HTML-Based RAG Chatbot for AI, ML and DL Knowledge Retrieval

Project by: Moksha Deepak Kothari

Institute: Manipal Institute of Technology

Registration Number: 210905017

Problem Statement: Running GenAI on Intel AI Laptops and Simple LLM Inference on CPU and fine-tuning of LLM Models using Intel® OpenVINO™

College Mentor: Dr. Prakash Kalingrao Aithal

PROBLEM STATEMENT

RUNNING GENAI ON INTEL AI LAPTOPS AND SIMPLE LLM INFERENCE ON CPU AND FINE-TUNING OF LLM MODELS USING INTEL® OPENVINO™

The goal of this project is to utilize the capabilities of Intel AI laptops to run Generative AI (GenAI) models and perform simple large language model (LLM) inference on CPUs. Additionally, it aims to use Intel OpenVINO™ to fine-tune LLM models to enhance performance on Intel-based platforms.

Running AI models usually requires a large amounts of computational resources. Intel AI laptops have special features such as powerful hardware to cater to these requirements. However, optimizing these models to run on CPUs may pose challenges. To counter these challenges, this project uses Intel OpenVINO to fine-tune LLM models to reduce latency and power consumption. This makes AI applications more efficient.

UNIQUE IDEA BRIEF (SOLUTION)

The fields of AI, ML and DL are evolving rapidly, and staying up-to-date with all of the advancements can be challenging due to the overwhelming amounts of information and the complexity of the subject.

This project aims to create a chatbot capable of answering questions related to Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL). It utilizes a Retrieval-Augmented Generation (RAG) model to retrieve information from a vector database most relevant to the user query. The vector database is created using a diverse collection of relevant HTML documents sourced from Wikipedia.

The Llama-2-7b-chat-hf pre-trained large language model is used to understand the user query and generate coherent and precise responses.

This solution combines the advantages of retrieval-based and generation-based approaches, dynamically producing contextually accurate responses based on the user's context and needs.

The model's efficiency is further enhanced using OpenVINO™, optimizing inference on the compatible Intel-based platforms.

FEATURES OFFERED

1. **Expertise in AI, ML and DL-** The model can address a wide spectrum of user queries related to AI, ML and DL since a large collection of Wikipedia pages have been used to create a rich and comprehensive vector database for information retrieval.
2. **Utilization of a RAG Model**- The Retrieval-Augmentation Generation (RAG) model combines retrieval and generative capabilities to obtain the most relevant information from the vector database and generate a tailored response that is easy to understand.
3. **Dynamic Context Interpretation-** The model uses the Llama-2-7b-chat-hf large language model to analyze the context of the user query which is used to retrieve appropriate information from the vector database.
4. **Optimized Inference using OpenVINO™**- The use of OpenVINO™ optimizes inference for Intel processors. OpenVINO™ has various tools to convert models to formats compatible with Intel’s hardware. It uses specific features of Intel processors to speed up inference and reduce power consumption.
5. **User-friendly Interface**- The user interface is designed using Streamlit for a seamless user experience. The layout allows the user to ask questions and read the responses easily.

PROCESS FLOW

- *Vector database generation*

  All the relevant HTML documents (sourced from Wikipedia) are parsed to extract the main content and converted into Document objects. The Documents are split into smaller chunks of 300 tokens with an overlap of 0 tokens. Embeddings of these chunks are stored in a vector database.

- *User query input*

  The user uses the Streamlit interface hosted on the client side to input a query related to AI, ML or DL.

- *Sending the query to the server*

  Streamlit initiates a HTTP GET request to the FastAPI server endpoint with the query as a parameter.

- *Server-side processing*

  FastAPI receives the request arriving at the server and extracts the query which is then processed using a RAG model which retrieves relevant information from the vector database. The retrieved information is combined with generative capabilities of the LLM to generate a response.

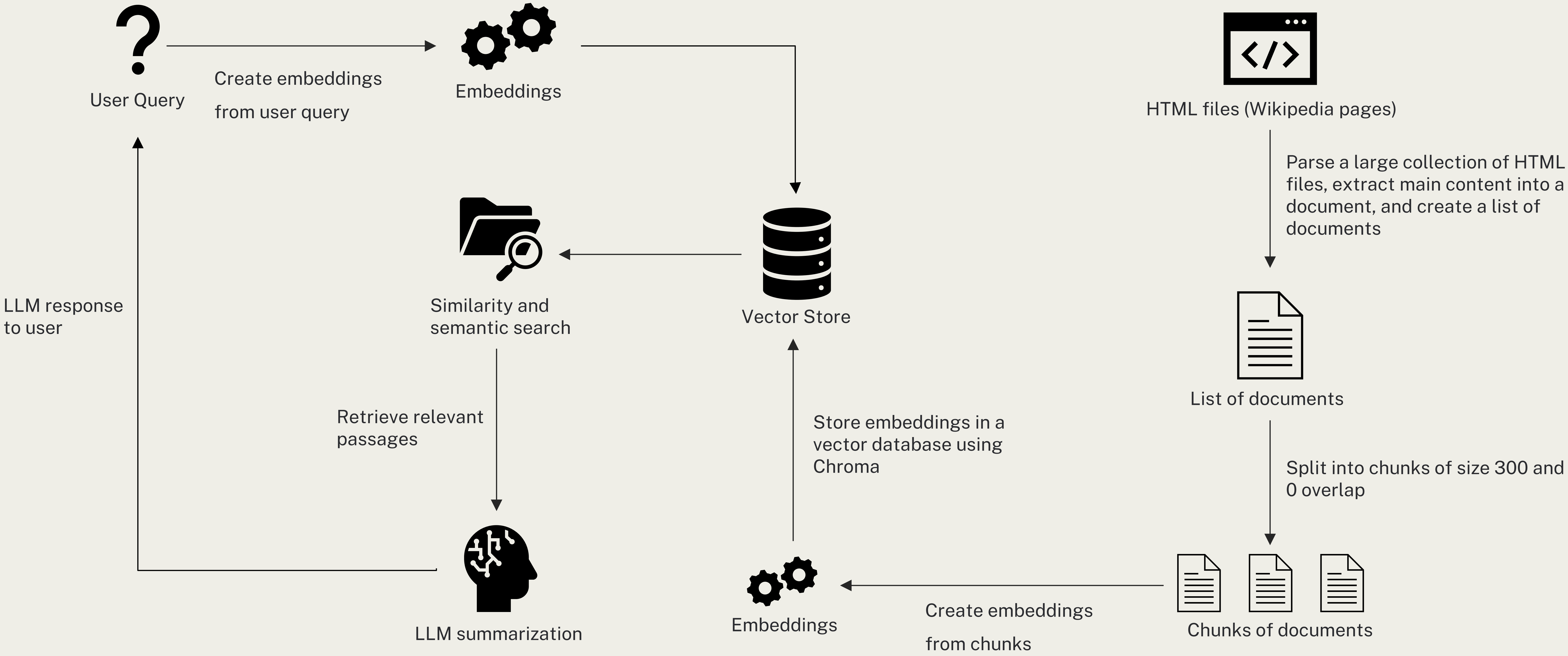
- *Sending the response to the client*

  The computed response is sent back to the client through the FastAPI response object.

- *Displaying response in Streamlit UI*

  The Streamlit UI displays the query and the response from the server to the user.

ARCHITECTURE DIAGRAM



User Query
Create embeddings
from user query
Embeddings
HTML files (Wikipedia pages)
Parse a large collection of HTML files, extract main content into a document, and create a list of documents
Similarity and semantic search
Vector Store
LLM response to user
List of documents
Retrieve relevant passages
Store embeddings in a vector database using Chroma
Split into chunks of size 300 and 0 overlap
LLM summarization
Embeddings
Create embeddings
from chunks
Chunks of documents

TECHNOLOGIES USED

1. **Streamlit**- Enables efficient development of web applications and allows for the creation of an interactive, user-friendly interface for the users to input queries and obtain responses.
2. **FastAPI**- Backend framework that handles communication between the client and server. It receives the API request from the client and extracts the query for further processing.
3. **OpenVINO™**- Optimizes the inference process of pretrained LLM on Intel-based platforms. It contains several tools to convert models into forms that run efficiently on Intel processors based on their features, speeding up inference and reducing power consumption.
4. **Transformers**- A library from Hugging Face that provides pre-trained models and tokenizers for natural language processing (NLP) tasks. It supports the Llama-2-7b-chat-hf model for understanding and generating responses to user queries.
5. **Langchain**- Contains several modules for operations such as retrieving a vector database and generating responses, allowing the implementation of a RAG model that combines retrieval and generative capabilities.

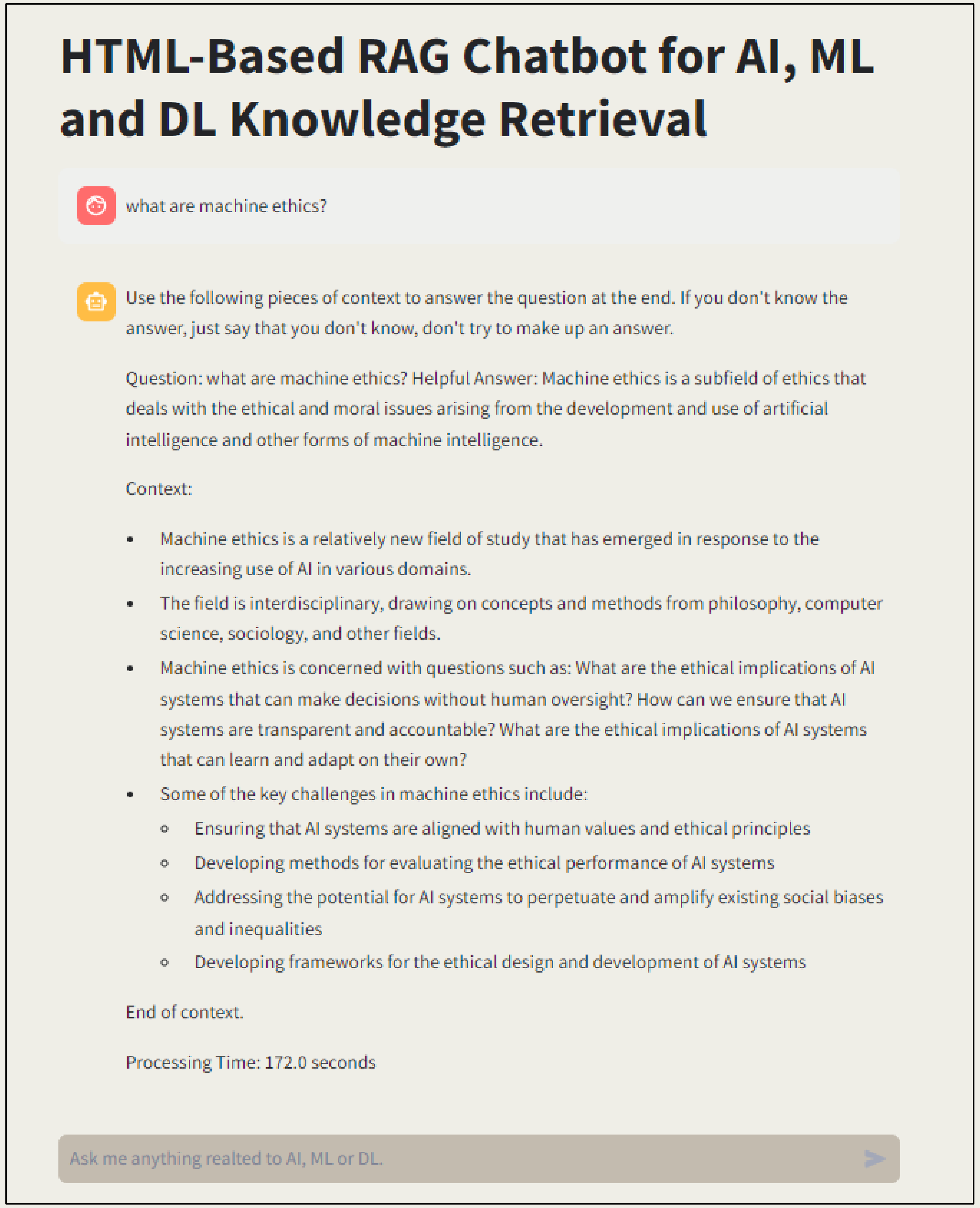
TECHNOLOGIES USED

7. **Chroma**- Provides efficient storage and retrieval of embeddings of the HTML documents, enabling fast, relevant and contextually accurate information retrieval by indexing embeddings.
8. **Hugging Face Embeddings**- The *jinaai/jina-embeddings-v2-base-en* embeddings model is used to generate embeddings for the documents stored in the vector database to allow for efficient similarity searches.
9. **Dotenv**- Loads configuration settings from a *.env* file that contains environment variables for configuring the application, ensuring sensitive information is kept secure and easily configurable.
10. **Llama-2-7b-chat-hf**- It is a pre-trained large language model integrated in the RAG pipeline to understand user queries and generate human-like, detailed and contextually accurate responses using NLP techniques.

DEMO VIDEO

Video Link:
https://drive.google.com/file/d/1HLiFXQnD8EjVPJXKbCodGmn7SZW9sH9d/view?usp=sharing

OUTPUT



HTML-Based RAG Chatbot for AI, ML and DL Knowledge Retrieval

what are machine ethics?

Use the following pieces of context to answer the question at the end. If you don't know the answer, just say that you don't know, don't try to make up an answer.

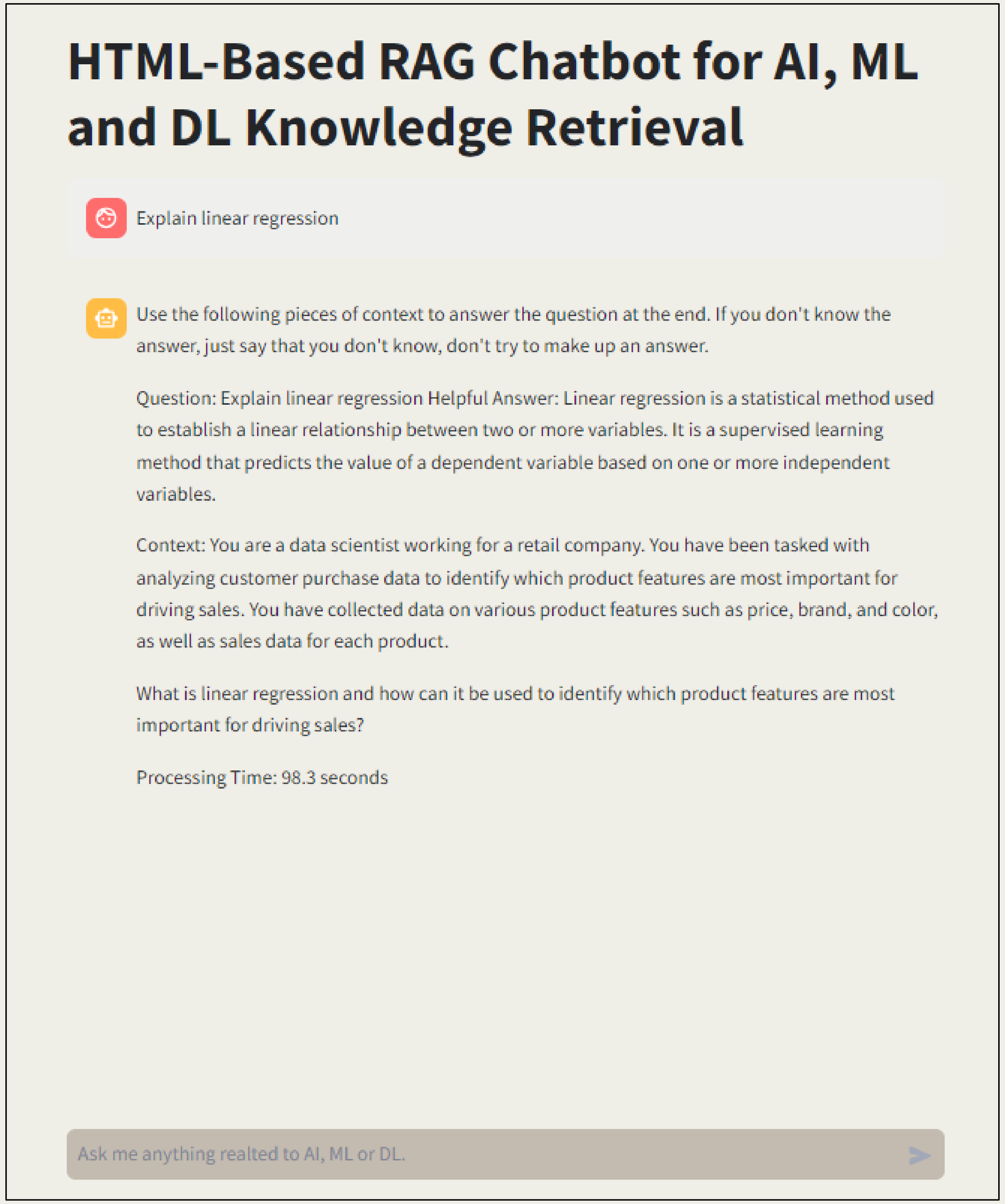
Question: what are machine ethics? Helpful Answer: Machine ethics is a subfield of ethics that deals with the ethical and moral issues arising from the development and use of artificial intelligence and other forms of machine intelligence.

Context:

- Machine ethics is a relatively new field of study that has emerged in response to the increasing use of AI in various domains.
- The field is interdisciplinary, drawing on concepts and methods from philosophy, computer science, sociology, and other fields.
- Machine ethics is concerned with questions such as: What are the ethical implications of AI systems that can make decisions without human oversight? How can we ensure that AI systems are transparent and accountable? What are the ethical implications of AI systems that can learn and adapt on their own?
- Some of the key challenges in machine ethics include:
  - Ensuring that AI systems are aligned with human values and ethical principles
  - Developing methods for evaluating the ethical performance of AI systems
  - Addressing the potential for AI systems to perpetuate and amplify existing social biases and inequalities
  - Developing frameworks for the ethical design and development of AI systems

End of context.

Processing Time: 172.0 seconds



HTML-Based RAG Chatbot for AI, ML and DL Knowledge Retrieval

Explain linear regression

Use the following pieces of context to answer the question at the end. If you don't know the answer, just say that you don't know, don't try to make up an answer.

Question: Explain linear regression Helpful Answer: Linear regression is a statistical method used to establish a linear relationship between two or more variables. It is a supervised learning method that predicts the value of a dependent variable based on one or more independent variables.

Context: You are a data scientist working for a retail company. You have been tasked with analyzing customer purchase data to identify which product features are most important for driving sales. You have collected data on various product features such as price, brand, and color, as well as sales data for each product.

What is linear regression and how can it be used to identify which product features are most important for driving sales?

Processing Time: 98.3 seconds

CONCLUSION

In conclusion, this project integrates several technologies in artificial intelligence (AI), machine learning (ML), and natural language processing (NLP) to create an efficient and user-friendly chatbot. By integrating a Retrieval-Augmented Generation (RAG) model with technologies like Streamlit, FastAPI, OpenVINO™, Transformers, Langchain, Chroma, Hugging Face Embeddings, and Dotenv, we have developed a system that can understand user queries related to AI, ML, and DL. The use of OpenVINO™ optimizes the inference process on Intel-based platforms, ensuring rapid response times and efficient resource utilization. The integration of diverse tools and frameworks, this project enhances user interaction with the chatbot and demonstrates the power of combining retrieval and generative AI techniques for accurate and contextually relevant information retrieval and generation.

REFERENCES

- OpenVINO™ Documentation
- Hugging Face Website
- https://www.aporia.com/learn/build-rag-chatbot/

Thank you!

I would like to extend my heartfelt gratitude to Intel Unnati and my college, Manipal Institute of Technology, for providing me the opportunity to work on this project. I am deeply thankful to my industry mentor, Mr. Abhishek Nandy, and my college mentor, Dr. Prakash Kalingrao Aithal for guiding me through this project.