# Appendix A

## Sample Code

Code File App.py

```
import streamlit as st import torch
import os import tempfile
import speech_recognition as sr import time
from transformers import pipeline import io

# Set page config st.set_page_config(
page_title="Speech Analysis App", page_icon="🎵", layout="centered"
)

# Initialize models @st.cache_resource
def load_gender_model():
return pipeline("audio-classification", model="audeering/wav2vec2-large- robust-24-ft-age-gender")

@st.cache_resource
def load_emotion_text_model():
return pipeline("sentiment-analysis", model="michellejieli/emotion_text_classifier")
# Function to process audio file for gender analysis
# Improved function to process audio for gender analysis def
process_audio_for_gender(audio_data, gender_model):
# Create a temporary file temp_audio_path = None try:
with tempfile.NamedTemporaryFile(delete=False, suffix='.wav') as tmp_file:
tmp_file.write(audio_data.getvalue())
temp_audio_path = tmp_file.name

# Print debug info
st.info(f"Processing audio file: {temp_audio_path}")
# Get audio info import wave
with wave.open(temp_audio_path, 'rb') as wf: channels = wf.getnchannels()
sampwidth = wf.getsampwidth() framerate = wf.getframerate()
st.write(f"Audio properties: Channels={channels}, Sample Width={sampwidth},
Rate={framerate}")
# Process with gender model
gender_results = gender_model(temp_audio_path)
# Print results for debugging
print("Gender model raw results:", gender_results)
# Remove temp file
if temp_audio_path and os.path.exists(temp_audio_path): os.unlink(temp_audio_path)
return gender_results except Exception as e:
# Remove temp file in case of error
if temp_audio_path and os.path.exists(temp_audio_path): os.unlink(temp_audio_path)
print(f"Error in gender processing: {e}") st.error(f"Audio processing error: {e}") raise e
# Function to convert speech to text def speech_to_text(audio_data):
"""Convert speech audio to text using Google's speech recognition""" r = sr.Recognizer()
```

```
temp_audio_path = None
try:
# Create temporary file for the audio
with tempfile.NamedTemporaryFile(delete=False, suffix='.wav') as temp_audio_file:
temp_audio_path = temp_audio_file.name

# Write audio data to the temporary file with open(temp_audio_path, 'wb') as f:
f.write(audio_data.getvalue())
# Use speech recognition on the file
with sr.AudioFile(temp_audio_path) as source: audio = r.record(source)
text = r.recognize_google(audio)

# Clean up the temporary file
if temp_audio_path and os.path.exists(temp_audio_path): os.unlink(temp_audio_path)

return text
except sr.UnknownValueError: return "Speech not recognized"
except sr.RequestError as e:
return f"Speech recognition service error: {e}" except Exception as e:
if temp_audio_path and os.path.exists(temp_audio_path): os.unlink(temp_audio_path)
return f"Error: {str(e)}"

# Function to display analysis results
# Updated display_results function with better gender detection logic
def display_results(gender_results, text=None, text_emotion_results=None): # Display
results in columns
if text_emotion_results:
col1, col2 = st.columns(2) else: col1 = st.container()
# Gender results with col1:
st.subheader("Gender Analysis")

# Show raw results for debugging
st.write("Raw model predictions:", gender_results)
# Get the top prediction top_prediction = gender_results[0]
 stop_prediction['score']

# Extract gender from the label if 'female' in label.lower():
gender = 'Female'
elif 'male' in label.lower(): gender = 'Male'
else:
gender = 'Unknown' confidence = score * 100
# Display result st.markdown(f"### {gender}")
st.markdown(f"Confidence: {confidence:.0f}%") st.progress(confidence/100)


# Text Emotion results (if available) if text_emotion_results:
with col2:
st.subheader("Emotion Analysis")
# Display transcribed text st.markdown("Transcribed Text:")
st.markdown(f"\"{text}\"")
```

```
# Display emotion from text
emotion = text_emotion_results[0]['label'] confidence = text_emotion_results[0]['score'] *
100

st.markdown(f"Detected Emotion: {emotion.capitalize()}") st.markdown(f"Confidence:
{confidence:.0f}%")
# Display progress bar st.progress(confidence/100)
# Map emotions to emojis emoji_map = {
'joy': '😊', 'sadness': '😢', 'anger': '😠',
'neutral': '😐', 'fear': '😨', 'disgust': '🤢', 'surprise': '😲'
}
def main():
st.title("🎵 Speech Gender and Text Emotion Analysis")

st.markdown("""
### Realtime Audio Analysis
Record your voice to analyze your speech:

1.        Select recording duration
2.        Choose whether to analyze text content
3.        Click "Start Recording" and speak clearly
4.        Review the analysis results """)

# Recording duration slider
duration = st.slider("Recording Duration (seconds)", 3, 10, 5)
# Text emotion detection checkbox with a unique key
analyze_text = st.checkbox("Analyze speech content (text emotion)", value=True,
key="analyze_text_checkbox")
# Start recording button
if st.button("Start Recording"): try:
# Record audio
with st.spinner(f"Recording for {duration} seconds..."): audio_data = record_audio(duration)
# Allow user to play back the recording st.audio(audio_data)

# Process the recording
with st.spinner("Analyzing audio..."): # Load gender model
gender_model = load_gender_model()

# Process audio for gender
gender_results = process_audio_for_gender(audio_data,

gender_model)



# Process text if enabled text = None text_emotion_results = None
if analyze_text:
# First convert speech to text
with st.spinner("Transcribing speech to text..."): text = speech_to_text(audio_data)
if text and text != "Speech not recognized": # Then analyze text emotion
```

```
text_emotion_model = load_emotion_text_model() text_emotion_results =
text_emotion_model(text)# Display results
display_results(gender_results, text, text_emotion_results)

except Exception as e:
st.error(f"Error recording or processing audio: {str(e)}")

st.info("Please make sure your microphone is connected and try
again.")

if _name_ == "_main_": main()
```

# Appendix B

## Output Screenshots

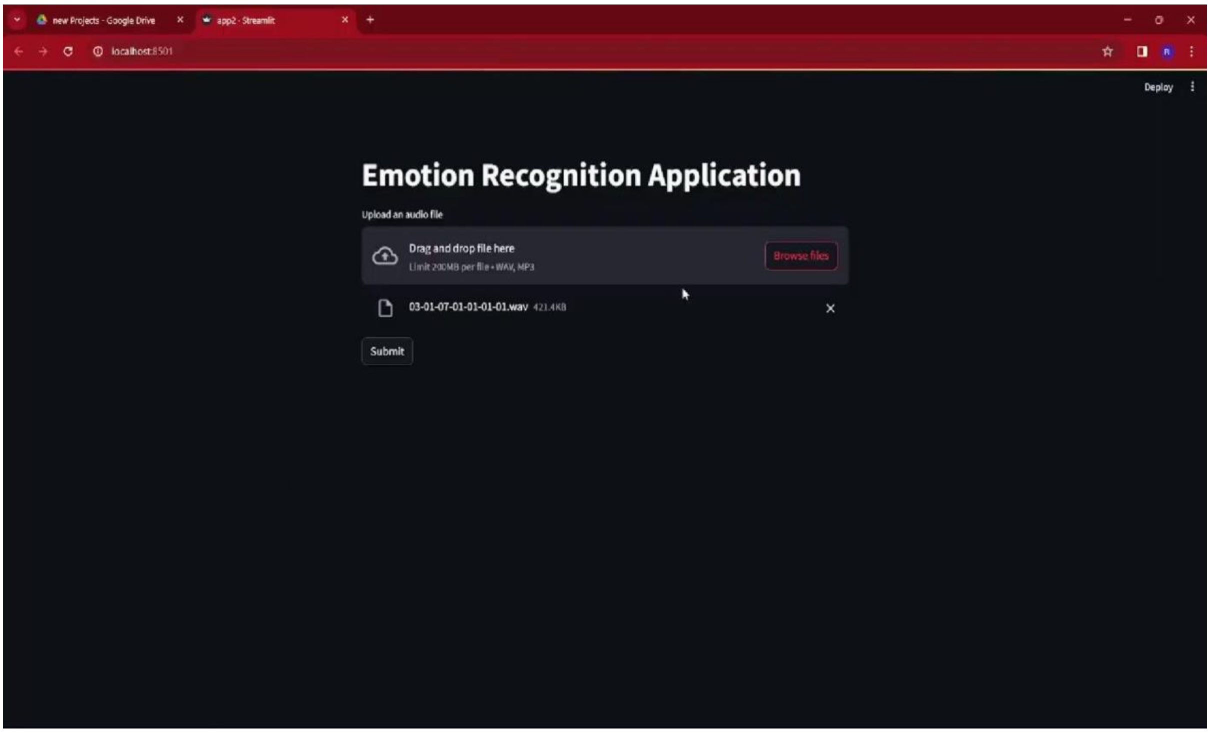

new Projects - Google Drive
app2 · Streamlit
localhost:8501
Deploy
Emotion Recognition Application
Upload an audio file
Drag and drop file here
Limit 200MB per file • WAV, MP3
Browse files
03-01-07-01-01-01-01.wav 421.4KB
Submit

Figure 5.1 Using Audio File Emotion Recognition Application

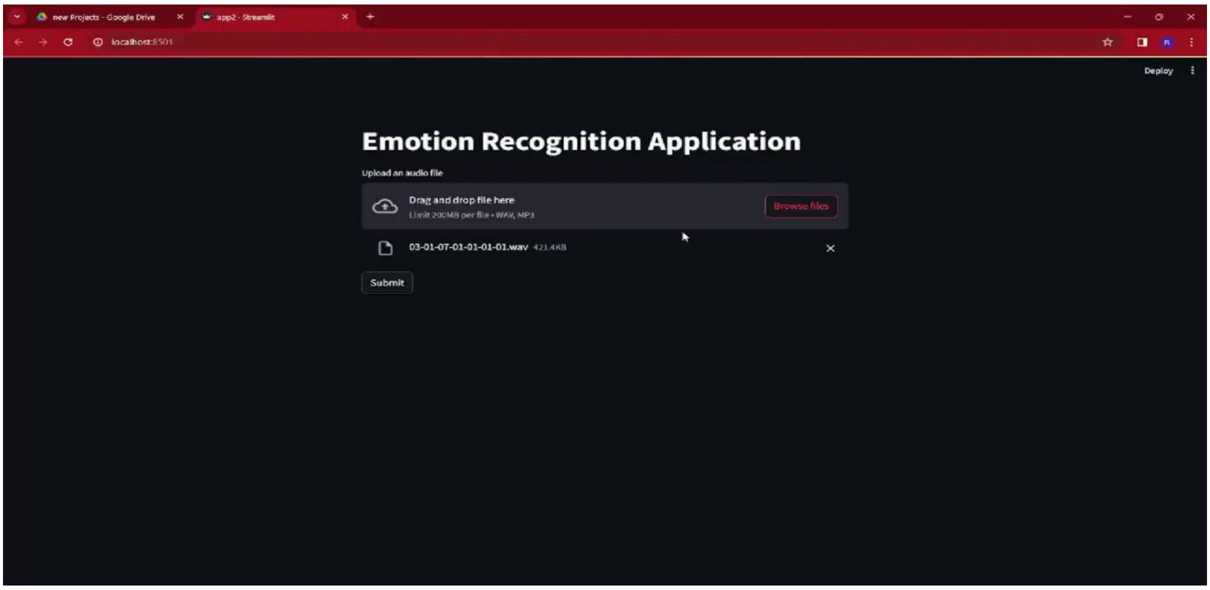
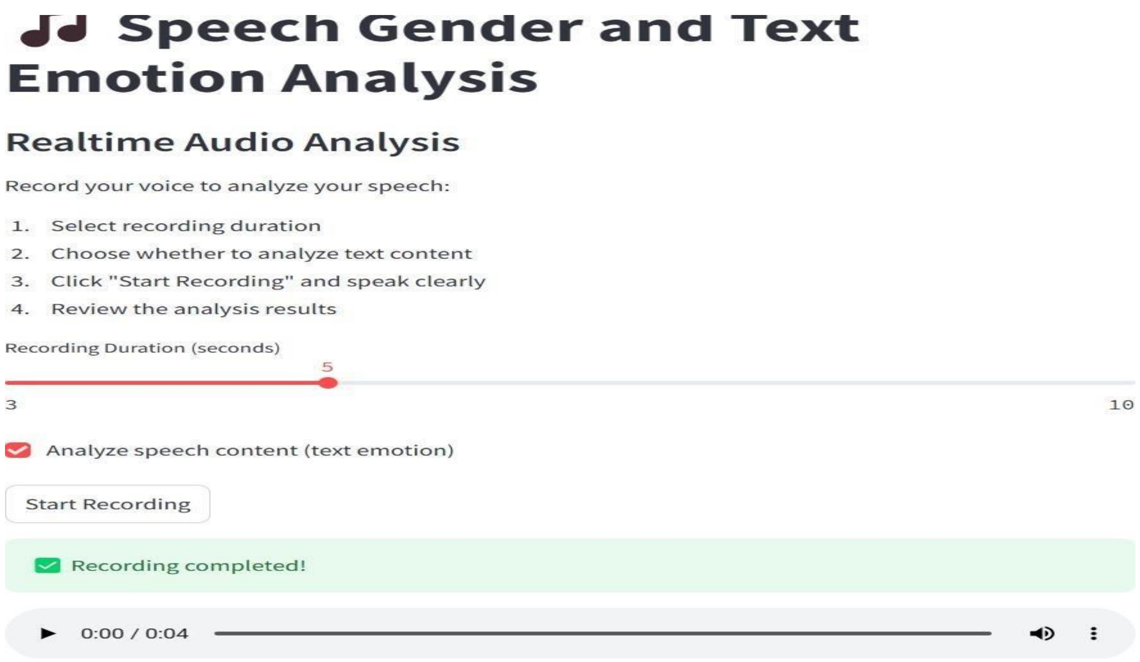

new Projects - Google Drive
app2 · Streamlit
localhost:8501
Deploy
Emotion Recognition Application
Upload an audio file
Drag and drop file here
Limit 200MB per file • WAV, MP3
Browse files
03-01-07-01-01-01-01.wav 421.4KB
Submit

Figure 5.2 Uploading a file for Emotion Recognition



# Speech Gender and Text Emotion Analysis

## Realtime Audio Analysis

Record your voice to analyze your speech:

1. Select recording duration
2. Choose whether to analyze text content
3. Click "Start Recording" and speak clearly
4. Review the analysis results

Recording Duration (seconds)

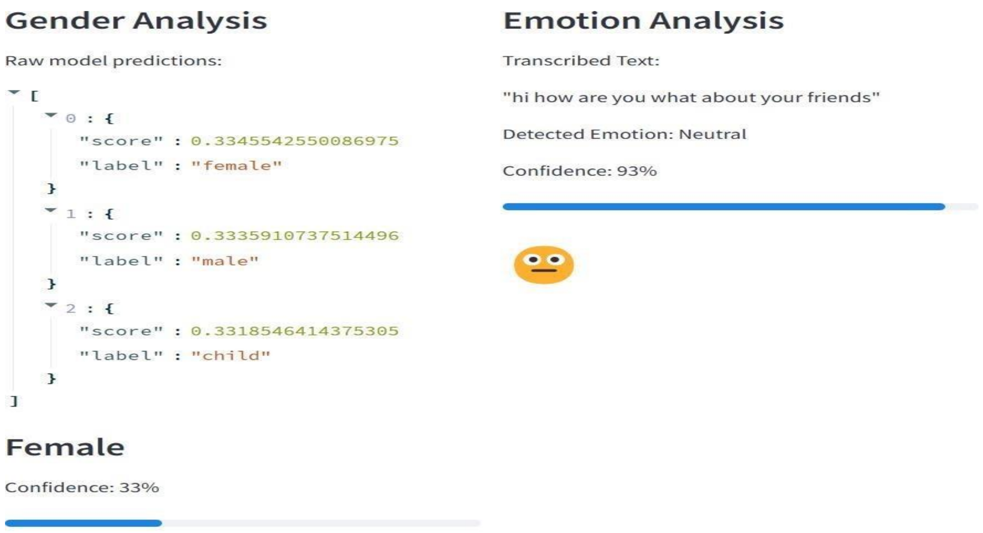
5

3 10

☑ Analyze speech content (text emotion)

✓ Recording completed!

Figure 5.3- Using an audio file to predict Gender and Emotion



## Gender Analysis

Raw model predictions:

```
[
  0 : {
    "score" : 0.3345542550086975
    "label" : "female"
  }
  1 : {
    "score" : 0.3335910737514496
    "label" : "male"
  }
  2 : {
    "score" : 0.3318546414375305
    "label" : "child"
  }
]
```

## Female

Confidence: 33%

## Emotion Analysis

Transcribed Text:

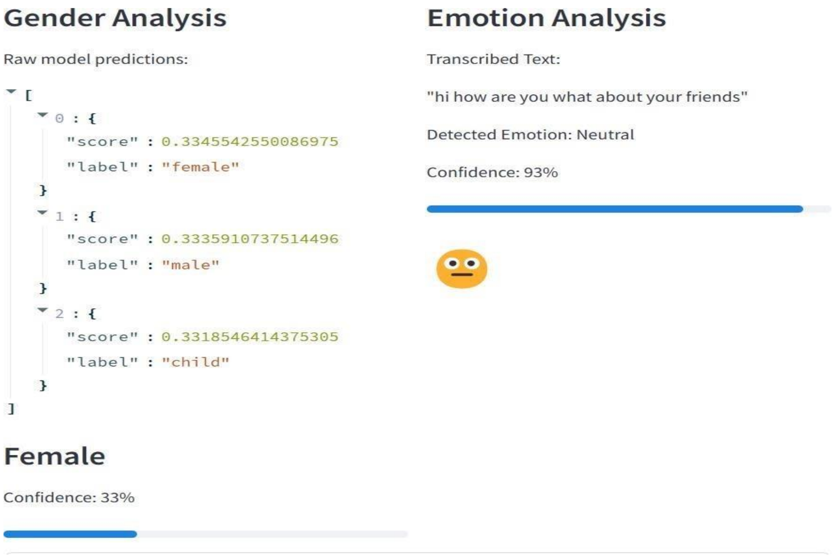
"hi how are you what about your friends"

Detected Emotion: Neutral

Confidence: 93%

😐

Fig 5.4 Emotion-Neutral; Confidence-93%



Gender Analysis

Raw model predictions:

```
[
  0 : {
    "score" : 0.3345542550086975
    "label" : "female"
  }
  1 : {
    "score" : 0.3335910737514496
    "label" : "male"
  }
  2 : {
    "score" : 0.3318546414375305
    "label" : "child"
  }
]
```

Female

Confidence: 33%

Emotion Analysis

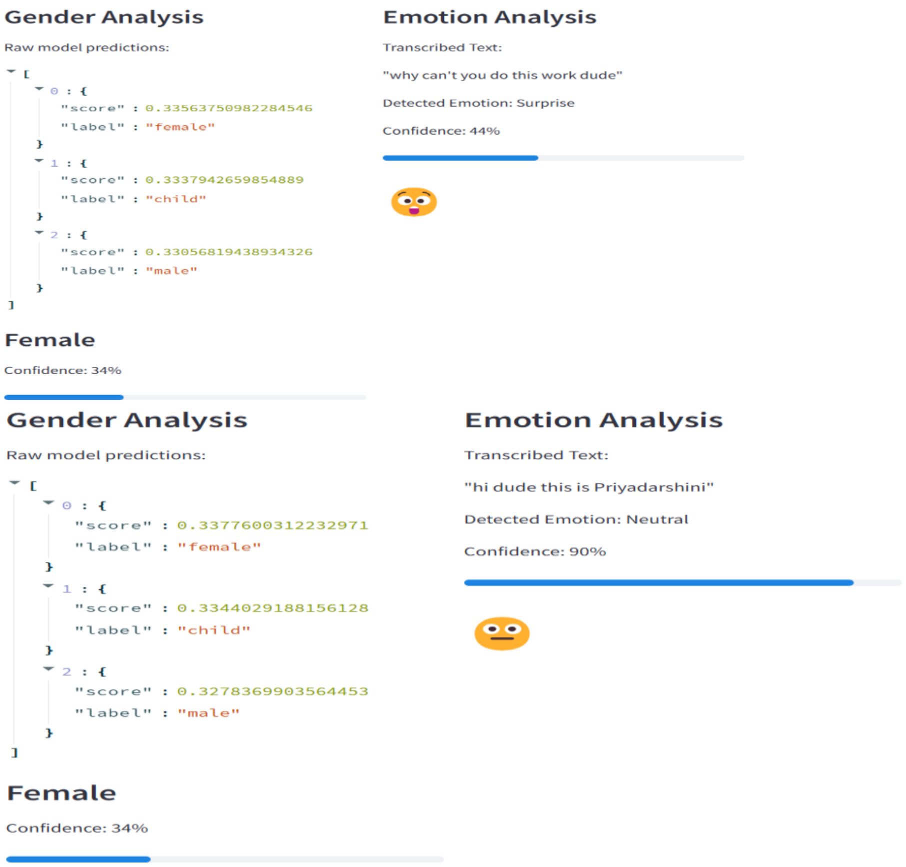
Transcribed Text:

"hi how are you what about your friends"

Detected Emotion: Neutral

Confidence: 93%

Figure 5.5: Both Gender and Emotion Analysis Prediction



Gender Analysis

Raw model predictions:

```
[
  0 : {
    "score" : 0.33563750982284546
    "label" : "female"
  }
  1 : {
    "score" : 0.3337942659854889
    "label" : "child"
  }
  2 : {
    "score" : 0.33056819438934326
    "label" : "male"
  }
]
```

Female

Confidence: 34%

Emotion Analysis

Transcribed Text:

"why can't you do this work dude"

Detected Emotion: Surprise

Confidence: 44%

Gender Analysis

Raw model predictions:

```
[
  0 : {
    "score" : 0.3377600312232971
    "label" : "female"
  }
  1 : {
    "score" : 0.3344029188156128
    "label" : "child"
  }
  2 : {
    "score" : 0.3278369903564453
    "label" : "male"
  }
]
```

Female

Confidence: 34%

Emotion Analysis

Transcribed Text:

"hi dude this is Priyadarshini"

Detected Emotion: Neutral

Confidence: 90%

Figure 5.6: Both Gender and Emotion Analysis Prediction
[Confidence-90% ;Emotion-Neutral]