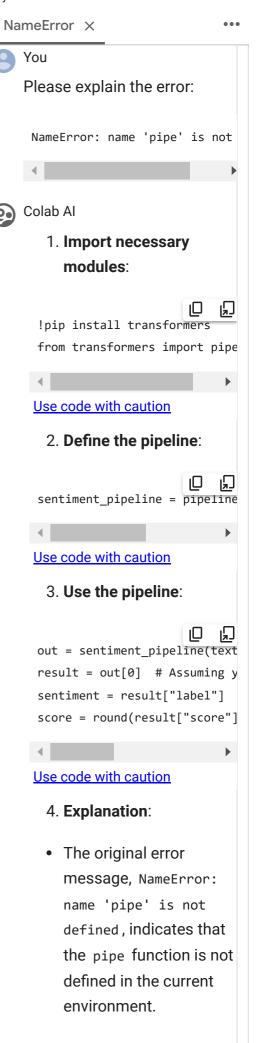
!pip install transformers

```
equirement already satisfied: transformers in /usr
equirement already satisfied: filelock in /usr/loc
equirement already satisfied: huggingface-hub<1.0,
equirement already satisfied: numpy>=1.17 in /usr/
equirement already satisfied: packaging>=20.0 in /
equirement already satisfied: pyyaml>=5.1 in /usr/
equirement already satisfied: regex!=2019.12.17 in
equirement already satisfied: requests in /usr/loc
equirement already satisfied: tokenizers<0.19,>=0.
equirement already satisfied: safetensors>=0.4.1 i
equirement already satisfied: tqdm>=4.27 in /usr/l
equirement already satisfied: fsspec>=2023.5.0 in
equirement already satisfied: typing-extensions>=3
equirement already satisfied: charset-normalizer<4
equirement already satisfied: idna<4,>=2.5 in /usr
equirement already satisfied: urllib3<3,>=1.21.1 i
equirement already satisfied: certifi>=2017.4.17 i
```

from transformers import pipeline, AutoModelForSequence

model_name = "nlptown/bert-base-multilingual-uncased-se
model = AutoModelForSequenceClassification.from_pretrai
tokenizer = AutoTokenizer.from_pretrained(model_name)
pipe = pipeline('sentiment-analysis')



/usr/local/lib/python3.10/dist-packages/huggingfac The secret `HF_TOKEN` does not exist in your Colat To authenticate with the Hugging Face Hub, create You will be able to reuse this secret in all of you Please note that authentication is recommended but warnings.warn(config.json: 100% 953/953 [00:00<00:00, 55.6kB/s] pytorch_model.bin: 100% 669M/669M [00:08<00:00, 75.7MB/s /usr/local/lib/python3.10/dist-packages/torch/ uti return self.fget.__get__(instance, owner)() tokenizer_config.json: 100% 39.0/39.0 [00:00<00:00, 486B/s] vocab.txt: 100% 872k/872k [00:00<00:00, 2.74MB/s] special tokens map.json: 100% 112/112 [00:00<00:00, 1.37kl No model was supplied, defaulted to distilbert-bas Using a pipeline without specifying a model name a config.json: 100% 629/629 [00:00<00:00, 7.27kB/s] model.safetensors: 100% 268M/268M [00:04<00:00, 78.0MB/s import random import pandas as pd def select_random_text(csv_file): # Read the CSV file into a pandas DataFrame df = pd.read_csv(csv_file) # Select a random index random_index = random.randint(0, len(df) - 1) # Get the random text from the 'text' column random_text = df.loc[random_index, 'text'] return random text # Example usage: random text = select random text('/content/rnn analysi print(random_text) text1=random_text very good data=pd.read_csv("/content/rnn_analysis.csv") data

- To fix this, we need to import the pipeline function from the transformers module.
- Once the pipeline
 function is imported, we
 can create a
 sentiment_pipeline
 object using the
 pipeline function and
 the desired task name,
 which is sentiment analysis in this case.
- Finally, we can use the sentiment_pipeline object to analyze the sentiment of the text in text1.

Rate this answer $\ \ \Box$

W.	(vvni			Trininpyrib C	A
14	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	5	classy pi	
15	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	5	mus	
16	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	5	fab	
17	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	5	worth I	
18	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	3	;	
19	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	5	great pi	
20	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	5	awe	
21	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	5	worth 	
22	Candes 60 L Room/Personal Air Cooler?????? (Whi	8999	2	just	
23	Candes 60 L Room/Personal Air	8999	5	awe	•