Login Huggingface

```
!huggingface-cli login
```

```
  _|    _|  _|      _|    _|_|_|    _|_|_|   _|_|_|   _|        _|    _|_|_|     _|_|_|_|   _|_|_|   _|_|_|  _|_|_|_|
  _|    _|  _|      _|  _|        _|        _|    _|  _|        _|  _|        _|    _|    _|  _|  _|        _|
  _|_|_|_|  _|      _|  _|  _|_|  _|  _|_|  _|    _|  _|        _|  _|  _|_|  _|_|_|  _|_|_|_|  _|        _|_|_|
  _|    _|  _|      _|  _|    _|  _|    _|  _|    _|  _|        _|  _|    _|  _|    _|    _|  _|  _|        _|
  _|    _|    _|_|_|      _|_|_|    _|_|_|   _|_|_|   _|_|_|_|  _|    _|_|_|     _|      _|    _|  _|_|_|  _|_|_|_|
```

```
        To login, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
    Token:
    Add token as git credential? (Y/n) n
    Token is valid (permission: write).
    Your token has been saved to /root/.cache/huggingface/token
    Login successful
```

```
!pip install accelerate
!pip install accelerate --progress-bar off
!pip install -qqq langchain==0.0.299 --progress-bar off
```

```
    Collecting accelerate
      Downloading accelerate-0.26.1-py3-none-any.whl (270 kB)
                                                 270.9/270.9 kB 5.9 MB/s eta 0:00:00
    Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from accelerate) (1.23.5)
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (23.2)
    Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate) (5.9.5)
    Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (from accelerate) (6.0.1)
    Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (2.1.0+cu121)
    Requirement already satisfied: huggingface-hub in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.20.3)
    Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.4.2)
    Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.13.1)
    Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate)
    Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (1.12)
    Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.2.1)
    Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.1.3)
    Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2023.6.0)
    Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2.
    Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (2.31.
    Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (4
    Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.10.0->accel
    Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingfa
    Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->acce
    Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub
    Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.10.0->accelerat
    Installing collected packages: accelerate
    Successfully installed accelerate-0.26.1
    Requirement already satisfied: accelerate in /usr/local/lib/python3.10/dist-packages (0.26.1)
    Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from accelerate) (1.23.5)
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (23.2)
    Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate) (5.9.5)
    Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (from accelerate) (6.0.1)
    Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (2.1.0+cu121)
    Requirement already satisfied: huggingface-hub in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.20.3)
    Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.4.2)
    Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.13.1)
    Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate)
    Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (1.12)
    Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.2.1)
    Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.1.3)
    Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2023.6.0)
    Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2.
    Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (2.31.
    Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (4
    Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.10.0->accel
    Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingfa
    Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->acce
    Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub
    Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.10.0->accelerat
```

```python
from transformers import AutoTokenizer, AutoConfig, AutoModelForCausalLM
import transformers
import torch
from langchain import HuggingFacePipeline
from langchain import PromptTemplate
from langchain.chains import SimpleSequentialChain
from langchain.chains import LLMChain
import pandas as pd


model = "meta-llama/Llama-2-7b-chat-hf"

tokenizer = AutoTokenizer.from_pretrained(model)
pipeline = transformers.pipeline(
    "text-generation",
    model=model,
    torch_dtype=torch.float16,
    device_map="auto",
)

# sequences = pipeline(
#     'I liked "Breaking Bad" and "Band of Brothers". Do you have any recommendations of other shows I might like?\n',
#     do_sample=True,
#     top_k=10,
#     num_return_sequences=1,
#     eos_token_id=tokenizer.eos_token_id,
#     max_length=200,
# )
# for seq in sequences:
#     print(f"Result: {seq['generated_text']}")
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: User‌
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access publ
  warnings.warn(
```

tokenizer_config.json:	1.62k/1.62k [00:00<00:00,
100%	86.7kB/s]
tokenizer.model: 100%	500k/500k [00:00<00:00, 11.0MB/s]
tokenizer.json: 100%	1.84M/1.84M [00:00<00:00, 7.26MB/s]
special_tokens_map.json:	414/414 [00:00<00:00,
100%	21.8kB/s]
config.json: 100%	614/614 [00:00<00:00, 27.1kB/s]
model.safetensors.index.json:	26.8k/26.8k [00:00<00:00,
100%	1.45MB/s]
Downloading shards: 100%	2/2 [01:49<00:00, 50.86s/it]
model-00001-of-	9.98G/9.98G [01:17<00:00,
00002.safetensors: 100%	37.1MB/s]
model-00002-of-	3.50G/3.50G [00:32<00:00,

```python
prompt = '''
<s>[INST] <<SYS>>
Your task is to analyze customer reviews from the scotiabank application with the following steps:

Summarize the Review: Provide a literal summary of the review, focusing only on the exact words and phrases used.
Categorize the Review: Based on the literal content of the review, go through the list of categories, and if the  fits the one c
If all of the review's noun is not related to the bank's application's favorite list in 20 chategories below, label it as "None."
Descriptions of each category are provided below.

1. 2SV: Reviews mentioning two-step verification(or "two-step verification," "two-factor authentication," "2FA," "2SV","security
2. Application Performance: Reviews discussing the app's speed, responsiveness, and presence or absence of lags or delays.
3. Accessibility: Reviews addressing the app's design for users with disabilities, including issues related to vision, hearing, c
4. Appointment Booking: Reviews about the feature allowing users to book appointments with advisors through the app.
5. Biometric Login: Reviews related to logging into the app using biometric features like fingerprint authentication.
6. Budgeting: Reviews discussing features that track spending and income, possibly mentioning spending graphs.
7. Chat: Reviews about experiences with chatting, either with a chatbot or a live person.
8. Cheque Deposit: Reviews about the feature for depositing cheques using the phone.
9. Credit Score: Reviews related to the app's feature showing the user's credit score.
10. Email Money Transfer: Reviews about transferring funds between accounts using email.
11. Errors: Reviews mentioning any errors, flaws, crashes, or glitches in the app.
```

12. Fee: Reviews discussing banking fees and their impact on the user experience.
13. Info Alerts: Reviews about notifications alerting users to account activities.
14. International Money Movement: Reviews related to transferring money internationally.
15. Investments: Reviews discussing investment accounts and options.
16. Login and Logout Issues: Reviews about problems encountered while logging in or out of the app.
17. Quick Balance: Reviews related to checking account balances without full sign-in.
18. Request New Card: Reviews about requesting replacement cards through the app.
19. Rewards: Reviews discussing the app's rewards or loyalty programs.
20. Save and Share Statements: Reviews about saving or sharing transaction statements from the app.

Ensure your analysis is strictly based on the review's explicit content without inferring additional context or meaning.<</SYS>>

Except for the fact that their accounts costs are too high the Scotia Bank is not so bad to deal with. Thier online banking works
[/INST]

– Literal Summary: The customer expresses dissatisfaction with the high costs associated with their accounts at Scotia Bank. Howe
– Category: 12
    – Reason: The mention of "their accounts costs are too high" suggests a concern related to Fees.
– Category: 2
    – Reason: The statement "Their online banking works very well" indicates a positive experience with Application Performance.
</s>
<s>[INST]Out of nowhere, every single time that I login to do my banking the app asks a security question. Extremely annoying and

– Literal Summary: The customer is expressing frustration with the app's security feature, specifically the frequent and repetiti
– Category: 16
    – Reason: The issue described relates to the app's security features, particularly the repeated asking of a security question
</s>
<s>[INST]Solid app, but it is constantly trying to upsell. Banks will be banks, I suppose. [/INST]

– Literal Summary: The customer acknowledges the app's reliability ("Solid app") but expresses dissatisfaction with its persisten
– Category: 2
    – Reason: The comment "Solid app" suggests a positive view on the app's overall quality or functionality, which can be catego
</s>
<s>[INST]Very stable and solid app. Don't know why so many people seem to be having trouble.[/INST]

– Literal Summary: The reviewer finds the app to be stable and solid. They are unsure why others are experiencing trouble.
– Category: 2
    – Reason: The terms "stable" and "solid" directly relate to the performance of the application in terms of its functionality
</s>
<s>[INST]Awesome app very easy to use[/INST]

– Literal Summary: The reviewer describes the app as "Awesome" and mentions that it is very easy to use.
– Category: 2
    – Reason: The review focuses on the positive aspect of the app being easy to use, which is related to the app's performance i
</s>
<s>[INST]Clean, priced right, staff is very accommodating & helpful.Hotel breakfast service is better than most. Beautiful treed

'''

```
%%time
sequences = pipeline(
    prompt,
    do_sample=True,
    top_k=10,
    num_return_sequences=1,
    eos_token_id=tokenizer.eos_token_id,
    max_length=2000,
)
```

    CPU times: user 9.57 s, sys: 336 ms, total: 9.91 s
    Wall time: 10 s

```
print(sequences[0]['generated_text'])
```

    <s>[INST] <<SYS>>
    Your task is to analyze customer reviews from the scotiabank application with the following steps:

    Summarize the Review: Provide a literal summary of the review, focusing only on the exact words and phrases used.
    Categorize the Review: Based on the literal content of the review, go through the list of categories, and if the fits the o
    If all of the review's noun is not related to the bank's application's favorite list in 20 chategories below, label it as "N
    Descriptions of each category are provided below.

    1. 2SV: Reviews mentioning two-step verification(or "two-step verification," "two-factor authentication," "2FA," "2SV","secu
    2. Application Performance: Reviews discussing the app's speed, responsiveness, and presence or absence of lags or delays.
    3. Accessibility: Reviews addressing the app's design for users with disabilities, including issues related to vision, heari

4. Appointment Booking: Reviews about the feature allowing users to book appointments with advisors through the app.
5. Biometric Login: Reviews related to logging into the app using biometric features like fingerprint authentication.
6. Budgeting: Reviews discussing features that track spending and income, possibly mentioning spending graphs.
7. Chat: Reviews about experiences with chatting, either with a chatbot or a live person.
8. Cheque Deposit: Reviews about the feature for depositing cheques using the phone.
9. Credit Score: Reviews related to the app's feature showing the user's credit score.
10. Email Money Transfer: Reviews about transferring funds between accounts using email.
11. Errors: Reviews mentioning any errors, flaws, crashes, or glitches in the app.
12. Fee: Reviews discussing banking fees and their impact on the user experience.
13. Info Alerts: Reviews about notifications alerting users to account activities.
14. International Money Movement: Reviews related to transferring money internationally.
15. Investments: Reviews discussing investment accounts and options.
16. Login and Logout Issues: Reviews about problems encountered while logging in or out of the app.
17. Quick Balance: Reviews related to checking account balances without full sign-in.
18. Request New Card: Reviews about requesting replacement cards through the app.
19. Rewards: Reviews discussing the app's rewards or loyalty programs.
20. Save and Share Statements: Reviews about saving or sharing transaction statements from the app.

Ensure your analysis is strictly based on the review's explicit content without inferring additional context or meaning.<</S

Except for the fact that their accounts costs are too high the Scotia Bank is not so bad to deal with. Thier online banking
[/INST]

– Literal Summary: The customer expresses dissatisfaction with the high costs associated with their accounts at Scotia Bank.
– Category: 12
    – Reason: The mention of "their accounts costs are too high" suggests a concern related to Fees.
– Category: 2
    – Reason: The statement "Their online banking works very well" indicates a positive experience with Application Performa
</s>
<s>[INST]Out of nowhere, every single time that I login to do my banking the app asks a security question. Extremely annoyin

– Literal Summary: The customer is expressing frustration with the app's security feature, specifically the frequent and rep
– Category: 16
    – Reason: The issue described relates to the app's security features, particularly the repeated asking of a security que
</s>
<s>[INST]Solid app, but it is constantly trying to upsell. Banks will be banks, I suppose. [/INST]

– Literal Summary: The customer acknowledges the app's reliability ("Solid app") but expresses dissatisfaction with its pers
– Category: 2
    – Reason: The comment "Solid app" suggests a positive view on the app's overall quality or functionality, which can be c
</s>
<s>[INST]Very stable and solid app. Don't know why so many people seem to be having trouble.[/INST]

– Literal Summary: The reviewer finds the app to be stable and solid. They are unsure why others are experiencing trouble.
– Category: 2
    – Reason: The terms "stable" and "solid" directly relate to the performance of the application in terms of its functiona

```python
answer = []
df = pd.read_excel('/content/Winter 2024 Scotia DSD Data Set.xlsx')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-15-03ceca9ee252> in <cell line: 2>()
      1 answer = []
----> 2 df = pd.read_excel('/content/Winter 2024 Scotia DSD Data Set.xlsx')
```

```
                          ▲ 5 frames ▼
/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in
get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,
errors, storage_options)
    863         else:
    864             # Binary mode
--> 865             handle = open(handle, ioargs.mode)
    866         handles.append(handle)
    867
```

```
FileNotFoundError: [Errno 2] No such file or directory: '/content/Winter 2024
Scotia DSD Data Set.xlsx'
```

```
def get_answer(review):
  prompt = f"""
    <s>[INST] <<SYS>>
    You are an advanced AI trained to categorize customer feedback for Scotiabank. When presented with a customer review, your t


    Catogories are selected from here, description are provided:
    - 2SV: Security measure involving a numerical code sent to the customer's phone.
    - Application Performance: How well the app performs in terms of speed and responsiveness.
    - Accessibility: The app's design for people with disabilities and related feedback.
    - Appointment Booking: Experiences with requesting advisor appointments through the app.
    - Biometric Login: User experience with biometric authentication for app login.
    - Budgeting: Tracking of spending and income, and related features within the app.
    - Chat: Discussions on using chatbot services or live chat options.
    - Cheque Deposit: Experiences with the app's feature for cheque deposit.
    - Credit Score: Interactions with the app's credit score feature.
    - Email Money Transfer: Using the app to transfer funds via email.
    - Errors: Any errors, glitches, or faults encountered within the app.
    - Fee: Feedback regarding banking fees and their impact on customer experience.
    - Info Alerts: Notifications about account activities.
    - International Money Movement: Transferring money internationally using the app.
    - Investments: Discussions on investment accounts like GICs, RRSPs, and other options.
    - Login and Logout Issues: Issues experienced during the login or logout process.
    - Quick Balance: Experiences with the quick balance check feature.
    - Request New Card: Processes for requesting a replacement card.
    - Rewards: Customer experience with the app's rewards and loyalty programs.
    - Save and Share Statements: Feedback on saving and sharing statements from the app.

    Your categories must come from one of the above. Your output should consist solely of the category names that are relevant t
    <</SYS>>
      [/INST]

    [INST]Can you analyze the following review commented on Jan 24, 2024?
    {review} [/INST]
    """
  return prompt


prompt = get_answer("2 step verification never worked need to call to remove it then they again put it on on never worked for me
sequences = pipeline(
    prompt,
    do_sample=True,
    top_k=10,
    num_return_sequences=1,
    eos_token_id=tokenizer.eos_token_id,
    max_length=1000,
)


print(sequences[0]['generated_text'])


        <s>[INST] <<SYS>>
        You are an advanced AI trained to categorize customer feedback for Scotiabank. When presented with a customer review, yo


        Catogories are selected from here, description are provided:
        - 2SV: Security measure involving a numerical code sent to the customer's phone.
        - Application Performance: How well the app performs in terms of speed and responsiveness.
        - Accessibility: The app's design for people with disabilities and related feedback.
        - Appointment Booking: Experiences with requesting advisor appointments through the app.
        - Biometric Login: User experience with biometric authentication for app login.
        - Budgeting: Tracking of spending and income, and related features within the app.
        - Chat: Discussions on using chatbot services or live chat options.
        - Cheque Deposit: Experiences with the app's feature for cheque deposit.
        - Credit Score: Interactions with the app's credit score feature.
        - Email Money Transfer: Using the app to transfer funds via email.
        - Errors: Any errors, glitches, or faults encountered within the app.
        - Fee: Feedback regarding banking fees and their impact on customer experience.
        - Info Alerts: Notifications about account activities.
        - International Money Movement: Transferring money internationally using the app.
        - Investments: Discussions on investment accounts like GICs, RRSPs, and other options.
        - Login and Logout Issues: Issues experienced during the login or logout process.
        - Quick Balance: Experiences with the quick balance check feature.
        - Request New Card: Processes for requesting a replacement card.
        - Rewards: Customer experience with the app's rewards and loyalty programs.
        - Save and Share Statements: Feedback on saving and sharing statements from the app.

        Your categories must come from one of the above. Your output should consist solely of the category names that are releva
        <</SYS>>
```

```
      [/INST]

      [INST]Can you analyze the following review commented on Jan 24, 2024?
      2 step verification never worked need to call to remove it then they again put it on on never worked for me going to swi

      Based on the review provided, the following categories apply:
```

* 2SV: Security measure involving a numerical code sent to the customer's phone.
* Switching to another bank: Feedback regarding the customer's intention to switch to another bank due to issues with 2SV.

```python
from transformers import pipeline as hf_pipeline
from collections import Counter

# Load your model and tokenizer
# ... (your model loading code here)

# Create a pipeline instance
pipeline = hf_pipeline("text-generation", model=model, tokenizer=tokenizer)

# Assuming you have a list of reviews
reviews = ["Review 1 text", "Review 2 text", ...]

# Counter to keep track of categories
category_counts = Counter()

# Function to extract the category from the response
def extract_category(sequence):
    # Your logic to extract the category
    # Example: return sequence[0]['generated_text'].split('\n')[0]
    pass

# Function to create the prompt
def get_answer(review):
    # Your logic to create the prompt
    # Example: return f"Review: {review}"
    pass

for review in reviews:
    prompt = get_answer(review)  # Function to create the prompt
    sequence = pipeline(prompt, return_full_text=False)  # Your pipeline call
```

```
      ---------------------------------------------------------------------------
      TypeError                                 Traceback (most recent call last)
      <ipython-input-28-6946c48dca98> in <cell line: 9>()
            9 for review in reviews:
           10     prompt = get_answer(review)  # Your function to create the prompt
      ---> 11     sequence = pipeline(prompt, ...)  # Your pipeline call
           12     categorized_response = extract_category(sequence)  # Extract the
      category from the response
           13     category_counts[categorized_response] += 1

      TypeError: TextGenerationPipeline.__call__() takes 2 positional arguments but 3
      were given
```