Table of Contents

1. Introduction: 2

  1.1 Background: 2

  1.2 Objective: 2

2. Data Set Summary 2

3. Approach 3

  3.1 Data Processing 3

  3.2 Process 3

    3.2.a Model Selection: 3

    3.2.b Chunking: 3

    3.2.c Embedding: 4

    3.2.d.PromptTemplate: 4

    3.2.e Query Modes: 4

    3.2.f Testing: 5

4. Limitations: 6

5. Future improvements: 6

Appendix 1 7

  1. a Figures Index 7

  1. b Tables Index 7

  2. Acronyms 7

  3. Testing 7

Appendix 2 9

References 12

1. Introduction:

1.1 Background:

Fitness encompasses exercise, nutrition, wellness, and lifestyle focusing on enhancing the overall quality of life. In this fast-paced world, fitness should be a necessary aspect of life. With the rapid developments in nutrition, workouts, and gym technologies, there has been a stark increase in the number of fitness instructors. From Personal Trainers to tailored applications on smartphones the industry is growing to be more efficient and impactful.

With the increased contribution of technology in fitness as a domain, people of different spectrums of life have been able to benefit from it. Technology and data have been able to cater to the unique needs of healthy individuals and individuals with chronic conditions or disabilities, ensuring a safe and inclusive approach to fitness. With wearable devices and other applications collecting data ranging from oxygen levels to sleep cycles, advanced analytics on these data points are used to tailor fitness or nutrition recommendations according to personal goals.

1.2 Objective:

With technologies from Analytics to Applications being implemented in the Fitness Domain, the next big thing to be implemented is Machine Learning and Artificial Intelligence. This Project aims to address the same. A RAG model combines ML and generation methods to provide accurate and relevant responses to queries. The application of an RAG model in the fitness domain can prove to be highly beneficial for people who wish to start fresh or professionals who want to tailor their workouts or fitness regimens. This RAG model can be applied in various ways in the domain, majorly in providing Personalized Workout Plans, Nutrition Guidance, and Lifestyle recommendations. Providing comprehensive and personalized responses in real-time makes RAG applications in the Fitness domain a necessity.

2. Data Set Summary

The data set used is Fitness_Unformatted. A dataset obtained from Hugging Face. The data set consists of 928 rows across 2 columns. The 2 columns are “Human” and “Assistant”. The “Human” column discusses the various fitness-related queries, and the “Assistant” column consists of answers for the respective questions. Essentially, the data set reflects Questions and Answers between a user and an assistant who answers the respective questions.

Human	Consists of various queries related to Fitness
Assistant	Answers to the relevant queries from the internet

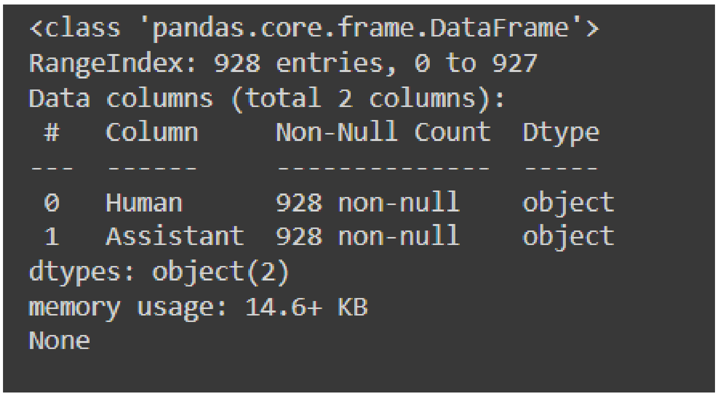
Table 1: Data Dictionary

The dataset discusses the various aspects of Fitness ranging from Exercise, Nutrition, and Well-Being. The Dataset has been structured consistently with multiple types of data, providing rich information about fitness. While focussing on the multi-faceted aspects of fitness as a whole, the data set also encompasses on technologies and Special Populations of people with chronic illness or disabilities.

3. Approach

3.1 Data Processing

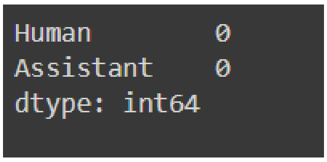
The data set “Fitness.csv” is checked for content and information, and from this, it is found that the distribution of all data points is across Human and Assistant as columns with 928 data points.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 928 entries, 0 to 927
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Human      928 non-null    object
 1   Assistant  928 non-null    object
dtypes: object(2)
memory usage: 14.6+ KB
None
```

Figure 1: Data Description

The data set is checked for NA values and consistency errors. It is found that the Human and Assistant columns are descriptive and consistent throughout the 928 data points. There are no NA or Null values present in the data set.



```
Human        0
Assistant    0
dtype: int64
```

Figure 2: NA Check

3.2 Process

3.2.a Model Selection:

Llama 3, an LLM developed by Meta is a highly functional open-source model fine-tuned and optimized in dialogue/ chat use cases. Since the Fitness dataset is similar to that of dialogues or chat use cases, Llama 3 is used as an LLM to develop the Fitness RAG model.

3.2.b Chunking:

The dataset after various checks and processing is then chunked. To implement an RAG model on the dataset, the dataset is chunked using sentence-based chunking in order to reduce its size and complexity.

Post data chunking, the parsed nodes output which is of the type TextNode has each node with a unique id and a chunk of text assigned. Although the TextNode output is crucial, for further usage in the development of the project, the output is changed and initialized as a list containing a Document object.

This conversion has been done to write the dataset into the vector database. Instead of writing the chunks into a file and then ingesting it into the vector database, the object is directly ingested.

3.2.c Embedding:

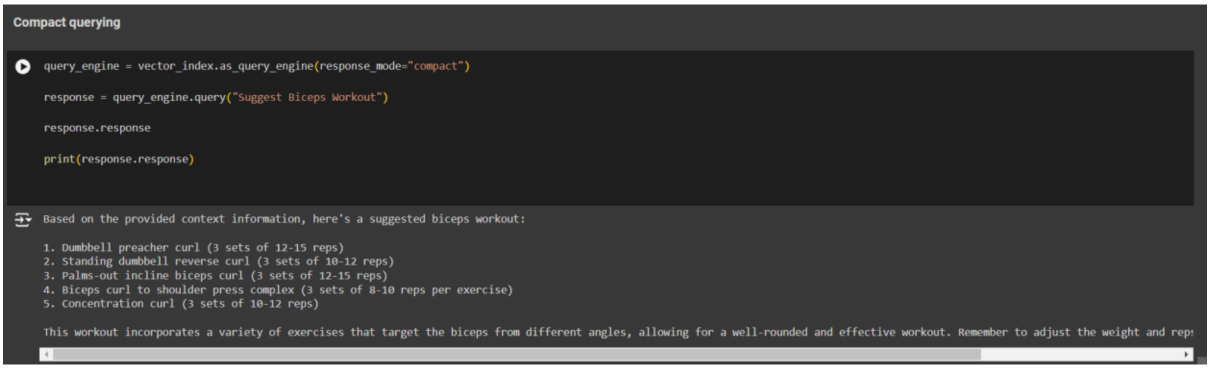
Following the chunking of the data, the chunked output is embedded using Sentence-Transformers as an embedding model. The Sentence-Transformer embedding model is a quick open-source embedding model used in the project to embed the various chunks sentence by sentence before writing them onto the vector database.

3.2.d.PromptTemplate:

A Prompt template has been set to standardize the outputs given by the RAG model. For queries that require the model to answer something out of the context of the data on which it was trained, the prompt template provides a structure to such situations. When the RAG model encounters such queries the prompt template instructs it to provide a generic answer quoting that there is no context for the query asked. The model has been tested with such queries as well.

3.2.e Query Modes:

Three different query modes namely compact, refine and tree-based modes have been tested and compared. The compact mode of querying provides a crisp and detailed answer to all queries making it the most suitable method for a Fitness-based RAG model as an application. While compact is more suitable, refine and tree-based modes also worked fine but the answers to the queries are much shorter or clear and to the point when compared to that of compact query mode.

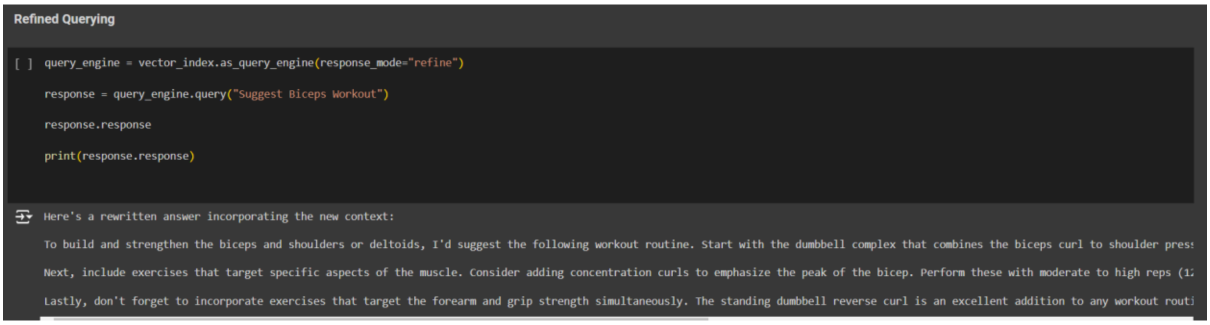


Compact querying

```
query_engine = vector_index.as_query_engine(response_mode="compact")

response = query_engine.query("Suggest Biceps Workout")

response.response

print(response.response)
```

```
Based on the provided context information, here's a suggested biceps workout:

1. Dumbbell preacher curl (3 sets of 12-15 reps)
2. Standing dumbbell reverse curl (3 sets of 10-12 reps)
3. Palms-out incline biceps curl (3 sets of 12-15 reps)
4. Biceps curl to shoulder press complex (3 sets of 8-10 reps per exercise)
5. Concentration curl (3 sets of 10-12 reps)

This workout incorporates a variety of exercises that target the biceps from different angles, allowing for a well-rounded and effective workout. Remember to adjust the weight and rep
```

Figure 3: Compact Query Mode

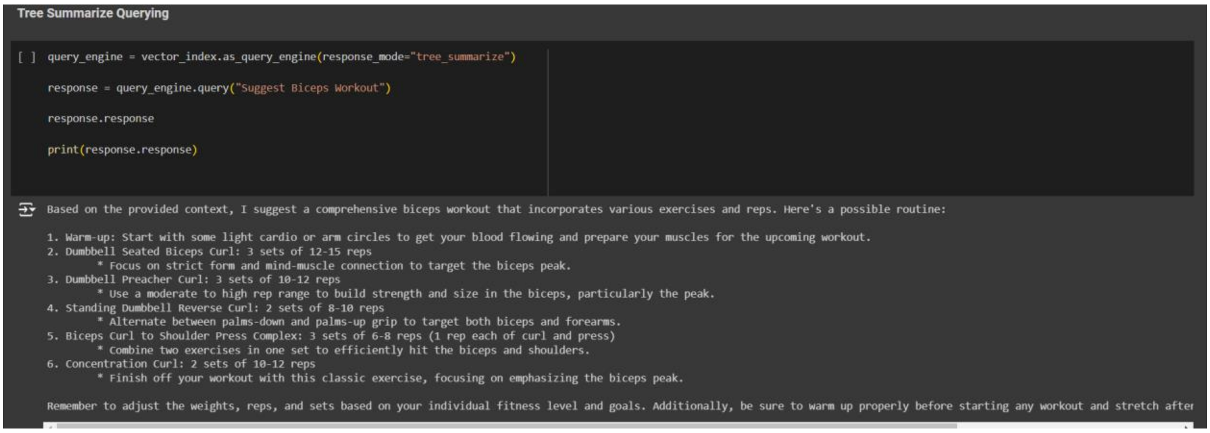


Refined Querying

```
[ ] query_engine = vector_index.as_query_engine(response_mode="refine")

    response = query_engine.query("Suggest Biceps Workout")

    response.response

    print(response.response)
```

```
Here's a rewritten answer incorporating the new context:

To build and strengthen the biceps and shoulders or deltoids, I'd suggest the following workout routine. Start with the dumbbell complex that combines the biceps curl to shoulder pres

Next, include exercises that target specific aspects of the muscle. Consider adding concentration curls to emphasize the peak of the bicep. Perform these with moderate to high reps (1

Lastly, don't forget to incorporate exercises that target the forearm and grip strength simultaneously. The standing dumbbell reverse curl is an excellent addition to any workout routi
```

Figure 4: Refined Querying Mode



```
[ ]  query_engine = vector_index.as_query_engine(response_mode="tree_summarize")

     response = query_engine.query("Suggest Biceps Workout")

     response.response

     print(response.response)
```

```
Based on the provided context, I suggest a comprehensive biceps workout that incorporates various exercises and reps. Here's a possible routine:

1. Warm-up: Start with some light cardio or arm circles to get your blood flowing and prepare your muscles for the upcoming workout.
2. Dumbbell Seated Biceps Curl: 3 sets of 12-15 reps
        * Focus on strict form and mind-muscle connection to target the biceps peak.
3. Dumbbell Preacher Curl: 3 sets of 10-12 reps
        * Use a moderate to high rep range to build strength and size in the biceps, particularly the peak.
4. Standing Dumbbell Reverse Curl: 2 sets of 8-10 reps
        * Alternate between palms-down and palms-up grip to target both biceps and forearms.
5. Biceps Curl to Shoulder Press Complex: 3 sets of 6-8 reps (1 rep each of curl and press)
        * Combine two exercises in one set to efficiently hit the biceps and shoulders.
6. Concentration Curl: 2 sets of 10-12 reps
        * Finish off your workout with this classic exercise, focusing on emphasizing the biceps peak.

Remember to adjust the weights, reps, and sets based on your individual fitness level and goals. Additionally, be sure to warm up properly before starting any workout and stretch after
```

Figure 5: Tree Summarize Querying

3.2.f Testing:

Testing of the RAG model has been done to ensure that the applicability of the model is unique and personalized, abiding by the objective of the project. The model was tested based on six test cases namely:

1. Workout routine for a specific goal and gender
2. Meal plan for a specific dietary requirement and goal
3. Overall suggestion for a person with chronic illness
4. Well-being and nutrition for specific conditions
5. Non-Context queries

Based on the results of the above-mentioned test cases (Refer to Appendix 1), the model has been seen to perform optimally, addressing each query in a personalized manner ensuring that the model meets the objective and expands the horizon of applicability.

4. Limitations:

Some of the major limitations of the project are namely:

1. Credibility of data
2. Evaluation challenges
3. Scalability

The source of how the data was collected or how credible the data is not known for the dataset used. The dataset descriptions in HuggingFace did not have any details regarding the same. Although the RAG model is completely functional, the evaluation's carried out are manual. There is no specific metric to measure the efficiency of its functioning.

Currently, the model is optimized to work on exercises, nutrition, and overall well-being but when it comes to scalability on a global level, a much more extensive dataset is required incorporating exercises and foods of various countries.

5. Future improvements:

The major improvements or advances that can be considered to improve the overall functionality of the RAG model are improved integration and user interaction, robustness, and integration across multiple languages.

The model can be integrated across multiple data sets like recipes or exercises which will increase the horizon of its usage. A chat-like user interface can be implemented to develop the model as a potential application. The model must be trained and tested across various datasets and queried to compare the efficiency. The application can be globalized by integrating across multiple languages.

Appendix 1

1. a Figures Index

Figure 1: Data Description 3
Figure 2: NA Check 3
Figure 3: Compact Query Mode 4
Figure 4: Refined Querying Mode 4
Figure 5: Tree Summarize Querying 5

1. b Tables Index

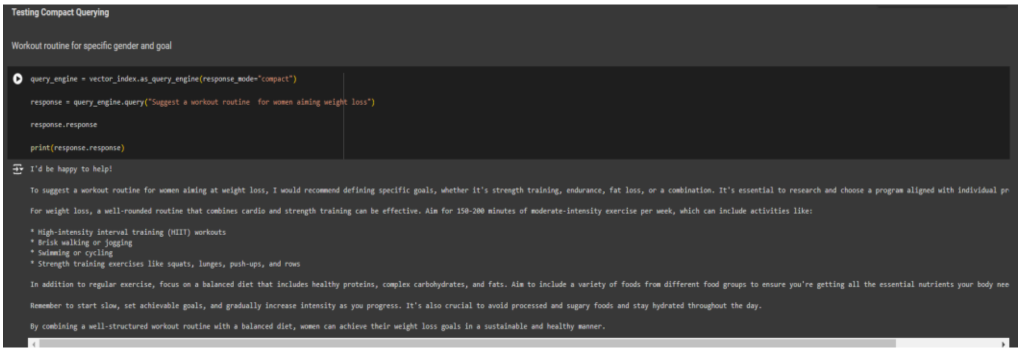
Table 1: Data Dictionary 2

2. Acronyms

LLM - Large Language Model

RAG – Retrieval Augmented Generation

3. Testing



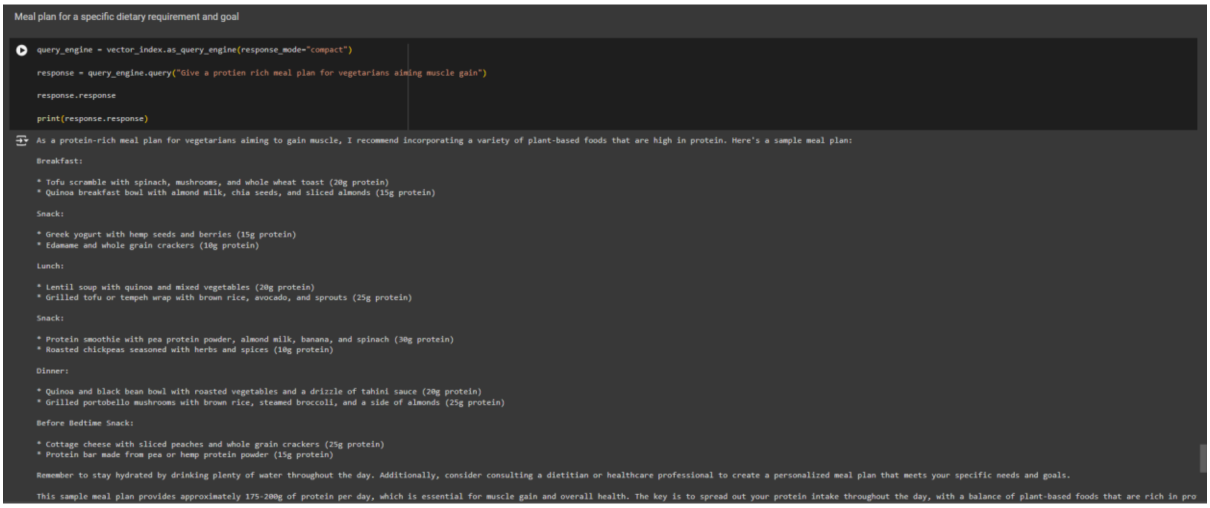
Testing Compact Querying

Workout routine for specific gender and goal

```
query_engine = vector_index.as_query_engine(response_mode="compact")

response = query_engine.query("Suggest a workout routine  for women aiming weight loss")

response.response

print(response.response)
```

```
I'd be happy to help!

To suggest a workout routine for women aiming at weight loss, I would recommend defining specific goals, whether it's strength training, endurance, fat loss, or a combination. It's essential to research and choose a program aligned with individual pr

For weight loss, a well-rounded routine that combines cardio and strength training can be effective. Aim for 150-200 minutes of moderate-intensity exercise per week, which can include activities like:

* High-intensity interval training (HIIT) workouts
* Brisk walking or jogging
* Swimming or cycling
* Strength training exercises like squats, lunges, push-ups, and rows

In addition to regular exercise, focus on a balanced diet that includes healthy proteins, complex carbohydrates, and fats. Aim to include a variety of foods from different food groups to ensure you're getting all the essential nutrients your body nee

Remember to start slow, set achievable goals, and gradually increase intensity as you progress. It's also crucial to avoid processed and sugary foods and stay hydrated throughout the day.

By combining a well-structured workout routine with a balanced diet, women can achieve their weight loss goals in a sustainable and healthy manner.
```

Gender and goal-specific testing

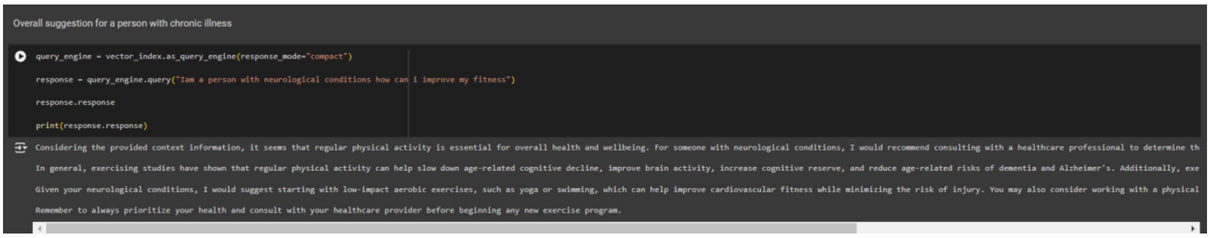


Meal plan for a specific dietary requirement and goal

```
query_engine = vector_index.as_query_engine(response_mode="compact")

response = query_engine.query("Give a protien rich meal plan for vegetarians aiming muscle gain")

response.response

print(response.response)
```

```
As a protein-rich meal plan for vegetarians aiming to gain muscle, I recommend incorporating a variety of plant-based foods that are high in protein. Here's a sample meal plan:

Breakfast:

* Tofu scramble with spinach, mushrooms, and whole wheat toast (20g protein)
* Quinoa breakfast bowl with almond milk, chia seeds, and sliced almonds (15g protein)

Snack:

* Greek yogurt with hemp seeds and berries (15g protein)
* Edamame and whole grain crackers (10g protein)

Lunch:

* Lentil soup with quinoa and mixed vegetables (20g protein)
* Grilled tofu or tempeh wrap with brown rice, avocado, and sprouts (25g protein)

Snack:

* Protein smoothie with pea protein powder, almond milk, banana, and spinach (30g protein)
* Roasted chickpeas seasoned with herbs and spices (10g protein)

Dinner:

* Quinoa and black bean bowl with roasted vegetables and a drizzle of tahini sauce (20g protein)
* Grilled portobello mushrooms with brown rice, steamed broccoli, and a side of almonds (25g protein)

Before Bedtime Snack:

* Cottage cheese with sliced peaches and whole grain crackers (25g protein)
* Protein bar made from pea or hemp protein powder (15g protein)

Remember to stay hydrated by drinking plenty of water throughout the day. Additionally, consider consulting a dietitian or healthcare professional to create a personalized meal plan that meets your specific needs and goals.

This sample meal plan provides approximately 175-200g of protein per day, which is essential for muscle gain and overall health. The key is to spread out your protein intake throughout the day, with a balance of plant-based foods that are rich in pro
```

Dietary and goal-specific meal plan

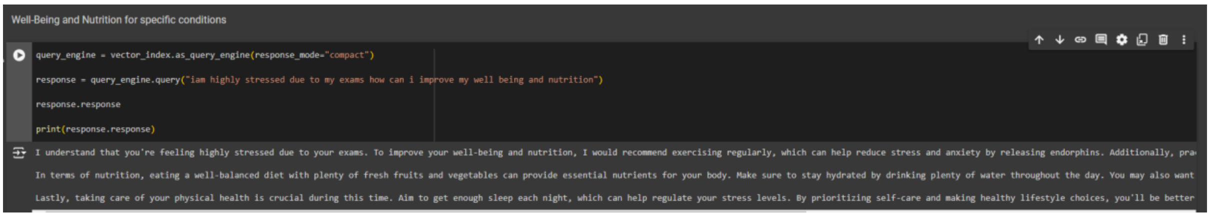


Overall suggestion for a person with chronic illness

```
query_engine = vector_index.as_query_engine(response_mode="compact")
response = query_engine.query("Iam a person with neurological conditions how can i improve my fitness")
response.response
print(response.response)
```

```
Considering the provided context information, it seems that regular physical activity is essential for overall health and wellbeing. For someone with neurological conditions, I would recommend consulting with a healthcare professional to determine th
In general, exercising studies have shown that regular physical activity can help slow down age-related cognitive decline, improve brain activity, increase cognitive reserve, and reduce age-related risks of dementia and Alzheimer's. Additionally, exe
Given your neurological conditions, I would suggest starting with low-impact aerobic exercises, such as yoga or swimming, which can help improve cardiovascular fitness while minimizing the risk of injury. You may also consider working with a physical
Remember to always prioritize your health and consult with your healthcare provider before beginning any new exercise program.
```

Chronic illness fitness suggestion

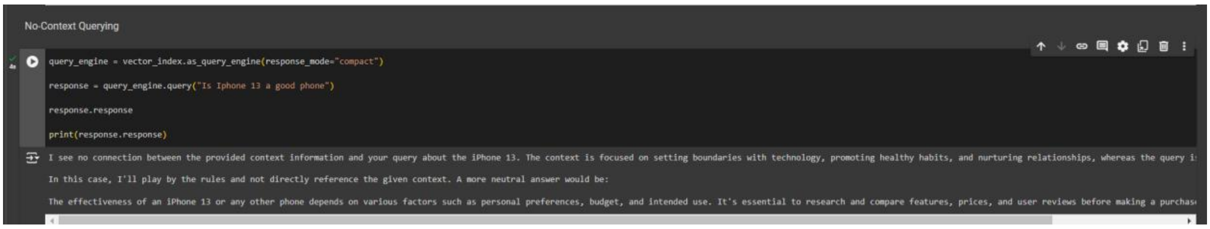


Well-Being and Nutrition for specific conditions

```
query_engine = vector_index.as_query_engine(response_mode="compact")
response = query_engine.query("iam highly stressed due to my exams how can i improve my well being and nutrition")
response.response
print(response.response)
```

```
I understand that you're feeling highly stressed due to your exams. To improve your well-being and nutrition, I would recommend exercising regularly, which can help reduce stress and anxiety by releasing endorphins. Additionally, pra
In terms of nutrition, eating a well-balanced diet with plenty of fresh fruits and vegetables can provide essential nutrients for your body. Make sure to stay hydrated by drinking plenty of water throughout the day. You may also want
Lastly, taking care of your physical health is crucial during this time. Aim to get enough sleep each night, which can help regulate your stress levels. By prioritizing self-care and making healthy lifestyle choices, you'll be better
```

Personalized Well-Being and Nutrition



No-Context Querying

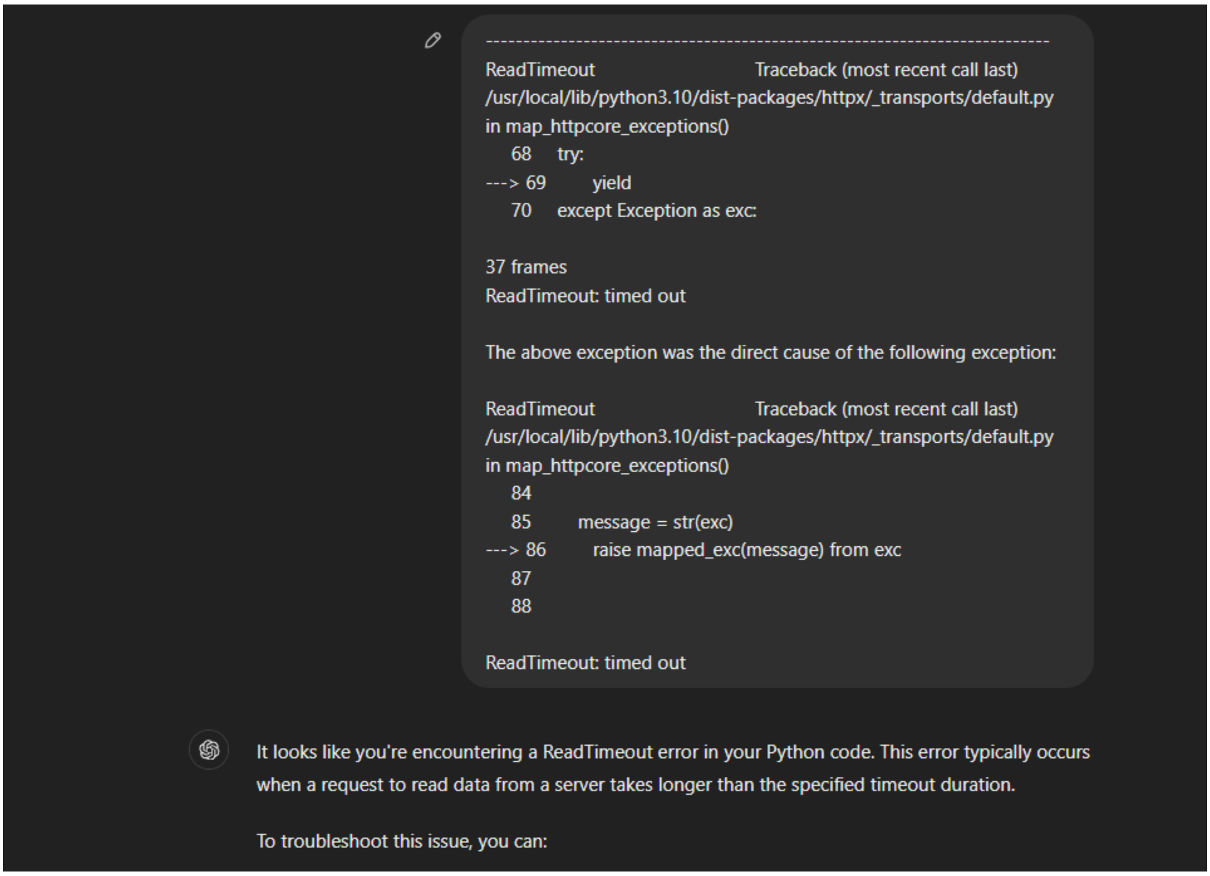
```
query_engine = vector_index.as_query_engine(response_mode="compact")
response = query_engine.query("Is Iphone 13 a good phone")
response.response
print(response.response)
```

```
I see no connection between the provided context information and your query about the iPhone 13. The context is focused on setting boundaries with technology, promoting healthy habits, and nurturing relationships, whereas the query i
In this case, I'll play by the rules and not directly reference the given context. A more neutral answer would be:
The effectiveness of an iPhone 13 or any other phone depends on various factors such as personal preferences, budget, and intended use. It's essential to research and compare features, prices, and user reviews before making a purchas
```

No-Context Querying

Appendix 2

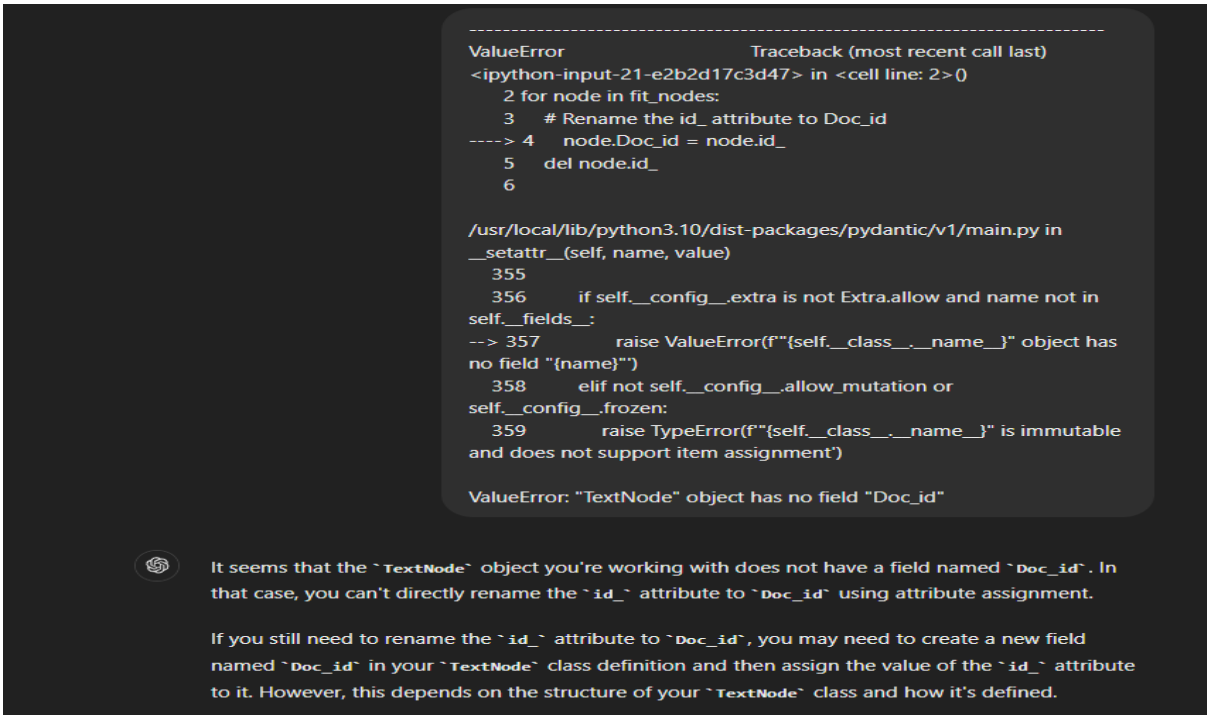
Error Handling



```
---------------------------------------------------------------------------
ReadTimeout                               Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/httpx/_transports/default.py
in map_httpcore_exceptions()
     68     try:
---> 69         yield
     70     except Exception as exc:

37 frames
ReadTimeout: timed out

The above exception was the direct cause of the following exception:

ReadTimeout                               Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/httpx/_transports/default.py
in map_httpcore_exceptions()
     84
     85         message = str(exc)
---> 86         raise mapped_exc(message) from exc
     87
     88

ReadTimeout: timed out
```

It looks like you're encountering a ReadTimeout error in your Python code. This error typically occurs when a request to read data from a server takes longer than the specified timeout duration.

To troubleshoot this issue, you can:

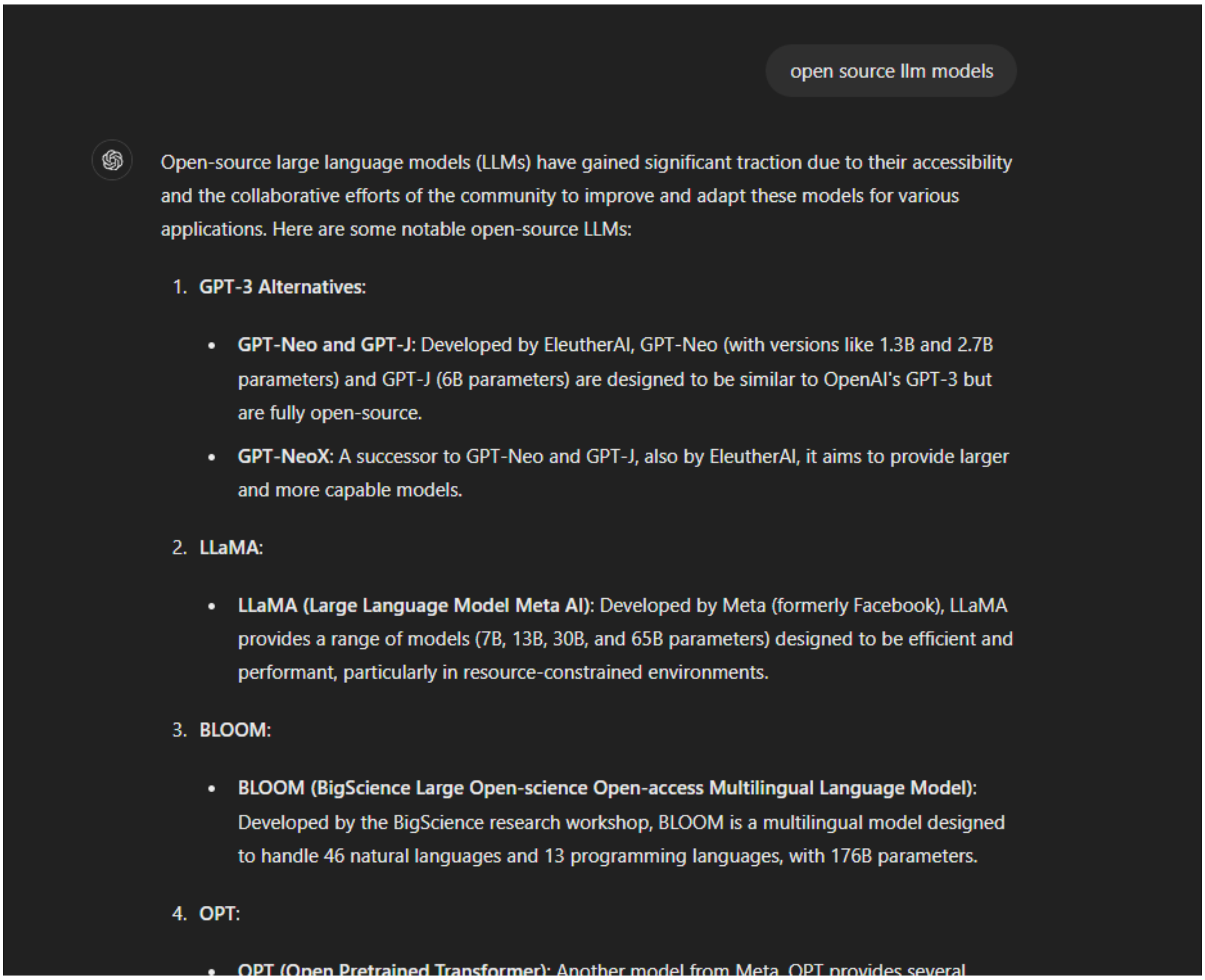


```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-21-e2b2d17c3d47> in <cell line: 2>()
      2 for node in fit_nodes:
      3     # Rename the id_ attribute to Doc_id
----> 4     node.Doc_id = node.id_
      5     del node.id_
      6

/usr/local/lib/python3.10/dist-packages/pydantic/v1/main.py in
__setattr__(self, name, value)
    355
    356         if self.__config__.extra is not Extra.allow and name not in
self.__fields__:
--> 357             raise ValueError(f'"{self.__class__.__name__}" object has
no field "{name}"')
    358         elif not self.__config__.allow_mutation or
self.__config__.frozen:
    359             raise TypeError(f'"{self.__class__.__name__}" is immutable
and does not support item assignment')

ValueError: "TextNode" object has no field "Doc_id"
```

It seems that the `TextNode` object you're working with does not have a field named `Doc_id`. In that case, you can't directly rename the `id_` attribute to `Doc_id` using attribute assignment.

If you still need to rename the `id_` attribute to `Doc_id`, you may need to create a new field named `Doc_id` in your `TextNode` class definition and then assign the value of the `id_` attribute to it. However, this depends on the structure of your `TextNode` class and how it's defined.

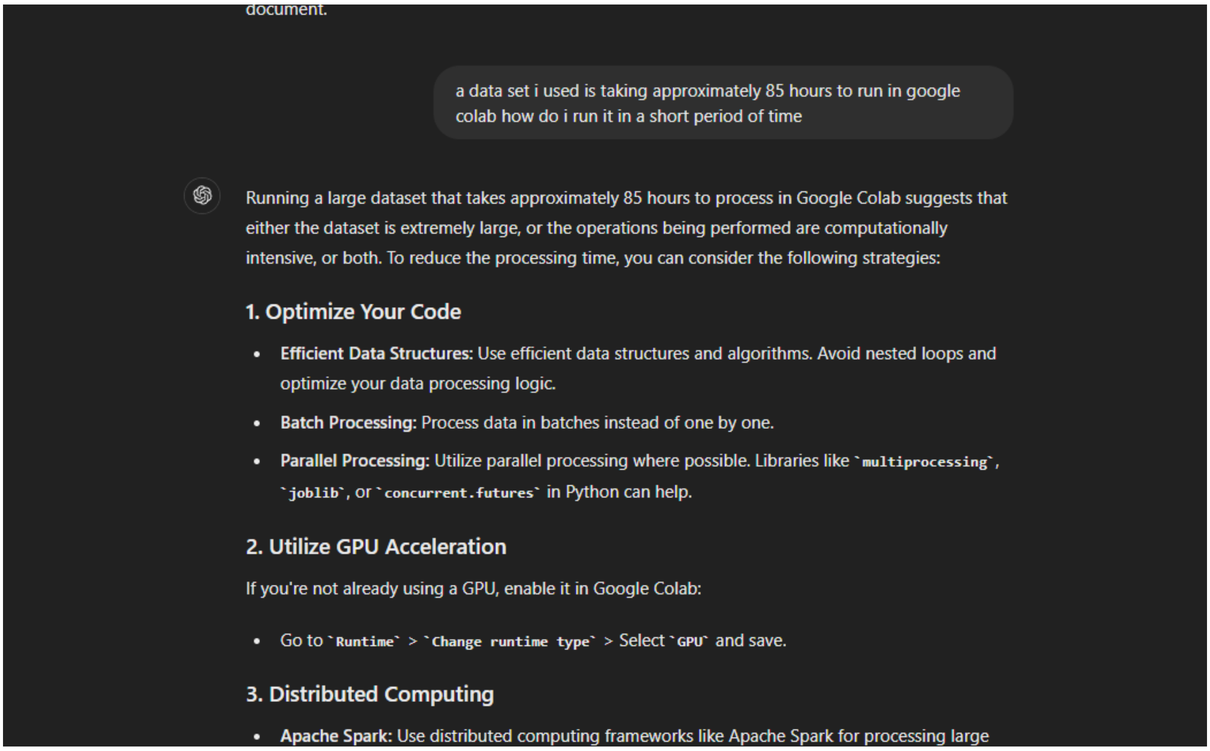
Research:



open source llm models

Open-source large language models (LLMs) have gained significant traction due to their accessibility and the collaborative efforts of the community to improve and adapt these models for various applications. Here are some notable open-source LLMs:

1. **GPT-3 Alternatives:**
   - **GPT-Neo and GPT-J:** Developed by EleutherAI, GPT-Neo (with versions like 1.3B and 2.7B parameters) and GPT-J (6B parameters) are designed to be similar to OpenAI's GPT-3 but are fully open-source.
   - **GPT-NeoX:** A successor to GPT-Neo and GPT-J, also by EleutherAI, it aims to provide larger and more capable models.
2. **LLaMA:**
   - **LLaMA (Large Language Model Meta AI):** Developed by Meta (formerly Facebook), LLaMA provides a range of models (7B, 13B, 30B, and 65B parameters) designed to be efficient and performant, particularly in resource-constrained environments.
3. **BLOOM:**
   - **BLOOM (BigScience Large Open-science Open-access Multilingual Language Model):** Developed by the BigScience research workshop, BLOOM is a multilingual model designed to handle 46 natural languages and 13 programming languages, with 176B parameters.
4. **OPT:**
   - **OPT (Open Pretrained Transformer):** Another model from Meta, OPT provides several



document.

a data set i used is taking approximately 85 hours to run in google colab how do i run it in a short period of time

Running a large dataset that takes approximately 85 hours to process in Google Colab suggests that either the dataset is extremely large, or the operations being performed are computationally intensive, or both. To reduce the processing time, you can consider the following strategies:

1. Optimize Your Code

- **Efficient Data Structures:** Use efficient data structures and algorithms. Avoid nested loops and optimize your data processing logic.
- **Batch Processing:** Process data in batches instead of one by one.
- **Parallel Processing:** Utilize parallel processing where possible. Libraries like `multiprocessing`, `joblib`, or `concurrent.futures` in Python can help.

2. Utilize GPU Acceleration

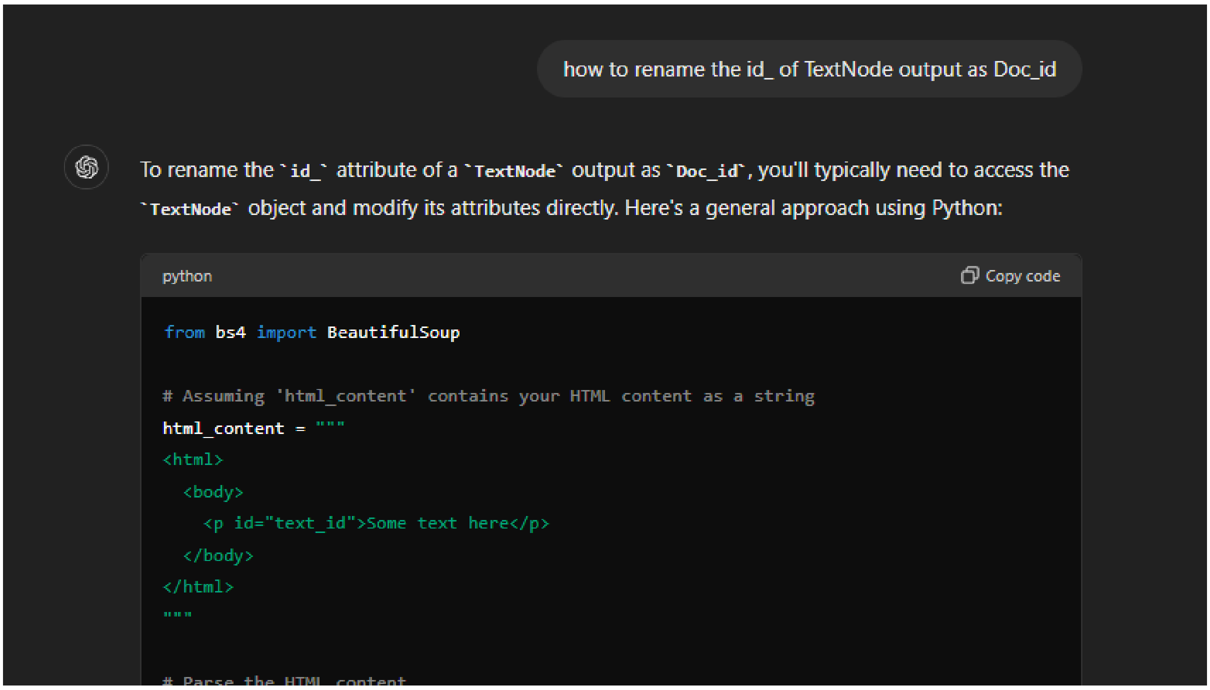
If you're not already using a GPU, enable it in Google Colab:

- Go to `Runtime` > `Change runtime type` > Select `GPU` and save.

3. Distributed Computing

- **Apache Spark:** Use distributed computing frameworks like Apache Spark for processing large

Code suggestions:



how to rename the id_ of TextNode output as Doc_id

To rename the `id_` attribute of a `TextNode` output as `Doc_id`, you'll typically need to access the `TextNode` object and modify its attributes directly. Here's a general approach using Python:

```python
from bs4 import BeautifulSoup

# Assuming 'html_content' contains your HTML content as a string
html_content = """
<html>
  <body>
    <p id="text_id">Some text here</p>
  </body>
</html>
"""

# Parse the HTML content
```

References

HazSylvia. (n.d.). *Fitness_Unformatted*. Retrieved from Hugging Face: https://huggingface.co/datasets/HazSylvia/Fitness_Unformatted

meta. (n.d.). *llama3*. Retrieved from ollama: https://ollama.com/library/llama3

nreimers. (n.d.). *Sentence Transformers*. Retrieved from Hugging Face: https://huggingface.co/sentence-transformers