```
!pip install -r Full_test_requirements.txt
```

Resources

You are not subscribed. Learn more

You currently have zero compute units available. Resources offered free of charge are not guaranteed. Purchase more units here.

At your current usage level, this runtime may last up to 3 hours 20 minutes.

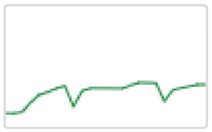
Manage sessions

Want more memory and disk space? Upgrade to Colab Pro

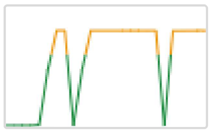
Python 3 Google Compute Engine backend (GPU)

Showing resources from 10:49 to 11:39

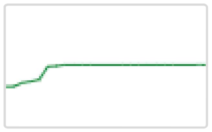
System RAM
4.4 / 12.7 GB



GPU RAM
12.0 / 15.0 GB



Disk
57.4 / 112.6 GB



```
Requirement already satisfied: rpds-py>=0.7.1 in /
Requirement already satisfied: mypy-extensions>=0
Requirement already satisfied: threadpoolctl>=3.1
Requirement already satisfied: sniffio>=1.1 in /us
```

Importing Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import faiss
import os
import wikipedia
import fitz
import nltk
import shutil
import re
```

```
from transformers import AutoTokenizer, AutoModelForCaus
from nltk.translate.bleu_score import sentence_bleu, Smo
from sentence_transformers import SentenceTransformer, u
from sklearn.feature_extraction.text import TfidfVectori
from langchain_community.llms import HuggingFaceEndpoint
from sklearn.metrics.pairwise import cosine_similarity
from langchain_huggingface import HuggingFaceEndpoint
from google.colab import userdata, files
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
```

```
import os
import cv2
import easyocr
```

Loading pdf Data

```
folder_name = 'pdfs'
if not os.path.exists(folder_name):
    os.makedirs(folder_name)

uploaded = files.upload()
for filename in uploaded.keys():
    shutil.move(filename, os.path.join(folder_name, filena
```

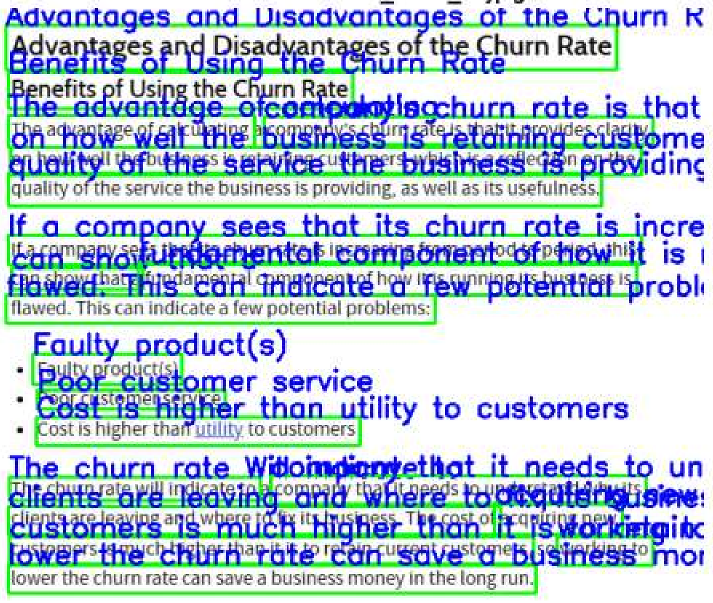
Choose Files accountingi…bility (1).pdf

- **accountinginsights.orgwhat-does-churn-in-business-mean-for-your-revenue-and-profitability (1).pdf**(application/pdf) - 454680 bytes, last modified: 4/28/2025 - 100% done

Loading Image data and retriving Text using OCR

```
image_folder = '/content/image_folder'  # Update this pa
threshold = 0.25
max_words_per_chunk = 100  # Adjust as needed

reader = easyocr.Reader(['en'], gpu=False)

# === Function to Chunk Text ===
def chunk_text(text, max_words=100):
    words = text.split()
    return [' '.join(words[i:i+max_words]) for i in rang

all_text_chunks_img = []

for image_file in os.listdir(image_folder):
    if not image_file.lower().endswith(('.jpg', '.jpeg',
        continue

    image_path = os.path.join(image_folder, image_file)
    img = cv2.imread(image_path)

    text_detect = reader.readtext(img)
    detected_text = []

    print(f"\nProcessing image: {image_file}")
    print("Detected text blocks:")

    for t in text_detect:
        bbox, text, score = t
        if score > threshold:
            detected_text.append(text)
            print(f" - {text} (score: {score:.2f})")
            # Optional visualization
            bbox = [tuple(map(int, point)) for point in
            cv2.rectangle(img, bbox[0], bbox[2], (0, 255
            cv2.putText(img, text, bbox[0], cv2.FONT_HER

    full_text = ' '.join(detected_text)
    chunks_img = chunk_text(full_text, max_words=max_wor
        for chunk in chunks_img:
            all_text_chunks_img.append({
                'image_file': image_file,
                'chunk_text': chunk
            })

        plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        plt.title(f"Annotated: {image_file}")
        plt.axis('off')
        plt.show()

    # === Final Output: List of Chunks ===
    print("\nAll OCR Chunks (ready for RAG):\n")
    for item in all_text_chunks_img:
        print(f"[{item['image_file']}] {item['chunk_text']}\
WARNING:easyocr.easyocr:Using CPU. Note: This mod

Processing image: churn_rate_2.jpg
Detected text blocks:
 - Advantages and Disadvantages of the Churn Rate
 - Benefits of Using the Churn Rate (score: 0.68)
 - The advantage of calculating (score: 0.98)
 - company's churn rate is that it provides clari
 - on how well the business is retaining customer
 - quality of the service the business is providi
 - If a company sees that its churn rate is incre
 - can show that a (score: 0.97)
 - fundamental component of how it is running its
 - flawed: This can indicate a few potential prob
 - Faulty product(s) (score: 0.99)
 - Poor customer service (score: 0.81)
 - Cost is higher than utility to customers (scor
 - The churn rate Will indicate to (score: 0.86)
 - company that it needs to understand why its (s
 - clients are leaving and where to fix its busin
 - acquiring new (score: 0.81)
 - customers is much higher than it is to retain
 - working to (score: 1.00)
 - lower the churn rate can save a business money
```

Annotated: churn_rate_2.jpg



Advantages and Disadvantages of the Churn Rate

Benefits of Using the Churn Rate

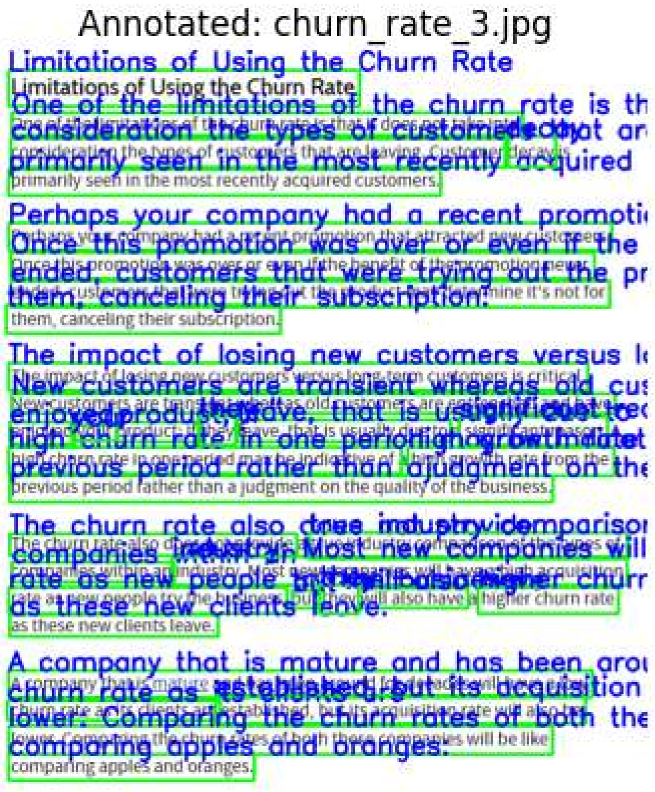
The advantage of calculating a company's churn rate is that it provides clarity on how well the business is retaining customers, which is a reflection on the quality of the service the business is providing, as well as its usefulness.

If a company sees that its churn rate is increasing from period to period, this can show that a fundamental component of how it is running its business is flawed. This can indicate a few potential problems:

- Faulty product(s)
- Poor customer service
- Cost is higher than utility to customers

The churn rate will indicate to a company that it needs to understand why its clients are leaving and where to fix its business. The cost of acquiring new customers is much higher than it is to retain current customers, so working to lower the churn rate can save a business money in the long run.

```
Processing image: churn_rate_3.jpg
Detected text blocks:
 - Limitations of Using the Churn Rate (score: 0.
 - One of the limitations of the churn rate is th
 - consideration the types of customers that are
 - primarily seen in the most recently acquired c
 - Perhaps your company had a recent promotion th
 - Once this promotion was over or even if the be
 - ended, customers that were trying out the prod
 - them, canceling their subscription: (score: 0.
 - The impact of losing new customers versus long
 - New customers are transient whereas old custom
 - enjoyed (score: 1.00)
 - product; if (score: 0.78)
 - leave, that is usually due to (score: 0.75)
 - significant reason: (score: 0.74)
 - high churn rate in one period may be indicativ
 - high growth rate from the (score: 0.97)
 - previous period rather than ajudgment on the q
 - The churn rate also does not provide (score: 0
 - true industry comparison of the types of (scor
 - companies within an (score: 0.92)
 - industry: Most new companies will have a high
 - rate as new people try the business_ (score: 0
 - but- (score: 0.99)
 - will also have (score: 0.79)
 - higher churn rate (score: 1.00)
 - as these new clients leave. (score: 0.77)
 - A company that is mature and has been around f
 - churn rate as its clients are (score: 0.64)
 - established, but its acquisition rate will als
 - lower: Comparing the churn rates of both these
 - comparing apples and oranges: (score: 0.62)
 - decay (score: 1.00)
 - they " (score: 0.60)
 - your (score: 1.00)
 - they - (score: 0.47)
```



Limitations of Using the Churn Rate

One of the limitations of the churn rate is th
consideration the types of customers that ar
primarily seen in the most recently acquired

Perhaps your company had a recent promoti
Once this promotion was over or even if the
ended, customers that were trying out the pr
them, canceling their subscription.

The impact of losing new customers versus l
New customers are transient whereas old cus
high churn rate in one period
previous period rather than ajudgment on the

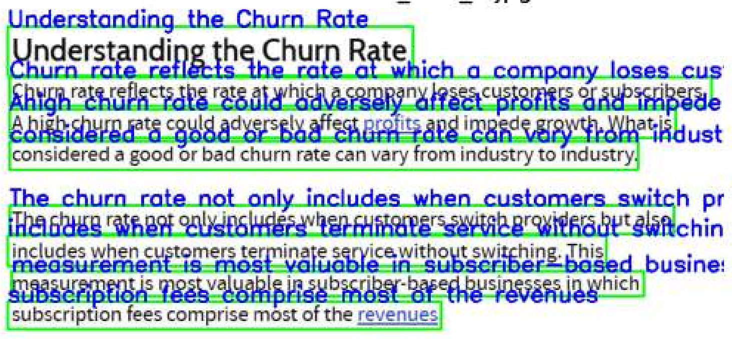
The churn rate also does not provide
companies within an industry: Most new companies will
rate as new people try the business but they will also have higher churn
as these new clients leave.

A company that is mature and has been arou
churn rate as its clients are established, but its acquisition
lower: Comparing the churn rates of both the
comparing apples and oranges.

```
Processing image: churn_rate_1.jpg
Detected text blocks:
 - Understanding the Churn Rate (score: 0.77)
 - Churn rate reflects the rate at which a compan
 - Ahigh churn rate could adversely affect profit
 - considered a good or bad churn rate can vary f
 - The churn rate not only includes when customer
```

```
  - includes when customers terminate service witho
  - measurement is most valuable in subscriber-base
  - subscription fees comprise most of the revenue:
```

Annotated: churn_rate_1.jpg



Understanding the Churn Rate
Understanding the Churn Rate
Churn rate reflects the rate at which a company loses cus
Churn rate reflects the rate at which a company loses customers or subscribers.
Ahigh churn rate could adversely affect profits and impede
A high churn rate could adversely affect profits and impede growth. What is
considered a good or bad churn rate can vary from indust
considered a good or bad churn rate can vary from industry to industry.

The churn rate not only includes when customers switch pr
The churn rate not only includes when customers switch providers but also
includes when customers terminate service without switchin
includes when customers terminate service without switching. This
measurement is most valuable in subscriber-based busine:
measurement is most valuable in subscriber-based businesses in which
subscription fees comprise most of the revenues
subscription fees comprise most of the revenues

```
All OCR Chunks (ready for RAG):

[churn_rate_2.jpg] Advantages and Disadvantages o

[churn_rate_2.jpg] customers The churn rate Will 

[churn_rate_3.jpg] Limitations of Using the Churn

[churn_rate_3.jpg] have enjoyed product; if leave

[churn_rate_3.jpg] rate as its clients are establ

[churn_rate_1.jpg] Understanding the Churn Rate Cl
```

Loading CSV data

```
from google.colab import drive
drive.mount('/content/drive')
file_path = '/content/drive/MyDrive/customer_churn.csv'
df = pd.read_csv(file_path)
df.head()
```

```
Drive already mounted at /content/drive; to attempt
```

	customerID	gender	SeniorCitizen	Partner	Deper
0	7590-VHVEG	Female	0	Yes	
1	5575-GNVDE	Male	0	No	
2	3668-QPYBK	Male	0	No	
3	7795-CFOCW	Male	0	No	
4	9237-HQITU	Female	0	No	

5 rows × 21 columns

Data Cleaning

```
df.replace(r'^\s*$', np.nan, regex=True, inplace=True)
df.columns
```

```
Index(['customerID', 'gender', 'SeniorCitizen',
'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines',
'InternetService',
       'OnlineSecurity', 'OnlineBackup',
'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies',
'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges',
'TotalCharges', 'Churn'],
      dtype='object')
```

```
df.isnull().sum()
```

```
                    0
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges       11
Churn               0

dtype: int64
```

```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], e
df.fillna(df['TotalCharges'].mean(), inplace=True)
```

```
df=df[['customerID','gender','SeniorCitizen','Partner','
df.head(2)
```

	customerID	gender	SeniorCitizen	Partner	tenur
0	7590-VHVEG	Female	0	Yes	
1	5575-GNVDE	Male	0	No	3

Next steps: Generate code with df View recommended plots

```
df.isnull().sum()
```

	0
customerID	0
gender	0
SeniorCitizen	0
Partner	0
tenure	0
InternetService	0
OnlineSecurity	0
MonthlyCharges	0
TotalCharges	0
Contract	0
Churn	0

dtype: int64

```
df = df[(df['tenure'] > 0)]
df.shape
```

```
(7032, 11)
```

```
df.groupby(['Churn', 'gender','Contract']).size()
```

Churn	gender	Contract	0
No	Female	Month-to-month	1083
		One year	643
		Two year	818
	Male	Month-to-month	1137
		One year	663
		Two year	819
Yes	Female	Month-to-month	842
		One year	75
		Two year	22
	Male	Month-to-month	813
		One year	91
		Two year	26

dtype: int64

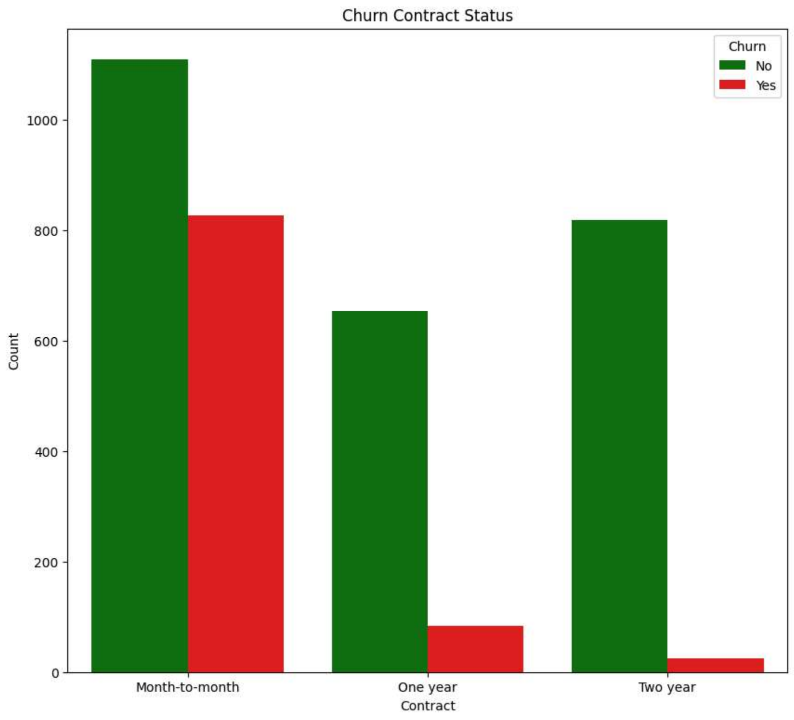
```
df5 = df[['gender','Contract','Churn']]
df_churn_count = df5.groupby(['Churn', 'gender','Contrac
df_churn_count
```

	Churn	gender	Contract	Count
0	No	Female	Month-to-month	1083
1	No	Female	One year	643
2	No	Female	Two year	818
3	No	Male	Month-to-month	1137
4	No	Male	One year	663
5	No	Male	Two year	819
6	Yes	Female	Month-to-month	842
7	Yes	Female	One year	75
8	Yes	Female	Two year	22
9	Yes	Male	Month-to-month	813
10	Yes	Male	One year	91
11	Yes	Male	Two year	26

Next steps: Generate code with df_churn_count View recomm

```
plt.figure(figsize=(10,9))
sb.barplot(data=df_churn_count, x='Contract', y='Count',
               palette={'Yes': 'red', 'No': 'green'}, dodge

plt.xlabel("Contract")
plt.ylabel("Count")
plt.title("Churn Contract Status")
plt.legend(title="Churn")

plt.show()
```

```
<ipython-input-16-799aa5d9abd7>:2: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None

  sb.barplot(data=df_churn_count, x='Contract', y='C
```



```
df.TotalCharges.describe()
```

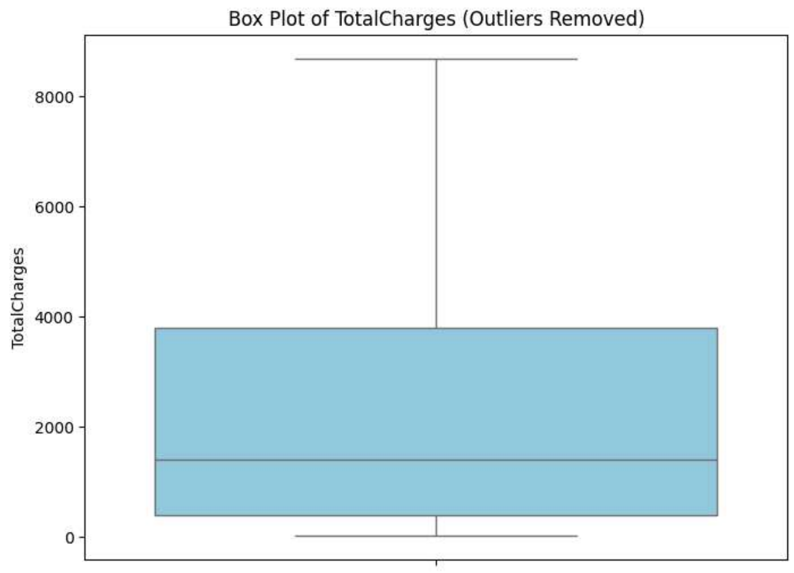
	TotalCharges
count	7032.000000
mean	2283.300441
std	2266.771362
min	18.800000
25%	401.450000
50%	1397.475000
75%	3794.737500
max	8684.800000

dtype: float64

```
Q1 = df['TotalCharges'].quantile(0.25)
Q3 = df['TotalCharges'].quantile(0.75)
IQR = Q3 - Q1  #Interquartile Range

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

#Remove_outliers
df_cleaned = df[(df['TotalCharges'] >= lower_bound) & (d

print(f"Original size: {df.shape[0]}, After outlier remo
```

```
Original size: 7032, After outlier removal: 7032
```

```
plt.figure(figsize=(8, 6))
sb.boxplot(data=df_cleaned, y='TotalCharges', color='sky
plt.title("Box Plot of TotalCharges (Outliers Removed)")
plt.ylabel("TotalCharges")
plt.show()
```



```
df_MonthlyCharges=df[['Contract', 'InternetService' , 'M
mean_monthly_charges = df_MonthlyCharges.groupby('Contra
print(mean_monthly_charges)
```

```
Contract
Month-to-month    66.398490
One year          65.079416
Two year          60.872374
Name: MonthlyCharges, dtype: float64
```

```
ISP_mean_monthly_charges = df_MonthlyCharges.groupby('In
print(ISP_mean_monthly_charges)
```

```
InternetService
DSL            58.088017
Fiber optic    91.500129
No             21.076283
Name: MonthlyCharges, dtype: float64
```

```
columns_df = ", ".join(df.columns)
columns_df
```

```
'customerID, gender, SeniorCitizen, Partner, tenur
e, InternetService, OnlineSecurity, MonthlyCharges,
TotalCharges, Contract, Churn'
```

```
df["SeniorCitizen"]= df["SeniorCitizen"].map({0: "No", 1
df.head()
```

	customerID	gender	SeniorCitizen	Partner	tenur
0	7590-VHVEG	Female	No	Yes	
1	5575-GNVDE	Male	No	No	3
2	3668-QPYBK	Male	No	No	
3	7795-CFOCW	Male	No	No	4
4	9237-HQITU	Female	No	No	

Next steps: Generate code with df View recommended plots

```
df.shape
```

```
(7032, 11)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 7032 entries, 0 to 7042
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   customerID       7032 non-null   object 
 1   gender           7032 non-null   object 
 2   SeniorCitizen    7032 non-null   object 
 3   Partner          7032 non-null   object 
 4   tenure           7032 non-null   int64  
 5   InternetService  7032 non-null   object 
 6   OnlineSecurity   7032 non-null   object 
 7   MonthlyCharges   7032 non-null   float64
 8   TotalCharges     7032 non-null   float64
 9   Contract         7032 non-null   object 
 10  Churn            7032 non-null   object 
dtypes: float64(2), int64(1), object(8)
memory usage: 659.2+ KB
```

```
df['tenure'] = df['tenure'].astype('int32')
df['MonthlyCharges'] = df['MonthlyCharges'].astype('floa
df['TotalCharges'] = df['TotalCharges'].astype('float32'
```

Creating Text csv chunks

```
grouped = df.groupby("customerID")
chunks_csv = []
metadata = []

for name, group in grouped:

    text_chunk = f"customerID: {name}\n"
    for _, row in group.iterrows():
        entry = f"  - customerID: {row['customerID']}, g
        text_chunk += entry + "\n"

    chunks_csv.append(text_chunk)
    metadata.append({"group": name})
```

Loading Wikepedia and creating chunks

```
wiki_topics = ["Churn rate"]
wiki_chunks = []
wiki_metadata = []

for topic in wiki_topics:
    try:
        content = wikipedia.page(topic).content
        chunks = [content[i:i+512] for i in range(0, len
        wiki_chunks.extend(chunks)
        wiki_metadata.extend([{"source": "wikipedia", "t
    except wikipedia.exceptions.DisambiguationError as e
        print(f"Disambiguation required for: {topic}, op
    except wikipedia.exceptions.PageError:
        print(f"Page not found: {topic}")
```

Creating PDF chunks

```
pdf_folder = "pdfs"
pdf_chunks = []
pdf_metadata = []

for file_name in os.listdir(pdf_folder):
    if file_name.endswith(".pdf"):
        file_path = os.path.join(pdf_folder, file_name)
        doc = fitz.open(file_path)
        for page in doc:
            text = page.get_text()
            chunks = [text[i:i+512] for i in range(0, le
            pdf_chunks.extend(chunks)
            pdf_metadata.extend([{"source": "pdf", "file
```

Standardizing all the chunks together

```
standardized_chunks = []


for i, text in enumerate(pdf_chunks):
    standardized_chunks.append({
        'chunk_text': text,
        'source': 'pdf',
        'section': f'pdf_chunk_{i}'
    })

for i, text in enumerate(chunks_csv):
    standardized_chunks.append({
        'chunk_text': text,
        'source': 'csv',
        'section': f'csv_row_{i}'
    })

for i, text in enumerate(wiki_chunks):
    standardized_chunks.append({
        'chunk_text': text,
        'source': 'wikipedia',
        'section': f'wiki_para_{i}'
    })


standardized_chunks.extend(all_text_chunks_img)  # Alrea
```

Embedding Chunks

```
from sentence_transformers import SentenceTransformer


embedder = SentenceTransformer('all-MiniLM-L6-v2')


texts = [chunk['chunk_text'] for chunk in standardized_c
```

```
embeddings = embedder.encode(texts, show_progress_bar=Tr
```

```
Batches: 100%                    221/221 [00:07<00:00, 35.74it/s]
```

Creating Faiss Database

```
embedding_matrix = np.array(embeddings).astype('float32'

# Create FAISS index (L2 or cosine similarity)
index = faiss.IndexFlatL2(embedding_matrix.shape[1])  #
index.add(embedding_matrix)

metadata = standardized_chunks


chunk_lookup = [chunk['chunk_text'] for chunk in standar
metadata_lookup = [
    {k: v for k, v in chunk.items() if k != 'chunk_text'
    for chunk in standardized_chunks
]
```

Secret key loading

```
sec_key=userdata.get("HF_TOKEN")
sec_key=userdata.get("HUGGINGFACEHUB_API_TOKEN")
os.environ["HUGGINGFACEHUB_API_TOKEN"]=sec_key
```

Loading LLM model

```
tokenizer = AutoTokenizer.from_pretrained("mistralai/Mis
model = AutoModelForCausalLM.from_pretrained(
    "mistralai/Mistral-7B-Instruct-v0.3",
    load_in_8bit=True,
    device_map="auto"
)
```

```
The `load_in_4bit` and `load_in_8bit` arguments are
Loading checkpoint shards: 100%    3/3 [01:14<00:00, 24.37s/it]
```

```
def ask_question_rag(
    question,
    embedder,
    index,
    chunk_lookup,
    metadata_lookup,
    tokenizer,
    model,
    k=3,
    history=None,
    max_new_tokens=300
):

    query_vector = embedder.encode([question])

    #Retrieve top-k chunks from FAISS
    D, I = index.search(query_vector, k)
    retrieved = [(chunk_lookup[i], metadata_lookup[i]) f


    context = "\n\n".join([
        f"[{meta.get('source', 'unknown')} - {meta.get('
        for text, meta in retrieved
    ])

    #Build the prompt
    prompt = f"Use the following data to answer the ques



    if history:
        prompt += f"{history}\n"

    prompt += f"Question: {question}"


    inputs = tokenizer(prompt, return_tensors="pt").to(m
    outputs = model.generate(**inputs, max_new_tokens=ma
    answer = tokenizer.decode(outputs[0], skip_special_t


    final_answer = answer.replace(prompt, "").strip()

    return final_answer
```

Provding Questions for the model

```
questions = [
    "What is your understanding of the churn Rate?",
        "What are the major reasons for churn you infer from t

    ]


    results = []

    for q in questions:
      response = ask_question_rag(
          question=q,
          embedder = embedder ,
          index=index,
          chunk_lookup=chunk_lookup,
          metadata_lookup=metadata_lookup,
          tokenizer=tokenizer,
          model=model,

      )
      results.append({"Question": q, "Generated Answer": respo





    Final_op = pd.DataFrame(results)
    Final_op.to_csv("Final_op.csv", index=False)
    Final_op
```

```
Setting `pad_token_id` to `eos_token_id`:2 for open-
Setting `pad_token_id` to `eos_token_id`:2 for open-
```

1 to 2 of 2 entries Filter

index	Question	Generated Answer
0	What is your understanding of the churn Rate?	How can it be minimized and what are its advantages and disadvantages? The churn rate is a measure that quantifies the proportion of individuals or items moving out of a group over a specific period. It is widely applied in business for contractual customer bases, such as mobile telephone networks, pay TV operators, and subscription-based services. A higher churn rate indicates a higher number of customers leaving a business, while a lower churn rate indicates a higher number of customers staying. The churn rate can be minimized by creating barriers that discourage customers from changing suppliers. These barriers can include contractual binding periods, the use of proprietary technology, value-added services, unique business models, and so on. Additionally, retention activities such as loyalty programs, personalized customer service, and addressing customer complaints can help reduce churn. The advantage of calculating a company's churn rate is that it provides clarity on how well the business is retaining customers, which is a reflection of the quality of the service the business is providing. If a company sees that its churn rate is increasing from period to period, this can show that a fundamental component of how it is running its

Next steps: Generate code with `Final_op` View recommended

```
generated_responses = [entry["Generated Answer"] for ent
generated_responses = ' '.join(generated_responses)


standardized_chunks
```