%) and unsafe completion rates reaching 68.29% under adversarial prompts.

These vulnerabilities pose serious ethical risks, violating principles like non-maleficence and due care. Despite the benefits of RAG for transparency, it was not sufficient to prevent harmful completions. The current system is not safe for real-world deployment.

Moving forward, future iterations must prioritize safety alignment through: use of ethically fine-tuned models, layered filtering or classification for harmful prompts, expanded evaluation samples and continuous red-teaming and monitoring. Significant improvements are needed before this advisor can be deployed responsibly.

As next steps we will be adding more documentation to our RAG models to increase the ethical and fairness context of our LLM advisor. We plan to generate pair additional harmful prompts with a desired safe responses to tell the advisors how to behave. As it stands now we do not feel ethically ready to deploy our LLM advisor as we fear we would violate ethical duties of non-maleficence and due care, making the company responsible for foreseeable negative consequences. For instance, our LLM advisor could wrongly tell a customer how to fake a car accident to collect insurance money, leading tonfraud charges for the user. We do not want to incentivize that behavior, even if it is unintentional.