KPMG

Data Science Institute
COLUMBIA UNIVERSITY

Spring 2025 Capstone Project: Midterm Progress Report

March 16, 2025

Model Freddie Mac data Chatbot with GenAI

Members: Yue Yang, Xinyue Zhang, Jie Hu, Han Qiang

Mentors: Jim Leach, Ben Carper

Data Science Institute Mentor: Sining Chen

Table of Contents:

1. Executive Summary

2. Introduction

a. Project Motivation and Scope

b. Past Project Overview

c. Current Project Status

d. Project Roadmap

3. Data Exploration and Initial Setup

Neo4j Database

4.Conclusion and next steps

Executive Summary

Background and Objective

The Chatbot project for KPMG aims to enhance the accessibility and accuracy of risk data by leveraging retrieval-augmented generation (RAG) and GenAI technologies. The project builds on existing capabilities in handling metadata but expands the scope to summarize and provide insights on risk-relevant data from Freddie Mac and controls. The goal is to deliver a chatbot that enables business users to efficiently query and understand complex risk data through a natural language interface. This solution will improve how risk data is organized, connected, and analyzed, empowering Freddie Mac to identify patterns and manage risks more effectively. The initiative includes refining the chatbot's ability to handle uncommon queries, improving response relevance, and increasing overall robustness.

Project Goals

- Enhance the chatbot's ability to retrieve relevant risk data from the database efficiently and accurately.
- Using embeddings, improve the chatbot’s intent-matching capabilities. This will help better classify and match user queries related to risk data, increasing the accuracy and relevance of generated responses.
- Deploy GraphRAG and other advanced retrieval methods to handle complex queries by leveraging graph-based data storage, ensuring more precise and contextually accurate answers.
- Refine and enhance the user interface to improve usability and user experience.
- Develop a structured framework to evaluate chatbot performance and measure improvements in response accuracy, relevance, and user satisfaction.

Technical Indicators

- Enhanced intent-matching process using embeddings and similarity measures.
- Optimized the prompt for LLM to extract data needed to answer the questions.
- Integration of GraphRAG workflow to leverage graph-based context for answering complex or uncommon queries.
- Development of a structured evaluation framework to measure chatbot performance.
- Integrate the updated workflow within the existing Streamlit interface.

Business Value & KPIs

- Improved efficiency when analyzing and interpreting risk-related data.
- Consistent and reliable access to accurate and relevant risk control information.
- Reduced time and effort required to gather insights from complex risk data.
- Generate a summary report on risks' information efficiently.

Milestones

- Development environment replicated
- Graph database initialized
- Experimentation on various prompts to generate queries (TBA)
- Generating a summary report for risk data (TBA)
- Research and develop model (TBA)
- Deployment of improved chatbot and assessment of performance (TBA)

Introduction

Project Motivation and Scope

Effective risk management is critical for businesses navigating complex and dynamic environments while aiming to minimize exposure to potential threats. Accurate and accessible risk data empowers organizations to identify patterns, assess potential risks, and make informed decisions. However, the sheer volume and complexity of risk-related data often make it difficult for business users to extract meaningful insights efficiently. This project aims to address this challenge by enhancing a chatbot designed to summarize and provide insights on risk-relevant data and controls through a natural language interface, improving both the accuracy and relevance of responses.

This project builds on a previous initiative focused on using retrieval-augmented generation (RAG) and GraphRAG to enhance metadata accessibility and improve response accuracy. The previous project created a chatbot that allowed users to query metadata about predictive models and reports in natural language. Our task is to modify the previous code to support Freddie Mac's risk and control data and to improve the chatbot's ability to provide accurate, relevant, and insightful answers, particularly for complex and uncommon queries related to risk controls for real data from Freddie Mac.

We have identified a few key ways to change from the previous application:

- **Prompt Generation:** We plan to create more effective prompts based on real data rather than relying solely on pre-written templates. By designing prompts that reflect actual data patterns, the chatbot will be better equipped to understand and respond to complex user queries.
- **Query Generation:** We aim to generate queries dynamically using real data instead of static templates. This will allow the chatbot to retrieve more accurate and contextually relevant information, improving the overall quality of responses.
- **Model Adjustment:** The original model was designed based on a template-driven approach. We aim to modify the underlying model to adapt to the structure and characteristics of the real data, allowing for more dynamic and flexible responses to

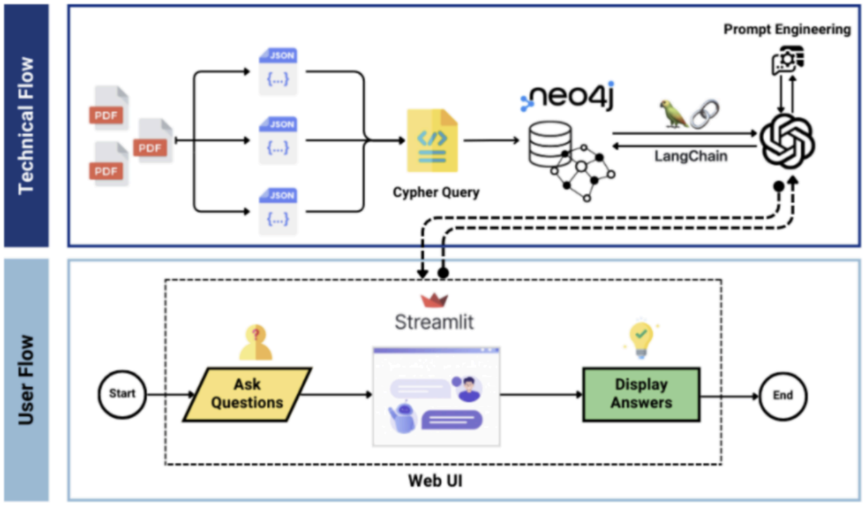
complex queries.

- **Risk Profile Summary Generation**: We will implement a system to automatically generate structured Risk Profile Summaries for different levels of risk. These summaries will follow a standardized format, capturing key details such as residual risk levels, control effectiveness, self-testing coverage, and any notable issues or profile impacts. This will ensure consistency in reporting and provide stakeholders with a clear, data-driven assessment of risk across different domains.
- **Evaluation Framework:** To measure the success of our improvements, we will develop a structured evaluation framework based on real data. This will allow us to assess chatbot performance based on accuracy, relevance, and user satisfaction, ensuring that our optimizations lead to measurable improvements.

Past Project Overview

Our project builds on the work of a previous team that developed a chatbot designed to retrieve and summarize information from structured data using retrieval-augmented generation (RAG) and a graph-based database. The previous project focused on creating a chatbot that could accurately respond to user queries about predictive models and reports by leveraging embeddings, graph-based data storage, and large language models (LLMs).

The backend process in the original project began with mock data generation. Real PDF reports were converted into Cypher queries using a JSON schema to preserve the structure and content of the original documents. This structured data was then integrated into a Neo4j graph database to enable efficient relationship-based querying at scale. The chatbot itself was developed using LangChain, which connected the Neo4j database to ChatGPT via the OpenAI API. This setup allowed the chatbot to process user queries, retrieve relevant information from the database, and generate natural language responses.



On the user side, the chatbot was deployed through a Streamlit Web UI. When a user submitted a query, the chatbot used the ChatGPT API to interpret the input and classify the question as "Common," "Uncommon," or "Irrelevant." If the question was deemed irrelevant, the chatbot returned an appropriate response automatically. For common queries, specific keywords were extracted and inserted into pre-defined Cypher queries to generate answers quickly. For uncommon queries, the chatbot used a RAG-based system to search the Neo4j graph database for the most relevant information and generate an appropriate response based on the retrieved context.

Current Project Status

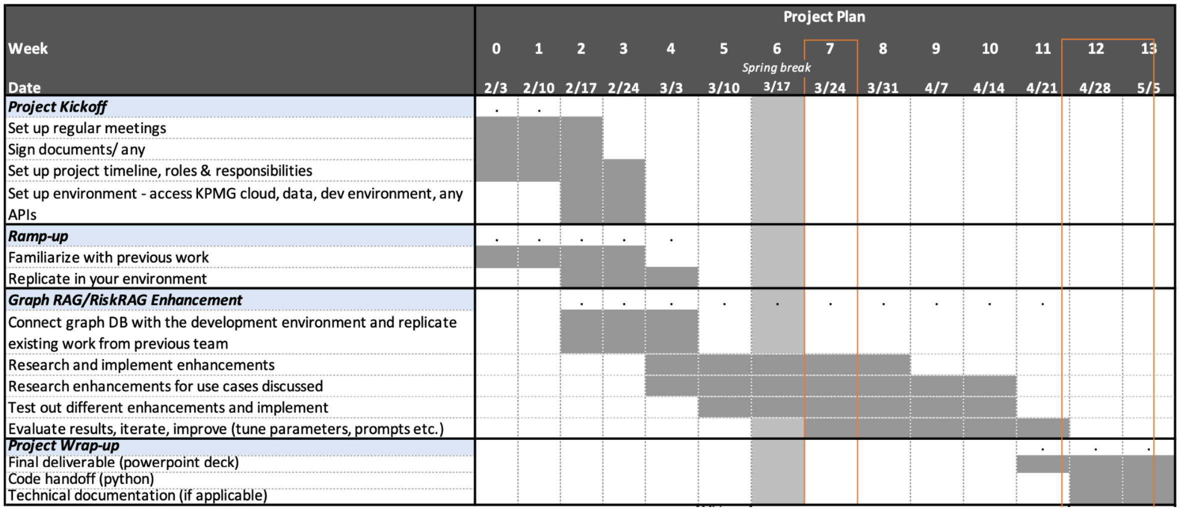
By the end of last semester, the previous team was able to create a Chatbot application with relatively high accuracy. Based on the previous work, we have done the following:

- Replicated the previous code and environment
- Understand approach & technical challenges(LangChian, LLM, Prompt Engineer)
- Imported the data to Neo4j
- Got access and set up VSCode in Virtual Desktop Infrastructure(VDI)
- Explored the real data using queries
- Extracted useful information for prompt engineering common questions

- Did experiments on prompts to generate queries for getting useful data for common questions(e.g, What is the risk profile of Cybersecurity Risk?) by using the OpenAI API

Project Roadmap

The Gantt Chart below outlines our tentative schedule for the semester. We've completed the initial project setup and begun working on query generation. However, we've encountered compatibility issues when trying to replicate past work in our VDI environment. Since resolving these issues may require significant time and effort from both our side and the KPMG side, we are considering starting the code from scratch while still taking the frameworks and insights from past teams. Our goal is to develop a functional summary chatbot application by the end of the semester.



	Project Plan													
Week	0	1	2	3	4	5	6	7	8	9	10	11	12	13
							Spring break							
Date	2/3	2/10	2/17	2/24	3/3	3/10	3/17	3/24	3/31	4/7	4/14	4/21	4/28	5/5
Project Kickoff	·	·					░							
Set up regular meetings	■	■	■				░							
Sign documents/ any	■	■	■				░							
Set up project timeline, roles & responsibilities	■	■	■	■			░							
Set up environment - access KPMG cloud, data, dev environment, any APIs			■	■			░							
Ramp-up	·	·	·	·	·		░							
Familiarize with previous work	■	■	■	■			░							
Replicate in your environment			■	■	■		░							
Graph RAG/RiskRAG Enhancement			·	·	·	·	·	·	·	·	·	·		
Connect graph DB with the development environment and replicate existing work from previous team			■	■	■		░							
Research and implement enhancements					■	■	■	■	■					
Research enhancements for use cases discussed					■	■	■	■	■	■	■			
Test out different enhancements and implement						■	■	■	■	■	■			
Evaluate results, iterate, improve (tune parameters, prompts etc.)							░	■	■	■	■	■		
Project Wrap-up							░					·	·	·
Final deliverable (powerpoint deck)							░					■	■	■
Code handoff (python)							░						■	■
Technical documentation (if applicable)							░						■	■

Data Exploration and Initial Setup

Neo4j database from freddie mac

We successfully connected our Neo4j database, leveraging Freddie Mac’s real risk data as the foundation for storing and managing relational information. We use Neo4j to track the relationships within risk data, ensuring accurate risk analysis. Using the credentials provided in the Neo4j console, we configured the database and established the necessary schema structures, enabling seamless integration and robust support for chatbot queries.

Due to confidentiality agreements, we do not display the data here.