TITLE: Human Emotion Detection from Voice

1. Abstract:

This project aims to develop a machine learning-based system that can accurately detect human emotions from voice recordings. Leveraging the RAVDESS dataset, audio features are extracted using MFCCs (Mel-frequency cepstral coefficients) and classified using a Random Forest algorithm. The end goal is a user-friendly web app built with Streamlit, allowing users to upload or record audio and receive instant emotion predictions. This system can serve as a foundation for applications in customer service, mental health monitoring, and affective computing.

2. Introduction:

Emotions are integral to human communication. Detecting them through voice enables machines to better understand user states, thus enabling more natural human-computer interactions. This project utilizes speech emotion recognition (SER) techniques by analyzing the spectral properties of audio signals to classify emotions such as happy, sad, angry, and more.

3. Tools Used:

- **Programming Language**: Python
- **Libraries**: librosa, numpy, scikit-learn, joblib, streamlit, pydub
- **Dataset**: RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song)
- **Model**: Random Forest Classifier
- **Deployment**: Streamlit Web App

4. Steps Involved in Building the Project:

- **Dataset Preparation**
    - Loaded the RAVDESS dataset containing labeled .wav files for different emotional states.
- **Feature Extraction**
    - Used librosa to extract 40 MFCCs from each audio file as key features.
- **Model Training**
    - Split the data into training and test sets (80:20).
    - Trained a Random Forest classifier on the extracted features.
    - Achieved high accuracy on validation using classification_report() and accuracy_score().
- **Model Saving**
    - Serialized the trained model using joblib for later inference.
- **Web Application Development**
    - Built an interactive UI using Streamlit.
    - Implemented audio uploading and format conversion using pydub.
    - Extracted features from user-uploaded audio and predicted the emotion in real-time.

5. Conclusion:

The project successfully demonstrates a functional pipeline for detecting emotions from speech using classical machine learning techniques. It combines signal processing, model training, and UI deployment into a cohesive system. Future enhancements could include real-time microphone input, use of deep learning models like CNNs or LSTMs, and multi-language support.