```
!pip install transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.37.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.19.3 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.20.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (23.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2023.12.25)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.19,>=0.14 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.15.2)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.2)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.19.3->tran
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.2.
```

```
from transformers import pipeline, AutoModelForSequenceClassification, AutoTokenizer
```

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

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
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 (https://huggingface.co/settings/tokens), set it as
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json:  100%                                  953/953  [00:00<00:00,  64.3kB/s]
pytorch_model.bin:  100%                            669M/669M  [00:08<00:00,  117MB/s]
/usr/local/lib/python3.10/dist-packages/torch/_utils.py:831: UserWarning: TypedStorage is deprecated. It will be removed in the fut
  return self.fget.__get__(instance, owner)()
tokenizer_config.json:  100%                        39.0/39.0  [00:00<00:00,  613B/s]
vocab.txt:  100%                                    872k/872k  [00:00<00:00,  7.10MB/s]
special_tokens_map.json:  100%                      112/112  [00:00<00:00,  3.91kB/s]
No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b (https://huggingface.co/di
Using a pipeline without specifying a model name and revision in production is not recommended.
config.json:  100%                                  629/629  [00:00<00:00,  13.6kB/s]
model.safetensors:  100%                            268M/268M  [00:02<00:00,  117MB/s]
tokenizer_config.json:  100%                        48.0/48.0  [00:00<00:00,  1.29kB/s]
vocab.txt:  100%                                    232k/232k  [00:00<00:00,  4.63MB/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_analysis.csv')
print(random_text)
text1=random_text
```

```
good
```

```
out = pipe(text1)
result = out[0]  # Assuming you want the first result if multiple are returned
sentiment = result["label"]
score = round(result["score"], 2)
```

```
print(f"Sentiment: {sentiment}")
print(score)
```

```
Sentiment: POSITIVE
1.0
```