PG6: Use a pre-trained Hugging Face model to analyze sentiment in text. Assume a real-world application, Load the sentiment analysis pipeline. Analyze the sentiment by giving sentences to input.

Soln:

```
%pip install --upgrade --quiet huggingface_hub

%pip install --upgrade langchain

from transformers import pipeline

# Load the sentiment analysis pipeline

sentiment_analyzer = pipeline("sentiment-analysis")

# Example sentences for analysis

sentences = [

    "The product quality is amazing! I'm very satisfied.",

    "I had a terrible experience with customer service.",

    "The delivery was quick, but the packaging was damaged.",

    "Absolutely love this! Best purchase I've made.",

    "Not worth the money, very disappointed."

]

# Analyze sentiment for each sentence

results = sentiment_analyzer(sentences)

# Print the results

for sentence, result in zip(sentences, results):

    print(f"Sentence: {sentence}\nSentiment: {result['label']}, Confidence: {result['score']:.2f}\n")
```

Output:

```
Sentence: The product quality is amazing! I'm very satisfied.
Sentiment: POSITIVE, Confidence: 1.00

Sentence: I had a terrible experience with customer service.
Sentiment: NEGATIVE, Confidence: 1.00

Sentence: The delivery was quick, but the packaging was damaged.
Sentiment: NEGATIVE, Confidence: 1.00
```

```
Sentence: Absolutely love this! Best purchase I've made.
Sentiment: POSITIVE, Confidence: 1.00

Sentence: Not worth the money, very disappointed.
Sentiment: NEGATIVE, Confidence: 1.00
```

Results

Output:

```
[{'label': 'POSITIVE', 'score': 0.9998825788497925},
 {'label': 'NEGATIVE', 'score': 0.9993104934692383},
 {'label': 'NEGATIVE', 'score': 0.9997345805168152},
 {'label': 'POSITIVE', 'score': 0.9998751878738403},
 {'label': 'NEGATIVE', 'score': 0.9998034834861755}]
```

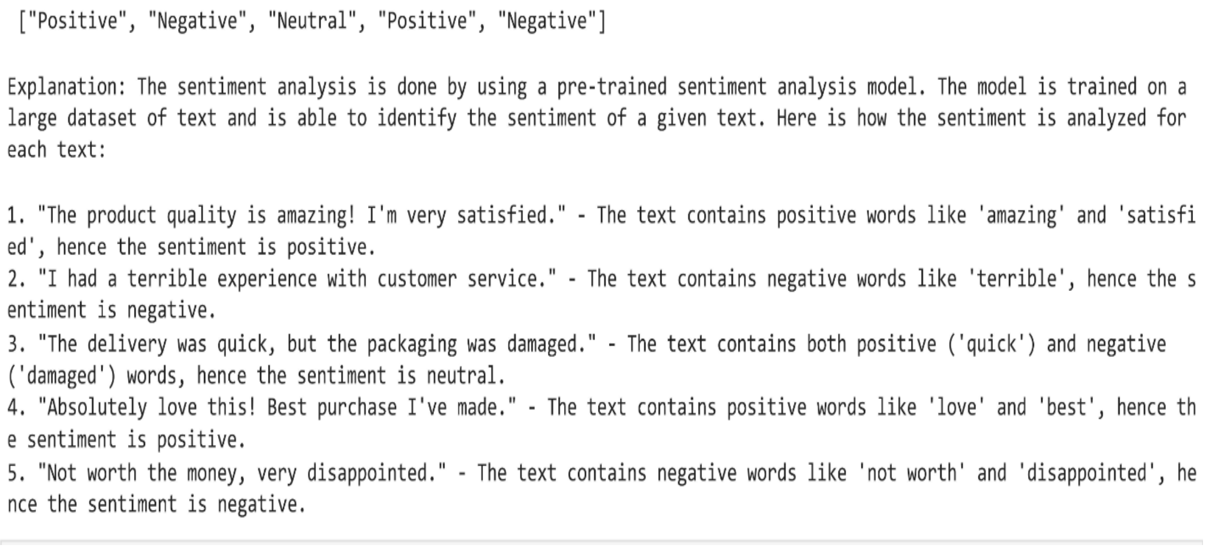
```
!pip install langchain-huggingface
```

Approach 2: Using API calls

```
from langchain_huggingface import HuggingFaceEndpoint

# get a token: https://huggingface.co/docs/api-inference/quicktour#get-your-api-token

from getpass import getpass

HUGGINGFACEHUB_API_TOKEN = getpass()

import os

os.environ["HUGGINGFACEHUB_API_TOKEN"] = HUGGINGFACEHUB_API_TOKEN

from langchain.chains import LLMChain

from langchain_core.prompts import PromptTemplate

text = ["The product quality is amazing! I'm very satisfied.",

    "I had a terrible experience with customer service.",

    "The delivery was quick, but the packaging was damaged.",

    "Absolutely love this! Best purchase I've made.",

    "Not worth the money, very disappointed."]

template = """Perform the sentiment analysis for the following:{text}.

Answer: Following is the sentiment for the given text:"""

prompt = PromptTemplate.from_template(template)
```

```
repo_id = "meta-llama/Llama-3.2-3B-Instruct" #"mistralai/Mistral-7B-Instruct-v0.2"
llm = HuggingFaceEndpoint(
    repo_id=repo_id,
    max_length=256,
    temperature=0.5,
    huggingfacehub_api_token=HUGGINGFACEHUB_API_TOKEN,
)
llm_chain = prompt | llm
print(llm_chain.invoke({"text": text}))
```

Output:



```
 ["Positive", "Negative", "Neutral", "Positive", "Negative"]

Explanation: The sentiment analysis is done by using a pre-trained sentiment analysis model. The model is trained on a
large dataset of text and is able to identify the sentiment of a given text. Here is how the sentiment is analyzed for
each text:

1. "The product quality is amazing! I'm very satisfied." - The text contains positive words like 'amazing' and 'satisfi
ed', hence the sentiment is positive.
2. "I had a terrible experience with customer service." - The text contains negative words like 'terrible', hence the s
entiment is negative.
3. "The delivery was quick, but the packaging was damaged." - The text contains both positive ('quick') and negative
('damaged') words, hence the sentiment is neutral.
4. "Absolutely love this! Best purchase I've made." - The text contains positive words like 'love' and 'best', hence th
e sentiment is positive.
5. "Not worth the money, very disappointed." - The text contains negative words like 'not worth' and 'disappointed', he
nce the sentiment is negative.
```