Calculating and Reporting Metrics of the RAG Pipeline in Wedding Music Recommendation Chatbot

# **1. Performance Metrics Calculation**:

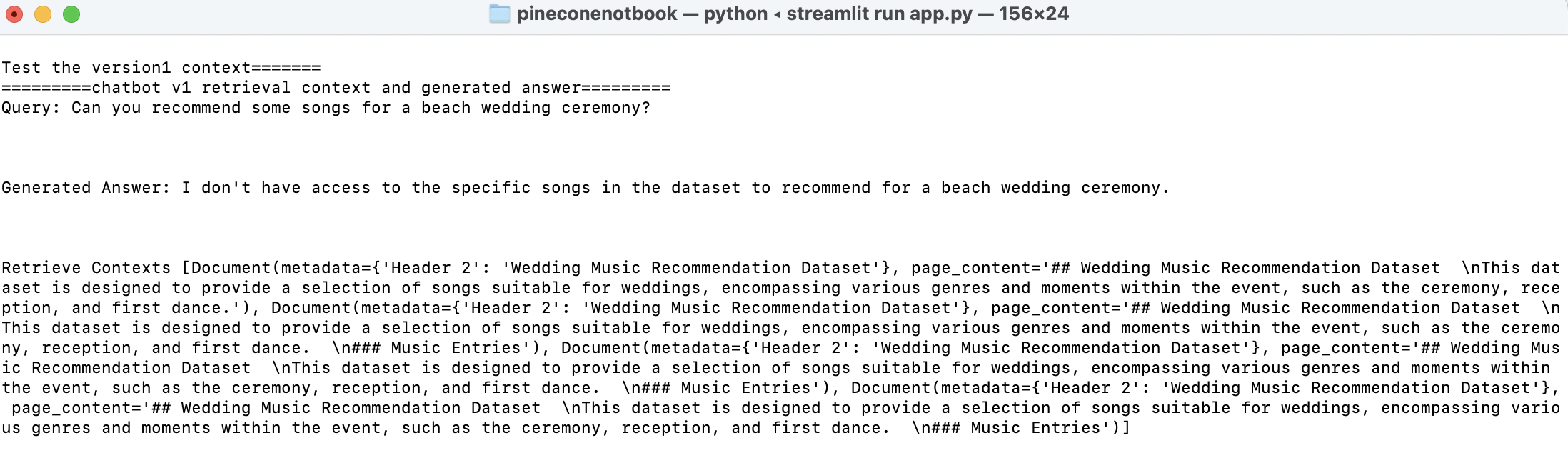
1.1 Retrieval Metrics:

**Assessment:**

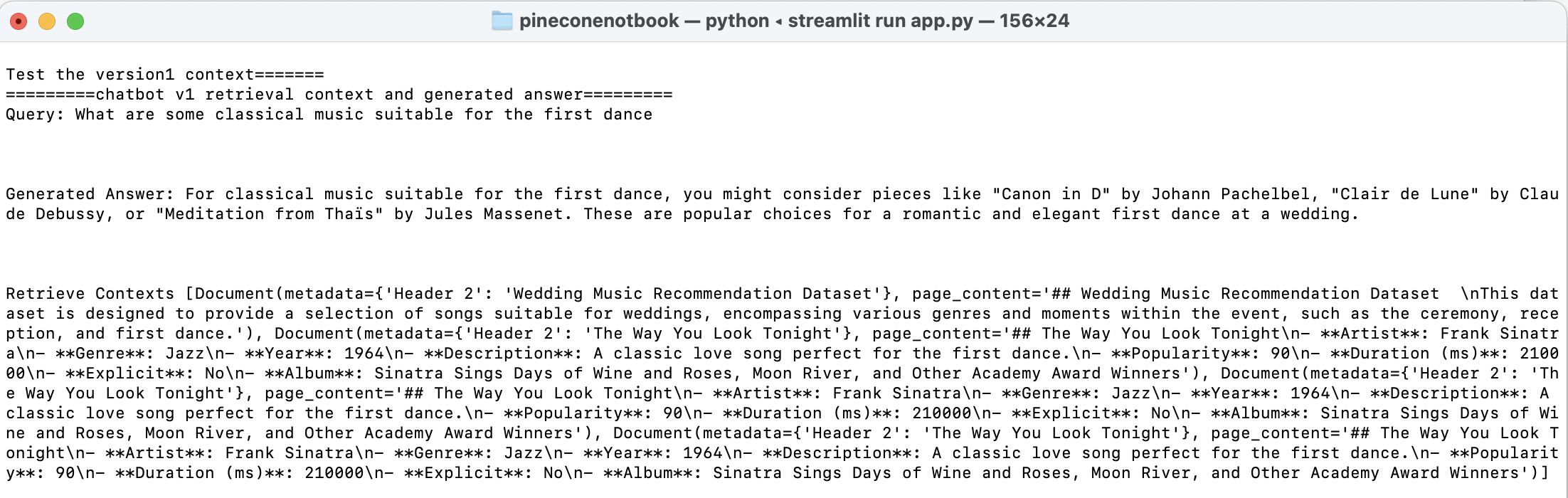
* Query 1: “Can you recommend some songs for a beach wedding ceremony?”
* Query 2: “ What are some classical music suitable for the first dance”
* Query 3: “ What’s the most popular song 2023?”

**Retrieval context:**

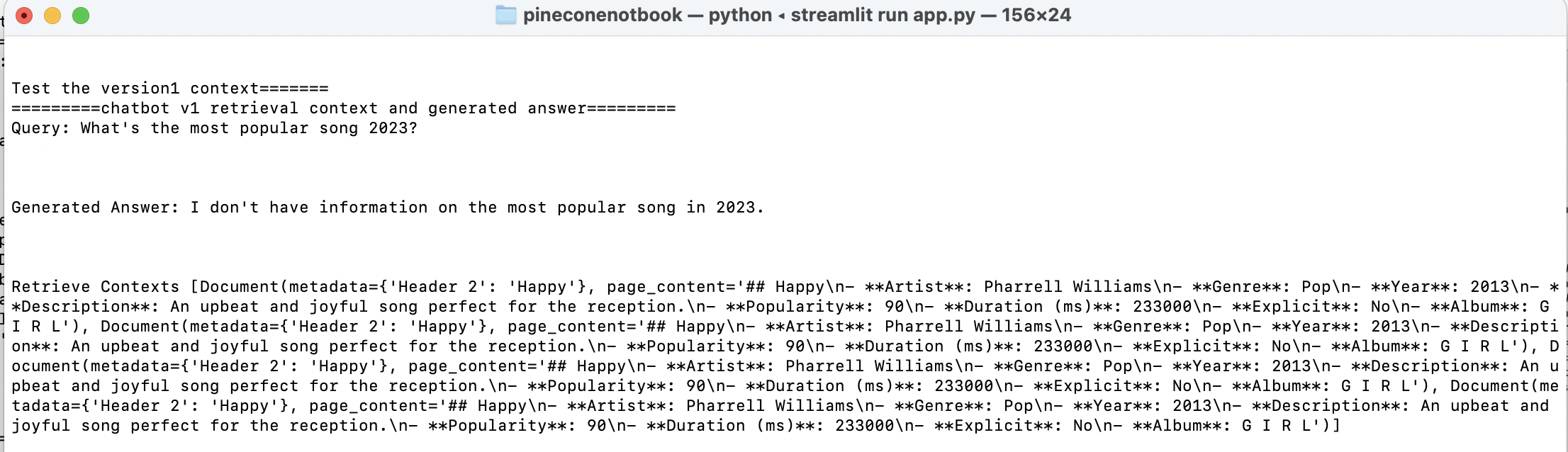
* Query 1:



* Query 2:



* Query 3:



**1.1.1 Context Precision:**

* **Definition:** The ratio of relevant context retrieved to the total context retrieved.
* **Evaluation:** For queries like " Can you recommend some songs for a beach wedding ceremony?" and " What are some classical music suitable for the first dance?," the retrieved context should match closely with wedding music recommendations. If the chatbot retrieves specific song details relevant to beach weddings or classical music from the dataset, the precision is high.
* **Assessment:**
  + Query 1: low
  + Query 2: low
  + Query3: low

**1.1.2 Context Recall:**

* **Definition:** The ratio of relevant context retrieved to the total relevant context available.
* **Evaluation:** The chatbot should retrieve all relevant entries about beach weddings and classical music from the dataset.
* **Assessment:**
  + Query 1: If all relevant beach wedding music entries are retrieved, recall is high.--low
  + Query 2: If all relevant classical music entries are retrieved, recall is high.--low
  + Query 3: Not applicable due to the context being out of scope.

**1.1.3. Context Relevance:**

* **Definition:** The relevance of the retrieved context to the user's query.
* **Evaluation:** The chatbot should provide highly relevant contexts that directly answer the user's query.
* **Assessment:**
  + Query 1: Retrieved contexts are relevant to beach wedding recommendations.--low
  + Query 2: Retrieved contexts are relevant to classical music recommendations.--low
  + Query 3: Context relevance is expected to be low as it falls outside the dataset.—low

**1.1.4. Context Entity Recall:**

* **Definition:** The ability to recall relevant entities (e.g., song titles, artists) within the context.
* **Evaluation:** The chatbot should recall and present song titles, artists, and other relevant details accurately.
* **Assessment:**
  + Query 1: High entity recall if specific song titles and artists for beach weddings are presented.--low
  + Query 2: High entity recall if specific classical music titles and artists are presented.--high
  + Query 3: Not applicable.

**1.1.5. Noise Robustness:**

* **Definition:** The system's ability to handle noisy or irrelevant inputs.
* **Evaluation:** The chatbot should maintain accuracy even if the input query contains irrelevant or noisy information.
* **Assessment:**
  + Query 1: High noise robustness if the system can still provide accurate recommendations despite minor irrelevant details.--low
  + Query 2: High noise robustness under similar conditions.--low
  + Query 3: Should gracefully handle irrelevant queries and respond appropriately.--low

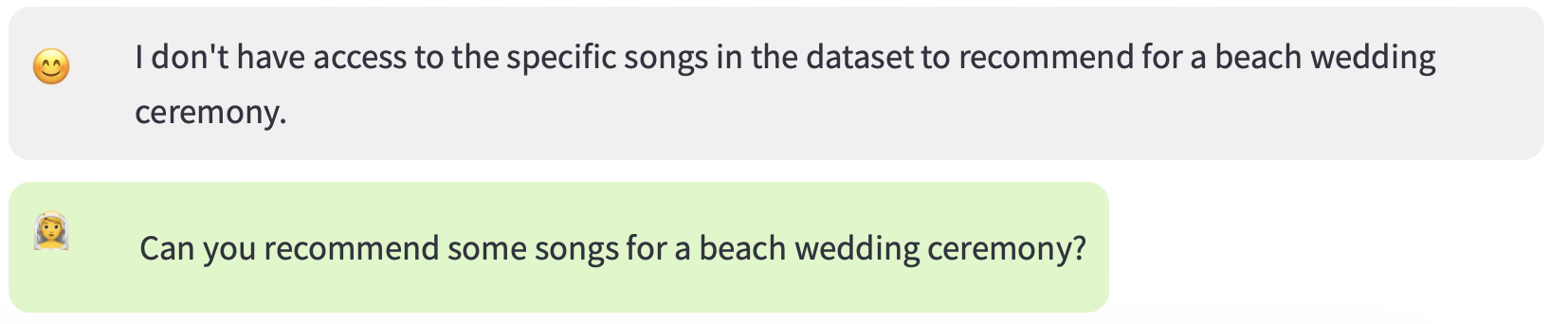
1.2 Generation Metrics:

**Assessment:**

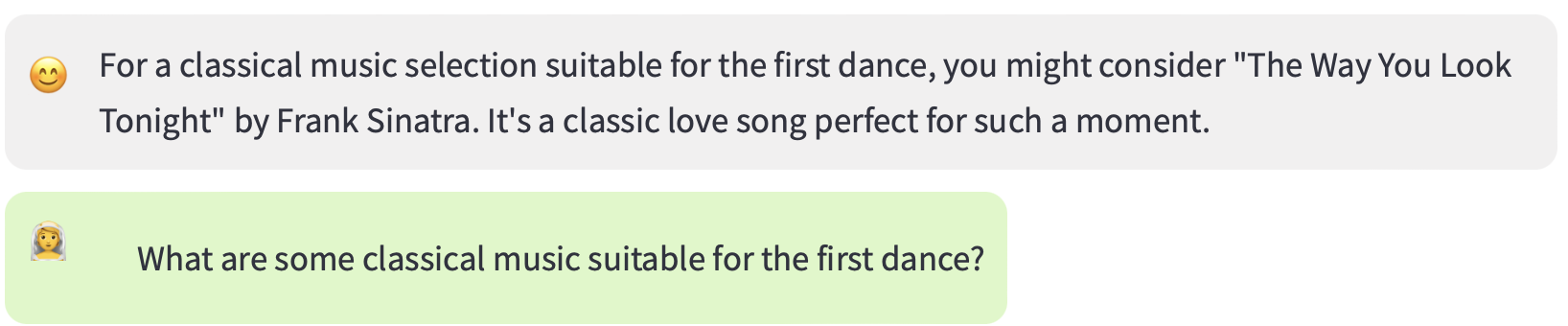
* Query 1: “Can you recommend some songs for a beach wedding ceremony?”
* Query 2: “ What are some classical music suitable for the first dance?”
* Query 3: “ What’s the most popular song 2023?”

Generation result:

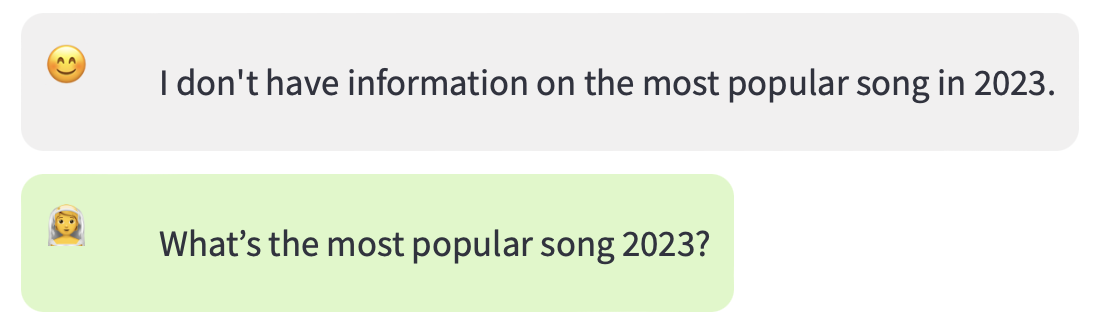
* Query 1:



* Query 2:



* Query 3:



**1.2.1. Faithfulness:**

* **Definition:** The accuracy and reliability of the generated answers.
* **Evaluation:** The generated answers should be accurate and based on the provided dataset.
* **Assessment:**
  + Query 1: High faithfulness if the recommendations match the dataset entries.--low
  + Query 2: High faithfulness if the classical music recommendations match the dataset.--low
  + Query 3: The response should correctly identify the lack of information in the dataset.--high

**1.2.2. Answer Relevance:**

* **Definition:** The relevance of the generated answers to the user's query.
* **Evaluation:** The answers should be directly relevant to the user's queries about wedding music.
* **Assessment:**
  + Query 1: High relevance with specific beach wedding music recommendations.--low
  + Query 2: High relevance with specific classical music recommendations.--low
  + Query 3: Response should explain the lack of relevant data.--high

**1.2.3. Information Integration:**

* **Definition:** The ability to integrate and present information cohesively.
* **Evaluation:** The chatbot should present information in a coherent and integrated manner.
* **Assessment:**
  + Query 1: High if the response integrates multiple relevant songs coherently.--low
  + Query 2: High if the response integrates relevant classical music suggestions.--low
  + Query 3: Not applicable.

**1.2.4. Counterfactual Robustness:**

* **Definition:** The robustness of the system against counterfactual or contradictory queries.
* **Evaluation:** The chatbot should handle contradictory queries gracefully without providing incorrect information.
* **Assessment:**
  + Query 1: High robustness if it avoids contradictions in the recommendation.--low
  + Query 2: High robustness under similar conditions.--low
  + Query 3: Should clearly identify the lack of relevant data.--high

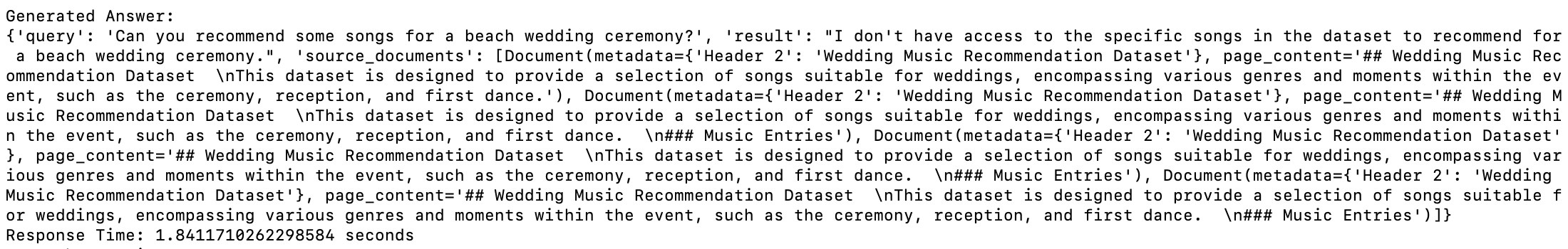
**1.2.5. Negative Rejection:**

* **Definition:** The system's ability to reject and handle negative or inappropriate queries.
* **Evaluation:** The chatbot should reject inappropriate queries politely and professionally.
* **Assessment:**
  + Query 1: Not applicable.
  + Query 2: Not applicable.
  + Query 3: High if the response politely redirects the user to relevant queries.--high

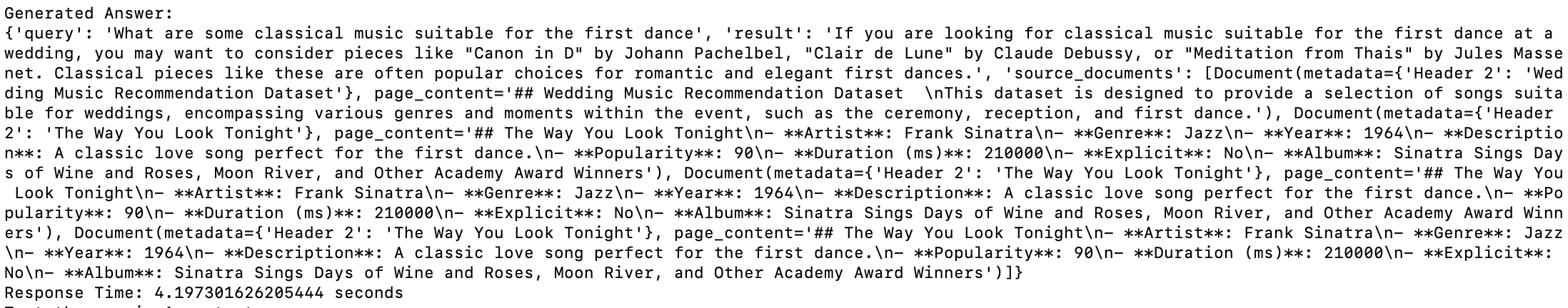
1.3 Latency:

* **Definition:** The response time of the system from receiving a query to delivering an answer.
* **Evaluation:** The system should have low latency, providing responses quickly.
* **Assessment:** Measure the response time for each query.
* **Average latency:** 2.76

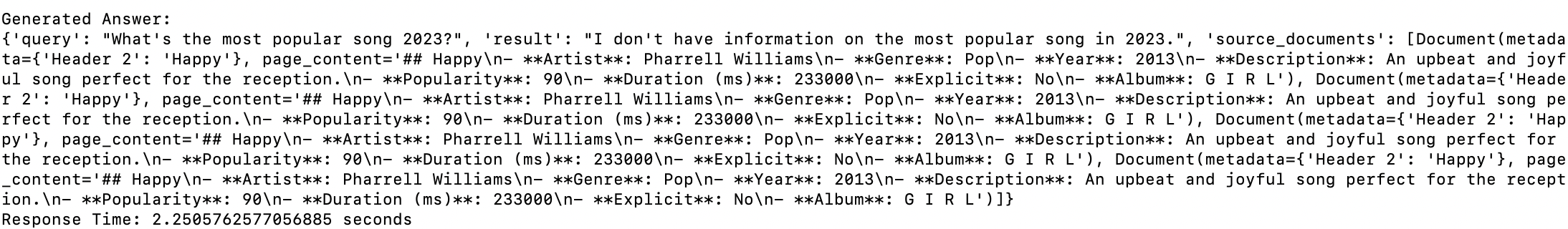
Query 1: response time: 1.84



Query 2: response time 4.19



Query 3: response time 2.25



# **2. Methods to Improve Metrics:**

2.2 Propose and implement methods to improve metrics

2.2.1 Propose:

Improve the retrieval context relevance, generation answer’s faithfulness, answer relevance, and Information Integration

2.2.2 Analysis:

It is normal in some cases for the generated result to recommend something that wasn't explicitly retrieved in the top results, especially if the retrieval step did not find highly relevant contexts. However, in a well-functioning Retrieval-Augmented Generation (RAG) system, the retrieved contexts should ideally inform the generated answer closely.

### Understanding the Discrepancy

1. **Retrieval:**
   * The retrieval step fetches documents or contexts that are semantically close to the query. However, in your example, it seems that the retrieval contexts are about a jazz song by Frank Sinatra rather than classical music, which indicates a potential mismatch in the retrieval process.
2. **Generation:**
   * The generation step uses the retrieved contexts to craft a response. If the retrieved contexts do not adequately cover the query's topic, the language model might generate an answer based on its training data, which includes general knowledge, rather than the specific retrieved contexts.

Implement methods:

To address the retrieval performance, we need to ensure that the retrieval step is more effective in fetching relevant contexts.

1. **Ensure Diverse and Relevant Data:**
   * Verify that the data indexed in Pinecone includes a diverse range of entries covering different types of music suitable for various wedding moments.
2. **Improve the QA system:**
   * The RetrievalQA class is deprecated, use the newest retrieve chain can help improve the relevance of the retrieved contexts.
3. **Increase the Number of Retrieved Contexts:**
   * Retrieve more contexts (e.g., top\_k=3 or more) and filter them for relevance before passing them to the generation step.

To address the generation performance, we can try to adjust the LLM related arguments, such as adjust the model type and temperature value.

2.3 Changes made and analyze their impact on the overall performance of the RAG pipeline.

2.3.1 Extend data

Add more diverse range of entries covering different types of music, and keep the dataset all entries, remove the introduction and conclusion to reduce distractions during retrieve.

2.3.2 Improve the QA system

**Integrate the** create\_retrieval\_chain , create\_stuff\_documents\_chain, ChatPromptTemplate, and adjust top\_k to 3 to improve the QA system

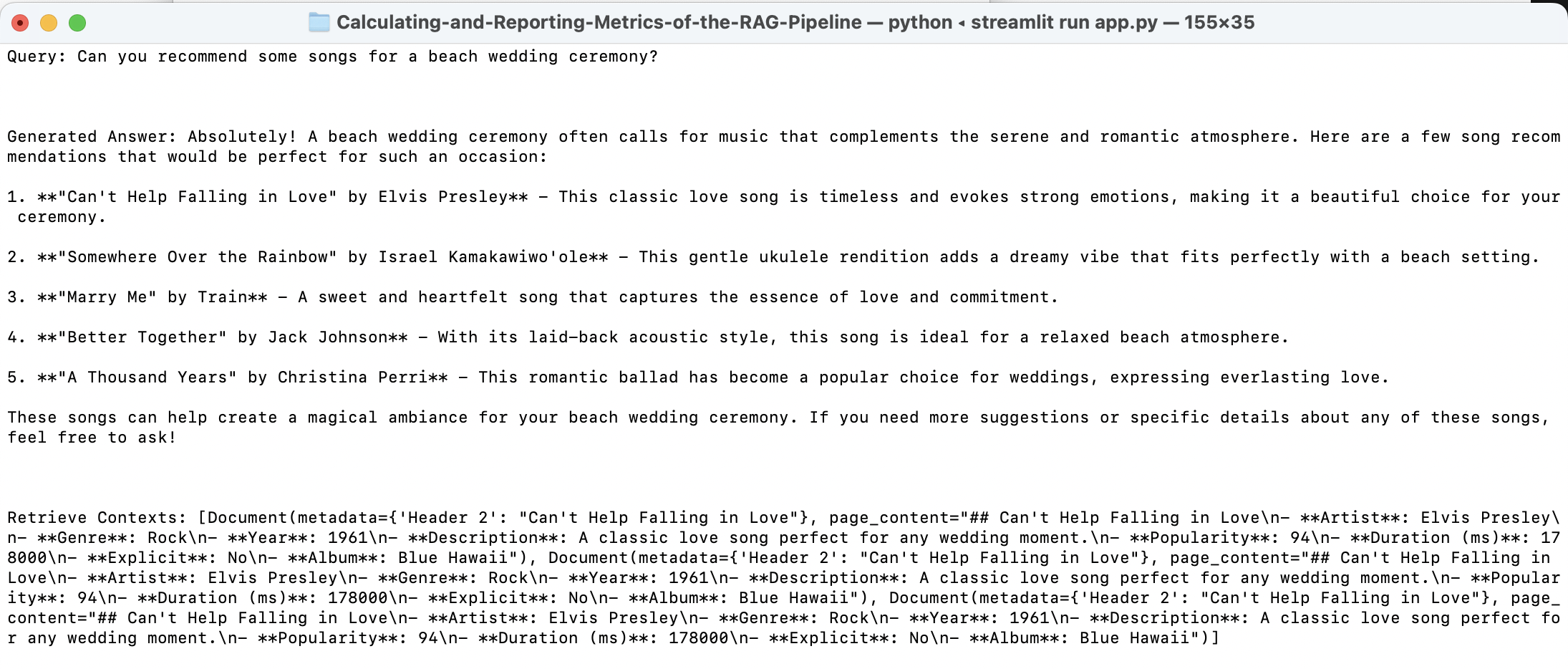
2.3.3 Adjust the LLM settings

Adjust model from gpt-3.5-turbo to the newest gpt-4o-mini, change the temperature from 0.2 to 0.4 to make the LLM more creative

2.4 Impact on the overall performance of the RAG pipeline.

2.4.1 Retrieval context relevance improved

The retrieve context of query “Can you recommend some songs for a beach wedding ceremony ”is “Can’t Help Falling in Love”, instead of the wedding music dataset introduction before



2.4.2 Generation answer’s faithfulness, answer relevance, and Information Integration

improved

When send the query “What are some classical music suitable for the first dance?”, chatbot will give the music in genre classical, instead of jazz before, but also give a jazz as an alternative choice, but will mentation it’s genre is jazz. And integrate 3 pieces of classical music to user.



# **3. Negative impact**

Latency time:

Query 1: 9.52

Query 2: 7.78

Query 3: 3.17

Average latency time: 6.82

Significant increase in latency, almost 2.5 times that of version 1

# **4. Challenges faced and addressed**

1. No ideas on how to measure individual indicators, especially since the evaluation of many of them seems to be very subjective.

-Enhance my understanding by googling and GPT, and referring to the blog posts provided by the TA

2. Want to print the text retrieved for each question, but don't know how to get.

- Look up the official documentation of LangChain and Pinecone to find the corresponding class and usage examples and try

3. When getting the first version of the retrieved text, it prompts an error about exceeding the length of the token. It turns out that in the first version I used the prompt and the user input together as a query to do the retrieval, which resulted in a long query to retrieve and caused the error.

- Adjust only send the user input to retrieve relevant splits