SPEECH EMOTION RECOGNITION

OVERVIEW

The Speech Emotion Recognition (SER) predicts emotional states from speech and can be used for personal voice emotion recognition. It processes raw speech audio (.wav) files, extracts acoustic features, and trains a machine-learning classifier. SER is trained and evaluated using the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset.

PROJECT EXECUTION

```
python3 src/extract_features_csv.py #extract features

python3 src/train.py #train model

python3 src/predict.py [.wav filepath] #predict model
```

Eg. python3 src/predict.py data/ ravdess_dataset/Audio_Speech_Actors_01-24/Actor_16/ 03-01-05-01-02-01-16.wav

---

DATASET

Dataset: Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

- Labels (what the model is trying to predict). Emotion categories from RAVDESS filename metadata.
    - These labels are stored in: labels.csv & column: emotion
    - Categories & target emotions: Neutral, happy, sad, angry
- Sample size: 672/ 2,892 .wav files

FEATURE EXTRACTION

Features are numerical representations of the audio signal extracted from each .wav file. They describe how the speech sounds, not what is being said. For each audio clip, we extract:

1. MFCCs (Mel-Frequency Cepstral Coefficients)
    a. Capture the spectral shape and timbre of speech

b. Widely used in speech and emotion recognition
c. Computation:
   i. Mean of each MFCC coefficient
   ii. Standard deviation of each MFCC coefficient
   iii. Total: 26 features (13 means + 13 stds)

2. RMS Energy (Root Mean Square)
   a. Measures speech loudness, intensity, and energy
   b. Computation:
      i. Mean RMS
      ii. Standard deviation of RMS
      iii. Total: 2 features
3. Zero-Crossing Rate (ZCR)
   a. Measures how often the audio signal changes sign
   b. Related to speech roughness and articulation
   c. Computation:
      i. Mean ZCR
      ii. Standard deviation of ZCR
      iii. Total: 2 features

- This results in total features per sample: 26 + 2 + 2 = 30

MODEL & PERFORMANCE (BASELINE)

Model
- Classifier: Support Vector Machine (SVM)
- Kernel: Radial Basis Function (RBF)
- Preprocessing: Feature standardization
- Split: Stratified train/test split

Model Performance
- Test Accuracy: 86.7%
- Strong performance on angry and happy emotions
- Expected confusion mainly between neutral and emotional speech

EVALUATION METRICS & PARAMETERS

F1-score: Harmonic mean of precision and recall
- Computation:

$$F1 = 2 \bullet \frac{precision \bullet recall}{precision + recall}$$

Precision: When the model predicts an emotion, how often is it correct?

Recall: Of all true samples of an emotion, how many did the model correctly identify?

Variables affecting emotion recognition:

- loudness, intensity, energy
- spectral shape and timbre
- roughness and articulation

PIPELINE

Process for training and testing the model:

1. .wav audio (RAVDESS)
2. Feature extraction (MFCC, RMS, ZCR)
3. Numerical feature vector
4. Trained machine learning model (training)
5. Predicted emotion
6. Evaluation and comparison to known label (Accuracy, F1, Confusion Matrix)

Process for new raw voices:

1. .wav audio (new raw voice)
2. Feature extraction (MFCC, RMS, ZCR)
3. Numerical feature vector
4. Trained machine learning model
5. Predicted emotion

PROJECT STRUCTURE (FILE DESCRIPTIONS & SCRIPTS)

ravdess_dataset/

- Has the raw RAVDESS .wav audio files used for training and evaluation.

features.csv

- Extracted acoustic features, with label and filepath columns.
- Input: labels.csv + corresponding .wav audio files
- Output: Extracted features (MFCC, RMS, ZCR) + label + filepath

labels.csv

- Parsed emotion labels, which are generated automatically from RAVDESS filenames.
- Columns:
    - Filepath: path to each .wav file
    - Emotion: parsed emotion label
        - Filtered to four target emotions: neutral, happy, sad, and angry.
        - 672 samples total
- Input: .wav files
- Output: filepath + emotion

make_labels_csv.py

- Reads through the RAVDESS dataset, parses emotion labels from filenames, filters to the selected emotions, and saves labels.csv.
- Input: .wav audio files
- Output: labels.csv

extract_features_csv.py

- Loads each audio file listed in labels.csv, extracts acoustic features (MFCC, RMS, ZCR), and saves them to features.csv.
- Input: audio + labels
- Output: features.csv

train_baseline.py

- Trains and evaluates a baseline emotion classifier.
- Input: features.csv
- Output: report (accuracy, confusion matrix, and classification)

train.py

- Trains an emotion recognition model from extracted audio features and saves the trained model for later use.
- Input: features.csv
- Output: model.joblib

predict.py

- Loads a trained emotion recognition model and predicts the emotion of a given audio file.
- Reads: model.joblib
- Output: stdout prediction
