Contents

[I. Introduction 2](#_Toc180270989)

[II. Pipeline Architecture 2](#_Toc180270990)

[2.1 Components: 2](#_Toc180270991)

[2.2 Approach to Retrieval 3](#_Toc180270992)

[III. Creation of Generative Response 4](#_Toc180270993)

[IV. Evaluation 4](#_Toc180270994)

[4.1. Document 1 – 4](#_Toc180270995)

[4.2. Document 2 – 7](#_Toc180270996)

[4.3. Document 3 – 9](#_Toc180270997)

[4.4. Document 4 – 11](#_Toc180270998)

[V. Approach and Decision-Making Process 13](#_Toc180270999)

[5.1. Decisions Made 13](#_Toc180271000)

[5.2. Challenges Faced 13](#_Toc180271001)

[5.3. Solutions Implemented 14](#_Toc180271002)

[VI. Deployment Instructions for the Text Analysis Pipeline 14](#_Toc180271003)

# Introduction

The Cohere LLM RAG model integrates document processing, retrieval, and generative AI to assist users in extracting relevant information from various text-based files. This model supports file types including PDF, CSV, Excel, TXT, MD, and JSON. The system is designed to efficiently process uploaded documents, create embeddings, and respond to user queries based on the document content.

# Pipeline Architecture

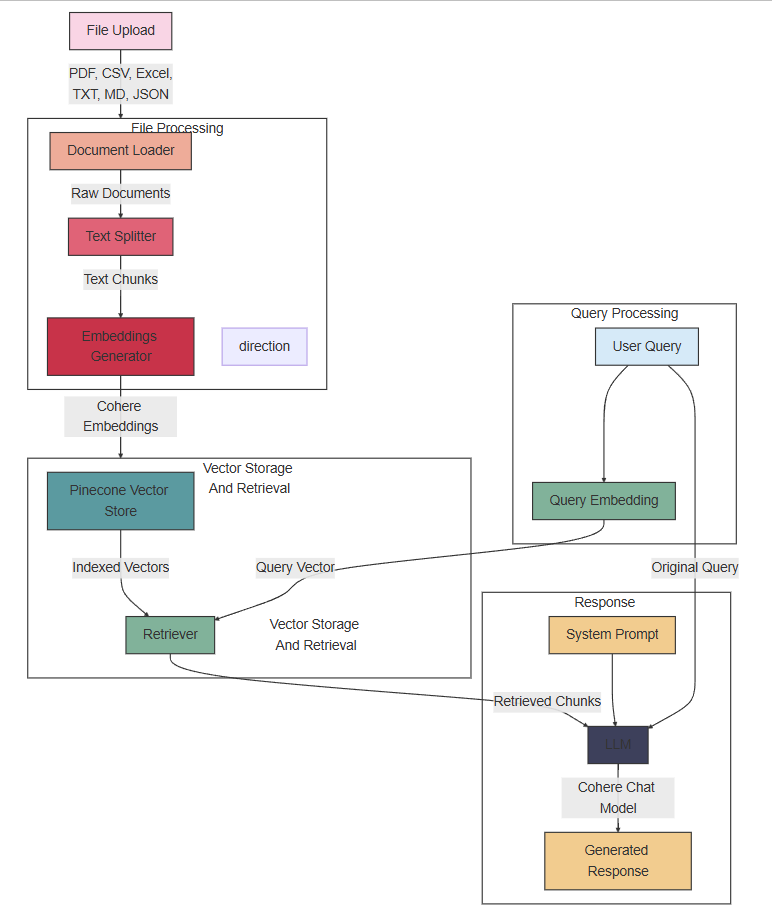


Figure 1: Pipeline Architecture (author generated)

## Components:

* **File Upload:** Users upload files in various formats (PDF, CSV, Excel, TXT, MD, JSON).
* **Document Loader:** Loads the raw documents from the uploaded files.
* **Text Splitter:** Splits the raw documents into smaller text chunks for efficient processing.
* **Embeddings Generator:** Generates embeddings for the text chunks using Cohere's embedding model.
* **Pinecone Vector Store:** Stores the embeddings as vectors in Pinecone for efficient retrieval.
* **Retriever:** Retrieves relevant document chunks based on the query embeddings.
* **User Query:** The user's question or query.
* **Query Embedding:** Generates embeddings for the user query.
* **LLM (Language Model):** Generates responses based on the retrieved document chunks and the user query.
* **System Prompt:** Provides initial instructions and guidelines for the LLM.
* **Generated Response:** The final response generated by the LLM.

## Approach to Retrieval

The retrieval process involves the following steps:

**File Processing:**

* The uploaded file is processed to extract raw documents.
* The raw documents are split into smaller text chunks.
* Embeddings are generated for each text chunk using Cohere's embedding model.
* The embeddings are stored as vectors in Pinecone.

**Query Processing:**

* The user's query is converted into an embedding using the same embedding model.
* The query embedding is used to retrieve relevant document chunks from Pinecone.

**Response Generation:**

* The retrieved document chunks are passed to the LLM along with the user's query and the system prompt.
* The LLM generates a response based on the retrieved information.

# Creation of Generative Response

The generative response creation process involves the following steps:

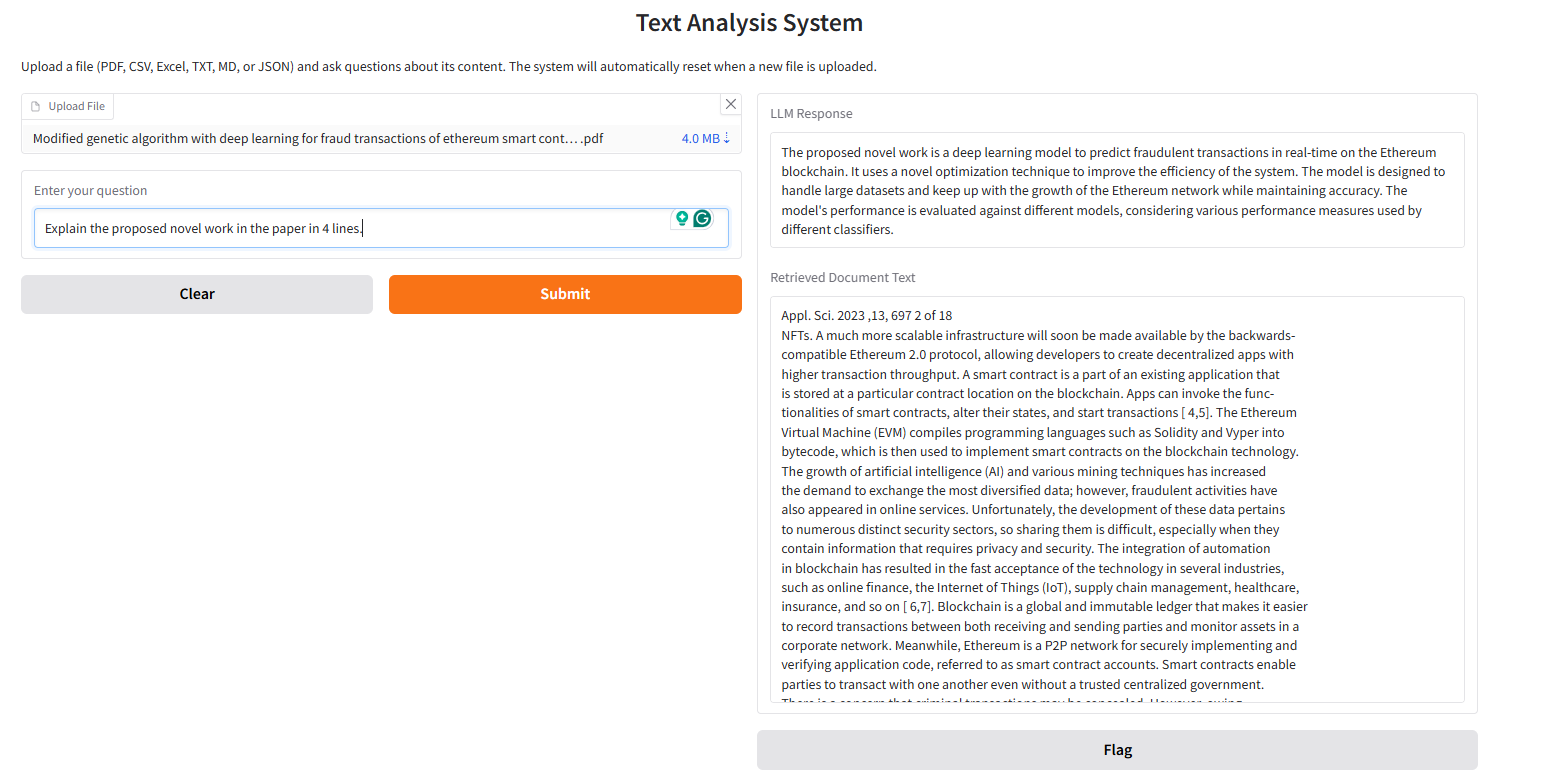
* **System Prompt:** The LLM is provided with an initial prompt that outlines the guidelines for generating responses. This prompt ensures that the LLM bases its answers solely on the information provided in the retrieved documents and follows a professional and helpful tone.
* **Query and Retrieved Documents:** The user's query and the retrieved document chunks are passed to the LLM. The LLM uses this information to generate a response that is accurate, concise, and relevant to the user's query.
* **Generated Response:** The LLM generates a response based on the retrieved information and the user's query. The response is designed to be helpful and informative, providing the user with the information they need.

# Evaluation

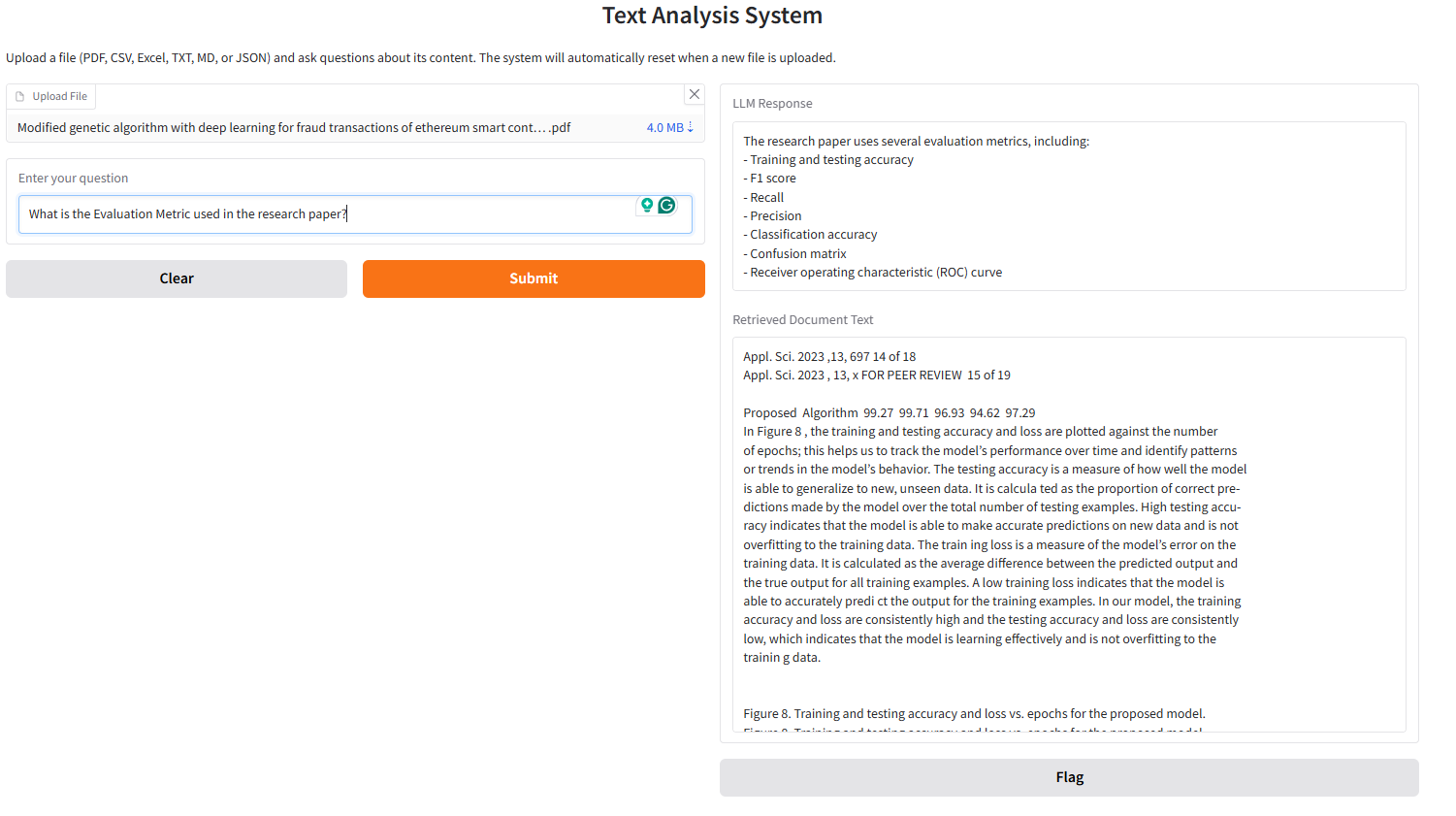
The evaluation of the Cohere LLM RAG model assessed its performance across 4 documents using 5 varied prompts per document though the pipeline can process 6 different types of files. The focus was on accuracy, relevance, and comprehensiveness in responses. The model was tested on its ability to handle both simple and complex queries, ensuring responses were solely based on the provided document content.

## Document 1 –

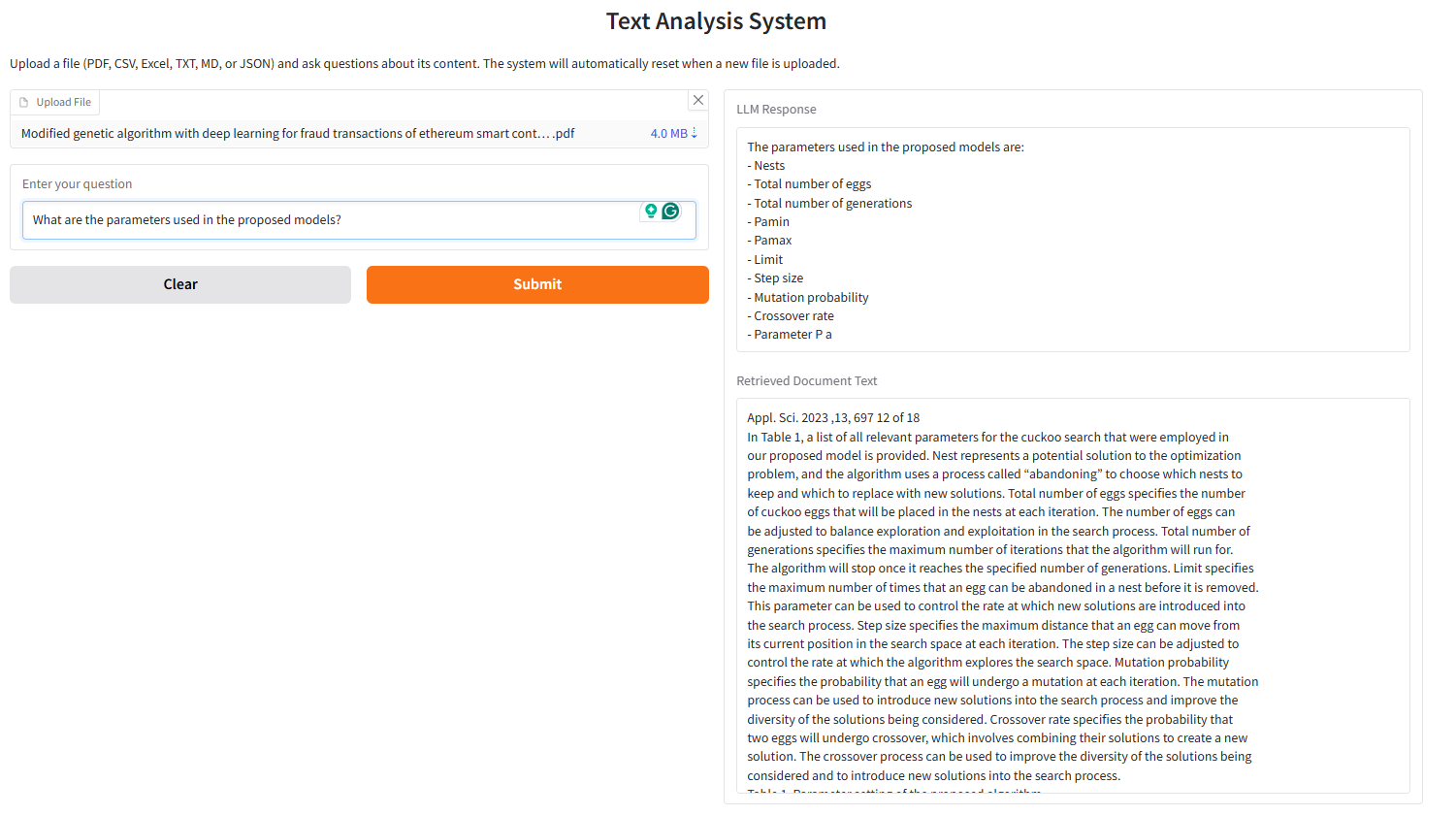
Link of the Document File - [LINK](https://github.com/arya-domain/Cohere-LLM-RAG/blob/main/Test%20set/DOC%201.pdf)



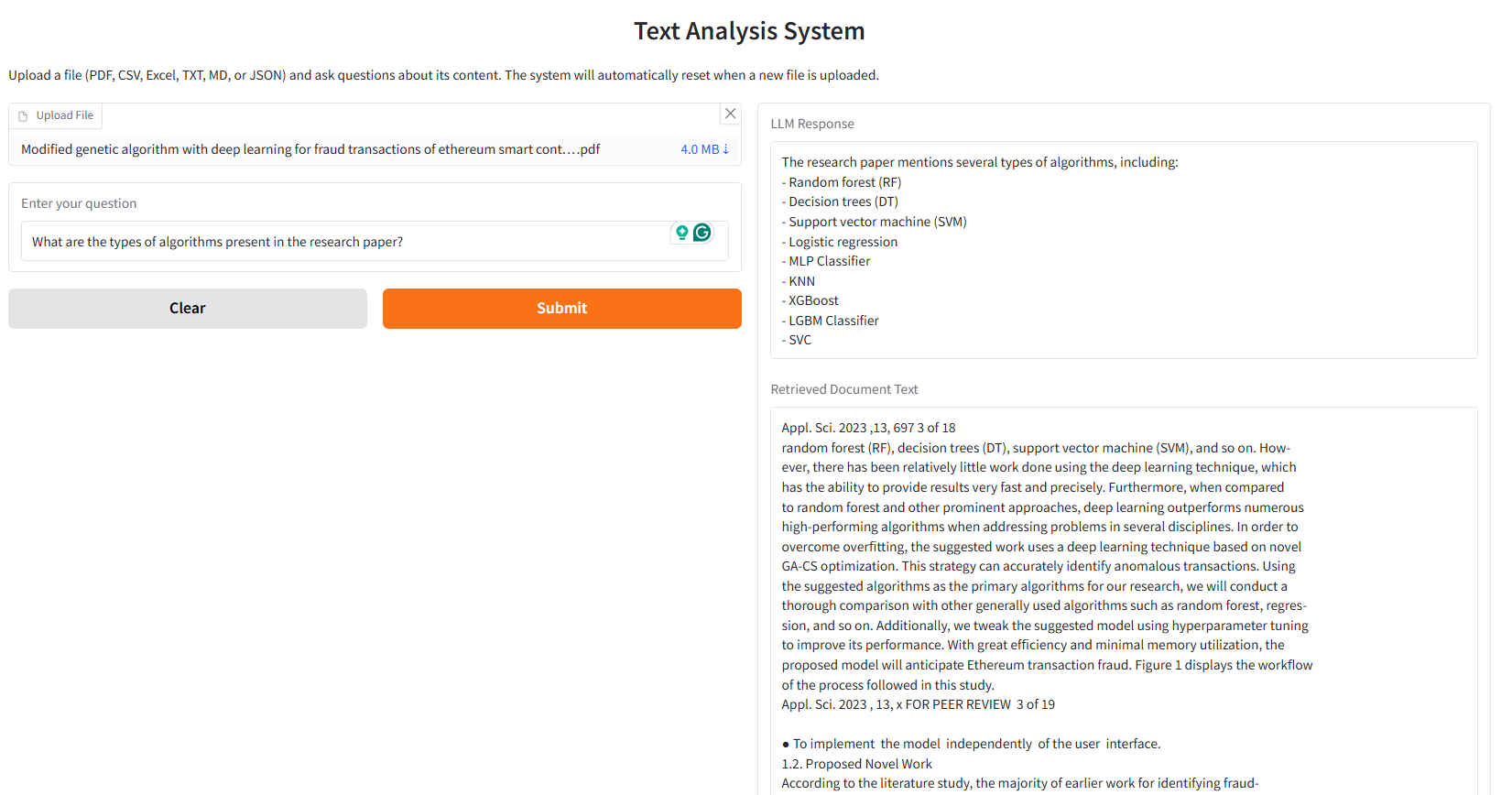
1. Prompt – Explain the proposed novel work in the paper.



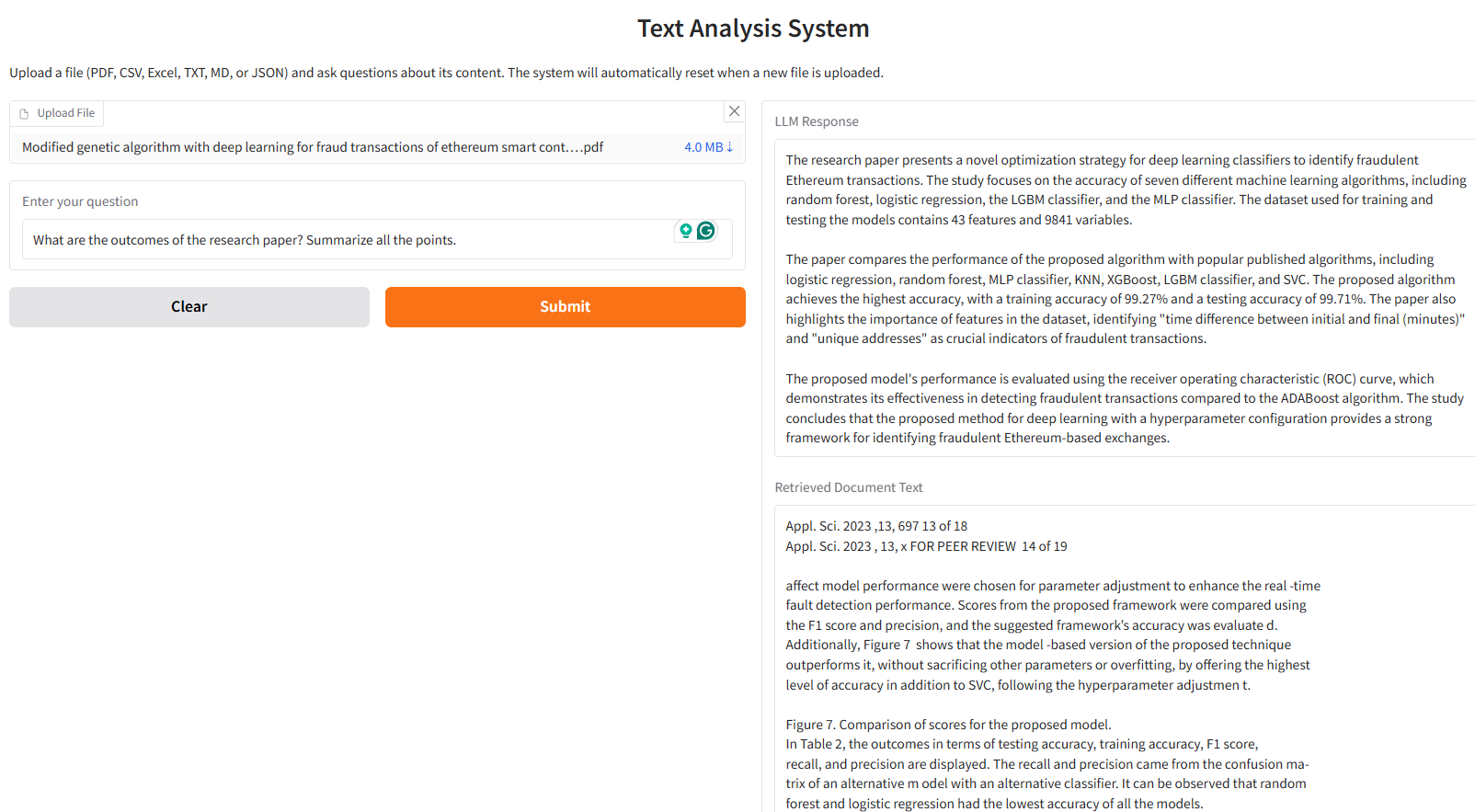
1. Prompt - What is the Evaluation Metric used in the research paper?



1. Prompt - What are the parameters used in the proposed models?



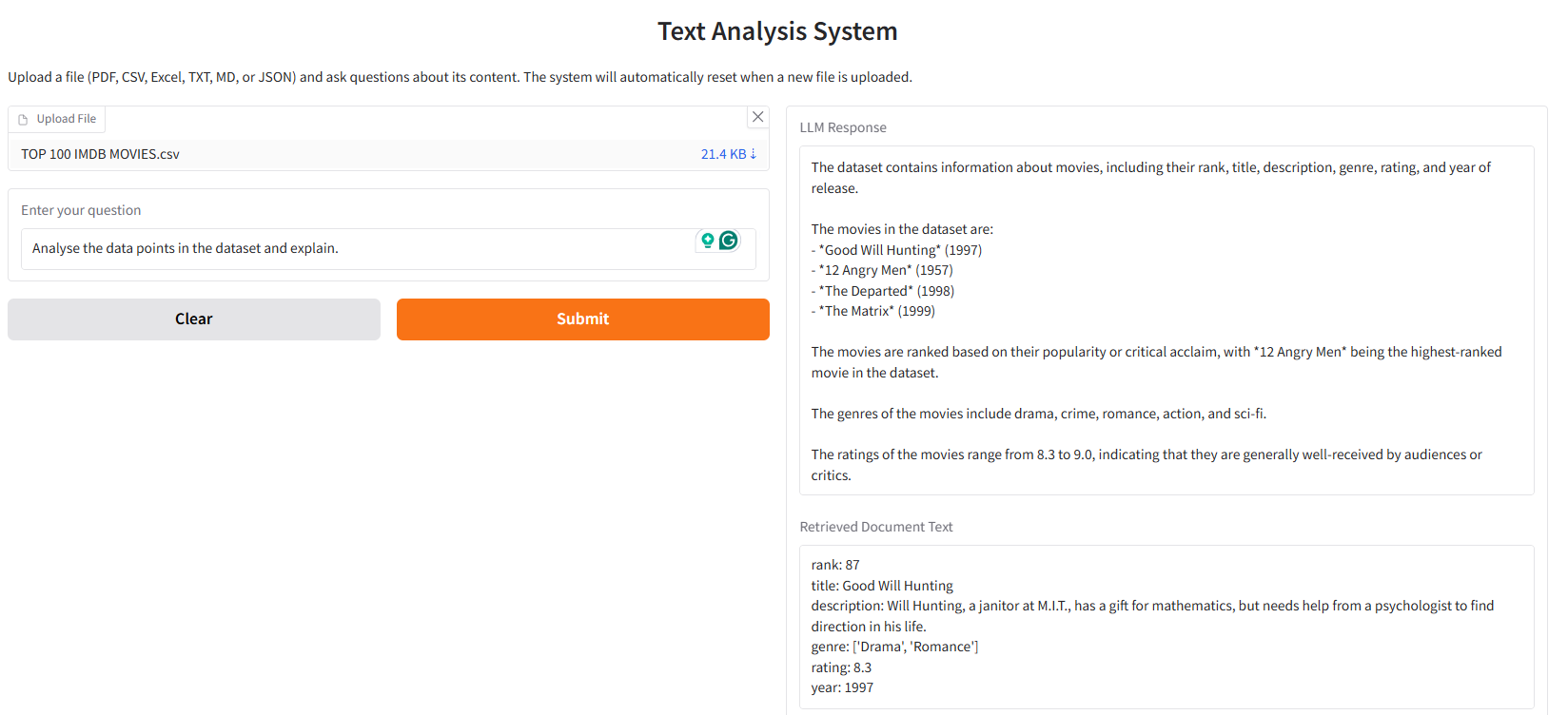
1. Prompt - What are the types of algorithms present in the research paper?



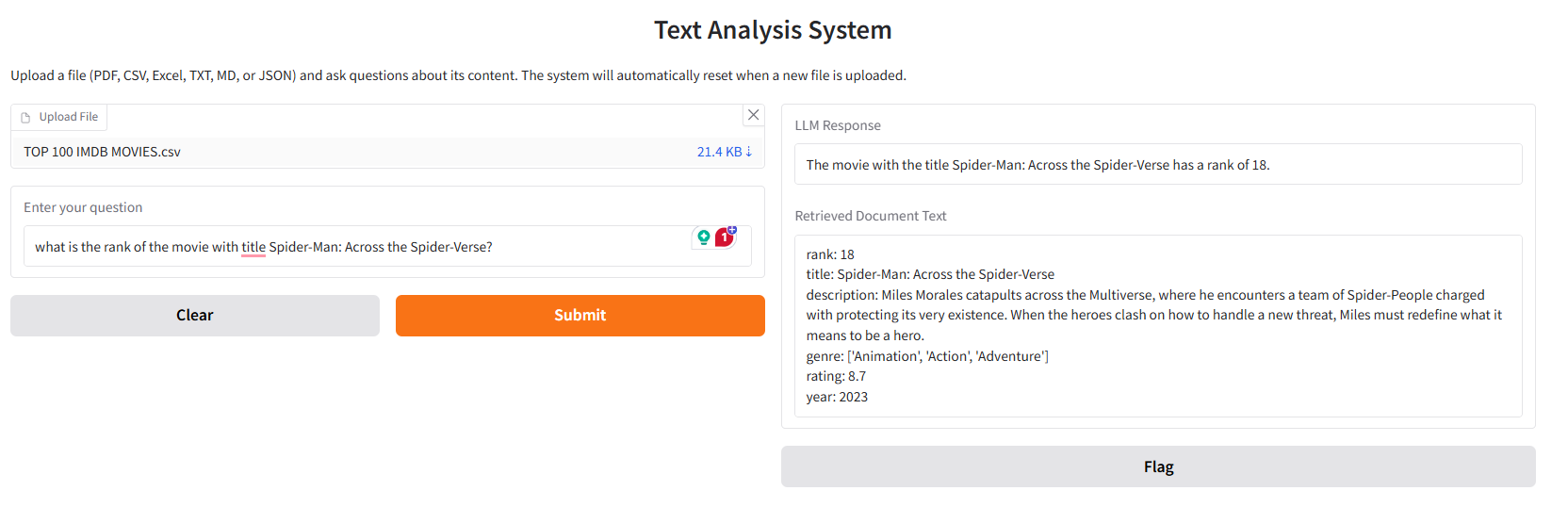
1. Prompt - What are the outcomes of the research paper? Summarize all the points.

## Document 2 –

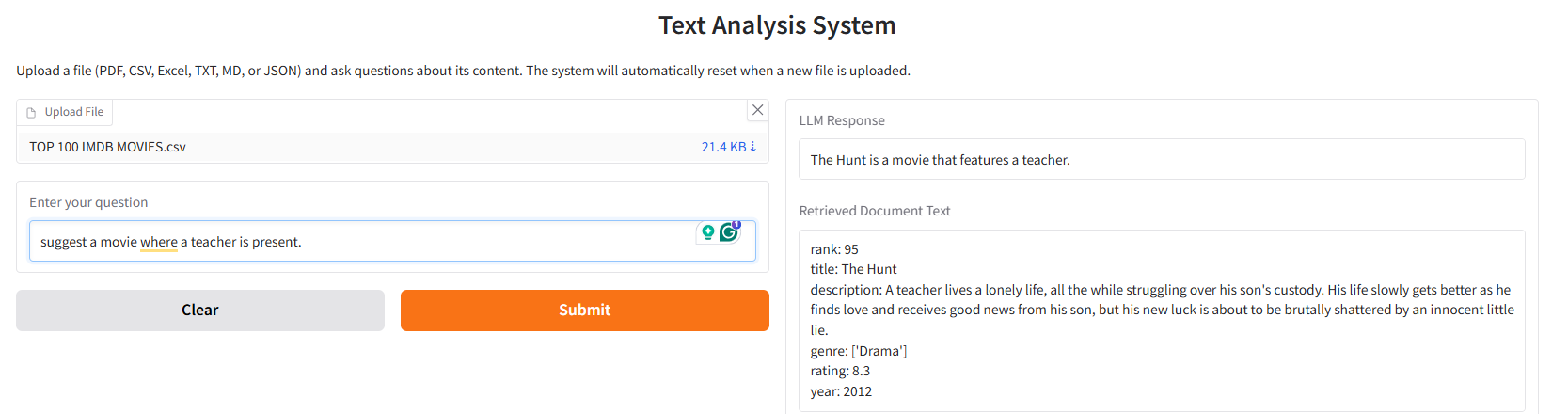
Link of the Document File - [LINK](https://github.com/arya-domain/Cohere-LLM-RAG/blob/main/Test%20set/DOC%202.csv)



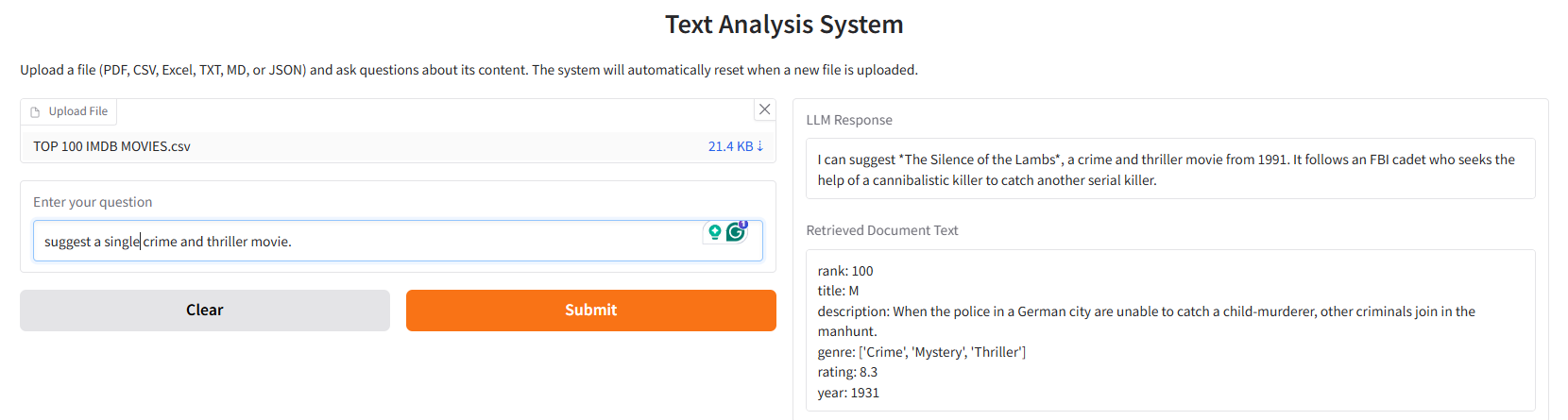
1. Prompt - Analyse the data points in the dataset and explain.



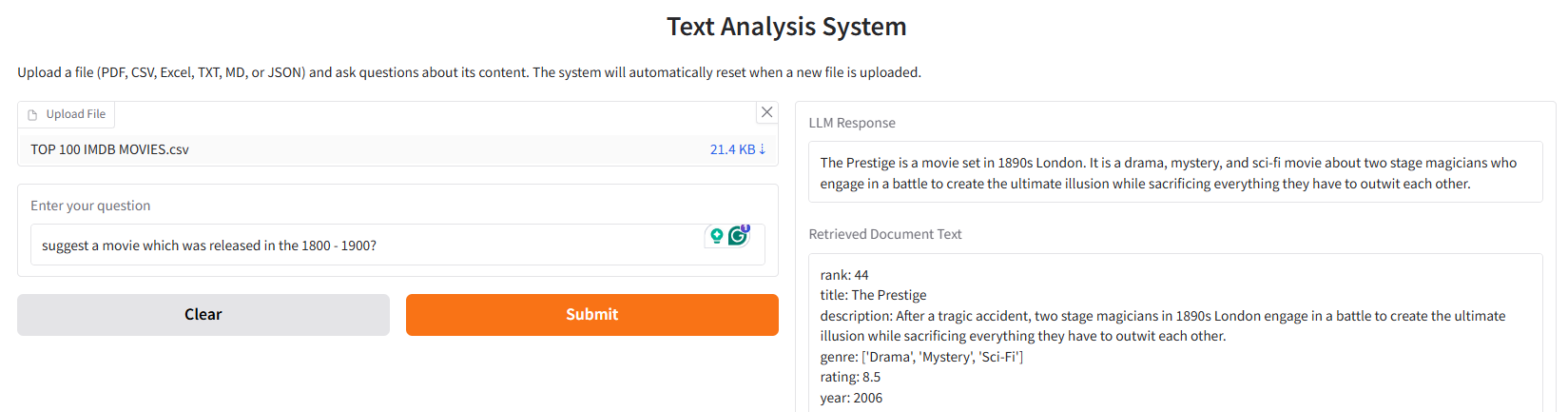
1. Prompt - What is the rank of the movie with title Spider-Man: Across the Spider-Verse?



1. Prompt - Suggest a movie where a teacher is present.



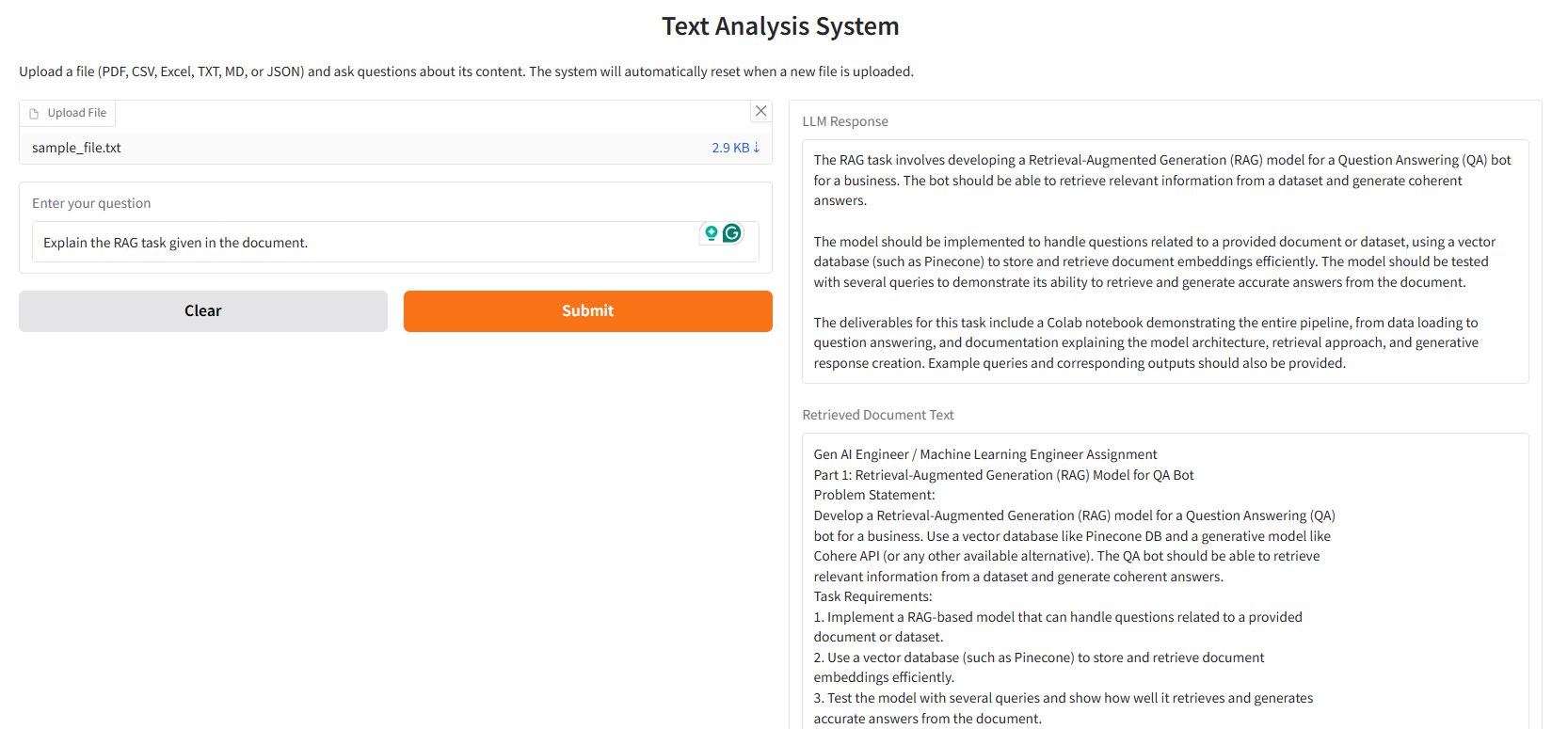
1. Prompt - Suggest a single crime and thriller movie.



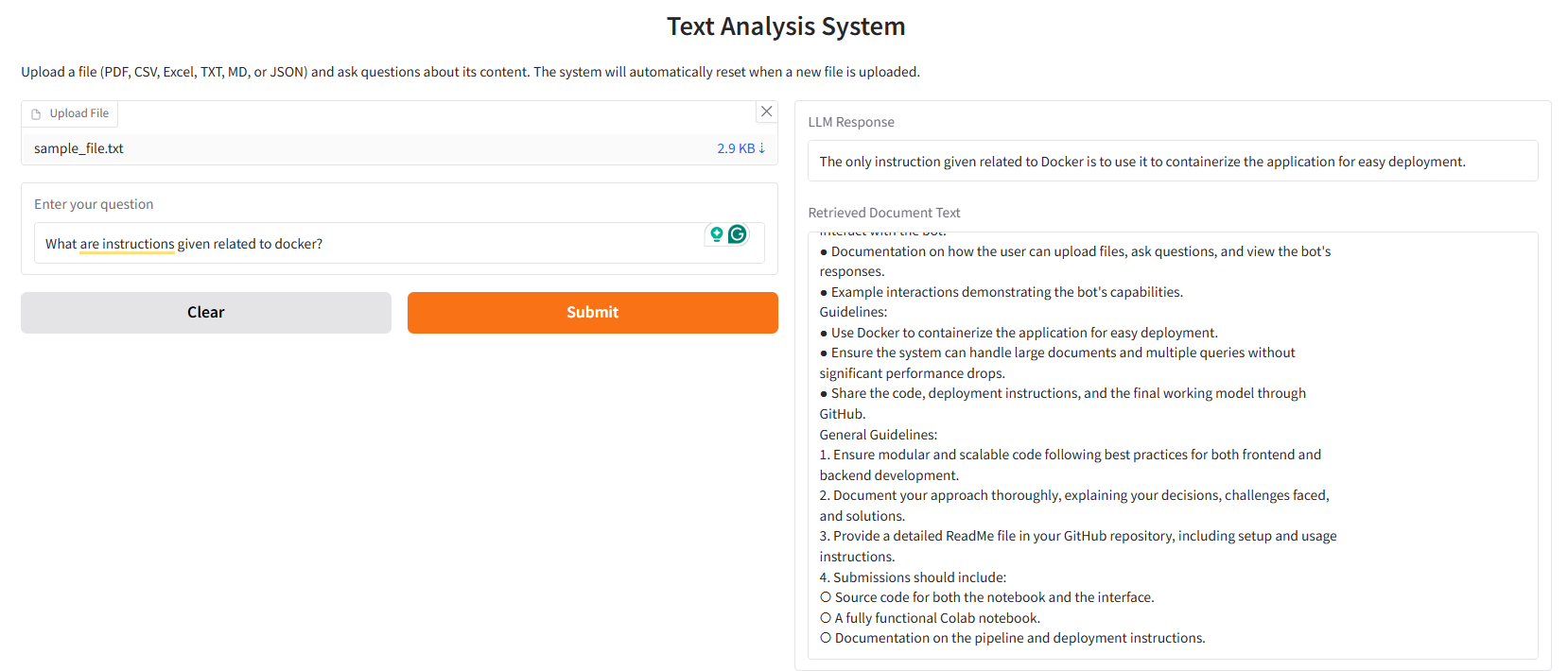
1. Prompt - Suggest a movie which was released in the 1800 - 1900?

## Document 3 –

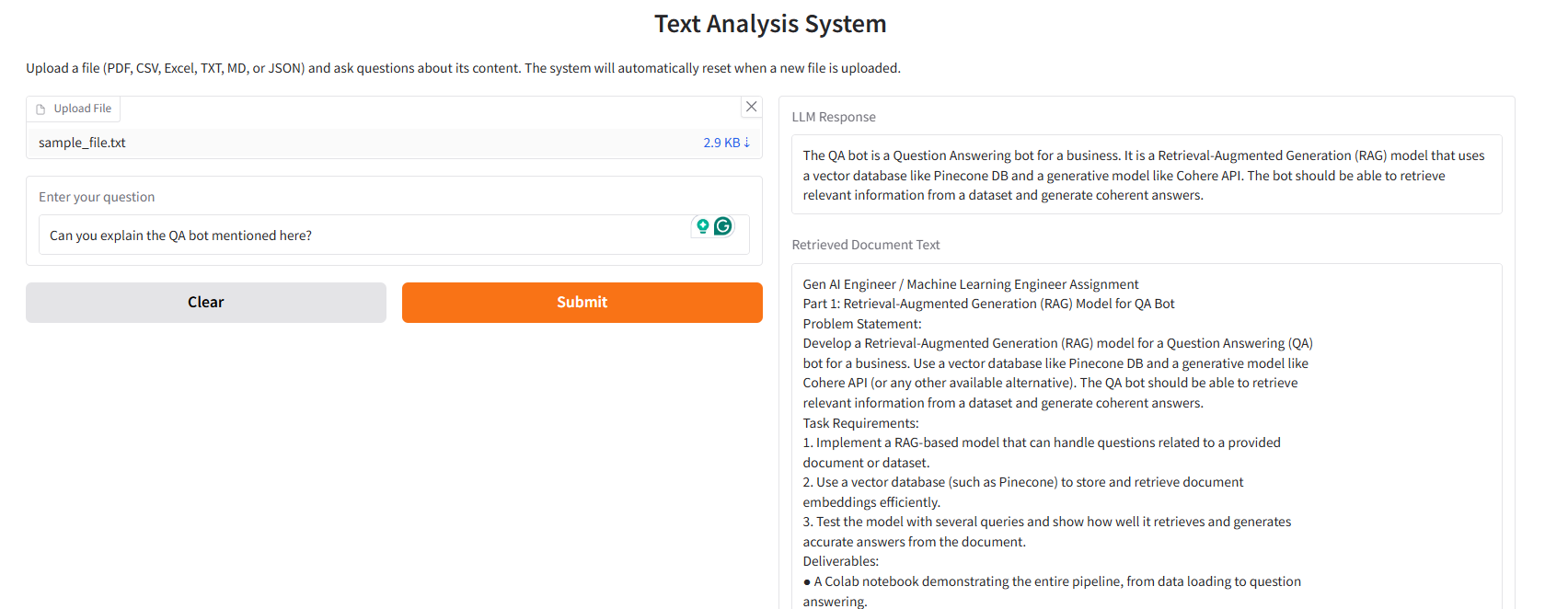
Link of the Document File - [LINK](https://github.com/arya-domain/Cohere-LLM-RAG/blob/main/Test%20set/DOC%203.txt)



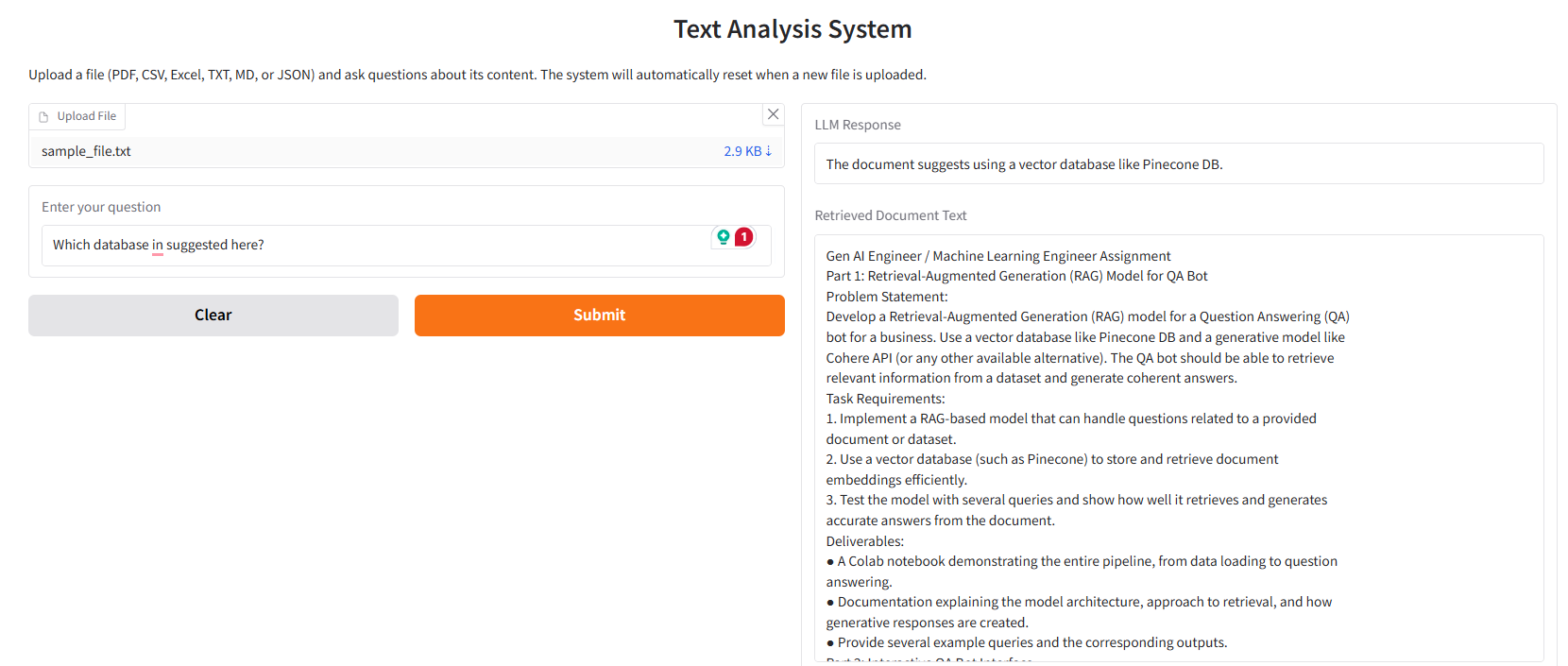
1. Prompt - Explain the RAG task given in the document.



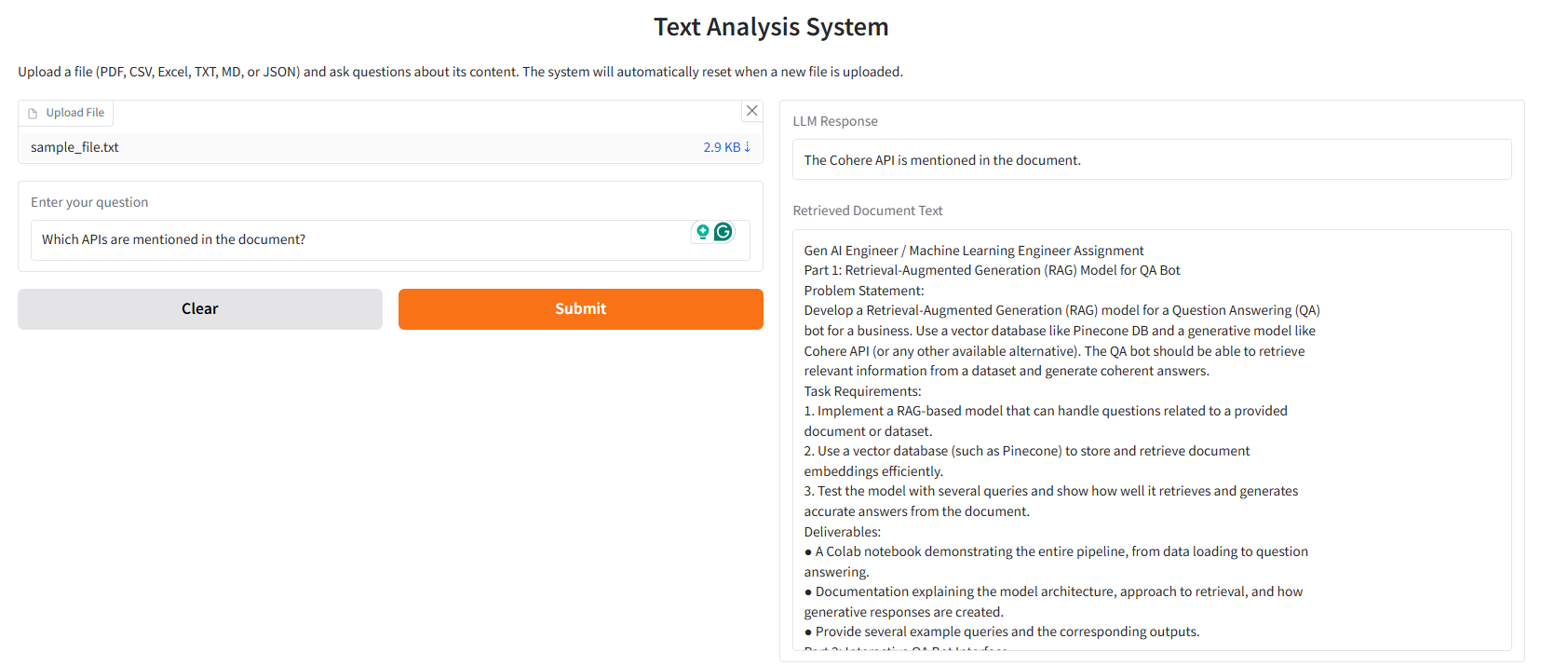
1. Prompt - What are instructions given related to docker?



1. Prompt - Can you explain the QA bot mentioned here?



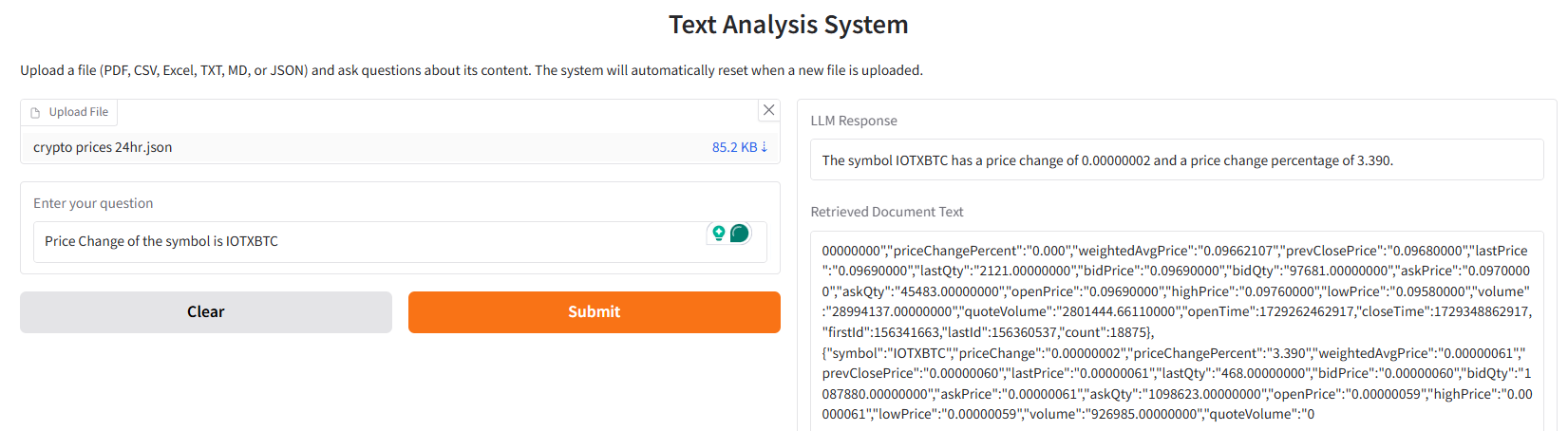
1. Prompt - Which database is suggested here?



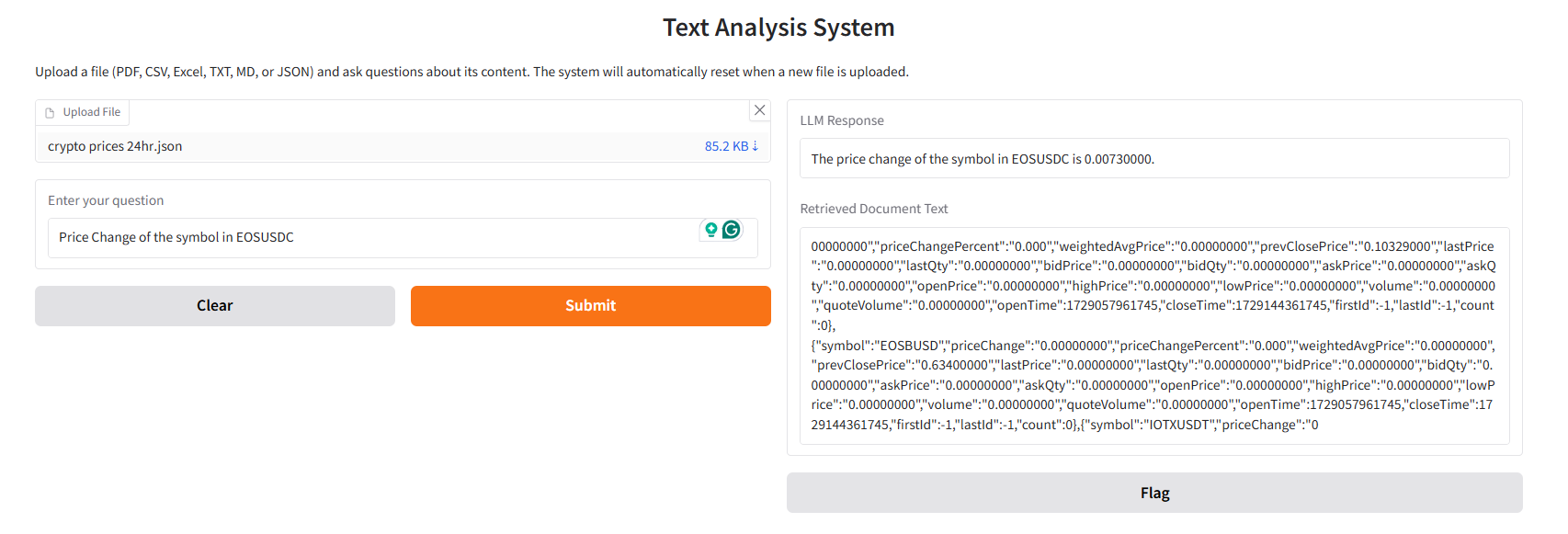
1. Prompt - Which APIs are mentioned in the document?

## Document 4 –

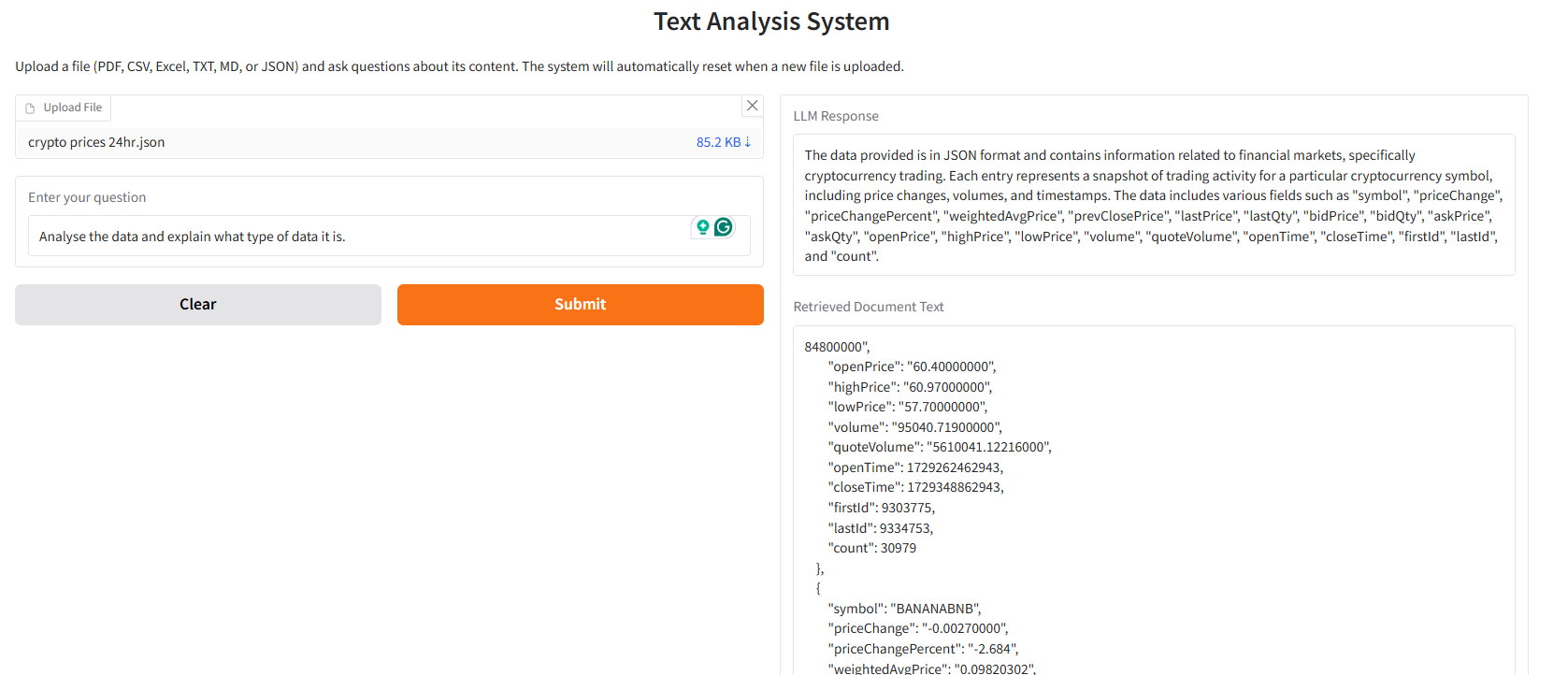
Link of the Document File - [LINK](https://github.com/arya-domain/Cohere-LLM-RAG/blob/main/Test%20set/DOC%204.json)



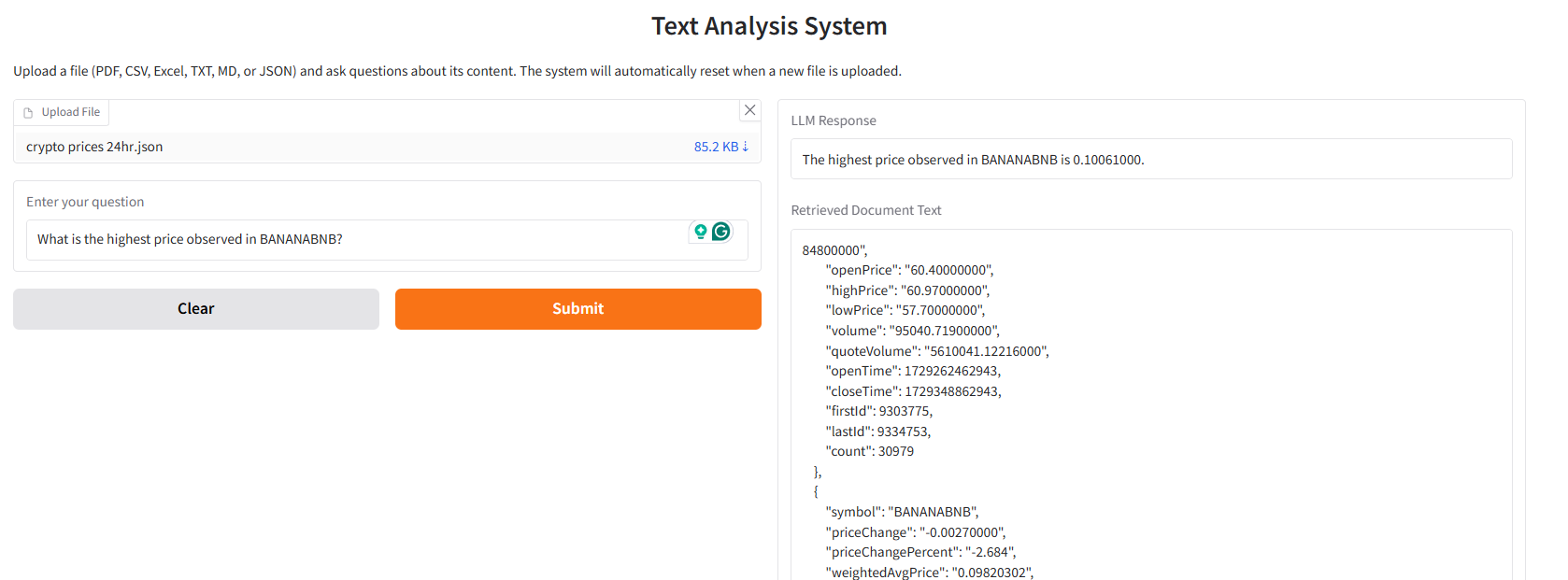
1. Prompt – Price Change of symbol in IOTXBTC



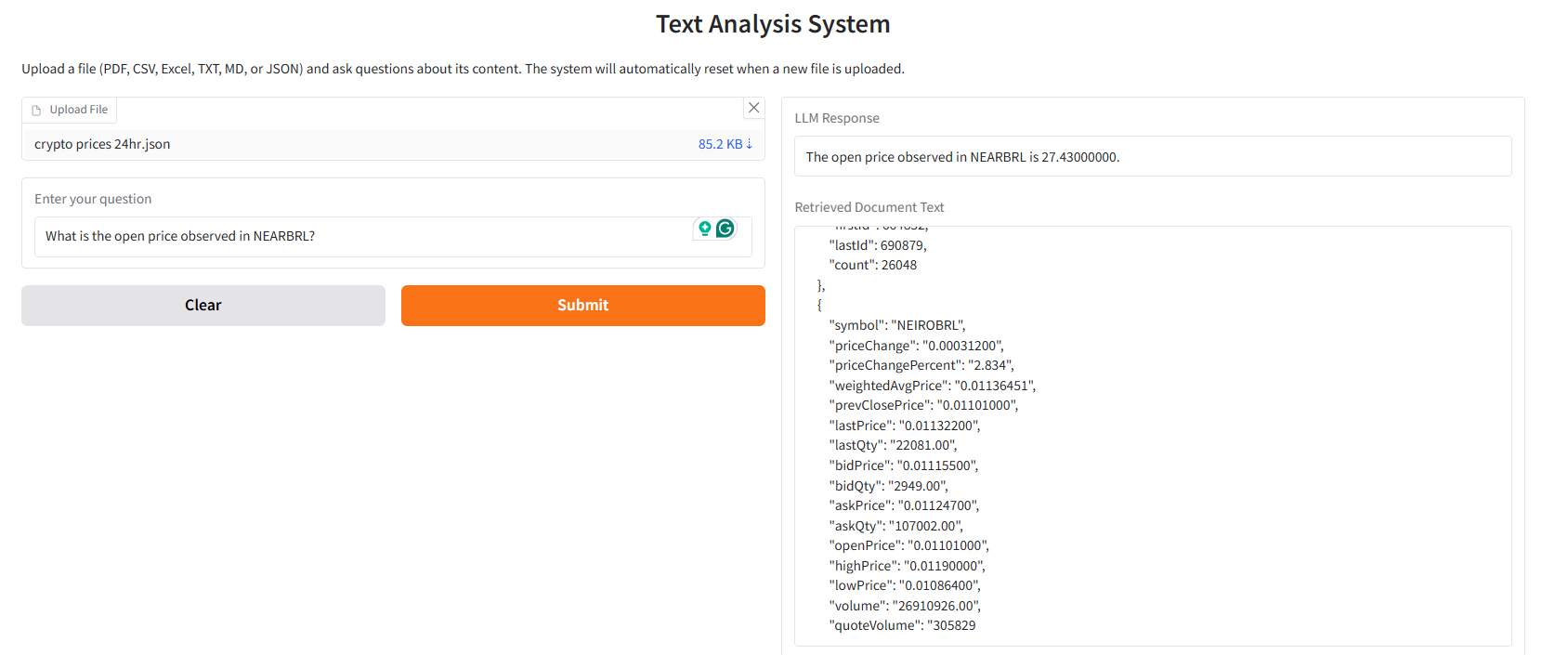
1. Prompt – Price Change of symbol in IOTXBTC



1. Prompt - Analyse the data and explain what type of data it is.



1. Prompt - What is the highest price observed in BANANABNB?



1. Prompt - What is the open price observed in NEARBRL?

# Approach and Decision-Making Process

## Decisions Made

Key technologies chosen include Cohere Embeddings for accurate retrieval and response generation, Pinecone for efficient vector storage, Gradio for creating a user-friendly interface, and Docker for consistent deployment across environments. Cohere and Pinecone were already suggested in the assignment file, so that is also a reason to be chosen for this assignment. A diverse file processing pipeline was set up with loaders for PDFs, CSVs, Excel, and text files. A text splitter was used to divide documents into manageable chunks for efficient processing and retrieval.

## Challenges Faced

Handling large documents posed memory and rate-limiting issues, which were addressed by implementing batch processing with delays. Ensuring accurate retrieval was crucial, requiring fine-tuning the retrieval process using high-quality Cohere embeddings. Maintaining consistency across various environments was resolved by using Docker, ensuring a standardized runtime environment.

## Solutions Implemented

Batch processing with delays was introduced to manage memory and avoid rate limits while processing large documents. Cohere’s embedding model was used to generate high-quality embeddings, ensuring precise retrieval and response generation. Docker containerization enabled consistent deployment across environments, and a comprehensive logging mechanism was implemented to improve error handling and system monitoring.

# Deployment Instructions for the Text Analysis Pipeline

**Prerequisites:**

* Ensure Docker is installed and running on your system.
* Download the Docker image from [Google Drive](https://drive.google.com/file/d/1-1ic3N-TJJeLDepEvOPQye6EKEQoshB8/) or using gdown.

**Step-by-Step Deployment:**

* **Download the Docker Image**

gdown 1-1ic3N-TJJeLDepEvOPQye6EKEQoshB8

* **Load the Docker Image**

docker load -i ./hosting\_image.tar

* **Run the Docker Container**

docker run -d -p 7860:7860 --name hosting\_container hosting

**Accessing the Pipeline:**

* **Access:** http://localhost:7860
* **Logs:** docker logs hosting\_container
* **Stopping the Container:** docker stop hosting\_container
* **Removing the Container:** docker rm hosting\_container

Finally, the model can be accessed through the PUBLIC IP ADDRESS of the Remote Server at port 7860.

**Example –**

* Server's public IP address is 192.168.1.100
* Link - http://192.168.1.100:7860