Report on Gemini App for SQL Data Retrieval

-Prarthana Ganesh Shetty
002833314

Introduction

Purpose

The Gemini App leverages advanced natural language processing (NLP) to facilitate querying a SQL database without the need for direct SQL command knowledge. This streamlines data retrieval for users who may not be familiar with SQL syntax.

Database Structure

The **loan_db** database comprises various columns such as **loan_id**, **no_of_dependents**, **education**, and more, which encapsulate information pertinent to loan applications and statuses.

- **Data Set**

The dataset encompasses a range of attributes pertinent to loans and borrowers, designed to facilitate the prediction of loan approval or rejection. It includes information such as loan identification numbers, the number of dependents associated with each borrower, educational levels, self-employment status, annual income, loan amounts, loan terms, credit scores, and various asset values possessed by borrowers. Notably, the dataset also includes a crucial feature denoting the loan status, with values Approved and Rejected, indicating whether their loans were approved or rejected. With a focus on predicting loan outcomes, this dataset provides a comprehensive array of borrower and loan characteristics essential for predictive modeling and risk assessment within lending institutions.

- **Variable Description**

The dataset contains information about loans and borrowers, with a total of 13 attributes and 4269 rows:

loan_id: Identification number for each loan (numerical)

no_of_dependents: Number of dependents of the borrower (numerical)

education: Level of education of the borrower (categorical)

self_employed: Indicates whether the borrower is self-em

ployed or not (categorical)

income_annum: Annual income of the borrower (numerical).

loan_amount: Amount of the loan (numerical)

loan_term: Term of the loan (numerical)

cibil_score: Credit score of the borrower (numerical)

residential_assets_value: Value of residential assets owned by the borrower (numerical)

commercial_assets_value: Value of commercial assets owned by the borrower (numerical)

luxury_assets_value: Value of luxury assets owned by the borrower (numerical).

bank_asset_value: Value of assets held in bank accounts by the borrower (numerical)

loan_status: Status of the loan, Approved or Rejected

Application Design and Workflow

Architecture

The app's architecture is a client-server model where the Streamlit interface serves as the client and the Python backend interacts with the SQL database.

Workflow

The user inputs a query in natural language, which is then interpreted by the Gemini model to formulate a corresponding SQL query. This SQL command is executed against the **loan_db** database, and results are returned to the user.

Conversion Process

The Gemini model receives the user's natural language query as input and outputs a SQL statement. It does so by understanding the intent behind the query and mapping it to the relevant SQL commands and database schema.

Streamlit Interface

Streamlit provides a simple and effective way to create a user-friendly interface that captures user inputs and displays outputs. It handles input validation and provides a seamless user experience.

Challenges and Solutions

Model Training and Accuracy

The initial challenge was ensuring the Gemini model correctly interpreted a variety of natural language queries.

Solution

To address this, extensive training data was prepared, encompassing various phrasings and terms used in typical SQL queries.

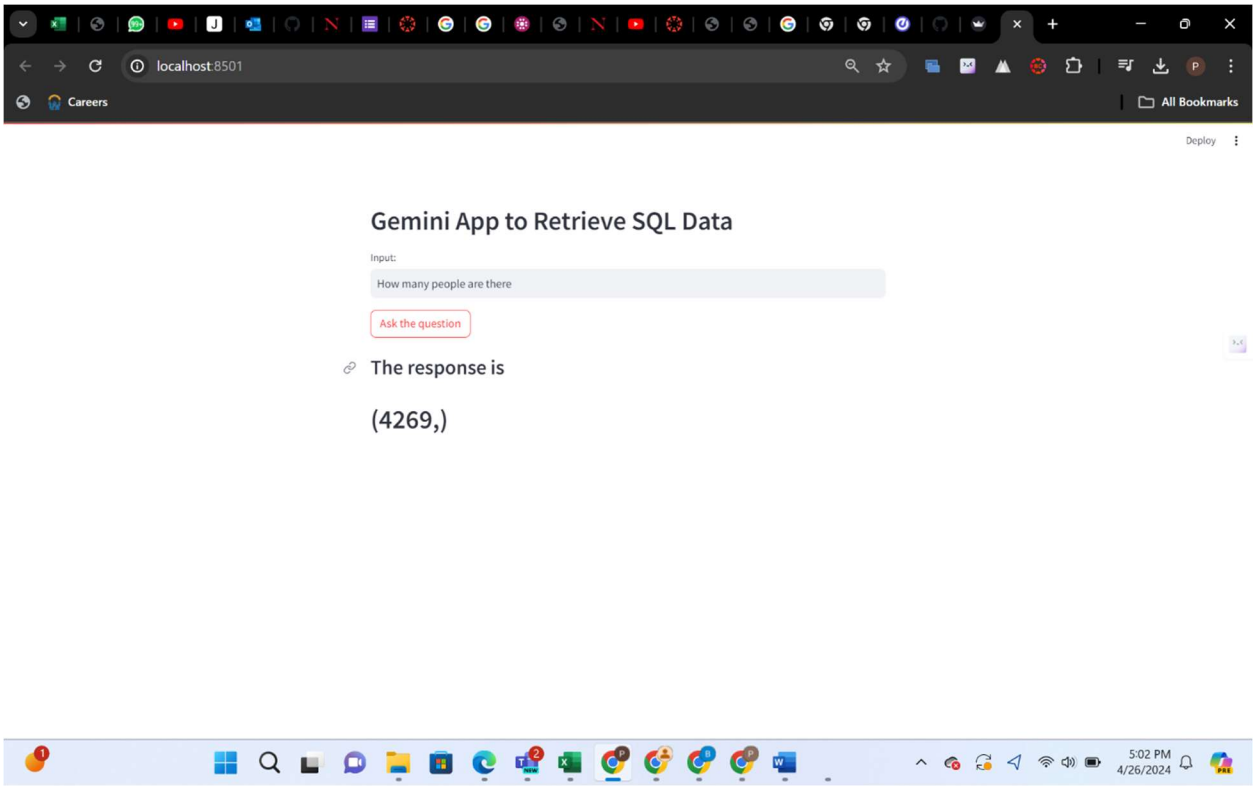
Integration with Streamlit

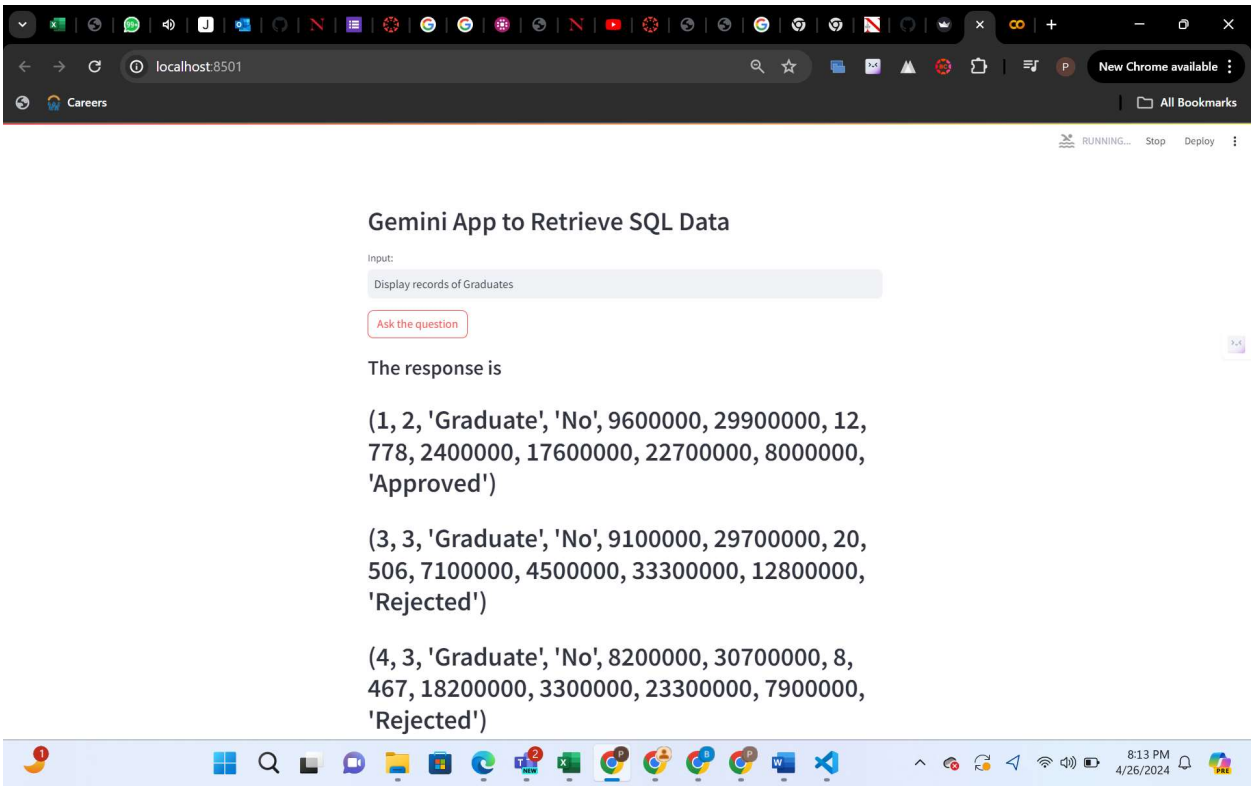
Another challenge was integrating the model's output with Streamlit's input components and ensuring synchronization between the natural language processing backend and the frontend interface.

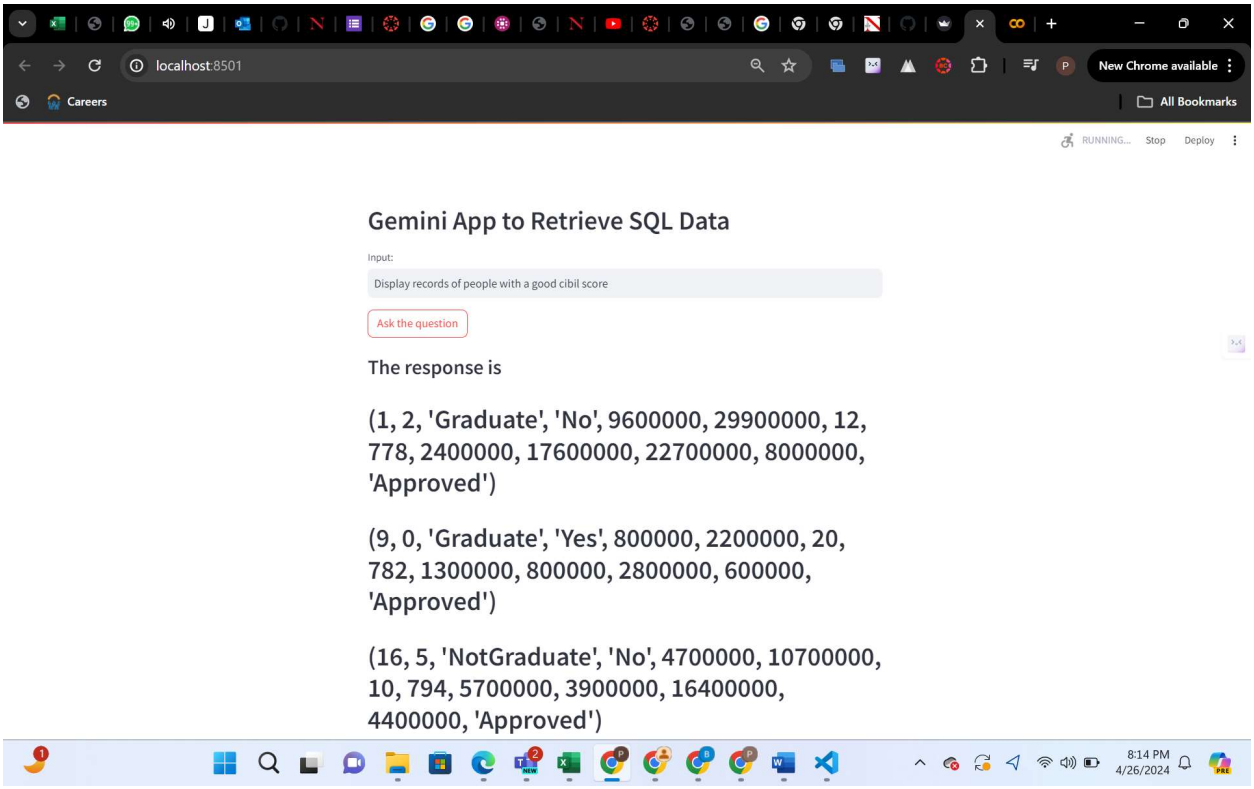
Solution

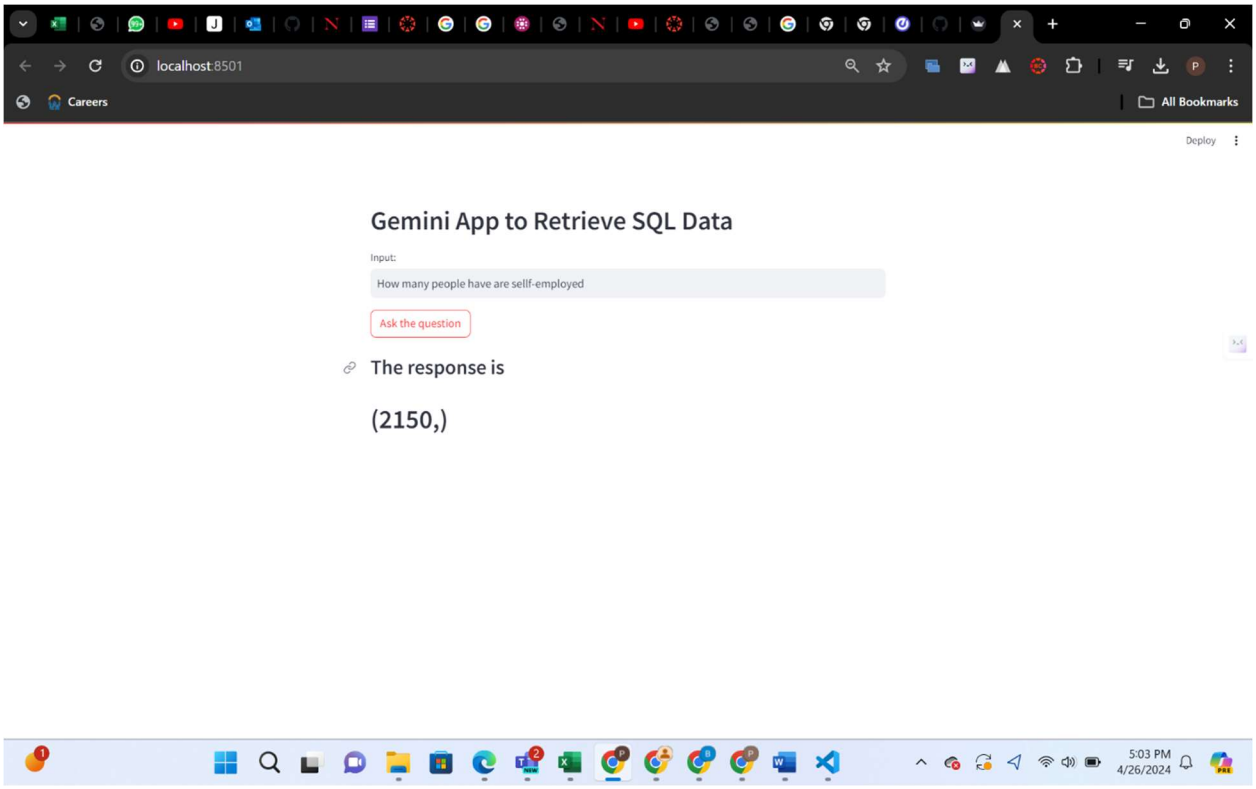
The backend was configured to trigger upon a button press in the Streamlit app, ensuring that the database is only queried when required, thus optimizing performance.

Demo Screenshots






localhost:8501
New Chrome available
Careers
All Bookmarks
RUNNING...
Stop
Deploy
Gemini App to Retrieve SQL Data
Input:
Display records of people with a good cibil score
Ask the question
The response is
(1, 2, 'Graduate', 'No', 9600000, 29900000, 12, 778, 2400000, 17600000, 22700000, 8000000, 'Approved')
(9, 0, 'Graduate', 'Yes', 800000, 2200000, 20, 782, 1300000, 800000, 2800000, 600000, 'Approved')
(16, 5, 'NotGraduate', 'No', 4700000, 10700000, 10, 794, 5700000, 3900000, 16400000, 4400000, 'Approved')
8:14 PM
4/26/2024


localhost:8501
Careers
All Bookmarks
Deploy
Gemini App to Retrieve SQL Data
Input:
How many people have are selff-employed
Ask the question
The response is
(2150,)
5:03 PM
4/26/2024

Results and Discussion

Query Examples

The report would detail several examples of user queries, the generated SQL statements, and the resulting outputs, illustrating the model's ability to accurately interpret and execute queries.

Conclusion

Capabilities

The Gemini App represents a significant advancement in making data accessible to non-technical users, democratizing data retrieval from complex databases.

Reflections

The development process highlighted the importance of robust NLP training and seamless integration between different software components.

Future Prospects

The application has room for further enhancement, such as supporting more complex queries and real-time learning from user interactions.