The code represents component level evaluation of RAG pipeline using RAGAS

1) Install Dependencies

```
!pip install langchain langchain-community langchain-huggingface langchain_groq chromadb pdfminer.six sentence-transformers tra
!pip install --upgrade ragas
```

Show hidden output

2) Import necessary packages

```
import os
from langchain_community.document_loaders import PDFMinerLoader
from langchain_text_splitters import RecursiveCharacterTextSplitter
from langchain_community.vectorstores import Chroma
from langchain_huggingface import HuggingFaceEmbeddings
from langchain_groq import ChatGroq
from langchain_core.messages import HumanMessage
from langchain_core.prompts import PromptTemplate
from langchain_community.llms import HuggingFacePipeline
from langchain_core.runnables import RunnableLambda, RunnablePassthrough
from langchain_core.output_parsers import StrOutputParser
from datasets import Dataset
from ragas import evaluate
from ragas.metrics import (faithfulness, context_recall, context_precision, answer_correctness)
from ragas.run_config import RunConfig
```

Show hidden output

3) Data Ingestion

```
Loader = PDFMinerLoader('/content/healthy_heart_disease_prevention.pdf')
doc = Loader.load()
```

4) Chunking

```
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size = 800,
    chunk_overlap = 100)
chunks = text_splitter.split_documents(doc)
```

5) View sample chunks

```
print(chunks[0])
print(chunks[1])
```

```
page_content='Healthy Eating for Heart Disease Prevention

A well-balanced diet is one of the most effective tools for preventing heart disease and promoting overall

cardiovascular health. The right dietary choices can help manage blood pressure, cholesterol, blood sugar

levels, and weight—all key factors in reducing the risk of heart disease. Below is a more detailed guide to

help you make heart-healthy food choices.

1. Focus on Heart-Healthy Fats

Fat is an essential nutrient that the body needs for energy, cell growth, and absorption of fat-soluble

vitamins. However, not all fats are the same. Healthy fats can help protect your heart, while unhealthy fats

increase the risk of heart disease.

- Monounsaturated Fats:' metadata={'producer': 'pdfmake', 'creator': 'pdfmake', 'creationdate': '2026-02-13T05:35:52+00:00', 'to
page_content='increase the risk of heart disease.
```

```
- Monounsaturated Fats:

  These fats help lower LDL (bad) cholesterol levels and raise HDL (good) cholesterol levels.

  - Sources: Olive oil, canola oil, peanut butter, avocados, almonds, and walnuts.

  - Health Benefits: Regular intake of monounsaturated fats can lower the risk of heart disease and stroke,

and also support weight loss when consumed in moderation.

- Polyunsaturated Fats:

  These fats can help reduce LDL cholesterol levels and decrease the risk of heart disease.

  - Sources: Fatty fish (salmon, mackerel, sardines, trout), flaxseeds, chia seeds, walnuts, sunflower seeds,

and soybean oil.

  - Health Benefits: Omega-3 fatty acids, a type of polyunsaturated fat found in fatty fish, help reduce' metadata={'producer':
```

6) View no of chunks

```
print(len(chunks))
```

```
13
```

7) Create embeddings

```
embeddings = HuggingFaceEmbeddings(model_name='sentence-transformers/all-MiniLM-L6-v2')
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
modules.json: 100% 349/349 [00:00<00:00, 40.8kB/s]
config_sentence_transformers.json: 100% 116/116 [00:00<00:00, 8.46kB/s]
README.md: 10.5k/? [00:00<00:00, 1.08MB/s]
sentence_bert_config.json: 100% 53.0/53.0 [00:00<00:00, 6.14kB/s]
config.json: 100% 612/612 [00:00<00:00, 73.4kB/s]
model.safetensors: 100% 90.9M/90.9M [00:00<00:00, 142MB/s]
tokenizer_config.json: 100% 350/350 [00:00<00:00, 37.9kB/s]
vocab.txt: 232k/? [00:00<00:00, 15.5MB/s]
tokenizer.json: 466k/? [00:00<00:00, 29.8MB/s]
special_tokens_map.json: 100% 112/112 [00:00<00:00, 12.8kB/s]
config.json: 100% 190/190 [00:00<00:00, 20.2kB/s]
```

8) Storing embeddings in ChromaDB

```
vectorstore = Chroma.from_documents(
    documents=chunks,
    embedding=embeddings)
```

9) Collection count in ChromaDB

```
print(vectorstore._collection.count())
```

```
13
```

10) Inspect vector collection

```
print(vectorstore._collection.get())
```

```
{'ids': ['adda324d-41b7-4db1-8e6d-de6f0bafe13f', 'fba6170a-350c-46a4-b1f6-d687f22cf7ee', 'eab0f7ce-d5f2-4ff3-a042-7ed974b223c3',
```

11) Set Retriever for semantic retrieval of document chunks

```
retriever = vectorstore.as_retriever(search_kwargs={"k":3})
```

12) Initialise Groq API to access Llama model for text generation

```
from getpass import getpass
import os

api_key = getpass("Enter your Groq API key: ")
os.environ["GROQ_API_KEY"] = api_key
```

```
Enter your Groq API key: ·····
```

```
chat_model = ChatGroq(
    model_name="llama-3.1-8b-instant",
    temperature=0
)
```

13) Generate custom template

```
template = """
You are an assistant for question-answering tasks. Use the following pieces of retrieved context to answer the question asked on
Keep the answer concise and If you don't know the answer, just say that you don't know.
DO NOT provide additional commentary or questions.
Context:
{context}

Question: {question}

Answer:"""

prompt = PromptTemplate(template=template, input_variables=["context", "question"])

output_parser = StrOutputParser()
```

14) Set RAG pipeline

```
def rag_with_context(query):
    retrieved_docs = retriever.invoke(query)
    context_texts = [doc.page_content for doc in retrieved_docs]

    formatted_context = "\n\n".join(context_texts)

    answer = (
        prompt
        | chat_model
        | output_parser
    ).invoke({
        "context": formatted_context,
        "question": query
    })

    return answer, context_texts
```

15) Prepare questions and ground truths

```
questions = [
    "What types of fats are considered heart healthy, and which fats should be avoided?",
    "How do whole grains help in preventing heart disease?",
    "Why is reducing sodium intake important for heart health?",
    "What are the benefits of including fruits and vegetables in a heart healthy diet?",
    "How does maintaining a healthy weight help prevent heart disease?"
]

ground_truths = [
    "Monounsaturated fats and polyunsaturated fats are heart healthy. Monounsaturated fats lower LDL cholesterol and raise HDL cl
```

```
    "Whole grains are rich in fiber, which lowers cholesterol, regulates blood sugar, and improves heart health. They reduce the
    "Excess sodium contributes to high blood pressure, increasing the risk of heart disease. Reducing sodium lowers blood pressu
    "Fruits and vegetables provide vitamins, minerals, antioxidants, and fiber. They support heart health, lower blood pressure,
    "Maintaining a healthy weight regulates blood pressure, cholesterol, and blood sugar levels, reducing strain on the heart an
]
```

16) Collect answers and contexts

```
answers = []
contexts = []

for query in questions:
    answer, context_texts = rag_with_context(query)
    answers.append(answer)
    contexts.append(context_texts)

answers = [
    a.replace("**", "").replace("\n", " ")
    for a in answers
]
```

17) Prepare the evaluation dataset

```
from datasets import Dataset

data = {
    "question": questions,
    "answer": answers,
    "contexts": contexts,
    "ground_truth": ground_truths
}

dataset = Dataset.from_dict(data)
```

18) Set evaluation runtime configuration

```
run_config = RunConfig(
    max_workers=1,
    timeout=300
)
```

19) Define retriever evaluation function

```
def evaluate_retriever(dataset, chat_model, run_config=None):
    return evaluate(
        dataset=dataset,
        metrics=[
            context_precision,
            context_recall
        ],
        llm=chat_model,
        run_config=run_config
    )
```

20) Define generator evaluation function

```
def evaluate_generator(dataset, chat_model, embeddings, run_config=None):
    return evaluate(
        dataset=dataset,
        metrics=[
            faithfulness,
            answer_correctness
        ],
        llm=chat_model,
        embeddings=embeddings,
        run_config=run_config
    )
```

21) Evaluate retriever performance

```
retriever_result = evaluate_retriever(
    dataset,
    chat_model,
    run_config
)

retriever_df = retriever_result.to_pandas()
```

```
Evaluating:  100%                                                    10/10  [03:24<00:00,  22.87s/it]
```

22) Evaluate generator performance

```
generator_result = evaluate_generator(
    dataset,
    chat_model,
    embeddings,
    run_config
)
generator_df = generator_result.to_pandas()
```

```
Evaluating:  100%                                                    10/10  [03:03<00:00,  21.45s/it]
```

23) Retriever evaluation results

```
retriever_df
```

	user_input	retrieved_contexts	response	reference	context_precision	context_recall
0	What types of fats are considered heart health...	[Healthy Eating for Heart Disease Prevention\n...	Monounsaturated Fats are considered heart heal...	Monounsaturated fats and polyunsaturated fats ...	1.0	0.5
1	How do whole grains help in preventing heart d...	[inflammation, lower triglycerides, and improv...	Whole grains help in preventing heart disease ...	Whole grains are rich in fiber, which lowers c...	1.0	1.0
2	Why is reducing sodium intake important for he...	[Oranges, grapefruits, lemons, and limes are h...	Excessive sodium intake is a major contributor...	Excess sodium contributes to high blood pressu...	1.0	1.0
3	What are the benefits of including	[- Leafy Greens: \n\n Vegetables like spinac	These fruits and vegetables are essential	Fruits and vegetables provide vitamins,	1.0	1.0

24) Generator evaluation results

```
generator_df
```

	user_input	retrieved_contexts	response	reference	faithfulness	answer_correctness
0	What types of fats are considered heart health...	[Healthy Eating for Heart Disease Prevention\n...	Monounsaturated Fats are considered heart heal...	Monounsaturated fats and polyunsaturated fats ...	1.00	0.417280
1	How do whole grains help in preventing heart d...	[inflammation, lower triglycerides, and improv...	Whole grains help in preventing heart disease ...	Whole grains are rich in fiber, which lowers c...	1.00	0.916846
2	Why is reducing sodium intake important for he...	[Oranges, grapefruits, lemons, and limes are h...	Excessive sodium intake is a major contributor...	Excess sodium contributes to high blood pressu...	1.00	0.777657
3	What are the benefits of including fruits and	[- Leafy Greens: \n\n Vegetables like spinac	These fruits and vegetables are essential	Fruits and vegetables provide vitamins,	1.00	0.648885

25) Analysis

Retriever Performance

- Achieved perfect context precision (1.0) across all five questions, indicating that all retrieved chunks were relevant.

- Achieved context recall of 1.0 for four out of five questions, demonstrating strong coverage of required information.
- For the multi-part question on heart-healthy fats, context recall dropped to 0.5, indicating partial retrieval of necessary information.
- The retriever performs strongly for single-topic questions but shows slight limitations when handling multi-aspect queries.

Generator Performance

- Achieved high faithfulness scores (1.0 for four questions and 0.75 for one), indicating strong grounding and minimal hallucination.
- Demonstrates consistent alignment between generated answers and retrieved context.
- Answer correctness scores ranged from 0.417 to 0.917, reflecting varying degrees of semantic alignment with reference answers.
- Lower correctness scores were observed in multi-aspect question, suggesting partial incompleteness.
- High correctness (0.909–0.917) in most cases indicates strong semantic alignment with reference answers.

Further improvements

- Increase k to improve recall for multi-part questions and reduce incomplete coverage.
- Refine the prompt to enforce completeness, ensuring the generator includes all relevant aspects mentioned in the context.
- Adjust chunk size and overlap (slightly larger chunks with higher overlap) to prevent splitting related concepts across chunks.
- Expand the evaluation dataset (20–30 questions) to obtain more stable and statistically meaningful performance metrics.