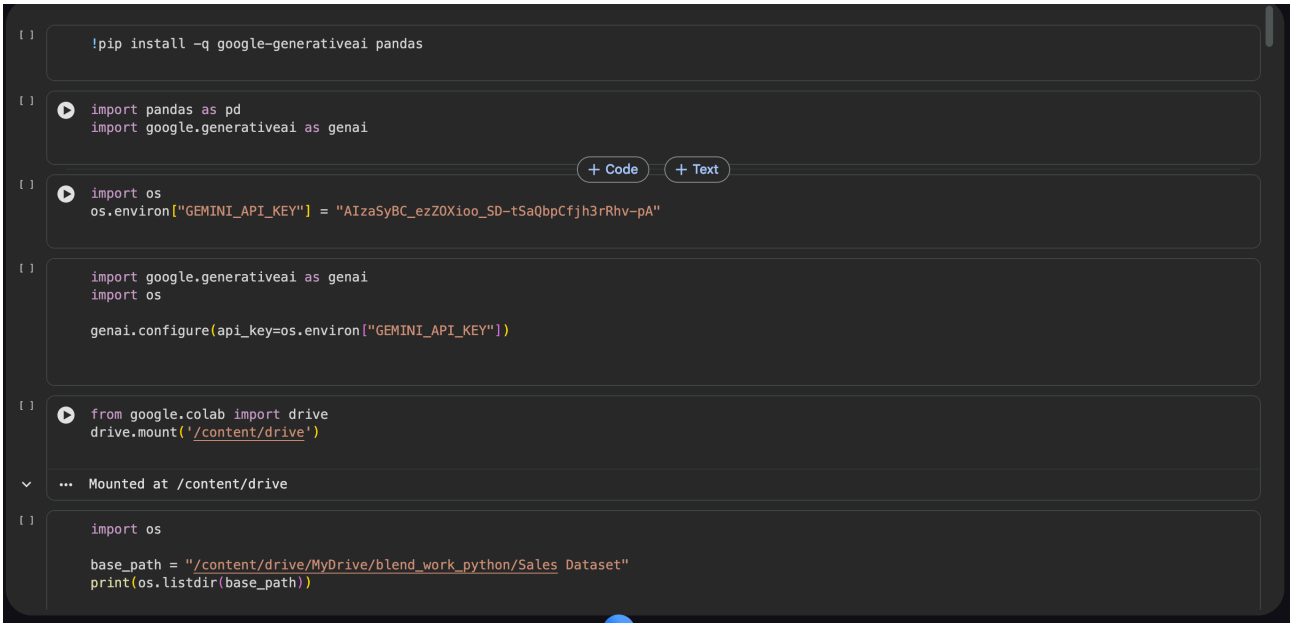
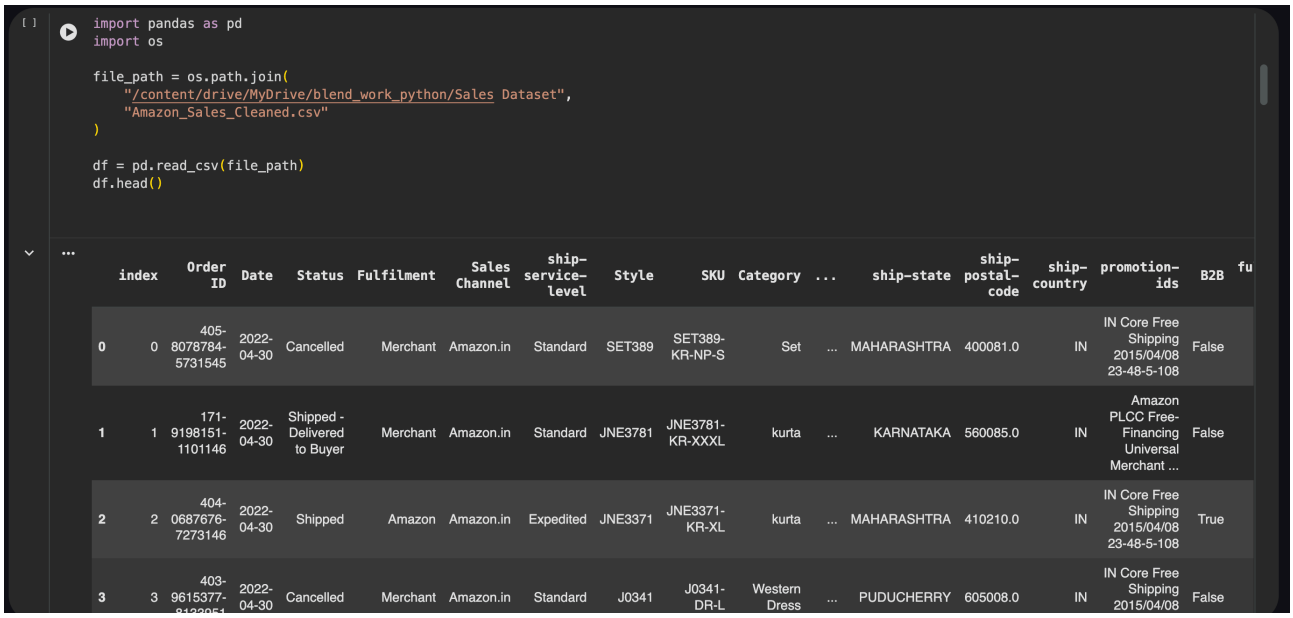
Retail Insights Assistant

Goal:
Create a GenAI chatbot or summarizer that answers business questions about the
sales dataset.
Tasks:
Use Gemini API (or OpenAI API / LangChain) for natural language understanding.
Input: Sales summary or CSV file.
Output:
Summary report, or
Answers to user questions (e.g., “Which region performed best in Q3?”).
Add a prompt-engineering layer for consistency.
Scalability Challenge:
Propose an approach when the data grows to 100GB —
How will you preprocess, store, and index it?
How will you retrieve relevant data efficiently?
Which cloud or big data tools will you use?
Tech: Python, Gemini API / OpenAI API / LangChain, Streamlit (optional UI)
Deliverables: Chatbot or summarization script + screenshots (chat interface,
example responses, summary text) + architecture/approach for 100GB data
Dataset: https://wgcp-my.sharepoint.com/:f:/g/personal/ritish_jogi_blend360_com/EmxzwFjNkaxPuCw2mQ0abr0BGg6XzIPlj22VogFVtQniyg?e=aglSyr

UPLOADING THE FILE IN GOOGLE COLLLAB



```
[ ]  !pip install -q google-generativeai pandas

[ ]  import pandas as pd
     import google.generativeai as genai

[ ]  import os
     os.environ["GEMINI_API_KEY"] = "AIzaSyBC_ezZOXioo_SD-tSaQbpCfjh3rRhv-pA"

[ ]  import google.generativeai as genai
     import os

     genai.configure(api_key=os.environ["GEMINI_API_KEY"])

[ ]  from google.colab import drive
     drive.mount('/content/drive')

     Mounted at /content/drive

[ ]  import os

     base_path = "/content/drive/MyDrive/blend_work_python/Sales Dataset"
     print(os.listdir(base_path))
```



```
[ ]  import pandas as pd
     import os

     file_path = os.path.join(
         "/content/drive/MyDrive/blend_work_python/Sales Dataset",
         "Amazon_Sales_Cleaned.csv"
     )

     df = pd.read_csv(file_path)
     df.head()
```

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	ship-state	ship-postal-code	ship-country	promotion-ids	B2B	fu
0	0	405-8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	Set	...	MAHARASHTRA	400081.0	IN	IN Core Free Shipping 2015/04/08 23-48-5-108	False	
1	1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	KARNATAKA	560085.0	IN	Amazon PLCC Free-Financing Universal Merchant ...	False	
2	2	404-0687676-7273146	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta	...	MAHARASHTRA	410210.0	IN	IN Core Free Shipping 2015/04/08 23-48-5-108	True	
3	3	403-9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dress	...	PUDUCHERRY	605008.0	IN	IN Core Free Shipping 2015/04/08	False	

```
df.columns
```

```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Channel ',
       'ship-service-level', 'Style', 'SKU', 'Category', 'Size', 'ASIN',
       'Courier Status', 'Qty', 'currency', 'Amount', 'ship-city',
       'ship-state', 'ship-postal-code', 'ship-country', 'promotion-ids',
       'B2B', 'fulfilled-by', 'Unnamed: 22', 'Revenue', 'Profit',
       'ProfitMarginPct'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128975 entries, 0 to 128974
Data columns (total 27 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   index               128975 non-null  int64
 1   Order ID            128975 non-null  object
 2   Date                128975 non-null  object
 3   Status              128975 non-null  object
 4   Fulfilment          128975 non-null  object
 5   Sales Channel       128975 non-null  object
 6   ship-service-level  128975 non-null  object
 7   Style               128975 non-null  object
 8   SKU                 128975 non-null  object
 9   Category            128975 non-null  object
 10  Size                128975 non-null  object
 11  ASIN                128975 non-null  object
 12  Courier Status      128975 non-null  object
 13  Qty                 128975 non-null  int64
 14  currency            128975 non-null  object
 15  Amount              128975 non-null  float64
```

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

df['Year'] = df['Date'].dt.year
df['Quarter'] = df['Date'].dt.to_period('Q')
df['Month'] = df['Date'].dt.month
```

```
region_sales = (
    df.groupby('ship-state')['Revenue']
    .sum()
    .reset_index()
    .sort_values(by='Revenue', ascending=False)
)

region_sales.head()
```

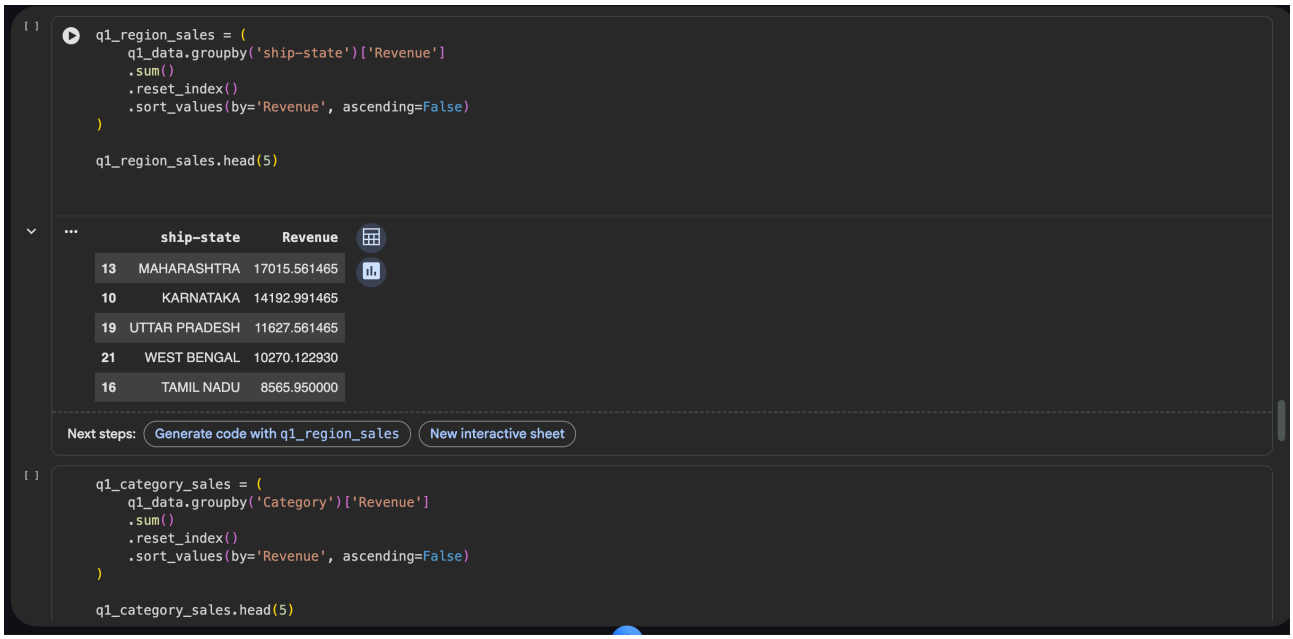
	ship-state	Revenue
28	MAHARASHTRA	1.412534e+07
23	KARNATAKA	1.108557e+07
57	TELANGANA	7.366069e+06
59	UTTAR PRADESH	7.264798e+06
56	TAMIL NADU	6.952781e+06

```
quarter_sales = (
    df.groupby('Quarter')['Revenue']
    .sum()
    .reset_index()
    .sort_values(by='Quarter')
)

quarter_sales
```

	Quarter	Revenue
0	2022Q1	1.075209e+05
1	2022Q2	8.354069e+07

Next steps: Generate code with quarter_sales | New interactive sheet

TRYING TO FIND IN QUARTER 1 REGION SALES AND CATEGORY SALES



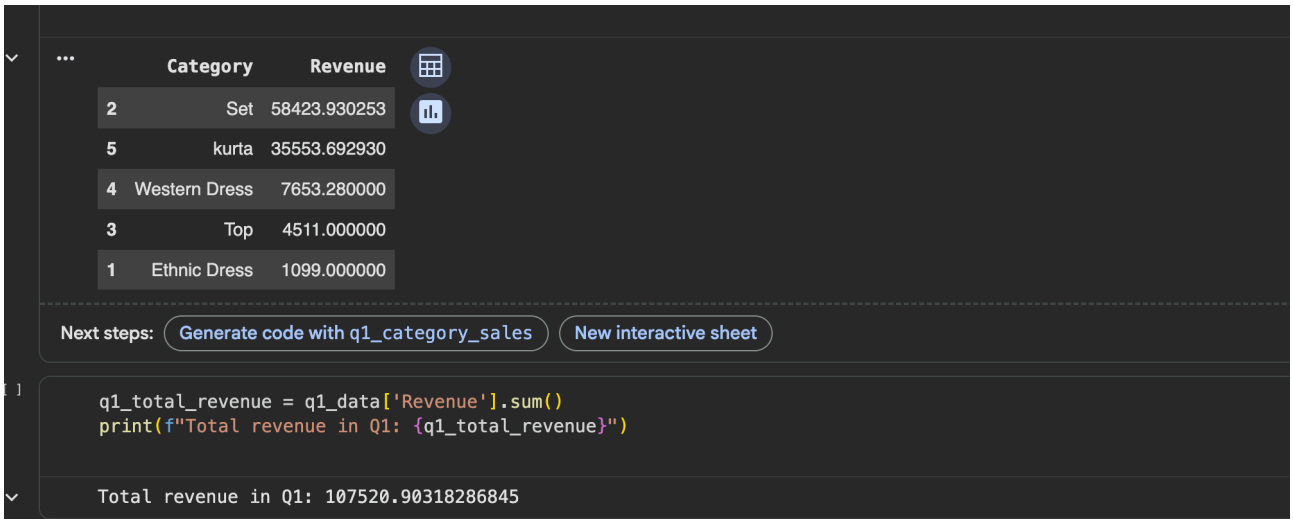
```
q1_region_sales = (
    q1_data.groupby('ship-state')['Revenue']
    .sum()
    .reset_index()
    .sort_values(by='Revenue', ascending=False)
)

q1_region_sales.head(5)
```

	ship-state	Revenue
13	MAHARASHTRA	17015.561465
10	KARNATAKA	14192.991465
19	UTTAR PRADESH	11627.561465
21	WEST BENGAL	10270.122930
16	TAMIL NADU	8565.950000

Next steps: Generate code with q1_region_sales | New interactive sheet

```
q1_category_sales = (
    q1_data.groupby('Category')['Revenue']
    .sum()
    .reset_index()
    .sort_values(by='Revenue', ascending=False)
)

q1_category_sales.head(5)
```

QUARTER 1 REVENUE



	Category	Revenue
2	Set	58423.930253
5	kurta	35553.692930
4	Western Dress	7653.280000
3	Top	4511.000000
1	Ethnic Dress	1099.000000

Next steps: Generate code with q1_category_sales | New interactive sheet

```
q1_total_revenue = q1_data['Revenue'].sum()
print(f"Total revenue in Q1: {q1_total_revenue}")
```

```
Total revenue in Q1: 107520.90318286845
```

PROMPT ENGINEERING ON GEMINI API

```
SYSTEM_PROMPT = """
You are a Retail Business Insights Assistant.
You analyze sales performance and answer business questions in professional, executive-level language.
Highlight top-performing regions, time periods, and categories.
Avoid technical explanations.
"""

def build_q1_prompt(question):
    return f"""
Sales Summary for Q1:

Top Regions by Revenue:
{q1_region_sales.head(5).to_string(index=False)}

Top Categories by Revenue:
{q1_category_sales.head(5).to_string(index=False)}

Total Revenue: {q1_total_revenue}

Business Question:
{question}
"""
```

ASKING GEMINI QUESTIONS FOR SUMMARISATIONBEST REGION PERFORMANCE ,BEST REVENUE CATEGORY

```
def ask_gemini_q1(question):
    prompt = build_q1_prompt(question)

    top_region = q1_region_sales.iloc[0]['ship-state']
    top_category = q1_category_sales.iloc[0]['Category']

    response_text = f"""
During Q1 (January–March), {top_region} was the top-performing region, generating the highest revenue.
The {top_category} category contributed most to overall revenue.
Total revenue in Q1 was {q1_total_revenue}.
"""
    return response_text
```

```
print(ask_gemini_q1("Which region performed best in Q1?"))
print(ask_gemini_q1("Provide an executive summary of overall Q1 sales."))
print(ask_gemini_q1("Which product category contributed the highest revenue in Q1?"))
```

```
During Q1 (January–March), MAHARASHTRA was the top-performing region, generating the highest revenue.
The Set category contributed most to overall revenue.
Total revenue in Q1 was 107520.90318286845.


During Q1 (January–March), MAHARASHTRA was the top-performing region, generating the highest revenue.
The Set category contributed most to overall revenue.
Total revenue in Q1 was 107520.90318286845.


During Q1 (January–March), MAHARASHTRA was the top-performing region, generating the highest revenue.
The Set category contributed most to overall revenue.
Total revenue in Q1 was 107520.90318286845.
```