

## ✓ Making Movie Mania

### ✓ Overview

Building a Movie Making Chatbot Assistance System with RAG, LangChain, LLM, and Vector Database

## Introduction

The goal of this project is to develop a movie recommendation system that leverages advanced machine learning techniques to provide personalized recommendations based on user inputs. The system integrates Retrieval-Augmented Generation (RAG) using a Large Language Model (LLM), LangChain, and a vector database to enhance the recommendation process.

## Components and Technologies

Large Language Model (LLM): Used for generating natural language responses and enhancing the recommendation process. LangChain: Manages the flow of interactions between the user, LLM, and vector database. Vector Database: Stores embeddings of movie data for efficient retrieval based on semantic similarity. Examples include Pinecone, Weaviate, or FAISS. Streamlit: Provides an interactive web interface for users to input queries and receive recommendations.

## Workflow

### Data Collection and Preprocessing

Collect movie data including titles, genres, ratings, and descriptions. Clean and preprocess the data to ensure it is in a suitable format for generating embeddings. Embedding Generation

### Pre-trained Embedding Model

Use a pre-trained embedding model to generate embeddings for movie titles and descriptions. Store these embeddings in a vector database for efficient retrieval. Vector Database Integration

### Vector Database

Initialize and configure the vector database. Index the movie embeddings for fast similarity searches. LangChain Integration

### LangChain

Set up LangChain to manage the interaction flow between the user inputs, LLM responses, and vector database retrievals. Define the prompts and response generation logic. Retrieval-Augmented Generation (RAG) Workflow

## LLM

User inputs a query through the Streamlit interface. LangChain processes the query and retrieves relevant movie embeddings from the vector database. The LLM uses the retrieved embeddings to generate a natural language response recommending movies.

## Streamlit Interface

Develop a user-friendly web interface for inputting queries and displaying recommendations. Implement input fields and submit buttons for user interaction.

## Objectives

This notebook provides a guide to building a Adaptive Recommendation Chatbot using multimodal retrieval augmented generation (RAG) and Vector Database.

The tasks that this notebook would perform:

1. Extract data from documents containing both text and images using Gemini Vision Pro, and generate embeddings of the data, store it in vector store
2. Search the vector store with text queries to find similar text data
3. Using Text data as context, generate answer to the user query using Gemini Pro Model.

### ✓ Begin with Vertex AI SDK Setup

### ✓ Setting Up Vertex AI SDK and Essential Packages

```
!pip install --upgrade --quiet pymupdf langchain gradio google-cloud-aiplatform lan
```

### ✓ Restart runtime

To use the newly installed packages in this Jupyter runtime, you must restart the runtime. You can do this by running the cell below, which restarts the current kernel.

The restart might take a minute or longer. After its restarted, continue to the next step.

```
# import IPython

# app = IPython.Application.instance()
# app.kernel.do_shutdown(True)

↻ {'status': 'ok', 'restart': True}
```

⚠ Wait for the kernel to finish restarting before you continue. ⚠

## ✓ Authenticate your notebook environment (Colab only)

If you are running this notebook on Google Colab, run the cell below to authenticate your environment.

This step is not required if you are using [Vertex AI Workbench](#).

```
import sys

# Additional authentication is required for Google Colab
if "google.colab" in sys.modules:
    # Authenticate user to Google Cloud
    from google.colab import auth

    auth.authenticate_user()
```

## ✓ Define Google Cloud project information and initialize Vertex AI

To get started using Vertex AI, you must have an existing Google Cloud project and [enable the Vertex AI API](#).


Learn more about [setting up a project and a development environment](#).


```
# Define project information
PROJECT_ID = "projectllm-430702" # @para
LOCATION = "us-central1" # @param {type:

# Initialize Vertex AI
import vertexai

vertexai.init(project=PROJECT_ID, locatio

!pip install langchain_community
```

**PROJECT\_ID:** "projectllm-430702" 

**LOCATION:** "us-central1" 



## Collecting langchain\_community

```

Downloading langchain_community-0.2.10-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/c
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.1
Collecting dataclasses-json<0.7,>=0.5.7 (from langchain_community)
Downloading dataclasses_json-0.6.7-py3-none-any.whl.metadata (25 kB)
Requirement already satisfied: langchain<0.3.0,>=0.2.9 in /usr/local/lib/python3
Requirement already satisfied: langchain-core<0.3.0,>=0.2.23 in /usr/local/lib/p
Requirement already satisfied: langsmith<0.2.0,>=0.1.0 in /usr/local/lib/python3
Requirement already satisfied: numpy<2,>=1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: tenacity!=8.4.0,<9.0.0,>=8.1.0 in /usr/local/lib/
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dis
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/di
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3
Collecting marshmallow<4.0.0,>=3.18.0 (from dataclasses-json<0.7,>=0.5.7->langch
Downloading marshmallow-3.21.3-py3-none-any.whl.metadata (7.1 kB)
Collecting typing-inspect<1,>=0.4.0 (from dataclasses-json<0.7,>=0.5.7->langch
Downloading typing_inspect-0.9.0-py3-none-any.whl.metadata (1.5 kB)
Requirement already satisfied: langchain-text-splitters<0.3.0,>=0.2.0 in /usr/lc
Requirement already satisfied: pydantic<3,>=1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.10
Requirement already satisfied: packaging<25,>=23.2 in /usr/local/lib/python3.10/
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in /usr/local/lib/python3.1
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/pythor
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/c
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/c
Requirement already satisfied: typing-extensions>=4.6.0 in /usr/local/lib/pythor
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dis
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.10/dis
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.
Requirement already satisfied: pydantic-core==2.20.1 in /usr/local/lib/python3.1
Collecting mpy-extensions>=0.3.0 (from typing-inspect<1,>=0.4.0->dataclasses-j
Downloading mpy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Downloading langchain_community-0.2.10-py3-none-any.whl (2.3 MB)
_____ 2.3/2.3 MB 22.6 MB/s eta 0:00:00
Downloading dataclasses_json-0.6.7-py3-none-any.whl (28 kB)
Downloading marshmallow-3.21.3-py3-none-any.whl (49 kB)
_____ 49.2/49.2 kB 3.2 MB/s eta 0:00:00
Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)
Downloading mpy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Installing collected packages: mpy-extensions, marshmallow, typing-inspect, dat
Successfully installed dataclasses-json-0.6.7 langchain_community-0.2.10 marshma

```

## ✓ Importing libraries

Let's start by importing the libraries that we will need for this tutorial

```
# File system operations and displaying images
import os

# Import utility functions for timing and file handling
import time

# Libraries for downloading files, data manipulation, and creating a user interface
import uuid
from datetime import datetime


import fitz
import gradio as gr
import pandas as pd

# Initialize Vertex AI libraries for working with generative models
from google.cloud import aiplatform
from PIL import Image as PIL_Image
from vertexai.generative_models import GenerativeModel, Image
from vertexai.language_models import TextEmbeddingModel

# Print Vertex AI SDK version
print(f"Vertex AI SDK version: {aiplatform.__version__}")

# Import LangChain components
import langchain

print(f"LangChain version: {langchain.__version__}")
from langchain.text_splitter import CharacterTextSplitter
from langchain_community.document_loaders import DataFrameLoader
```

 Vertex AI SDK version: 1.60.0  
LangChain version: 0.2.11

## ▼ Initializing Gemini Vision Pro and Text Embedding models

```
# Loading Gemini Pro Vision Model
multimodal_model = GenerativeModel("gemini-1.0-pro-vision")

# Initializing embedding model
text_embedding_model = TextEmbeddingModel.from_pretrained("textembedding-gecko@003")

# Loading Gemini Pro Model
model = GenerativeModel("gemini-1.0-pro")
```

```

!wget https://www.hitachi.com/rev/archive/2023/r2023_04/pdf/04a02.pdf
!wget https://img.freepik.com/free-vector/hand-drawn-no-data-illustration_23-2150696455.jpg

# Create an "Images" directory if it doesn't exist
Image_Path = "./Images/"
if not os.path.exists(Image_Path):
    os.makedirs(Image_Path)

!mv hand-drawn-no-data-illustration_23-2150696455.jpg {Image_Path}/blank.jpg

--2024-07-27 16:37:31-- https://www.hitachi.com/rev/archive/2023/r2023_04/pdf/04a02.pdf
Resolving www.hitachi.com (www.hitachi.com)... 13.35.35.11, 13.35.35.93, 13.35.35.11
Connecting to www.hitachi.com (www.hitachi.com)|13.35.35.11|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://www.hitachihyoron.com/rev/notice/index.html [following]
--2024-07-27 16:37:31-- https://www.hitachihyoron.com/rev/notice/index.html
Resolving www.hitachihyoron.com (www.hitachihyoron.com)... 13.35.7.61, 13.35.7.11, 13.35.7.61
Connecting to www.hitachihyoron.com (www.hitachihyoron.com)|13.35.7.61|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13148 (13K) [text/html]
Saving to: '04a02.pdf'

04a02.pdf          100%[=====>]  12.84K  --.-KB/s    in 0s

2024-07-27 16:37:32 (228 MB/s) - '04a02.pdf' saved [13148/13148]

--2024-07-27 16:37:32-- https://img.freepik.com/free-vector/hand-drawn-no-data-illustration_23-2150696455.jpg
Resolving img.freepik.com (img.freepik.com)... 23.46.63.122, 23.46.63.129, 23.46.63.122
Connecting to img.freepik.com (img.freepik.com)|23.46.63.122|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32694 (32K) [image/jpeg]
Saving to: 'hand-drawn-no-data-illustration_23-2150696455.jpg'

hand-drawn-no-data- 100%[=====>]  31.93K  104KB/s    in 0.3s

2024-07-27 16:37:33 (104 KB/s) - 'hand-drawn-no-data-illustration_23-2150696455.jpg' saved [32694/32694]

```

## ✓ Convert PDF to Images and Extract Data Using Gemini Vision Pro

This module processes a set of images, extracting text and tabular data using the multimodal model Gemini Vision Pro. It manages potential errors, stores the extracted information in a DataFrame, and saves the results to a CSV file.

```

# Run the following code for each file
PDF_FILENAME = "Making-Movies-Manual.pdf" # Replace with the filename for making movies

```

```
# To get better resolution
zoom_x = 2.0 # horizontal zoom
zoom_y = 2.0 # vertical zoom
mat = fitz.Matrix(zoom_x, zoom_y) # zoom factor 2 in each dimension

doc = fitz.open(PDF_FILENAME) # open document
for page in doc: # iterate through the pages
    pix = page.get_pixmap(matrix=mat) # render page to an image
    outpath = f"./Images/{PDF_FILENAME}_{page.number}.jpg"
    pix.save(outpath) # store image as a PNG

# Define the path where images are located
image_names = os.listdir(Image_Path)
Max_images = len(image_names)

# Create empty lists to store image information
page_source = []
page_content = []
page_id = []

p_id = 0 # Initialize image ID counter
rest_count = 0 # Initialize counter for error handling

while p_id < Max_images:
    try:
        # Construct the full path to the current image
        image_path = Image_Path + image_names[p_id]

        # Load the image
        image = Image.load_from_file(image_path)

        # Generate prompts for text and table extraction
        prompt_text = "Extract all text content in the image"
        prompt_table = (
            "Detect table in this image. Extract content maintaining the structure"
        )

        # Extract text using your multimodal model
        contents = [image, prompt_text]
        response = multimodal_model.generate_content(contents)
        text_content = response.text

        # Extract table using your multimodal model
        contents = [image, prompt_table]
        response = multimodal_model.generate_content(contents)
        table_content = response.text

        # Log progress and store results
        print(f"processed image no: {p_id}")
        page_source.append(image_path)
        page_content.append(text_content + "\n" + table_content)
```

```
    page_id.append(p_id)
    p_id += 1

except Exception as err:
    # Handle errors during processing
    print(err)
    print("Taking Some Rest")
    time.sleep(1) # Pause execution for 1 second
    rest_count += 1
    if rest_count == 5: # Limit consecutive error handling
        rest_count = 0
        print(f"Cannot process image no: {image_path}")
        p_id += 1 # Move to the next image

# Create a DataFrame to store extracted information
df = pd.DataFrame(
    {"page_id": page_id, "page_source": page_source, "page_content": page_content}
)
del page_id, page_source, page_content # Conserve memory
df.head() # Preview the DataFrame
```





Response candidate content has no parts (and thus no text). The candidate is like  
Content:

```
{}
```

Candidate:

```
{
  "finish_reason": "RECITATION",
  "safety_ratings": [
    {
      "category": "HARM_CATEGORY_HATE_SPEECH",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.09619517,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.1046602
    },
    {
      "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.104294725,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.07382972
    },
    {
      "category": "HARM_CATEGORY_HARASSMENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.0996453,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.073164724
    },
    {
      "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.18922126,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.17050801
    }
  ],
  "citation_metadata": {
    "citations": [
      {
        "start_index": 1,
        "end_index": 220,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 131,
        "end_index": 415,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 316,
        "end_index": 573,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {

```

```

    "start_index": 480,
    "end_index": 1035,
    "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
  },
  {
    "start_index": 1004,
    "end_index": 1163,
    "uri": "https://www.studocu.com/en-us/document/american-film-institute-c
  }
]
}
}
Response:
{
  "candidates": [
    {
      "finish_reason": "RECITATION",
      "safety_ratings": [
        {
          "category": "HARM_CATEGORY_HATE_SPEECH",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.09619517,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.1046602
        },
        {
          "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.104294725,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.07382972
        },
        {
          "category": "HARM_CATEGORY_HARASSMENT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.0996453,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.073164724
        },
        {
          "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.18922126,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.17050801
        }
      ],
      "citation_metadata": {
        "citations": [
          {
            "start_index": 1,
            "end_index": 220,
            "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
          },
          {
            "start_index": 131,
            "end_index": 415.

```

```

      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-",
    },
    {
      "start_index": 316,
      "end_index": 573,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-",
    },
    {
      "start_index": 480,
      "end_index": 1035,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-",
    },
    {
      "start_index": 1004,
      "end_index": 1163,
      "uri": "https://www.studocu.com/en-us/document/american-film-institu",
    }
  ]
}
},
],
"usage_metadata": {
  "prompt_token_count": 265,
  "total_token_count": 265
}
}

```

Taking Some Rest

Cannot get the response text.

Cannot get the Candidate text.

Response candidate content has no parts (and thus no text). The candidate is lik  
Content:

```
{}
```

Candidate:

```

{
  "finish_reason": "RECITATION",
  "safety_ratings": [
    {
      "category": "HARM_CATEGORY_HATE_SPEECH",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.09619517,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.1046602
    },
    {
      "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.104294725,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.07382972
    },
    {
      "category": "HARM_CATEGORY_HARASSMENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.0996453,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.073164724
    }
  ]
}

```

```

    },
    {
      "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.18922126,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.17050801
    }
  ],
  "citation_metadata": {
    "citations": [
      {
        "start_index": 1,
        "end_index": 220,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 131,
        "end_index": 415,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 316,
        "end_index": 573,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 480,
        "end_index": 1035,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 1004,
        "end_index": 1163,
        "uri": "https://www.studocu.com/en-us/document/american-film-institute-c
      }
    ]
  }
}

```

Response:

```

{
  "candidates": [
    {
      "finish_reason": "RECITATION",
      "safety_ratings": [
        {
          "category": "HARM_CATEGORY_HATE_SPEECH",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.09619517,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.1046602
        },
        {
          "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.104294725,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",

```

```

    "severity_score": 0.07382972
  },
  {
    "category": "HARM_CATEGORY_HARASSMENT",
    "probability": "NEGLIGIBLE",
    "probability_score": 0.0996453,
    "severity": "HARM_SEVERITY_NEGLIGIBLE",
    "severity_score": 0.073164724
  },
  {
    "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
    "probability": "NEGLIGIBLE",
    "probability_score": 0.18922126,
    "severity": "HARM_SEVERITY_NEGLIGIBLE",
    "severity_score": 0.17050801
  }
],
"citation_metadata": {
  "citations": [
    {
      "start_index": 1,
      "end_index": 220,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 131,
      "end_index": 415,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 316,
      "end_index": 573,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 480,
      "end_index": 1035,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 1004,
      "end_index": 1163,
      "uri": "https://www.studocu.com/en-us/document/american-film-institu
    }
  \]
}
}
},
"usage\_metadata": {
  "prompt\_token\_count": 265,
  "total\_token\_count": 265
}
}

```

Taking Some Rest

Cannot get the response text.

Cannot get the Candidate text.

Response candidate content has no parts (and thus no text). The candidate is nil.

```

response candidate content has no parts (and thus no text). The candidate is in
Content:
{}
Candidate:
{
  "finish_reason": "RECITATION",
  "safety_ratings": [
    {
      "category": "HARM_CATEGORY_HATE_SPEECH",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.09619517,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.1046602
    },
    {
      "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.104294725,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.07382972
    },
    {
      "category": "HARM_CATEGORY_HARASSMENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.0996453,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.073164724
    },
    {
      "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.18922126,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.17050801
    }
  ],
  "citation_metadata": {
    "citations": [
      {
        "start_index": 1,
        "end_index": 220,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 131,
        "end_index": 415,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 316,
        "end_index": 573,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
      },
      {
        "start_index": 480,
        "end_index": 1035,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres

```

```

    },
    {
      "start_index": 1004,
      "end_index": 1163,
      "uri": "https://www.studocu.com/en-us/document/american-film-institute-c
    }
  ]
}
}
Response:
{
  "candidates": [
    {
      "finish_reason": "RECITATION",
      "safety_ratings": [
        {
          "category": "HARM_CATEGORY_HATE_SPEECH",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.09619517,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.1046602
        },
        {
          "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.104294725,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.07382972
        },
        {
          "category": "HARM_CATEGORY_HARASSMENT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.0996453,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.073164724
        },
        {
          "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.18922126,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.17050801
        }
      ],
      "citation_metadata": {
        "citations": [
          {
            "start_index": 1,
            "end_index": 220,
            "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
          },
          {
            "start_index": 131,
            "end_index": 415,
            "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
          },
        ]
      }
    }
  ]
}

```

```

    {
      "start_index": 316,
      "end_index": 573,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 480,
      "end_index": 1035,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 1004,
      "end_index": 1163,
      "uri": "https://www.studocu.com/en-us/document/american-film-institu
    }
  \]
}
},
\],
"usage\_metadata": {
  "prompt\_token\_count": 265,
  "total\_token\_count": 265
}
}

```

Taking Some Rest

Cannot get the response text.

Cannot get the Candidate text.

Response candidate content has no parts (and thus no text). The candidate is lik  
Content:

```
{}
```

Candidate:

```

{
  "finish_reason": "RECITATION",
  "safety_ratings": [
    {
      "category": "HARM_CATEGORY_HATE_SPEECH",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.09619517,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.1046602
    },
    {
      "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.104294725,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.07382972
    },
    {
      "category": "HARM_CATEGORY_HARASSMENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.0996453,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.073164724
    },
    {
      "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",

```



```

    "probability": "NEGLIGIBLE",
    "probability_score": 0.18922126,
    "severity": "HARM_SEVERITY_NEGLIGIBLE",
    "severity_score": 0.17050801
  }
],
"citation_metadata": {
  "citations": [
    {
      "start_index": 1,
      "end_index": 220,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
    },
    {
      "start_index": 131,
      "end_index": 415,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
    },
    {
      "start_index": 316,
      "end_index": 573,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
    },
    {
      "start_index": 480,
      "end_index": 1035,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres
    },
    {
      "start_index": 1004,
      "end_index": 1163,
      "uri": "https://www.studocu.com/en-us/document/american-film-institute-c
    }
  ]
}
}

```

Response:

```

{
  "candidates": [
    {
      "finish_reason": "RECITATION",
      "safety_ratings": [
        {
          "category": "HARM_CATEGORY_HATE_SPEECH",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.09619517,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.1046602
        },
        {
          "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.104294725,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.07382972
        }
      ]
    }
  ]
}

```

```

    "category": "HARM_CATEGORY_HARASSMENT",
    "probability": "NEGLIGIBLE",
    "probability_score": 0.0996453,
    "severity": "HARM_SEVERITY_NEGLIGIBLE",
    "severity_score": 0.073164724
  },
  {
    "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
    "probability": "NEGLIGIBLE",
    "probability_score": 0.18922126,
    "severity": "HARM_SEVERITY_NEGLIGIBLE",
    "severity_score": 0.17050801
  }
],
"citation_metadata": {
  "citations": [
    {
      "start_index": 1,
      "end_index": 220,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 131,
      "end_index": 415,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 316,
      "end_index": 573,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 480,
      "end_index": 1035,
      "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-
    },
    {
      "start_index": 1004,
      "end_index": 1163,
      "uri": "https://www.studocu.com/en-us/document/american-film-institu
    }
  \]
}
},
"usage\_metadata": {
  "prompt\_token\_count": 265,
  "total\_token\_count": 265
}
}
Taking Some Rest
Cannot get the response text.
Cannot get the Candidate text.
Response candidate content has no parts \(and thus no text\). The candidate is lik
Content:
{}

```

```

candidate:
{
  "finish_reason": "RECITATION",
  "safety_ratings": [
    {
      "category": "HARM_CATEGORY_HATE_SPEECH",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.09602549,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.104294725
    },
    {
      "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.10484337,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.07382972
    },
    {
      "category": "HARM_CATEGORY_HARASSMENT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.100524865,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.07329728
    },
    {
      "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
      "probability": "NEGLIGIBLE",
      "probability_score": 0.18892181,
      "severity": "HARM_SEVERITY_NEGLIGIBLE",
      "severity_score": 0.16954333
    }
  ],
  "citation_metadata": {
    "citations": [
      {
        "start_index": 1,
        "end_index": 221,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres"
      },
      {
        "start_index": 132,
        "end_index": 416,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres"
      },
      {
        "start_index": 317,
        "end_index": 574,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres"
      },
      {
        "start_index": 481,
        "end_index": 1036,
        "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-pres"
      },
      {
        "start_index": 1005,

```

```

    "end_index": 1164,
    "uri": "https://www.studocu.com/en-us/document/american-film-institute-c"
  }
]
}
}
Response:
{
  "candidates": [
    {
      "finish_reason": "RECITATION",
      "safety_ratings": [
        {
          "category": "HARM_CATEGORY_HATE_SPEECH",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.09602549,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.104294725
        },
        {
          "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.10484337,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.07382972
        },
        {
          "category": "HARM_CATEGORY_HARASSMENT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.100524865,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.07329728
        },
        {
          "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
          "probability": "NEGLIGIBLE",
          "probability_score": 0.18892181,
          "severity": "HARM_SEVERITY_NEGLIGIBLE",
          "severity_score": 0.16954333
        }
      ],
      "citation_metadata": {
        "citations": [
          {
            "start_index": 1,
            "end_index": 221,
            "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-"
          },
          {
            "start_index": 132,
            "end_index": 416,
            "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-"
          },
          {
            "start_index": 317,
            "end_index": 574,
            "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-"
          }
        ]
      }
    }
  ]
}

```

```
uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-",
},
{
  "start_index": 481,
  "end_index": 1036,
  "uri": "https://artsdocbox.com/Movies/102435996-The-film-foundation-",
},
{
  "start_index": 1005,
  "end_index": 1164,
  "uri": "https://www.studocu.com/en-us/document/american-film-institu
}
]
}
},
],
"usage_metadata": {
  "prompt_token_count": 265,
  "total_token_count": 265
}
}
```

Next steps:

Generate code index

View recommended plots

New interactive sheet

Taking Some Rest

Cannot process image no: ./Images/Making-Movies-Manual.pdf\_1.jpg

processed image no: 1

processed image no: 2

processed image no: 3

processed image no: 4

processed image no: 5

page_id		page_source	page_content
0	1	./Images/Making-Movies-Manual.pdf_3.jpg	A Word from Your Sponsor\nDo you like going t...
1	2	./Images/Making-Movies-Manual.pdf_0.jpg	013322\nThe Film Foundation presents:\nMAKING...
2	3	./Images/Making-Movies-Manual.pdf_2.jpg	Introduction\nThis manual will help you make ...
3	4	./Images/blank.jpg	? \n? \nX \n     \n   :---   :---: \n   ?     ...

## Generate Text Embeddings

Leverage a powerful language model textembedding-gecko to generate rich text embeddings that helps us find relevant information from a dataset.

```
def generate_text_embedding(text) -> list:
    """Text embedding with a Large Language Model."""
    embeddings = text_embedding_model.get_embeddings([text])
    vector = embeddings[0].values
    return vector

# Create a DataFrameLoader to prepare data for LangChain
loader = DataFrameLoader(df, page_content_column="page_content")

# Load documents from the 'page_content' column of your DataFrame
documents = loader.load()

# Log the number of documents loaded
print(f"# of documents loaded (pre-chunking) = {len(documents)}")

# Create a text splitter to divide documents into smaller chunks
text_splitter = CharacterTextSplitter(
    chunk_size=10000, # Target size of approximately 10000 characters per chunk
    chunk_overlap=200, # overlap between chunks
)

# Split the loaded documents
doc_splits = text_splitter.split_documents(documents)

# Add a 'chunk' ID to each document split's metadata for tracking
for idx, split in enumerate(doc_splits):
    split.metadata["chunk"] = idx

# Log the number of documents after splitting
print(f"# of documents = {len(doc_splits)}")

texts = [doc.page_content for doc in doc_splits]
text_embeddings_list = []
id_list = []
page_source_list = []
for doc in doc_splits:
    id = uuid.uuid4()
    text_embeddings_list.append(generate_text_embedding(doc.page_content))
    id_list.append(str(id))
    page_source_list.append(doc.metadata["page_source"])
    time.sleep(1) # So that we don't run into Quota Issue

# Creating a dataframe of ID, embeddings, page_source and text
embedding_df = pd.DataFrame(
    {
        "id": id_list,
        "embedding": text_embeddings_list,
        "page_source": page_source_list,
        "text": texts,
    }
)
```