Prototype Documentation

Approach:

The prototype is a simple Python script that takes audio files as input and extracts audio features such as MFCCs and pitch. These features are then used to train a simple machine learning model to classify the input audio as either "happy" or "sad." The script also integrates a solution for detecting confidence, either an open-source algorithm or a commercialized third-party solution. Once the audio features have been extracted and the confidence has been detected, the script prints the results in a table and a smooth line chart.

Features:

- MFCCs: Mel-frequency cepstral coefficients are a type of audio feature that is commonly used for speech recognition and other audio processing tasks. MFCCs are calculated by taking the Fourier transform of the audio signal, converting it to a mel-frequency scale, and then taking the logarithm of the magnitude spectrum.
- Pitch: Pitch is the perceived fundamental frequency of a sound. It is an important audio feature for emotion detection, as different emotions are often associated with different pitch ranges.

Preprocessing Steps:

Before extracting the audio features, the audio signal is normalized to a standard amplitude level. This is done to ensure that the features are not affected by the volume of the audio signal.

Libraries and Tools:

The prototype uses the following Python libraries and tools:

- scikit-learn: scikit-learn is a free, open-source machine learning library in Python. It is used to train and evaluate the machine learning model for emotion detection.
- OpenSmile: OpenSmile is a free, open-source toolkit for audio feature extraction. It is used to extract the MFCC and pitch features from the audio signal.
- Librosa: Librosa is a free, open-source Python library for audio analysis. It is used to normalize the audio signal and to perform other audio processing tasks.

Challenges:

One of the challenges faced in developing the prototype was selecting the optimal solution for detecting confidence. There are a number of different solutions available, both open-source and commercialized. The selected solution should be accurate, efficient, and cost-effective.

Another challenge was integrating the confidence detection solution into the prototype. The solution should be integrated in a way that is efficient and easy to use.

User Manual:

To use the prototype, users need to first install the required Python libraries and tools. Once the required libraries and tools have been installed, users can upload their audio files to the prototype. The prototype accepts audio files in the following formats:

- WAV
- MP3
- OGG

Once the audio files have been uploaded, users can click the "Analyze" button to start the analysis process. The prototype will extract the audio features and detect the emotion and confidence of the input audio. The results will be printed in a table and a smooth line chart.

Tables and Line Charts:

The prototype prints the results of the analysis in two tables and two line charts:

- Emotion Table: The emotion table shows the predicted emotion ("happy" or "sad") based on the input audio. The table has two columns: Audio File Name and Predicted Emotion.
- Confidence Table: The confidence table shows the predicted confidence of the input audio. The table has two columns: Audio File Name and Predicted Confidence.
- Emotion Line Chart: The emotion line chart shows the predicted emotion over time. The positive y-axis of the chart is "happy" and the negative y-axis is "sad."
- Confidence Line Chart: The confidence line chart shows the predicted confidence over time.

Conclusion:

The prototype is a simple but effective tool for detecting emotion and confidence in audio recordings. It can be used by a variety of users, including researchers, developers, and businesses.