Prototype Documentation

Introduction

The "Audio Analysis App" is a Python application built using the Streamlit library. It allows users to upload an audio file (in WAV or MP3 format) for analysis. The analysis includes transcription of the audio, prediction of emotion in the transcribed text, and visualisation of the results.

Code Overview

The code can be divided into the following sections:

1. User Interface

The user interface of the application is created using Streamlit. It consists of the following components:

Title: A title that introduces the app as "Audio Analysis App."
File Upload Widget: A file upload widget that allows users to upload an audio file (supported formats: WAV and MP3) for analysis.

2. Audio Processing

Once an audio file is uploaded, it undergoes several processing steps:

Audio Loading: The uploaded audio file is loaded using the `torchaudio` library.
Transcription: The audio is transcribed into text using a pre-trained Wav2Vec2 model from the Hugging Face Transformers library.
Emotion Prediction: The transcribed text is classified for emotion (currently binary: "happy" or "sad") using a pre-trained BERT-based model from the Hugging Face Transformers library.

3. Results Display

The results of the analysis are displayed to the user:

Audio Playback: The uploaded audio file can be played back to the user using `st.audio`.
Transcription: The transcribed text is displayed to the user.
Emotion Prediction: The predicted emotion (either "happy" or "sad") is displayed along with a table containing the audio file name and emotion label.
Emotion Prediction Chart: A line chart showing the predicted emotion over time (currently set to time 0) is displayed.

Research Findings

Audio Transcription

- The code successfully transcribes the uploaded audio file using the pre-trained Wav2Vec2 model.
- Transcription accuracy may vary depending on the quality of the audio and the complexity of the spoken content.

Emotion Prediction

- The code predicts the emotion in the transcribed text using the pre-trained BERT-based model.
- The emotion prediction is binary, classifying emotions as either "happy" or "sad."
- Complex emotions are not captured, and there is room for improvement in emotion classification.

User Interface

- The Streamlit user interface is user-friendly and allows for easy uploading of audio files.
- However, there is potential for user interface enhancements, such as providing more detailed feedback during analysis.

Visualisation

- The code uses line charts to visualise the predicted emotion over time. However, it currently only displays emotion at time 0.
- There is an opportunity to extend this visualisation to show emotional changes over the duration of the audio.

Conclusion

The "Audio Analysis App" demonstrates the potential of using pre-trained models for audio transcription and text classification to analyse content and predict emotions in audio recordings. With further development and improvements, this application could become a valuable tool for various applications, including sentiment analysis, emotional support, and content moderation.