Title: DSCI 551 Final Project Proposal

Team Details (ChatDB 17)

Members: Qingyi Feng & Chuqi(Angel) Jin

Background and Skills:

Qingyi Feng **(USC ID:8401362266 Email:qingyife@usc.edu)**

This is Qingyi Feng. During my undergraduate studies, I gained experience with basic programming languages, including R and Python. I also participated in several group projects that strengthened my problem-solving and collaboration skills. One of the most significant projects I worked on involved developing an interpreter for a subset of the Scheme language. Throughout this project, my team carefully considered various challenges in programming language design, ensuring that our implementation decisions for both the interpreter and compiler directly influenced the language's behavior and properties. While I have not learned SQL or NoSQL yet, I am eager to explore database management systems. Before now, I had no experience with MongoDB or Firebase. But, I am always open to learning new technologies that will enhance my programming and data-handling capabilities. Last semester, I took DSCI 552, where I studied machine learning techniques, including classification methods and tree-based algorithms. This course provided me with hands-on experience in implementing models and analyzing data, strengthening my analytical and computational thinking skills. Additionally, I worked on projects that required applying these concepts to real-world datasets, which helped me develop a more intuitive understanding of machine learning principles.

Chuqi (Angel) Jin (**USC ID: 9740105151 Email: chuqijin@usc.edu*****)***

I am a first-year applied data science master's student at USC. I have a bachelor's degree in statistics with a computational concentration and a minor in accounting from UCR. Last summer, I graduated with a variety of project experience and research skills in statistical analysis, data analysis, and data science. Some of the classes I've taken in recent years include probability and statistical theories, regression analysis, statistical computing, applied optimization, numerical analysis, introduction to data structure and algorithms, and data science using R, SAS, Python, and C++. My expertise is in R for massive dataset analysis and regression model building. The majority of projects include data cleaning, combining, organizing, exploratory data analysis, exploration of interaction terms, variable transformation, model building, model comparison and residual diagnosis. Furthermore, I took DSCI 552 last semester, which taught me more about fundamental machine learning methods. However, prior to taking this class, I had very little understanding of database systems and management, and I only knew the very basics of SQL and knew nothing about NoSQL. So, I believe this project will be a little challenging for me, but I'm prepared to learn throughout the semester and overcome each challenge one by one in order to lay the solid foundation for my future studies.

Project Requirements:

As a two-person team, we need to choose a SQL database and a NoSQL database to complete the following tasks. First, we must select a real-world dataset, design and construct a well-structured database model using code, and store the dataset in separate SQL and NoSQL databases. Both databases should allow users to ask questions to retrieve statistical information/insights from our dataset using natural language. To achieve this, we need to use a large language model API to process those natural language questions and interpret and convert the user's questions into database commands using code. Following that, our code will also execute the generated commands on each database, retrieve the requested information/data, and return the results to the user. Furthermore, we should allow users to edit our structured database by writing additional commands, such as updating, adding, and deleting existing data.

In addition to completing the coding and database part, we have specific assignments to report our project progress. These include a proposal outlining the general project plan before starting our coding, a midterm progress report to update our project status and challenges faced, an in-class demonstration of our work, a final report to explain in detail our project steps and learning outcomes, and finally, an implementation to upload our code and supporting materials.

Planned Implementation

- **RDBMS (SQL-based):** MySQL
- **NoSQL:** MongoDB
- **Language Model API:** GPT-3.5 Turbo or Llama 2(Meta)
- **Dataset:** Apartment Management
  https://www.kaggle.com/datasets/adithyaawati/apartments-for-rent-classifie
  This dataset contains 22 variables with 100000 observations. Variables include both numerical and categorical variables, including id, category, title, body, amenities, bathrooms, bedrooms, currency, fee, has_photo, pets_allowed, price, price_display, price_type, square_feet, address, cityname, state, latitude, longitude, source and time.

  We plan to clean the dataset by removing observations with empty or null value in some of the variables, ensuring that each observation contains complete information. Also, we plan to reorganize the data to improve usability. Then, we will filter apartments based on US regions and select one or two regions, like the west and east, to design our database structure. Moreover, we will delete certain variables to reduce the data size. Hopefully, we can reduce the dataset size to a manageable range for easy implementation.

  We plan to organize our dataset into three tables:
  1. **Apartments basic information**: id, location, size, price, category
  2. **Apartments facilities**: amenities, bathrooms, bedrooms
  3. **Additional information about the apartments**: has_photo, pets_allowed

This system will support the below function:

- Schema and Data Exploration
    - Users can ask questions such as
        - "Show me the attributes of the apartment table"
        - "Give me a sample of apartment in LA"
        - "What is the highest and lowest priced apartment?"
        - "How many apartments in LA allow me to live with my pets?"
- Query Processing
    - Convert user queries into SQL or MongoDB queries
    - Supported SQL operations
        - Like SELECT, FROM, WHERE, etc.
    - Supported MongoDB operations
        - Like find(), aggregates(), etc.

- **Natural Language Processing**
    - Entity Recognition: Extract keywords
    - Synonym Matching
        - Like "cheap apartments" instead of "low-price apartment"
    - Conversational Context Memory
        - Users ask "show me apartments in LA" and then "What's the price range?" the system should understand "price range of apartment in LA"

Team responsibilities:

Since we only have two people on the team, we decided to collaborate and equally share our responsibilities throughout the entire process. Both of us will be responsible for database management and backend development. We will work together to design the database structure for both SQL and NoSQL databases, clean and organize the dataset, integrate an AI API to help retrieve data from our built structure, and write code together to develop the project. Finally, we'll test and debug our code together before writing our final reports.

Timeline:

Date	**Phase**	**Checkpoints**
Due 02/14	Data Cleaning	• Collect, check, and clean data. • Remove observations with empty values.

		• Reorganize the dataset, ensuring that each variable value is in its appropriate column.
Due 02/21	Data Cleaning	• Remove unimportant variables. • Select and filter a subset before proceeding with the next phases.
Due 02/28	LLM Setup Data Structure (mySQL)	• Start writing the LLM test prompt. • Learn how to turn text into SQL and NoSQL queries. • Design MySQL schema • Organize the data into tables. • Code out the queries command.
Due 03/07	Data Structure (MongoDB)	• Define the NoSQL schema • Optimize queries
Due 03/14	Data Structure (mySQL & MongoDB)	• Complete the code for data entry in both databases. • Examine the produced database structures in both databases. • Test data retrieval from both databases.
Due 03/17	Submission	Midterm Progress Report
Due 03/21	LLMs	• Use the LLM API to

		process natural language queries. ● Fine-tune prompt engineering
Due 03/28	LLMs	● Improve our written code based on the test results from the previous phase. ● In both databases, Test LLM generated SQL and NoSQL queries.
Due 04/04	LLMs	● Edit the code to obtain more improvements. ● Test the entire system.
Due 04/11	Implementation	● Unit Testing ● Debugging
Due 04/18	Performance optimization	● Optimize query execution time for large datasets ● Improve database
Due 04/20	Implementation	● Run the code to see its overall performance
Due 04/21	Presentation	Demo (In-class)
Due 04/21	Presentation	Demo (In-class)
Due 04/25	Final Report	● Report writing
Due 05/02	Final Report	● Test system ● Provide link to the code repository and necessary documentation
Due 05/09	Submission	Final report submission