GOVERNMENT COLLEGE OF TECHNOLOGY, COIMBATORE

TEAM MEMBERS : YAZHINI K (71772117151)
KENEISOU KEDITSU (71772117153)

DEPARTMENT : B.E COMPUTER SCIENCE ENGG.

SUBJECT : ARTIFICIAL INTELLIGENCE

COURSE CODE : 18SPC702

PROJECT NAME : AI-POWERED EXCEL DATA ASSISTANT

ASSIGNMENT NO : 02

PROJECT DESCRIPTION

The AI-powered Excel Data Assistant is an interactive application designed to assist users in extracting insights from CSV data files. Leveraging advanced natural language processing (NLP) techniques, the system enables users to ask questions related to the contents of their CSV files. This project integrates Google's generative AI API to understand user queries and retrieve answers based on the uploaded data. The application provides a seamless experience by enabling file uploads, displaying the loaded data, and allowing users to ask questions about it, with the answers generated by the AI.

Approach Used

1. Document Loading and Preprocessing:

The user uploads a CSV file, which is processed using the `pandas` library. The file is loaded into a DataFrame, and its content is converted into a string format that is easy to integrate into the generative model's prompt. This transformation allows the system to efficiently handle and work with the data.

2. Query Generation and Interaction:

After loading the data, the system prompts users to input a question about the data. The user's question is sent along with the data context to a generative AI model. The model, using its trained capabilities, processes the question and context to generate an appropriate response. The response is then displayed in the app.

3. Streamlit Integration:

The entire system is integrated with Streamlit, providing an intuitive user interface for file upload and interaction with the AI assistant. The user can upload a CSV file, ask questions, and receive answers in real time. This interface is designed to be simple, interactive, and user-friendly.

Challenges Faced

1. Data Size and Performance:

A major challenge is handling large CSV files. Large datasets can slow down the system due to the amount of text being processed. Optimizing the data handling and ensuring that the system can process large files efficiently without compromising on performance was crucial.

2. Understanding Context:

The generative AI model sometimes struggled to capture the full context of the data, especially with complex or ambiguous questions. This required finetuning of the prompts and the model’s settings to ensure that it could answer questions in a meaningful and relevant manner.

3. API Limitations:

The use of an external generative AI API means that there are some limitations on the size and complexity of the responses. There were times when responses were too long or not sufficiently concise, requiring adjustments to the system's configuration to provide clearer answers.

4. User Interaction:

Ensuring a smooth user experience, especially when interacting with larger files, was challenging. For instance, optimizing the file upload process and making sure the file data was parsed correctly without errors was a key area of focus.

Technologies Used

1. Google Generative AI:

The core AI functionality comes from Google's generative AI models, which are capable of understanding and responding to user queries. The `google.generativeai` library allows for easy integration of the AI models into the project.

2. Pandas:

The `pandas` library is used to load, manipulate, and process CSV data. It is an essential tool for handling structured data and transforming it into a format that can be processed by the AI model.

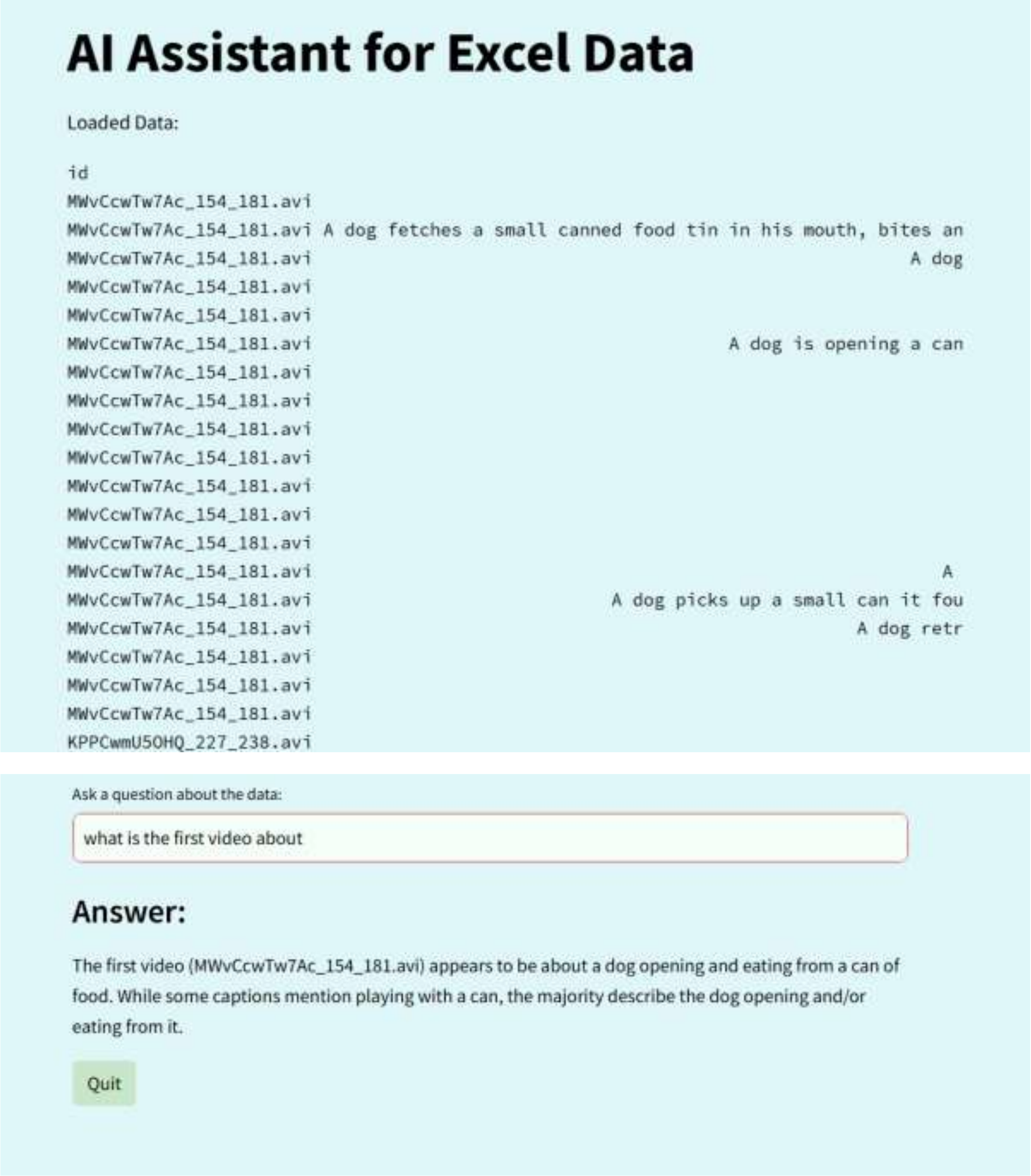
3. Streamlit:

Streamlit is used for building the frontend of the application, providing an interactive, user-friendly interface for users to upload files, ask questions, and view the results. It simplifies the development of web apps, particularly for data science applications.

4. Time and Error Handling:

The system employs error handling techniques to manage potential issues, such as API timeouts or file processing errors. This ensures that the system remains robust and can handle exceptions without crashing.

SAMPLE OUTPUT:



AI Assistant for Excel Data

Loaded Data:

```
id
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi A dog fetches a small canned food tin in his mouth, bites an
MWvCcwTw7Ac_154_181.avi                                                        A dog
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi                                         A dog is opening a can
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi                                                            A
MWvCcwTw7Ac_154_181.avi                                  A dog picks up a small can it fou
MWvCcwTw7Ac_154_181.avi                                                    A dog retr
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
MWvCcwTw7Ac_154_181.avi
KPPCwmU5OHQ_227_238.avi
```

Ask a question about the data:

what is the first video about

Answer:

The first video (MWvCcwTw7Ac_154_181.avi) appears to be about a dog opening and eating from a can of food. While some captions mention playing with a can, the majority describe the dog opening and/or eating from it.

Conclusion

The AI-powered Excel Data Assistant effectively combines modern NLP techniques with an easy-to-use interface, allowing users to interact with and extract insights from their CSV data. By leveraging Google's generative AI, the system can understand and respond to a wide variety of questions based on the provided data. While there were challenges related to performance and data context understanding, these were overcome through optimizations and adjustments to the system. This project showcases how generative AI can be applied to solve real-world problems by making data interaction more intuitive and accessible. The integration with Streamlit further enhances the user experience, making the system easy to use and interact with.