LAB2:PART1

SAANVI VENKATESH KULKARNI,1RVU23CSE391

a)LANGCHAIN

CODE:

```
!pip install -q sentence-transformers faiss-cpu pypdf transformers

from google.colab import files
files.upload()

import os
from pypdf import PdfReader
from sentence_transformers import SentenceTransformer
import faiss
import numpy as np
from transformers import pipeline

# Read PDF text
text = ""
for f in os.listdir():
    if f.endswith(".pdf"):
        reader = PdfReader(f)
        for page in reader.pages:
            text += page.extract_text() + "\n"

# Split into chunks
chunks = [text[i:i+500] for i in range(0, len(text), 500)]

# Embeddings
model = SentenceTransformer("all-MiniLM-L6-v2")
embeddings = model.encode(chunks)

# FAISS index
index = faiss.IndexFlatL2(embeddings.shape[1])
index.add(np.array(embeddings))

# LLM
qa_pipeline = pipeline("text-generation", model="google/flan-t5-base")

def ask(question):
    q_emb = model.encode([question])
    D, I = index.search(np.array(q_emb), k=3)
    context = " ".join([chunks[i] for i in I[0]])

    prompt = f"Answer the question using this
context:\n{context}\n\nQuestion: {question}"
    answer = qa_pipeline(prompt, max_length=200)[0]["generated_text"]
```

```
    print("\nANSWER:\n", answer)

ask("Summarize the document")
```

OUTPUT

```
ANSWER:
 Answer the question using this context:
like a helpful assistant that can read, explain, and summarize information across multiple formats.

 textbook figure, label parts of a system, or create a short study
note from a screenshot.

User support and troubleshooting: interpret screenshots of errors and guide a fix step-by-step.

Basic reasoning using visual information: compare objects, count simple items, and follow
relationships shown in the image (with some limits).
These models are strongest when the question is clear and the visual information is readable (high
resolution, good lighting, and not too cluttered).
6. Limitations (w (similar to reading a printed page).

Combine text + image information to produce one final answer (for example: read a table in a PDF
and summarize it).

In some systems: understand audio (speech) and video (a sequence of images).
These abilities help when the answer depends on something you can only learn by looking at the
content, such as a figure in a report or a chart in a slide.
3. How multimodal LLMs are built
Most multimodal LLMs are not trained from scratch as one giant model. Instead
```

b)KAGGLE

CODE:

```
!pip install -q kaggle

import kagglehub

# Download dataset
path = kagglehub.dataset_download("himanshunakrani/iris-dataset")

print("Path to dataset files:", path)

import pandas as pd
import os

# Check files inside folder
print(os.listdir(path))

# Load CSV
df = pd.read_csv(os.path.join(path, "iris.csv"))

df.head()
```

```
print("Dataset shape:", df.shape)
print("\nColumns:", df.columns)

print("\nInfo:")
df.info()

print("\nStatistics:")
df.describe()

df["species"].value_counts()

import matplotlib.pyplot as plt

plt.figure()
df["sepal_length"].hist()
plt.title("Sepal Length Distribution")
plt.xlabel("Sepal Length")
plt.ylabel("Count")
plt.show()
```

OUTPUT

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
Dataset shape: (150, 5)

Columns: Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')

Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB

Statistics:
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000

species	count
setosa	50
versicolor	50
virginica	50

dtype: int64