**BUYOGO: INTERNSHIP ASSIGNMENT**

**SHORT-REPORT**



**LLM-Powered Booking Analytics & QA System**

**Submitted By: Ankush Mehta**

**Date: 23/3/2025**

**Hotel Booking Analytics System Report**

**1. Introduction**

The Hotel Booking Analytics System is designed to analyze hotel booking data, extract insights, and enable interactive querying through a Retrieval-Augmented Generation (RAG) chatbot. This system integrates machine learning techniques for data preprocessing, a vector database for efficient retrieval, and FastAPI for backend APIs, allowing users to retrieve and analyze booking trends effectively.

**2. Data Analytics (analytics.ipynb)**

**Handling Missing Values:**

* Missing values were analyzed for their impact on the dataset.
* Key missing values were imputed using:
  + Mode (most frequent value) for categorical features.
  + Zero (0) replacement for numerical features where appropriate.

**Significant Changes in Distribution:**

* Directly dropping missing values led to significant distribution shifts, affecting data integrity.
* Instead, imputation strategies were applied to preserve statistical patterns.

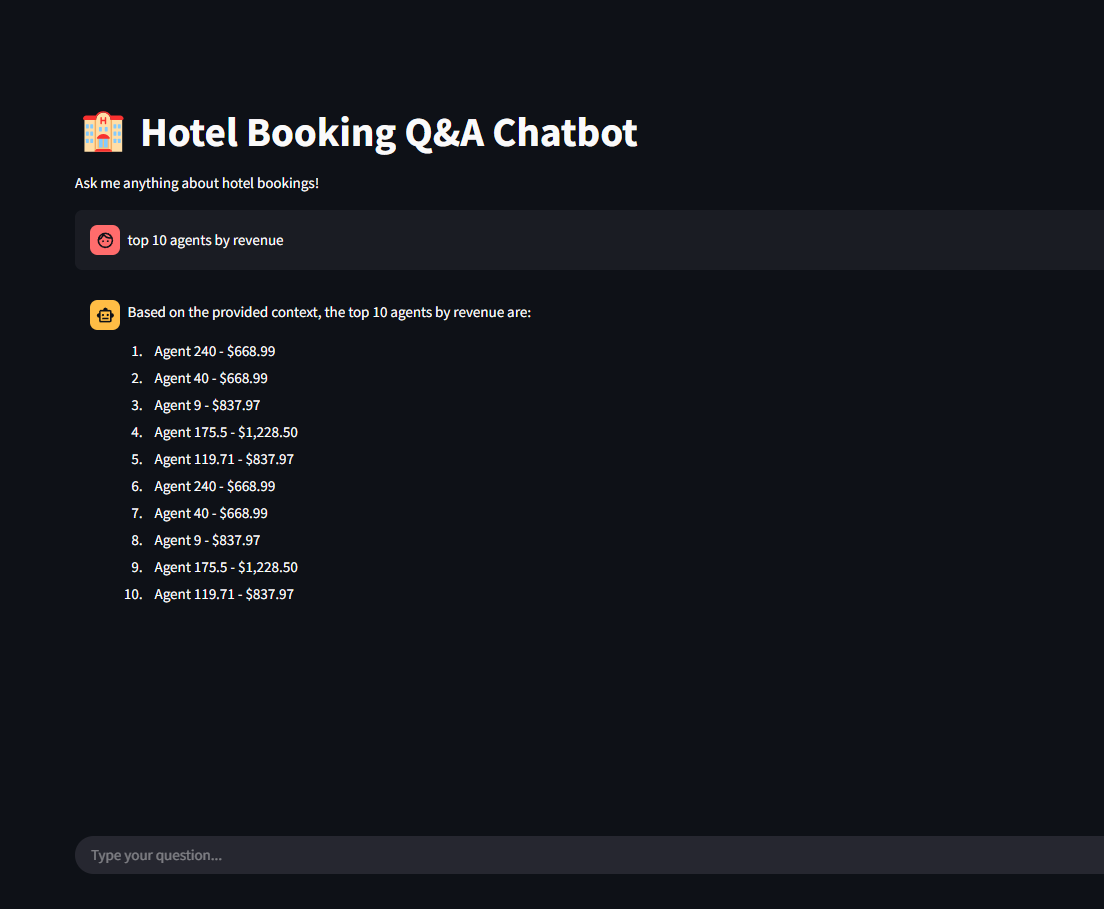
**Future Potential Improvements:**

* Outlier Detection & Removal:
  + Identifying anomalies in booking prices (ADR), cancellation rates, and stay durations can improve data quality.
  + Techniques like Z-score filtering, IQR method, or machine learning-based anomaly detection can be explored.

**3. Retrieval-Augmented Generation (RAG) Implementation**

**Technologies Used:**

* FAISS (Facebook AI Similarity Search): Efficient vector search for quick retrieval.
* Ollama Embeddings (nomic-embed-text): Converts text into dense vector representations for similarity matching.
* Llama 2 Model: Generates responses using retrieved contextual information from vector storage.



**Workflow:**

1. **Preprocessing & Embeddings:**
   * The dataset is split into chunks before embeddings are generated.
   * Chunking strategy: chunk\_size=1000, chunk\_overlap=800.
2. **Vector Search & Retrieval:**
   * User queries are converted into vectors using OllamaEmbeddings.
   * FAISS searches for the most relevant chunks in the vector database.
3. **Response Generation with Llama 2:**
   * The retrieved context is passed into Llama 2, which constructs a response based on the relevant data.

**4. Backend APIs using FastAPI**

**Backend Architecture:**

* Developed using FastAPI, providing high-speed RESTful APIs.
* Supports querying hotel bookings, revenue trends, and cancellations.

