Mohammed Adil
https://huggingface.co/spaces/aadil732/Analytics-Vidya-Free-Courses

Analytics Vidya Free Course Search Engine

1. Introduction

This report outlines the development of a free courses search engine using Retrieval-Augmented Generation (RAG). The project aims to address the challenge of retrieving relevant course information from Analytics Vidya website with Web Scraping while providing an efficient and user-friendly experience.

Objectives

- Develop a semantic search engine leveraging RAG.
- Ensure fast, accurate, and contextually relevant course recommendations based on keywords and natural language.

2. Approach and Solution

2.1 Problem Understanding

Traditional keyword-based search methods often fail to capture the semantic nuances of user queries. This project addresses these limitations by implementing RAG to enhance search capabilities and deliver more relevant results.

2.2 Proposed Solution

The solution integrates a retrieval mechanism with an LLM-based generation process, improving accuracy and providing detailed, context-aware responses.

3. Architecture and Workflow

3.1 System Architecture

Key Components:

- **Embedding Model:** Generates dense vector representations of queries and course data.
- **Vector Database:** Efficiently stores and retrieves course embeddings.
- **Large Language Model (LLM):** Processes retrieved data to generate coherent responses.
- **Web UI:** Ensures efficient communication between user and backend.

3.2 Workflow

1. System work:
    a. uploading_to_database.py ensures the fetching, filtering, embedding and storing the course data to **Pinecone Vector Database.**
    b. This file utilizes extract_all_courses.py and extract_single_course.py files to extract and filter the course details.
2. User Interaction:
    a. app.py is a Streamlit python file for Web UI.
    b. retriever.py is a FastAPI endpoint file that accepts the user keyword, fetches the similar keywords from the vector database, and provides the results to the LLM model.
    c. The LLM model make the results to be well-structured and easily readable in markdown.

4. Methodology

4.1 Embedding Model Selection

Criteria:

- Accuracy in representing semantic meaning.
- Speed and efficiency.
- Compatibility with vector database.

Selected Model: Google's `text-embedding-004` embedding model, chosen for its high performance and reliability.

4.2 Vector Database

Criteria:

- Accurate similarity search on remote database.

- Speed and efficiency.
- Free to use upto limit.

Selected Database: Pinecone vector database, for its scalable and efficient nearest-neighbor search capabilities. It is free to use upto 5 indexes and a high number of requests supported.

4.3 LLM Selection

Considerations:

- Fluency and coherence in generating responses.
- Free to use upto 15 requests per minute.

Selected Model: Google Gemini 1.5 Flash's API, chosen for their superior language understanding and response generation.

5. Implementation Details

5.1 Tools and Technologies

- **Languages & Frameworks:** Python, Streamlit, FastAPI.
- **APIs & Libraries:** LangChain, Pinecone, Google generative ai.
- **Deployment:** Deployed using Docker on Hugging Face Space.

5.2 Code Structure

Modules:

- **Frontend:** Built with Streamlit for interactive user interfaces.
- **Backend:** FastAPI handles query processing and integration.
- **Integration:** Seamless communication between vector database and LLM.

5.3 Challenges and Solutions

Challenges:

- Managing latency for queries.
- Ensuring the integrity of course details.
- Ensuring relevance of retrieved courses.

Solutions:

- Google’s LLM and embedding model for fast response.
- Keeping each course in a single vector for data integrity.
- Using Pinecone’s Cosine similarity search for most relevant courses search.

7. Conclusion

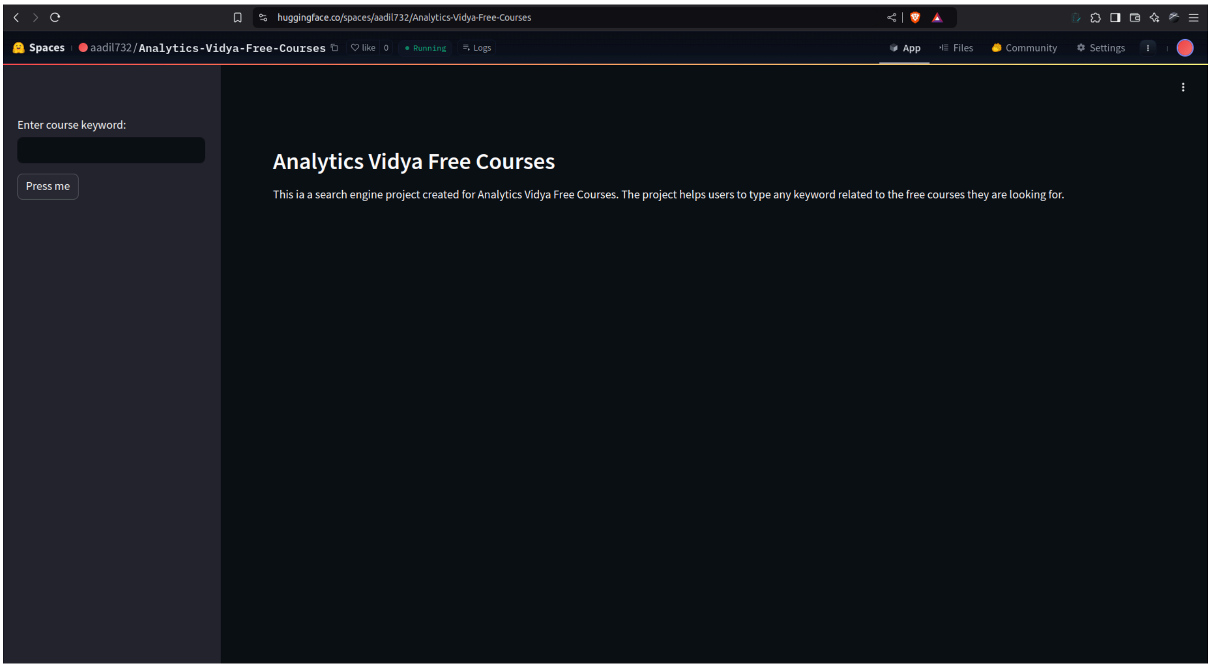
The project successfully delivers an advanced search engine that bridges the gap between user intent and course availability. By leveraging RAG, it ensures contextually relevant and accurate recommendations, greatly enhancing user experience.

8. References

- LangChain documentation.
- Pinecone API guides.
- Langchain Gemini API documentation.
- Streamlit documentation.

Appendix

- **UI Screenshots:**



huggingface.co/spaces/aadil732/Analytics-Vidya-Free-Courses

Spaces | aadil732/Analytics-Vidya-Free-Courses | like 0 | Running | Logs | App | Files | Community | Settings

Enter course keyword:

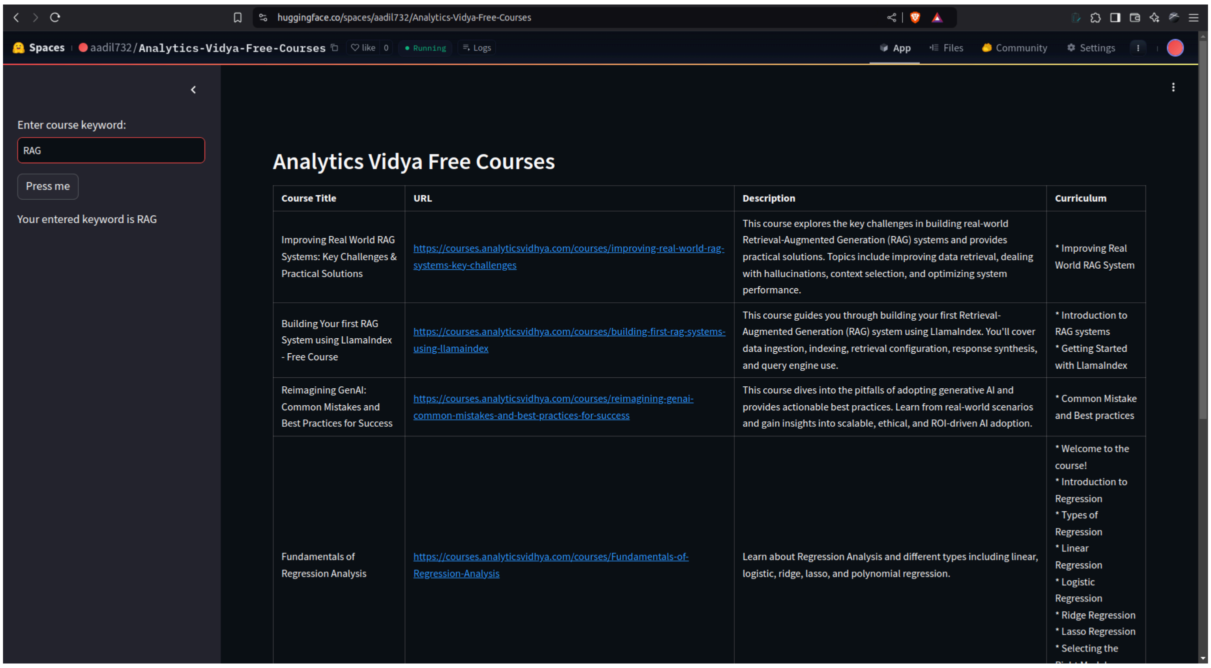
Press me

Analytics Vidya Free Courses

This ia a search engine project created for Analytics Vidya Free Courses. The project helps users to type any keyword related to the free courses they are looking for.

Mohammed Adil

https://huggingface.co/spaces/aadil732/Analytics-Vidya-Free-Courses


huggingface.co/spaces/aadil732/Analytics-Vidya-Free-Courses
Spaces
aadil732/Analytics-Vidya-Free-Courses
like 0
Running
Logs
App
Files
Community
Settings

Enter course keyword:

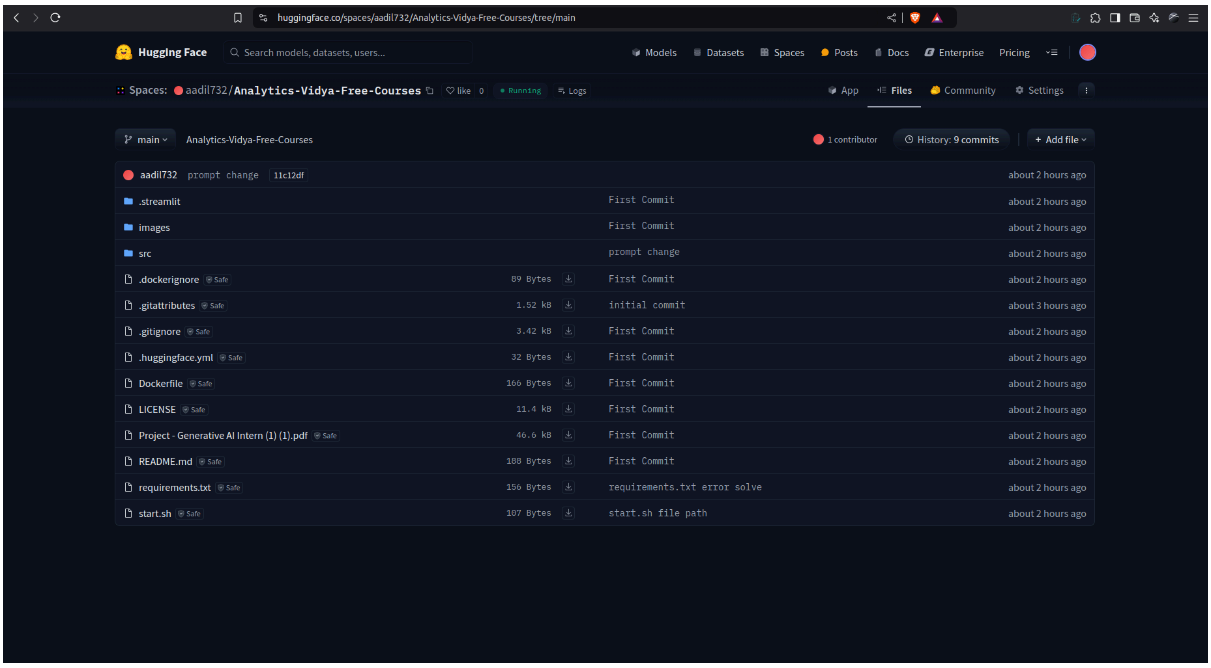
RAG

Press me

Your entered keyword is RAG

Analytics Vidya Free Courses

Course Title	URL	Description	Curriculum
Improving Real World RAG Systems: Key Challenges & Practical Solutions	https://courses.analyticsvidhya.com/courses/improving-real-world-rag-systems-key-challenges	This course explores the key challenges in building real-world Retrieval-Augmented Generation (RAG) systems and provides practical solutions. Topics include improving data retrieval, dealing with hallucinations, context selection, and optimizing system performance.	* Improving Real World RAG System
Building Your first RAG System using LlamaIndex - Free Course	https://courses.analyticsvidhya.com/courses/building-first-rag-systems-using-llamaindex	This course guides you through building your first Retrieval-Augmented Generation (RAG) system using LlamaIndex. You'll cover data ingestion, indexing, retrieval configuration, response synthesis, and query engine use.	* Introduction to RAG systems * Getting Started with LlamaIndex
Reimagining GenAI: Common Mistakes and Best Practices for Success	https://courses.analyticsvidhya.com/courses/reimagining-genai-common-mistakes-and-best-practices-for-success	This course dives into the pitfalls of adopting generative AI and provides actionable best practices. Learn from real-world scenarios and gain insights into scalable, ethical, and ROI-driven AI adoption.	* Common Mistake and Best practices
Fundamentals of Regression Analysis	https://courses.analyticsvidhya.com/courses/Fundamentals-of-Regression-Analysis	Learn about Regression Analysis and different types including linear, logistic, ridge, lasso, and polynomial regression.	* Welcome to the course! * Introduction to Regression * Types of Regression * Linear Regression * Logistic Regression * Ridge Regression * Lasso Regression * Selecting the


huggingface.co/spaces/aadil732/Analytics-Vidya-Free-Courses/tree/main
Hugging Face
Search models, datasets, users...
Models
Datasets
Spaces
Posts
Docs
Enterprise
Pricing
Spaces: aadil732/Analytics-Vidya-Free-Courses
like 0
Running
Logs
App
Files
Community
Settings
main
Analytics-Vidya-Free-Courses
1 contributor
History: 9 commits
+ Add file

aadil732	prompt change	11c12df		about 2 hours ago
.streamlit			First Commit	about 2 hours ago
images			First Commit	about 2 hours ago
src			prompt change	about 2 hours ago
.dockerignore	Safe	89 Bytes	First Commit	about 2 hours ago
.gitattributes	Safe	1.52 kB	initial commit	about 3 hours ago
.gitignore	Safe	3.42 kB	First Commit	about 2 hours ago
.huggingface.yml	Safe	32 Bytes	First Commit	about 2 hours ago
Dockerfile	Safe	166 Bytes	First Commit	about 2 hours ago
LICENSE	Safe	11.4 kB	First Commit	about 2 hours ago
Project - Generative AI Intern (1) (1).pdf	Safe	46.6 kB	First Commit	about 2 hours ago
README.md	Safe	188 Bytes	First Commit	about 2 hours ago
requirements.txt	Safe	156 Bytes	requirements.txt error solve	about 2 hours ago
start.sh	Safe	107 Bytes	start.sh file path	about 2 hours ago

https://www.linkedin.com/in/mohammed-adil-silawat/