```
!pip install openai
```

```
Collecting openai
  Downloading openai-1.1.1-py3-none-any.whl (217 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 217.8/217.8 kB 3.9 MB/s eta 0:00:00
Requirement already satisfied: anyio<4,>=3.5.0 in /usr/local/lib/python3.10/dist-packages (from openai) (3.7.1)
Requirement already satisfied: distro<2,>=1.7.0 in /usr/lib/python3/dist-packages (from openai) (1.7.0)
Collecting httpx<1,>=0.23.0 (from openai)
  Downloading httpx-0.25.1-py3-none-any.whl (75 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 75.0/75.0 kB 8.0 MB/s eta 0:00:00
Requirement already satisfied: pydantic<3,>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from openai) (1.10.13)
Requirement already satisfied: tqdm>4 in /usr/local/lib/python3.10/dist-packages (from openai) (4.66.1)
Requirement already satisfied: typing-extensions<5,>=4.5 in /usr/local/lib/python3.10/dist-packages (from openai) (4.5.0)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.5.0->openai) (3.4)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.5.0->openai) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.5.0->openai) (1.1.3)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.23.0->openai) (2023.7.22)
Collecting httpcore (from httpx<1,>=0.23.0->openai)
  Downloading httpcore-1.0.1-py3-none-any.whl (76 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.9/76.9 kB 8.5 MB/s eta 0:00:00
Collecting h11<0.15,>=0.13 (from httpcore->httpx<1,>=0.23.0->openai)
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 7.5 MB/s eta 0:00:00
Installing collected packages: h11, httpcore, httpx, openai
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
llmx 0.0.15a0 requires cohere, which is not installed.
llmx 0.0.15a0 requires tiktoken, which is not installed.
Successfully installed h11-0.14.0 httpcore-1.0.1 httpx-0.25.1 openai-1.1.1
```

```
!pip install -U sentence-transformers
```

```
Collecting sentence-transformers
  Downloading sentence-transformers-2.2.2.tar.gz (85 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 86.0/86.0 kB 1.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting transformers<5.0.0,>=4.6.0 (from sentence-transformers)
  Downloading transformers-4.35.0-py3-none-any.whl (7.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.9/7.9 MB 45.5 MB/s eta 0:00:00
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (4.66.1)
Requirement already satisfied: torch>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (2.1.0+cu118)
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (0.16.0+cu118)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (1.23.5)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (1.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (1.11.3)
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (3.8.1)
Collecting sentencepiece (from sentence-transformers)
  Downloading sentencepiece-0.1.99-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 52.4 MB/s eta 0:00:00
Collecting huggingface-hub>=0.4.0 (from sentence-transformers)
  Downloading huggingface_hub-0.18.0-py3-none-any.whl (301 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 302.0/302.0 kB 24.4 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.4.0->sentence-transformer
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.4.0->sentence-tra
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.4.0->sentence-transformer
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.4.0->sentence-transfor
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.4.0->se
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.4.0->sentence-trar
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.6.0->sentence-transformers) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.6.0->sentence-transformers) (3.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.6.0->sentence-transformers) (3.1.2)
Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.6.0->sentence-transformers) (2
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.6.0->sentenc
Collecting tokenizers<0.15,>=0.14 (from transformers<5.0.0,>=4.6.0->sentence-transformers)
  Downloading tokenizers-0.14.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.8/3.8 MB 101.6 MB/s eta 0:00:00
Collecting safetensors>=0.3.1 (from transformers<5.0.0,>=4.6.0->sentence-transformers)
  Downloading safetensors-0.4.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 82.7 MB/s eta 0:00:00
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk->sentence-transformers) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk->sentence-transformers) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->sentence-transform
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.10/dist-packages (from torchvision->sentence-transform
Collecting huggingface-hub>=0.4.0 (from sentence-transformers)
  Downloading huggingface_hub-0.17.3-py3-none-any.whl (295 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 295.0/295.0 kB 32.8 MB/s eta 0:00:00
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.6.0->sentence-transf
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0.4.0->senter
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0.4.0->
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0.4.0->
```

```
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.6.0->sentence-transforme
Building wheels for collected packages: sentence-transformers
  Building wheel for sentence-transformers (setup.py) ... done
  Created wheel for sentence-transformers: filename=sentence_transformers-2.2.2-py3-none-any.whl size=125923 sha256=e8bdf3213dba64b17
  Stored in directory: /root/.cache/pip/wheels/62/f2/10/1e606fd5f02395388f74e7462910fe851042f97238cbbd902f
Successfully built sentence-transformers
Installing collected packages: sentencepiece, safetensors, huggingface-hub, tokenizers, transformers, sentence-transformers
Successfully installed huggingface-hub-0.17.3 safetensors-0.4.0 sentence-transformers-2.2.2 sentencepiece-0.1.99 tokenizers-0.14.1 tr
```

```
import openai
import numpy as np
import pandas as pd
```

```
from sklearn.metrics.pairwise import cosine_similarity
```

```
api_key ="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"
openai.api_key = api_key
```

```
product_data =pd.read_excel('/content/Dataset.xlsx')
product_data_df=pd.DataFrame(product_data)
```

```
product_data_df.head()
```

	Company_Name	Product_Name	Color	Rear_Camera	Front_Camera	Space	RAM	CPU_Speed	Price
0	Samsung	Galaxy S23	Black	16	12	128	8	3.2	999
1	Samsung	Galaxy S23	Silver	16	12	128	8	3.2	999
2	Samsung	Galaxy S23	Blue	16	12	128	8	3.2	999
3	Samsung	Galaxy S23	Black	16	12	256	12	3.2	1199
4	Samsung	Galaxy S23	Silver	16	12	256	12	3.2	1199

```
product_data_df['features'] = product_data_df.apply(
   lambda row: f"{row['Rear_Camera']},{row['Front_Camera']},{row['Space']},{row['RAM']},{row['CPU_Speed']},{row['Price']}",
    axis=1
)
product_data_df.head()
```

	Company_Name	Product_Name	Color	Rear_Camera	Front_Camera	Space	RAM	CPU_Speed	Price	features
0	Samsung	Galaxy S23	Black	16	12	128	8	3.2	999	16,12,128,8,3.2,999
1	Samsung	Galaxy S23	Silver	16	12	128	8	3.2	999	16,12,128,8,3.2,999
2	Samsung	Galaxy S23	Blue	16	12	128	8	3.2	999	16,12,128,8,3.2,999
3	Samsung	Galaxy S23	Black	16	12	256	12	3.2	1199	16,12,256,12,3.2,1199
4	Samsung	Galaxy S23	Silver	16	12	256	12	3.2	1199	16,12,256,12,3.2,1199

```
# Initialize user_query_vector with default values (0 for unspecified specs)
user_query_vector = [0, 0, 0, 0, 0, 0]

print("Enter Following Details of mobile [Enter 0 if specs not decided]")
print("Rear Camera Specification :")
rear = int(input())
user_query_vector[0] = rear

print("Front Camera Specification :")
front = int(input())
user_query_vector[1] = front

print("Space :")
space = int(input())
user_query_vector[2] = space

print("RAM :")
ram = int(input())
user_query_vector[3] = ram
```

```
print("CPU Speed :")
cpu = float(input())
user_query_vector[4] = cpu

print("Price :")
price = int(input())
user_query_vector[5] = price

print("User Query Vector:", user_query_vector)
4
```

```
 Enter Following Details of mobile [Enter 0 if specs not decided]
 Rear Camera Specification :
 48
 Front Camera Specification :
 12
 Space :
 128
 RAM :
 6
 CPU Speed :
 3
 Price :
 399
 User Query Vector: [48, 12, 128, 6, 3.0, 399]
 4
```

```
# input_data = [rear,front,space,ram,cpu,price]
# # changing the input_data to numpy array
# input_data_as_numpy_array = np.asarray(input_data)

# # reshape the array as we are predicting for one instance
# user_query_vector = input_data_as_numpy_array.reshape(1,-1)


# Compute the cosine similarity matrix
cosine_sim = cosine_similarity(product_data_df[['Rear_Camera', 'Front_Camera', 'Space', 'RAM', 'CPU_Speed', 'Price']])

def recommend_products(user_query_vector, product_name, cosine_sim_matrix, data, top_n=5):
    # Vectorize the user query, including the product name
   # user_query_vector = [48,20,128,6,0,499]  # Replace with the actual user query vector



    # Calculate the cosine similarity between the user query vector and all product feature vectors
    similarity_scores = cosine_similarity([user_query_vector], product_data_df[['Rear_Camera', 'Front_Camera', 'Space', 'RAM', 'CPU_Speed', '

    # Get the indices of the top N product recommendations
    product_indices = similarity_scores.argsort()[0][-top_n:][::-1]

    # Retrieve the product names and features associated with the recommended product indices
    recommended_products = []
    for i in product_indices:
        product_info = {
            'Name': data.loc[i, 'Product_Name'],
            'Rear Camera': data.loc[i, 'Rear_Camera'],
            'Front Camera': data.loc[i, 'Front_Camera'],
            'Space': data.loc[i, 'Space'],
            'RAM': data.loc[i, 'RAM'],
            'CPU Speed': data.loc[i, 'CPU_Speed'],
            'Price': data.loc[i, 'Price']
        }
        recommended_products.append(product_info)

    return recommended_products

def generate_chatbot_response(user_query, recommended_products):
    chatbot_response = f"Based on your query '{user_query}', we recommend the following products:\n"

    for i, product_info in enumerate(recommended_products, 1):
        chatbot_response += f"{i}. {product_info['Name']} - "
        chatbot_response += f"Rear Camera: {product_info['Rear Camera']}, "
        chatbot_response += f"Front Camera: {product_info['Front Camera']}, "
        chatbot_response += f"Space: {product_info['Space']}, "
        chatbot_response += f"RAM: {product_info['RAM']}, "
        chatbot_response += f"CPU Speed: {product_info['CPU Speed']}, "
        chatbot_response += f"Price: {product_info['Price']}\n"
```

```
    return chatbot_response

# User's query with potential null values
user_query = "I want a smartphone with a good camera."
product_name = "LG"  # Replace with the specific product name

# Get the top 5 recommended products along with their features
top_5_recommended_products = recommend_products(user_query_vector, product_name, cosine_sim, product_data_df, top_n=6)

# Generate a chatbot response
chatbot_response = generate_chatbot_response(user_query, top_5_recommended_products)


print(chatbot_response)
```

```
Based on your query 'I want a smartphone with a good camera.', we recommend the following products:
1. Galaxy A32 - Rear Camera: 48, Front Camera: 16, Space: 128, RAM: 8, CPU Speed: 2.2, Price: 399
2. Y73 - Rear Camera: 48, Front Camera: 16, Space: 128, RAM: 8, CPU Speed: 2.2, Price: 399
3. F19 Pro - Rear Camera: 48, Front Camera: 16, Space: 128, RAM: 8, CPU Speed: 2.2, Price: 399
4. Galaxy A32 - Rear Camera: 48, Front Camera: 16, Space: 128, RAM: 8, CPU Speed: 2.2, Price: 399
5. F19 Pro - Rear Camera: 48, Front Camera: 16, Space: 128, RAM: 8, CPU Speed: 2.2, Price: 399
6. Galaxy A32 - Rear Camera: 48, Front Camera: 16, Space: 128, RAM: 8, CPU Speed: 2.2, Price: 399
```