* **Updated Instructions for Running the Model**

This model allows users to ask questions based on a **PDF document** using **TF-IDF and cosine similarity** to retrieve the most relevant answer.

* **Install Required Dependencies**

Before running the model, install the required Python libraries:

pip install PyPDF2, numpy, joblib scikit-learn

* **Extract Text from a PDF**

Run the following code to extract text from your uploaded **PDF file**:

**from** PyPDF2 **import** PdfReader

pdf\_path **=** "/content/PYTHON PROGRAMMING NOTES (1).pdf"

reader **=** PdfReader(pdf\_path)

text **=** "\n"**.**join([page**.**extract\_text() **for** page **in** reader**.**pages **if** page**.**extract\_text()])

text[:1000]

* **Preprocess and Chunk the Text**

Once the text is extracted, clean it and split it into smaller **chunks** for better searchability.

import re

def clean\_text(text):

text = re.sub(r'\n+', '\n', text)

text = re.sub(r'[^\x00-\x7F]+', ' ', text)

text = re.sub(r'\s+', ' ', text)

return text.strip()

cleaned\_text = clean\_text(text)

words = cleaned\_text.split()

chunk\_size = 500

chunks = [" ".join(words[i:i + chunk\_size]) for i in range(0, len(words), chunk\_size)]

chunks[0]

import PyPDF2

pdf\_path = "/content/PYTHON PROGRAMMING NOTES (1).pdf"

with open(pdf\_path, "rb") as file:

reader = PyPDF2.PdfReader(file)

text = "\n".join([page.extract\_text() for page in reader.pages if page.extract\_text()])

with open("book\_text.txt", "w", encoding="utf-8") as text\_file:

text\_file.write(text)

print(" PDF text extracted and saved to 'book\_text.txt'")

import numpy as np

with open("book\_text.txt", "r", encoding="utf-8") as file:

book\_text = file.read()

chunk\_size = 500

words = book\_text.split()

chunks = [" ".join(words[i:i + chunk\_size]) for i in range(0, len(words), chunk\_size)]

np.save("chunks.npy", np.array(chunks))

print(" Text has been split into chunks and saved as 'chunks.npy'")

* **Create and Train the TF-IDF Model**

Now, we transform the text chunks into **TF-IDF vectors** to allow efficient question-answering.

from sklearn.feature\_extraction.text import TfidfVectorizer

import numpy as np

import joblib

chunks = np.load("chunks.npy", allow\_pickle=True)

vectorizer = TfidfVectorizer(stop\_words='english', max\_features=5000)

tfidf\_matrix = vectorizer.fit\_transform(chunks)

joblib.dump(vectorizer, "vectorizer.pkl")

np.save("tfidf\_matrix.npy", tfidf\_matrix.toarray())

print(" TF-IDF model is now ready for question answering!")

* **Run the Question Answering Model**

Now, you can ask questions based on the **uploaded PDF**.

import numpy as np

import joblib

from sklearn.metrics.pairwise import cosine\_similarity

vectorizer = joblib.load("vectorizer.pkl")

tfidf\_matrix = np.load("tfidf\_matrix.npy")

chunks = np.load("chunks.npy", allow\_pickle=True)

print("💬 Type your questions below. Type 'exit' to stop.\n")

while True:

question = input("Ask a question: ")

if question.lower() == "exit":

print("🚪 Exiting the Q&A system. Have a great day! 😊")

break

question\_vector = vectorizer.transform([question])

similarities = cosine\_similarity(question\_vector, tfidf\_matrix)

best\_match\_idx = np.argmax(similarities)

best\_answer = chunks[best\_match\_idx]

print("\n📖 Answer from the book:\n")

print(best\_answer)

print("\n" + "-"\*80 + "\n")

* **Sample Input & Expected Output**

Once the model is running, here’s an example interaction:

**Example Input**

Ask a question: define While Loop?

**Expected Output**

📖 Answer from the book:

**inside a nested loop ( loop inside another loop), break will terminate the innermost loop.**

* **How to Upload to GitHub**
* **Initialize a Git Repository**

Run the following command inside your project folder:

**git init**

* **Add the Files**

**git add .**

* **Commit the Changes**

**git commit -m "Added TF-IDF Q&A Model"**

* **Create a GitHub Repository**
* Go to [GitHub](https://github.com/)
* Click on **"New Repository"**
* Name your repository (e.g., QA\_Model\_TFIDF)
* Copy the repository URL (e.g., **https://github.com/yourusername/QA\_Model\_TFIDF.git)**