**Reddit Insights Chatbot with RAG  
MSDS692 – Data Science Practicum 1**

**Jonish Bishwakarma**

**Progress Report for Week6**

**Project Details**This project aims to build a research tool that uses Reddit discussions to surface industry-specific insights, particularly about commonly used software and related pain points in law firms, construction, and tech. Posts and comments are scraped from selected subreddits, cleaned, and analyzed for software/tool mentions. A Retrieval-Augmented Generation (RAG) chatbot will later be developed so users can query insights conversationally. The project also compares RAG performance with an LLM-only baseline.

**Project Timeline:**

Week 1 – Project definition and submit proposal (DONE)

Week 2 – Data collection (DONE)

Week 3 – Data cleaning, preprocessing, and initial keyword filtering (DONE)

Week 4 – Build software dictionary and extract mentions (DONE)

Week 5 – Sentiment analysis and identification of pain points (DONE)

Week 6 – Develop and test RAG + LLM pipelines (DONE)

Week 7 – Evaluation of RAG vs LLM and refinement (Pending)

Week 8 – Final analysis, presentation preparation, and submission (Pending)

**Planned Work for the Week:**  
This week’s primary goal was to integrate the cleaned dataset into a working RAG pipeline and test chatbot functionality. Specific tasks included setting up document chunking, embedding generation, and Pinecone vector storage, followed by connecting the retrieval chain to an OpenAI LLM for contextual responses. The plan also included developing a simple web interface to query the chatbot and test retrieval performance.

**Progress for the Week:**

* Built RAG pipeline integrating LangChain, HuggingFace embeddings, and Pinecone vector database.
* Created retrieval chain combining context-aware search with OpenAI GPT-3.5-turbo responses.
* Developed and deployed a Flask-based web chatbot interface with a responsive front-end design.
* Integrated real-time streaming logs to monitor progress across different pipeline stages (data collection, cleaning, sentiment, indexing).
* Added buttons for sequential execution of all pipeline steps, improving user interactivity.
* Conducted initial RAG accuracy checks using test prompts.

**Roadblocks/Issues:**

* Long execution times during data collection caused browser requests to time out; addressed by adding live log streaming via server-sent events (SSE).
* Some buffering issues in subprocess logs persist, causing delayed updates for long-running scripts.
* Need for better feedback mechanisms (e.g., progress indicators or log tailing) during extended scraping sessions.
* Limited context granularity in certain Reddit posts occasionally leads to incomplete answers from the RAG model.

**Plan for Next Week:**

* Evaluate and compare RAG vs direct LLM performance using a fixed question set.
* Analyze retrieval accuracy and contextual relevance metrics (e.g., precision/recall of correct tool mentions).
* Begin preparing documentation and visual components for the final presentation.

**Resources for the Week:**

Utilized LangChain, Pinecone, Flask, OpenAI API, and HuggingFace embeddings. Existing datasets and scripts were refactored for modular execution (data\_collection.py, data\_clean.py, data\_sentiment.py, and store\_index.py).

nxr-deen. (2024). *GitHub - nxr-deen/AI-Chatbot: AI Chatbot in HTML CSS & JavaScript*. GitHub. https://github.com/nxr-deen/AI-Chatbot