Project Title: Gender Classification from Speech Using MFCCs

1.Introduction This Signals and Systems project began with a simple requirement: perform Fourier transforms on audio files of spoken vowels to analyze their frequency content.To extend the scope,we generalized the task to classify speaker gender from short speech utterances using Mel-Frequency Cepstral Coefficients (MFCCs),a widely adopted feature in speech and audio processing.The project uses the RAVDESS dataset,which contains 1,440 high-quality recordings of professional actors speaking emotionally neutral or expressive statements.

2.Dataset

- **Source**: RAVDESS Audio Speech Actors 01-24 (Audio-only, speech modality).
- **Structure**: 24 actors (12 male,12 female),each uttering two statements with 8 emotions at two intensity levels,yielding 1,440 `.wav` files.
- **Goal**: Use only the speech files (modality code "03" ,channel "01") to avoid song or video data,yielding 1,440 samples balanced by gender.

3 .Feature Extraction with MFCCs

- **Rationale**: MFCCs mimic human auditory perception by applying a mel-scale filter bank that emphasizes lower frequencies.They capture formant structure and spectral envelope—key cues that differ between male and female voices due to vocal-tract length and pitch.

- **Pipeline**:

  1. **Load audio** at 16 kHz.
  2. **Compute static MFCCs** (13 coefficients) via short-time FFT (2048-sample window,512-sample hop).This yields a 13×T matrix per file.

3. **Compute derivatives**: first-order delta and second-order delta-delta to capture temporal dynamics.
4. **Stack features** into a 39× T matrix.
5. **Summarize** each matrix into fixed-length vectors by concatenating per-coefficient mean and standard deviation,resulting in 78-dimensional feature vectors.

- **Implementation**: Python script `ravdess_mfcc_pipeline.py` extracts and filters only the modality=03,channel=01 files,constructs gender labels,and saves features in a compressed `.npz` archive.

4.Data Preparation

- **Train/ Test Split**: Stratified 80/20 split (1,152 train,288 test), preserving gender balance.
- **Normalization**: Fit a `StandardScaler` on the training features (zero mean,unit variance) and transform both train and test sets to prevent data leakage.

5.Classification with XGBoost

- **Model Choice**: `XGBClassifier` offers tree-based ensemble learning with built-in regularization (L1,L2,gamma) to control overfitting.

- **Hyperparameter Tuning**: Performed grid search with 5-fold cross-validation over:

  - Number of trees (50,100,200)
  - Max depth (3,5,7)
  - Learning rate (0.01,0.1,0.2)
  - Gamma (0,1,5)
  - L1 (`reg_alpha`: 0,0.1,1) and L2 (`reg_lambda`: 1,5,10) regularization

- **Best Performance**: Achieved ~98.6% accuracy on the test set,with high precision (99.3%) and recall (97.9%).

6.Evaluation

- **Metrics**: Accuracy,precision,recall (sensitivity),specificity,F1 score.
- **Visualization**: Confusion matrix plotted to show true/false positives and negatives.
- **Insights**: Low-order MFCCs show systematic shifts between male and female voices (formant differences).Even when spectrograms look similar,mean-MFCC comparison reveals gender cues.

7.Implementation Notes

- **Script Invocation**: Use `python3 ravdess_mfcc_pipeline.py --dataset-path <path> --out-features gender_features.npz --task gender`.
- **Notebook Workflow**: Jupyter notebook cells run the pipeline, normalize data,train XGBoost,and plot results.
- **Model Persistence**: Save the trained XGBoost model and scaler with `joblib.dump` for reproducible inference.

8 .Conclusion By leveraging MFCCs—rooted in Fourier analysis and perceptual filter banks—this project extends a basic frequency-content exercise into a practical speaker-gender classification task. The pipeline demonstrates how Signals and Systems concepts (windowed FFT,spectral analysis) feed directly into ML features, enabling high-accuracy classification on a standard speech corpus.