NAOMI EDEM KUMAH
10211100289
COMPUTER SCIENCE
INTRODUCTION TO ARTIFICIAL INTELLIGENCE
END OF SEMESTER EXAMINATION

PROJECT DOCUMENTATION

HOW TO USE EACH FEATURE

1) REGRESSION

- Go to the **Services** page.
- Click **Regression**.
- Upload your CSV file.
- Select your target column and feature columns.
- Optionally preprocess (handle missing values, normalize data).
- Click **Run Regression** to train and evaluate.
- View regression results and plots.
- Enter a value for prediction if one feature is selected.

2) CLUSTERING

- Go to **Services**.
- Click **Clustering**.
- Upload a CSV.
- Select number of clusters.
- Click **Run Clustering**.

- View cluster labels and plots.

3) **NEURAL NETWORKS**

- Go to **Services**.
- Click **Neural Networks**.
- Upload your CSV dataset.
- Choose the target column.
- Set epochs and learning rate.
- Train the model.
- View accuracy and loss plots.
- Optionally upload a model or test data to make predictions.
- Download predictions if needed.

4) **LLM RAG (Q&A System)**

- Go to **Services**.
- Click **Large Language Model**.
- Ask a question about the **Academic City Student Policy**.
- Get AI-generated responses based on relevant document passages.
- View confidence scores for the answer.

DATASET AND MODELS USED

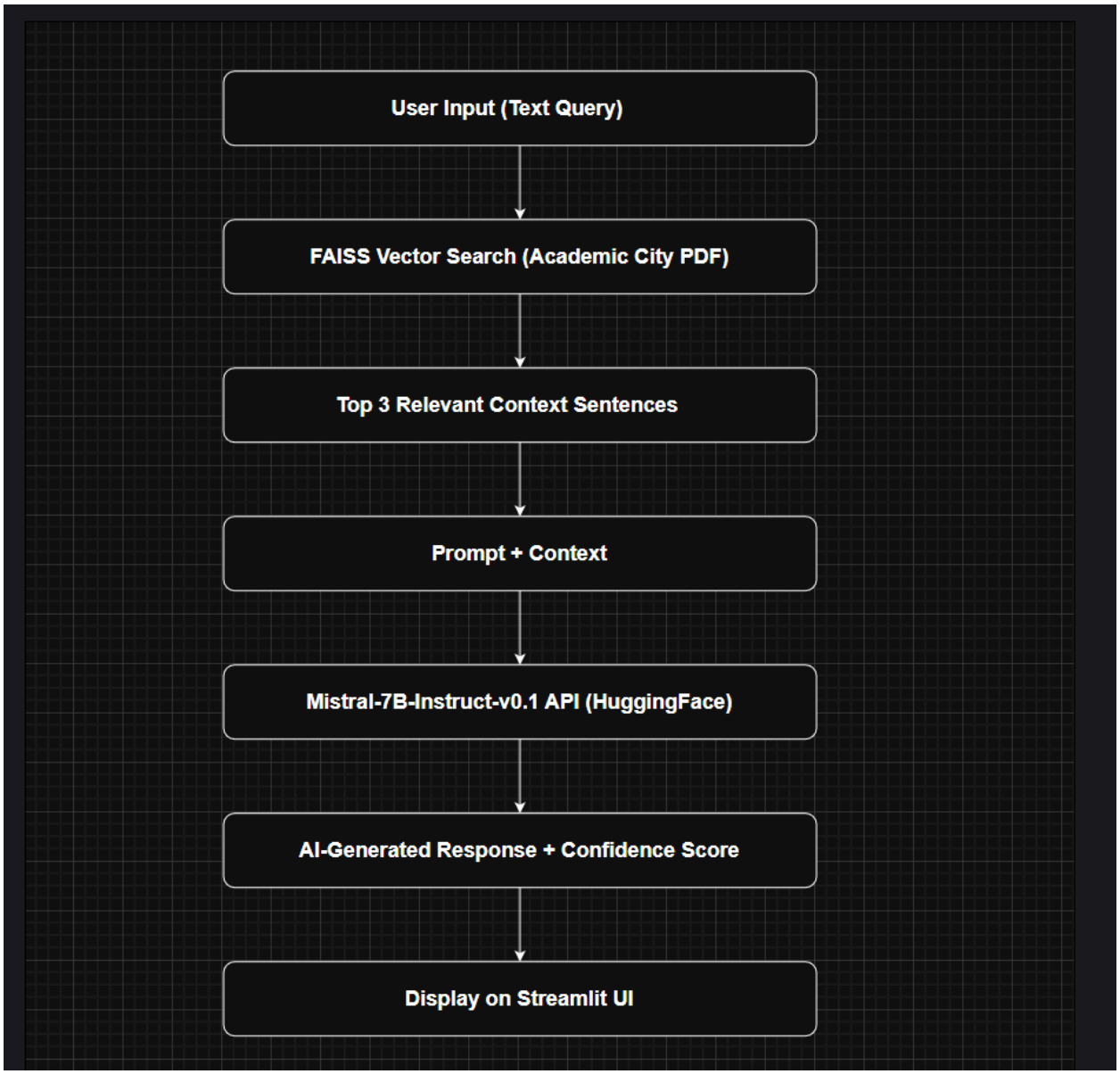
Dataset

- **Academic City Student Policy (PDF)** for LLM RAG.
- User-uploaded CSV files for Regression, Clustering, and Neural Networks.

Models

- **Linear Regression** (from scikit-learn)
- **KMeans Clustering** (from scikit-learn)
- **Feedforward Neural Network** (Keras Sequential API)
- **Mistralai/Mistral-7B-Instruct-v0.1** via HuggingFace API for LLM RAG, integrated with a FAISS semantic search index.

ARCHITECTURE FOR THE LLM RAG SYSTEM



- **Novelty:** Combined document retrieval via FAISS and open LLM API for accurate, relevant Q&A, with confidence scoring derived from semantic similarity.

METHODOLOGY

Document Preprocessing

The document is cleaned and extracted from the Academic City Student Policy PDF by removing numbers, unnecessary spaces, and dividing the content into individual sentences.

Vectorization & Indexing

Sentences are converted to TF-IDF vectors, and a FAISS index is created for efficient similarity searching.

Query Handling

User inputs are also transformed to vectors, and the FAISS index is probed to yield the top three closest sentences. Similarity-based confidence scores are then computed.

LLM Inference

The user input and the recalled context are compounded together to produce a prompt and subsequently passed through the HuggingFace API for inference to the Mistral-7B-Instruct-v0.1 model.

Response Display

The response, along with the confidence, is provided in real-time, and the question history and answer history are recorded

EVALUATION AND COMPARISON

LLM RAG Results

The LLM RAG system gives context-specific, accurate responses based on the actual policy document, and the confidence measure reflects the relevance of the context, computed as a function of distance-based similarity.

Comparison with ChatGPT

The standard ChatGPT model generates generic answers and doesn't have any specific policy references unless added manually. LLM RAG, on the other hand, generates document-specific answers based on relevant passages and indicates higher accuracy in policy-specific Q&A compared to ChatGPT.

EXAMPLE

Query	ChatGPT Response	LLM RAG Response
“What is the dress code?”	“It typically includes modest attire.”	“The policy specifies students must wear formal attire on designated days.”
“Who to contact for absences?”	“Generally your lecture or head”	“The policy states contact the Registrar’s office via email at acity.edu.gh.”

LLM RAG clearly outperforms ChatGPT for document-grounded queries.