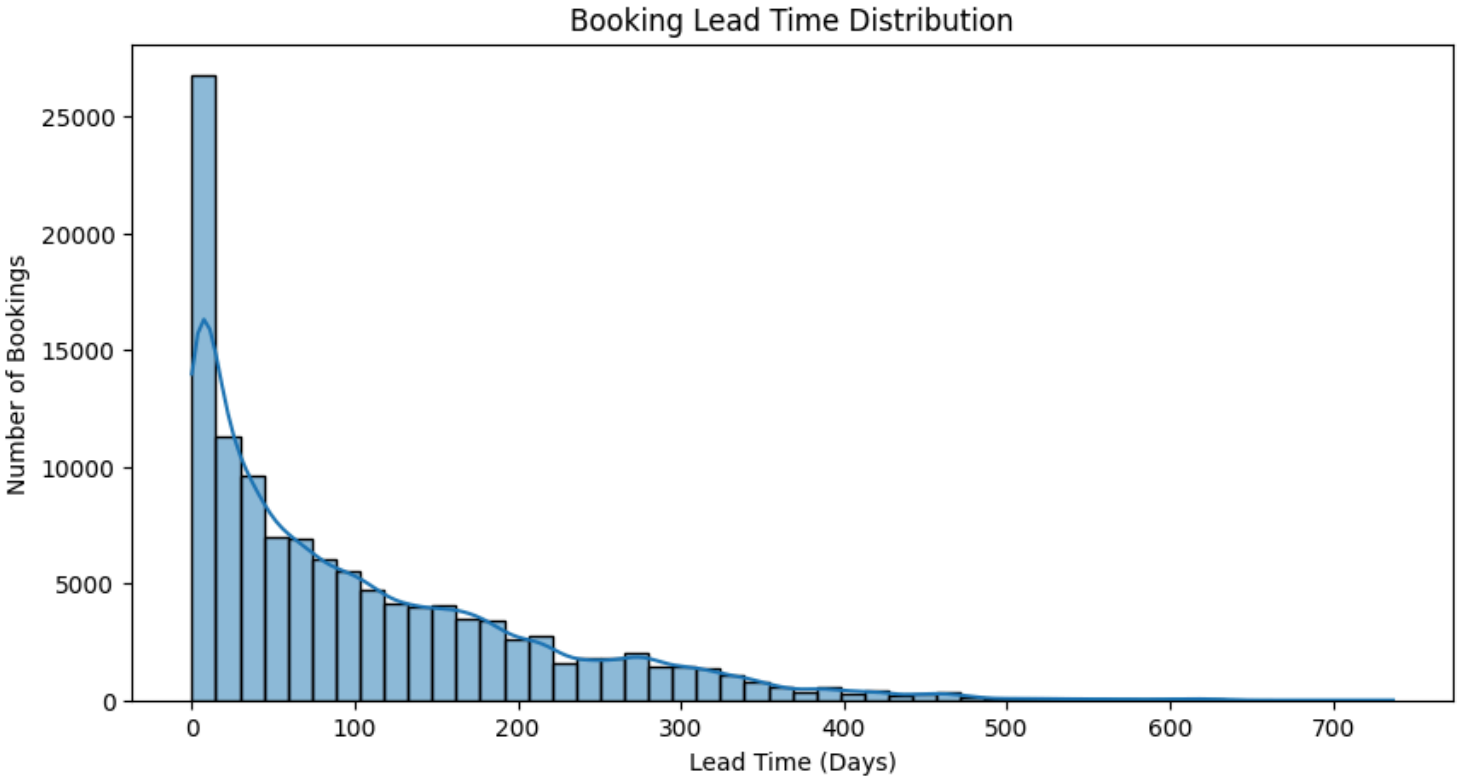
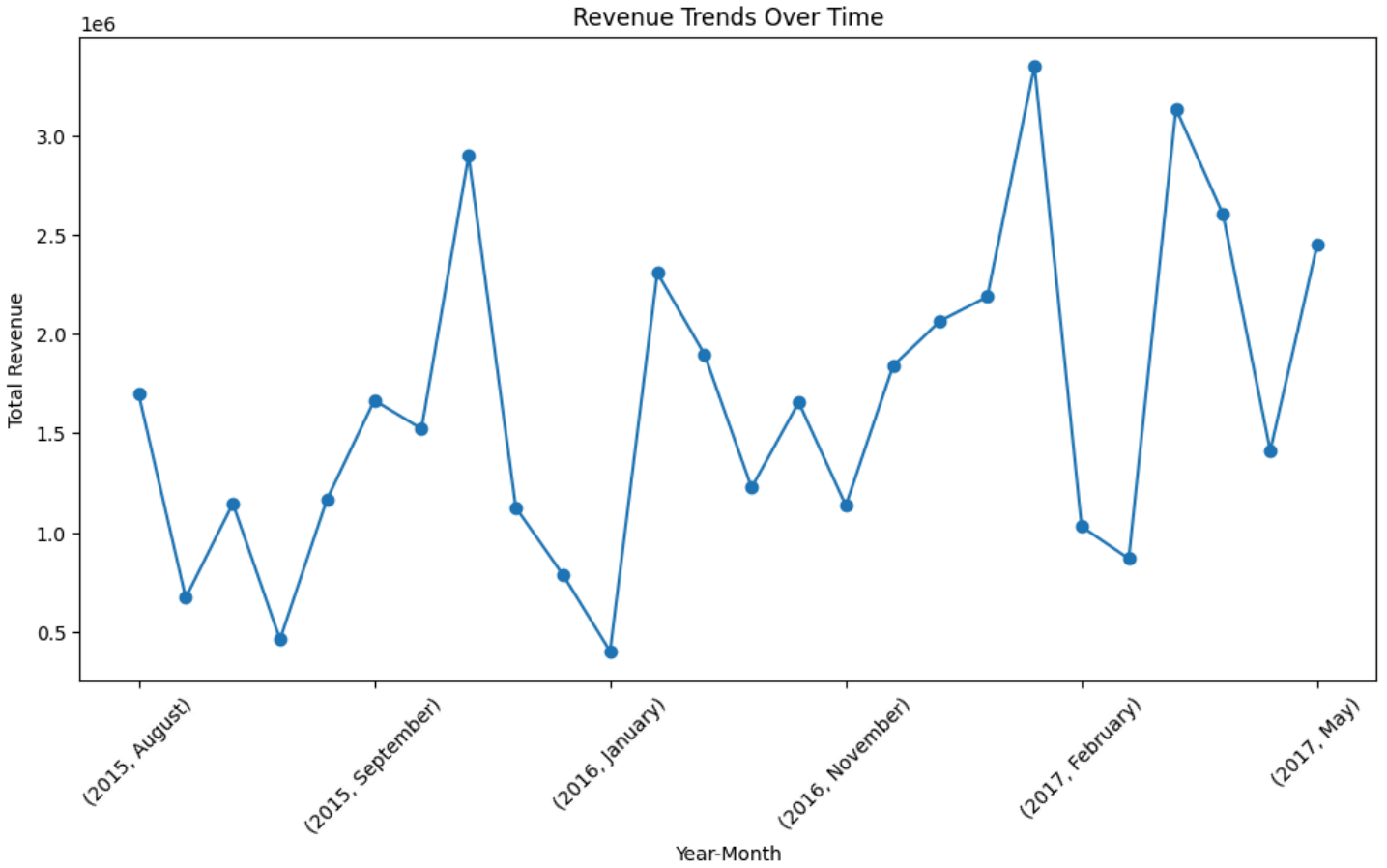
Hotel Bookings Report

Analytics & Reporting



- There is a very pronounced peak at the shortest lead times (0-20 days), with approximately 27,000 bookings made with minimal advance notice.
- The distribution has a strong right-skewed pattern, with a steep decline as lead time increases.
- Most bookings occur within the first 100 days before the stay date.

- The frequency steadily decreases as the lead time increases, becoming quite low after about 400 days.
- There are still some bookings made with very long lead times (600-700 days), but these are comparatively rare.

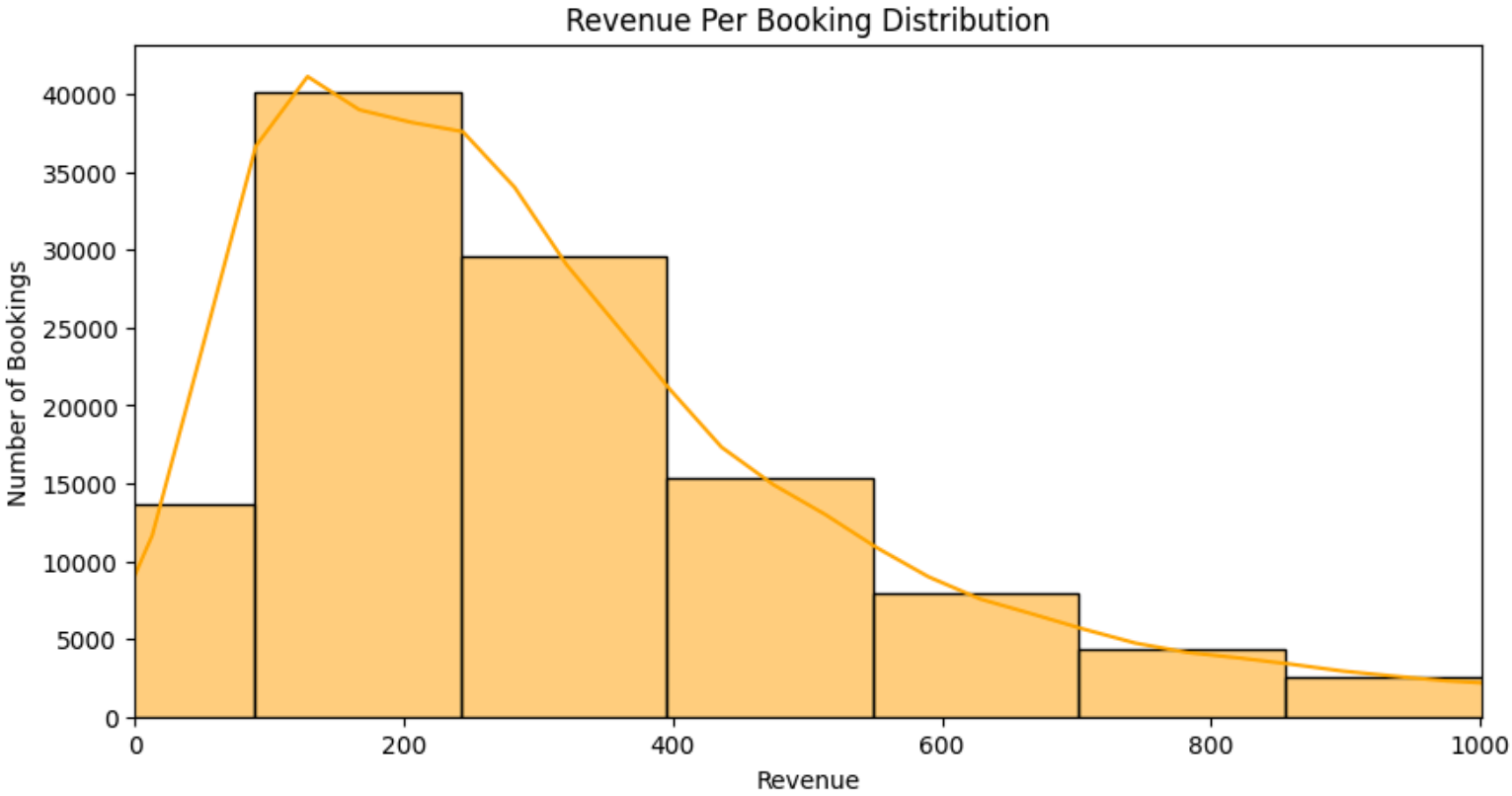


- The revenue pattern shows significant volatility throughout the period.

- There are three notable revenue peaks: one around October 2015 (approximately 2.9 million), one in early 2017 (reaching about 3.5 million, the highest point on the chart), and another peak in March 2017 (approximately 3.1 million).
- The lowest revenue point occurs around January 2016, dropping to approximately 0.4 million.
- After each peak, there tends to be a sharp decline in revenue.

- Portugal (PRT) dominates with nearly 48,000 bookings, significantly higher than any other country.
- Great Britain (GBR) is in second place with approximately 12,000 bookings.
- France (FRA) follows in third place with about 10,000 bookings.

- Spain (ESP) and Germany (DEU) are in fourth and fifth places with around 9,000 and 7,000 bookings respectively.
- The remaining countries (Italy, Ireland, Belgium, Brazil, and Netherlands) each have fewer than 5,000 bookings.
- There is a steep drop-off after Portugal, with a more gradual decline among the remaining countries.
- The distribution is heavily skewed toward Portugal, which has approximately four times more bookings than the second-ranking country.



- The histogram shows revenue distributed across several bins, with yellow color filled bars.
- The most frequent revenue range is between approximately 100-220 units, with nearly 40,000 bookings in this category.
- The second most common revenue range is between 220-400 units, with about 30,000 bookings.
- There is a smooth density curve (in darker orange) overlaid on the histogram that peaks around 170 revenue units.
- The distribution is right-skewed (positively skewed), with a long tail extending toward higher revenue values.

Retrieval-Augmented Question Answering (RAG)

I have implemented RAG effectively with the following components:

- **Vector Database**: I have implemented FAISS as my vector database in model_db.py, where I generate embeddings for each booking record and store them in a FAISS index for semantic similarity search.
- **Embedding Model**: My code uses Sentence Transformers with the "all-MiniLM-L6-v2" model to create embeddings of text descriptions generated from booking data.
- **LLM Integration**: I have tested three models:
    - mistralai/Mistral-7B-Instruct-v0.2 - On Kaggle
    - meta-llama/Llama-2-7b-chat-hf - On Kaggle
    - TinyLlama/TinyLlama-1.1B-Chat-v1.0 - On Local Machine

  I first implemented mistralai/Mistral-7B-Instruct-v0.2, but the results were not good, so I used the model meta-llama/Llama-2-7b-chat-hf, and the results were better than before. I tried these models on Kaggle as my local

machine does not have a powerful GPU. On my local machine, I have implemented an open-source LLM using TinyLlama ("TinyLlama/TinyLlama-1.1B-Chat-v1.0") for answering questions based on the retrieved context. My system can fall back to traditional filtering if the LLM is unavailable.

The notebooks for Llama are available on: https://github.com/ganesh-stem/HotelBookingsRAG/tree/main/Notebooks

Text Generation:

- The performance of Mistral and TinyLlama is not good. Mistral performs better than TinyLlama.
- I tried Llama 2 7B on the dataset with 2,500 instances and on full data. I asked seven questions:

On the Full Dataset:

Response time ranged from 9 to 15 seconds

Hotel Booking RAG Evaluation

Metric	Expected Output	RAG Output	Result
Total revenue for July 2017	3,132,959.07	3,132,959.07	✅
Cancellation rate for resort hotels	27.76%	28%	✅
Country with highest bookings	PRT (48,590 bookings)	PRT (7,438)	❌
Number of Bookings with special requests	49,072	17,699	❌
Country with highest cancellations	PRT (27,519 cancellations)	United States (223 cancellations)	❌
Percentage of repeat guests	3.19%	4.4%	❌
Average price of a hotel booking	101.83	101.83	✅

Score: 3/7 (42.86%)

On dataset with 25,00 instances:

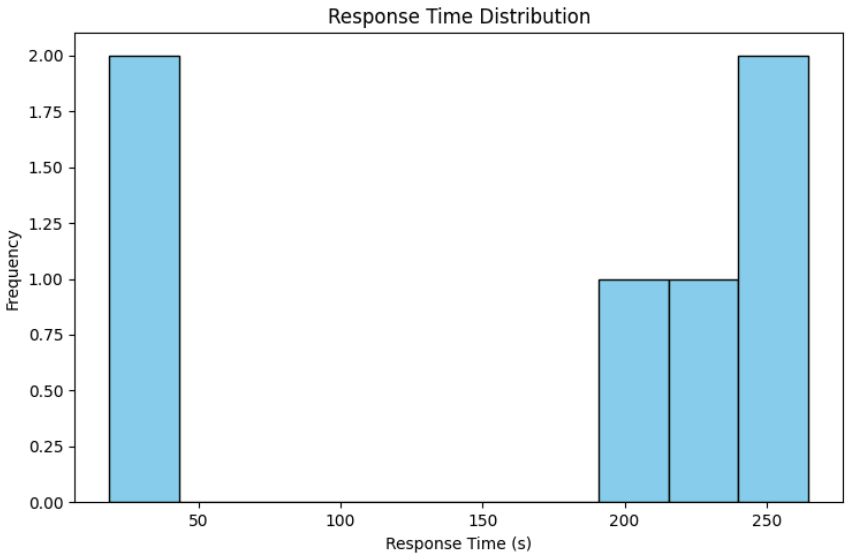
Hotel Booking RAG Evaluation

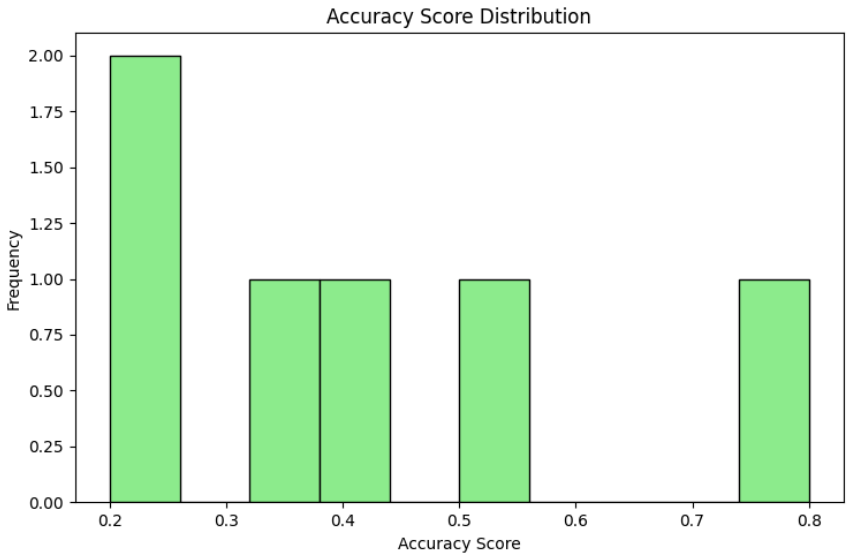
Metric	Expected Output	RAG Output	Result
Total revenue for July 2017	62,197.98	$62,197.98	✅
Cancellation rate for resort hotels	30.25%	30%	✅
Country with highest bookings	PRT (1,030 bookings)	PRT (621)	❌
Number of Bookings with special requests	994	994	✅
Country with highest cancellations	PRT (621 cancellations)	PRT (621 cancellations)	✅
Percentage of repeat guests	2.68%	2.6%	✅
Average price of a hotel booking	101.75	101.75	✅

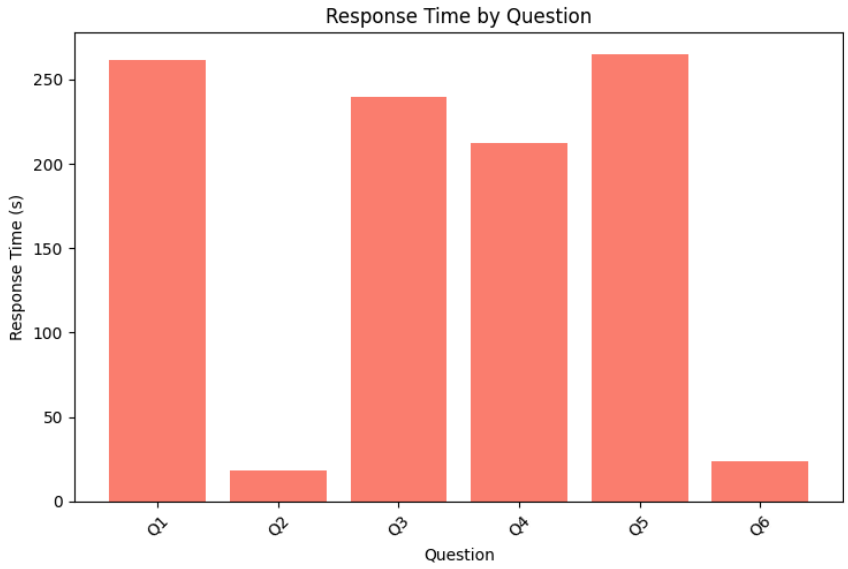
Score:

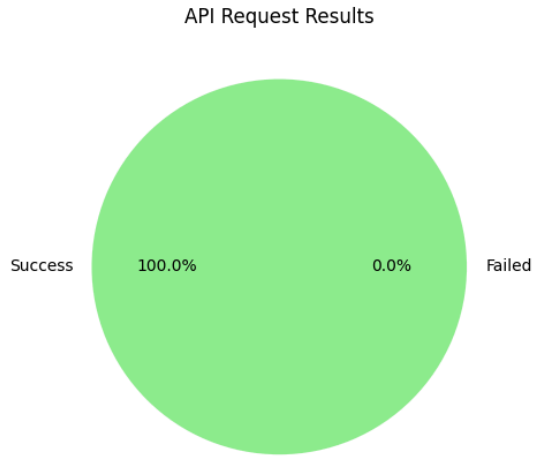
- **Correct:** 6
- **Incorrect:** 1
- **Total Score:** 6/7
- **Percentage: (6/7) × 100 = 85.71%**

By Tiny ML On Dataset with 2,500 instances:


Response Time Distribution
Frequency
Response Time (s)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
50
100
150
200
250


Accuracy Score Distribution
Frequency
Accuracy Score
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Response Time by Question
Response Time (s)
Question
0
50
100
150
200
250
Q1
Q2
Q3
Q4
Q5
Q6


API Request Results
Success
100.0%
0.0%
Failed

API Development

My API is built with Flask and includes all required endpoints:

- **POST /analytics**: Implemented in main.py, this endpoint returns analytics reports based on natural language queries or specific filters and metrics. It leverages the RAG system to process both types of analytics requests.
- **POST /ask**: This endpoint answers booking-related questions by using the RAG system to find relevant data and generate natural language answers.

Ask

curl -X POST http://localhost:8008/analytics -H "Content-Type: application/json" -d "{\"query\": \"Show me the cancellation rate for resort hotels\"}"

```
(llama_env) E:\workspace\flask_app\llama_rag\src>curl -X POST http://localhost:8008/ask -H "Conten
t-Type: application/json" -d "{\"question\": \"Which country has the highest number of bookings?\"
}"
{"answer":"Based on the data provided, the distribution of channel types is: TA/TO, Direct, Corpor
ate, GDS, Online TA, Offline TA and Groups.","processing_time_seconds":6.023974418640137,"status":
"success","thread_id":41888,"timestamp":1743167202.3025708}
```

Analytics

curl -X POST http://localhost:8008/analytics -H "Content-Type: application/json" -d "{\"query\": \"Show me the cancellation rate for resort hotels\"}"

```
(llama_env) E:\workspace\flask_app\llama_rag\src>curl -X POST http://localhost:8008/analytics -H "
Content-Type: application/json" -d "{\"query\": \"Show me the cancellation rate for resort hotels\
"}"
{"processing_time_seconds":0.13668060302734375,"record_count":165,"results":{"adr_by_hotel":{"Reso
rt Hotel":95.76609480812641},"avg_price":95.76609480812641,"canceled_bookings":268,"cancellation_r
ate":0.30248306997742663,"filtered_records":165,"hotel_distribution":{"Resort Hotel":886},"max_pri
ce":437.0,"median_price":75.0,"min_price":0.0,"revenue_by_channel":{"Corporate":11768.6,"Direct":7
2212.79,"TA/TO":289442.58},"revenue_by_segment":{"Complementary":0.0,"Corporate":5992.05,"Direct":
64059.87,"Groups":38131.1,"Offline TA/TO":84516.13,"Online TA":180724.82},"top_companies":{"-1.0":
820,"86.0":1,"135.0":2,"154.0":2,"204.0":1,"223.0":17,"281.0":3,"307.0":2,"331.0":3,"498.0":2},"to
tal_bookings":886,"total_records":886},"status":"success","thread_id":28884,"timestamp":1743169268
.4203658}
```

This API response presents a detailed analysis of resort hotel bookings with a focus on cancellations. The system processed the query in just 0.14 seconds, analyzing 886 total resort hotel bookings, of which 268 were canceled, resulting in a cancellation rate of 30.25%. The average daily rate for these resort hotel stays was $95.77, with prices ranging from free stays to a maximum of $437, and a median price of $75. Revenue analysis shows that the majority of bookings came through travel agencies and tour operators (TA/TO), generating $289,442.58, followed by direct bookings at $72,212.79, and corporate bookings at $11,768.60. The market segment breakdown further reveals that online travel agencies contributed the most revenue at $180,724.82. Most bookings were from individual consumers rather than corporate accounts, as indicated by the predominance of the -1.0 company ID in the data.

I built a REST API using Flask in `main.py` with the following endpoints:

- `POST /ask` - Answers natural language questions about hotel bookings
- `POST /analytics` - Provides analytical reports based on queries or specific filters
- `GET /health` - Checks system health and component status
- `POST/PUT/DELETE /bookings` - Endpoints for managing booking data
- `POST /bookings/batch` - Batch import of booking data
- `POST /refresh` - Force refresh of the data from the database
- `GET /metrics` - API usage metrics and query history

Performance Evaluation

I have created a comprehensive performance evaluation framework in performance_evaluation.py:

python performance_evaluation.py --url http://localhost:8008

- **Accuracy Evaluation**: My system evaluates the accuracy of answers by checking for expected keywords in responses.
- **Response Time Measurement**: The framework measures and records response times for all API endpoints and operations.
- **Visualization**: The evaluation generates visualizations of performance metrics, including response time distributions, accuracy scores, and error rates.

I have provided information about the accuracy in the Retrieval-Augmented Question Answering (RAG) section.

Additional Features

Real-time Data Updates: My system supports real-time data updates through SQLite with thread-safe operations. When new bookings are added, updated, or deleted, the database is updated, and the FAISS index is refreshed to reflect the changes.

I have created a file named data_update.py to perform this operation.

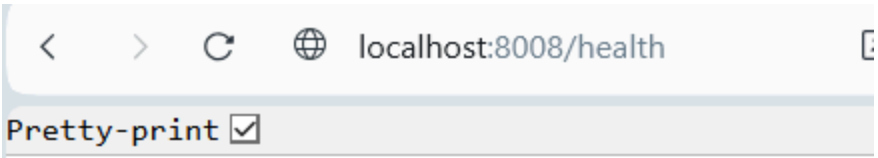
Query History Tracking: I have implemented query history tracking in the query_history table, recording timestamps, queries, responses, and processing times.

We can see the questions that have been tracked by this command: **curl http://localhost:8008/metrics**


```
(llama_env) E:\workspace\flask_app\llama_rag\src>curl http://localhost:8008/metrics
{"database":{"recent_updates":[{"details":"Imported from data/hotel_bookings.csv","id":1,"operation":"initial_import","record_count":2500,"timestamp":"2025-03-28T16:38:13.213234"}],"stats":{"canceled_bookings":964,"cancellation_rate":0.3856,"distinct_countries":68,"hotel_distribution":{"City Hotel":1614,"Resort Hotel":886},"last_update":"2025-03-28T16:38:13.060349","total_bookings":2500,"total_updates":1,"year_distribution":{"2015":494,"2016":1164,"2017":842}}},"metrics":{"analytics_queries":1,"avg_response_time":0,"questions":2,"total_queries":3},"query_history":[{"question":"Which country has the highest number of bookings?","thread_id":25460,"timestamp":1743166870.1737318,"type":"question"},{"question":"Which country has the highest number of bookings?","thread_id":41888,"timestamp":1743167196.2785964,"type":"question"},{"query":"Show me the cancellation rate for resort hotels","thread_id":28884,"timestamp":1743169268.3727083,"type":"analytics"}],"recent_questions":["Which country has the highest number of bookings?","Which country has the highest number of bookings?"],"status":"success","thread_id":28880,"timestamp":1743169696.1384864}
```

This key will show only after questions that have been asked.

Health Check Endpoint: The /health endpoint checks the status of all system components, returning detailed information about the database, embedding model, FAISS index, and LLM.



Pretty-print ☑


```
{
  "components": {
    "database": "ok",
    "embedding_model": "ok",
    "faiss_index": "ok",
    "llm": "ok"
  },
  "database_stats": {
    "canceled_bookings": 964,
    "cancellation_rate": 0.3856,
    "distinct_countries": 68,
    "hotel_distribution": {
      "City Hotel": 1614,
      "Resort Hotel": 886
    },
    "last_update": "2025-03-28T16:38:13.060349",
    "total_bookings": 2500,
    "total_updates": 1,
    "year_distribution": {
      "2015": 494,
      "2016": 1164,
      "2017": 842
    }
  },
  "dataset_size": 2500,
  "llm_status": "LLM initialized",
  "message": "All systems operational",
  "status": "ok",
  "thread_id": 30172,
  "timestamp": 1743166694.3814
}
```

Challenges

GPU: My local machine does not have a powerful GPU, which is why I use the TinyLLama model, and its performance is not good. However, I have successfully implemented RAG with the Mistral and Llama models on Kaggle.

Thread Safety: While working on this, I was getting thread-related problems which I resolved later. I have implemented thread locks and connection management to ensure safe concurrent access to the database and FAISS index.