EVOLVE - A Podcast Based On Your Emotions using Streamlit

```
!pip install transformers --quiet

from transformers import pipeline

# Take input from the user
user_input = input("Enter your text: ")

# Run the text classification pipeline
classifier = pipeline("text-classification",
model='bhadresh-savani/distilbert-base-uncased-emotion',
return_all_scores=True)
prediction = classifier(user_input)

# Print the prediction
print(prediction)
```

```
Enter your text: I feel overwhelmed by everything happening around me.

Hardware accelerator e.g. GPU is available in the environment, but no
`device` argument is passed to the `Pipeline` object. Model will be on
CPU.

[[{'label': 'sadness', 'score': 0.0023926233407109976}, {'label':
'joy', 'score': 0.0013033674331381917}, {'label': 'love', 'score':
0.0011179584544152021}, {'label': 'anger', 'score':
0.0010860546026378887}, {'label': 'fear', 'score': 0.7283236384391785},
{'label': 'surprise', 'score': 0.2657763957977295}]]
```

```
k=prediction[0][:]

res = []
for i in k:
    res.append(i)

result = max(res, key=lambda x: x['score'])
Result=result['label']
Result
```

```
{"type":"string"}
```

```
!pip install google-generativeai --quiet


import pathlib
import textwrap

import google.generativeai as genai

# Used to securely store your API key
from google.colab import userdata
```

```
from IPython.display import display
from IPython.display import Markdown


def to_markdown(text):
  text = text.replace('•', '  *')
  return Markdown(textwrap.indent(text, '> ', predicate=lambda _:
True))

import google.generativeai as genai
from google.colab import userdata
GOOGLE_API_KEY=userdata.get('GOOGLE_API_KEY')

genai.configure(api_key=GOOGLE_API_KEY)

for m in genai.list_models():
  if 'generateContent' in m.supported_generation_methods:
    print(m.name)
```

```
models/gemini-1.0-pro-latest
models/gemini-1.0-pro
models/gemini-pro
models/gemini-1.0-pro-001
models/gemini-1.0-pro-vision-latest
models/gemini-pro-vision
models/gemini-1.5-pro-latest
models/gemini-1.5-pro-001
models/gemini-1.5-pro-002
models/gemini-1.5-pro
models/gemini-1.5-pro-exp-0801
models/gemini-1.5-pro-exp-0827
models/gemini-1.5-flash-latest
models/gemini-1.5-flash-001
models/gemini-1.5-flash-001-tuning
models/gemini-1.5-flash
models/gemini-1.5-flash-exp-0827
models/gemini-1.5-flash-002
models/gemini-1.5-flash-8b
models/gemini-1.5-flash-8b-001
models/gemini-1.5-flash-8b-latest
models/gemini-1.5-flash-8b-exp-0827
models/gemini-1.5-flash-8b-exp-0924
```

```
model = genai.GenerativeModel('gemini-1.5-pro-latest')

%%time
response = model.generate_content(f"Give emotional support based on
this sentiment {Result} in only 120 words")
```

```
CPU times: user 68.4 ms, sys: 4.96 ms, total: 73.3 ms
Wall time: 4.92 s
```

```
to_markdown(response.text)
```

```
<IPython.core.display.Markdown object>
```

```
# Do not forget to install all dependencies first:
!pip install -Uqq WhisperSpeech
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/3.7 MB ? eta -:--:--
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╸ 3.2/3.7 MB 94.3 MB/s eta
0:00:01 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.7/3.7 MB 54.9 MB/s
eta 0:00:00
etadata (setup.py) ... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
74.6/74.6 kB 6.6 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 630.6/630.6 kB 41.1 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 117.8/117.8 kB 9.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 526.7/526.7 kB 35.2 MB/s eta
0:00:00
```

```
# def is_colab():
#     try: import google.colab; return True
#     except: return False

# import torch
# if not torch.cuda.is_available():
#     if is_colab(): raise BaseException("Please change the runtime
type to GPU. In the menu: Runtime -> Change runtime type (the free T4
instance is enough)")
#     else:          raise BaseException("Currently the example
notebook requires CUDA, make sure you are running this on a machine
with a GPU.")

%load_ext autoreload
%autoreload 2

import torch
import torch.nn.functional as F

from IPython.display import Markdown, HTML

!pip install webdataset
```

```
Collecting webdataset
  Downloading webdataset-0.2.100-py3-none-any.whl.metadata (12 kB)
Collecting braceexpand (from webdataset)
  Downloading braceexpand-0.1.7-py2.py3-none-any.whl.metadata (3.0 kB)
Requirement already satisfied: numpy in
```

```
/usr/local/lib/python3.10/dist-packages (from webdataset) (1.26.4)
Requirement already satisfied: pyyaml in
/usr/local/lib/python3.10/dist-packages (from webdataset) (6.0.2)
Downloading webdataset-0.2.100-py3-none-any.whl (74 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/74.8 kB ? eta -:--:--
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 74.8/74.8 kB 6.1 MB/s eta
0:00:00
```

```
# check "7. Pipeline.ipynb"
from whisperspeech.pipeline import Pipeline

# let's start with the fast SD S2A model
pipe = Pipeline(s2a_ref='collabora/whisperspeech:s2a-q4-tiny-
en+pl.model')
```


```
{"model_id":"770df5f48f104e9cb71d31ea3ee523bd","version_major":2,"vers
ion_minor":0}
```


```
/usr/local/lib/python3.10/dist-packages/whisperspeech/
t2s_up_wds_mlang_enclm.py:365: FutureWarning: You are using
`torch.load` with `weights_only=False` (the current default value),
which uses the default pickle module implicitly. It is possible to
construct malicious pickle data which will execute arbitrary code
during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  spec = torch.load(local_filename, map_location=device)
```


```
{"model_id":"d9cf03411fcb4a348ae64bc544e151cf","version_major":2,"vers
ion_minor":0}
```


```
/usr/local/lib/python3.10/dist-packages/whisperspeech/inference.py:38:
FutureWarning: You are using `torch.load` with `weights_only=False`
(the current default value), which uses the default pickle module
implicitly. It is possible to construct malicious pickle data which
will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
```

```
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  return torch.load(local_filename, map_location=device)

{"model_id":"9d6564e01de447dead32f969292904eb","version_major":2,"vers
ion_minor":0}

{"model_id":"e02ec360f0904f8e88b0f24ed4110240","version_major":2,"vers
ion_minor":0}

/usr/local/lib/python3.10/dist-packages/torch/nn/utils/
weight_norm.py:134: FutureWarning: `torch.nn.utils.weight_norm` is
deprecated in favor of `torch.nn.utils.parametrizations.weight_norm`.
  WeightNorm.apply(module, name, dim)
Downloading: "https://dl.fbaipublicfiles.com/encodec/v0/encodec_24khz-
d7cc33bc.th" to /root/.cache/torch/hub/checkpoints/encodec_24khz-
d7cc33bc.th
100%|██████████| 88.9M/88.9M [00:02<00:00, 42.8MB/s]
/usr/local/lib/python3.10/dist-packages/vocos/pretrained.py:70:
FutureWarning: You are using `torch.load` with `weights_only=False`
(the current default value), which uses the default pickle module
implicitly. It is possible to construct malicious pickle data which
will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  state_dict = torch.load(model_path, map_location="cpu")

speech=pipe.generate_to_file("output_audio.wav", response.text)

/usr/lib/python3.10/contextlib.py:103: FutureWarning:
`torch.backends.cuda.sdp_kernel()` is deprecated. In the future, this
context manager will be removed. Please see
`torch.nn.attention.sdpa_kernel()` for the new context manager, with
updated signature.
  self.gen = func(*args, **kwds)

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
pip install pydub
```

```
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub
Successfully installed pydub-0.25.1
```

```
with open("output_audio.wav", "rb") as f:
    audio_data = f.read()
```

```
pip install streamlit --quiet
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.9/41.9 kB 3.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.7/8.7 MB 40.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 17.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.9/6.9 MB 104.1 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.3/79.3 kB 6.1 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 kB 5.1 MB/s eta
0:00:00
```

```
%%writefile app.py
import streamlit as st

# Define the title of the podcast
podcast_title = " Evolve: A Podcast Based on Your Emotions "

# Read the audio data
audio_data = open("output_audio.wav", "rb").read()

# Set Streamlit app title and page configuration
st.set_page_config(page_title="Evolve Podcast",
page_icon=":microphone:", layout="centered")

# Add title and subtitle
st.markdown(f"<h1 style='text-align: center; color:
#f63366;'>{podcast_title}</h1>", unsafe_allow_html=True)
st.write("Welcome to Evolve, where we create mini moodcasts under 30
sec based on your emotions for fast paced world. Let's dive in and
explore the journey of self-discovery together!")

# Add audio player
st.subheader("Listen to Your moodcast:")
st.audio(audio_data, format="audio/wav")
```

```
# Add footer with teammates' names
st.write("---")
st.write("Created with ❤ by Boddu Harshitha, Harini J, K M Sindhu
Priya, Mahalakshmi J")
```

```
Writing app.py
```

```
!streamlit run app.py & npx localtunnel --port 8501
```

```
Collecting usage statistics. To deactivate, set
browser.gatherUsageStats to false.


  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.28.0.12:8501
  External URL: http://34.147.69.182:8501

  Stopping...
^C
```