Project Documentation

Name: John Ray Tran

Index Number: 10211100247

GitHub Repository: <https://github.com/JRTran16/AI_10211100247>

Deployed App URL: <https://jrtran16-ml-ai-explorer-app-n0gcfs.streamlit.app/>

# Project Overview

This Streamlit-based application enables users to interactively explore and solve AI/ML problems including regression, clustering, neural networks, and retrieval-augmented generation using large language models. Each module is available via a unified dashboard with intuitive interfaces.

# Feature Summary & Usage

## Regression

- Upload a CSV file with numeric data.

- Select target and feature columns.

- Apply preprocessing (remove outliers, normalize, log-transform).

- Train linear regression and visualize predictions.

- View performance (MAE, R²), and make custom predictions.

## Clustering

- Upload a CSV file with 2+ columns.

- Choose feature columns (automatic label encoding for categorical columns).

- Use a slider to pick several clusters (K).

- Visualize clusters (2D/3D) and download labeled dataset.

## Neural Network

- Upload CSV for classification task.

- Select target and feature columns (auto encoding for categorical variables).

- Adjust epochs and learning rate.

- Train Feedforward MLP (sklearn).

- Visualize training loss and make predictions using the form.

## Large Language Model (LLM) Q&A

LLM Approach: Retrieval-Augmented Generation (RAG)

Model Used: Gemini-1.5-Pro (Google Generative AI)

Embedding Model: sentence-transformers/all-MiniLM-L6-v2

Indexing: FAISS

Framework: LangChain

Workflow Methodology:

Step 1: Document Ingestion

* Accepts PDF or CSV. Text is extracted via PyPDF2 or converted from CSV row data.

Step 2: Text Chunking

* Uses LangChain RecursiveCharacterTextSplitter (chunk size: 1000, overlap: 200)

Step 3: Semantic Embedding

* Each chunk is embedded using all-MiniLM-L6-v2 from HuggingFace

Step 4: FAISS Vector Indexing

* FAISS (Facebook AI Similarity Search) is an efficient library for fast vector similarity search. It indexes embedded chunks and retrieves top-matching documents for any user question.

Step 5: Contextual Retrieval

* Uses FAISS similarity search to fetch top relevant chunks

Step 6: Query Gemini with Context

* Gemini generates answers based solely on retrieved context.

Step 7: Real-time Answer Display

* The application shows the answer and an expandable view of relevant text chunks.

**FAISS** (Facebook AI Similarity Search) is a library developed by **Meta AI (Facebook Research)** that enables **fast and scalable vector similarity search**. It's widely used in AI systems for tasks like:

* Semantic search
* Recommendation systems
* Retrieval-augmented generation (RAG)

In this project, **FAISS is used to index the embeddings** of all the document chunks. When a user asks a question, FAISS retrieves the top 3 chunks **most semantically similar** to the question. This allows the LLM to ground its answers in the most relevant parts of the document.

# Architecture Diagram (LLM RAG)

The following image shows the overall RAG system architecture used in the LLM Q&A module:

