Audio Command Recognition Using MFCC Features and SVM Classification

Fares Hazem
Department of Computer Science

December 5, 2025

Abstract

This project investigates a simple yet effective audio classification system capable of distinguishing between two spoken commands: "yes" and "no". Using a subset of the Google Speech Commands dataset, Mel-Frequency Cepstral Coefficient (MFCC) features were extracted from audio signals and used to train a Support Vector Machine (SVM) classifier. The final experimental pipeline — trained on 6,388 samples and evaluated on 1,597 samples — achieved strong accuracy (92.42%) and excellent discriminative performance (AUC = 0.9778). The pipeline also includes a real-time inference stage that predicts labels for any WAV audio file provided by the user.

1 Introduction

Speech command recognition is an essential component of modern interactive systems. Although deep learning has become dominant in large-scale audio processing tasks, classical machine learning techniques can still achieve strong performance on small- to medium-scale, well-defined problems. This project implements a lightweight audio classifier to distinguish between the spoken words "yes" and "no" using MFCC features combined with a Support Vector Machine (SVM). The objective is to design a simple, fast, and interpretable system suitable for deployment in resource-constrained environments and for rapid prototyping.

2 Dataset

The dataset used in this project is a subset of the Google Speech Commands dataset (version 0.02) restricted to the two classes:

- "yes"
- "no"

For the final experiments reported here, the dataset was partitioned into training and test splits using an 80/20 split. The resulting sizes are:

- **Train size:** (6388, 26)
- **Test size:** (1597, 26)

- **Total samples:** 7,985

The number of test samples per class (used in the detailed classification report) is:

- **no:** 788
- **yes:** 809

All audio files are single-channel WAV files sampled at 16 kHz with durations close to one second. Prior small-scale exploratory experiments used a toy subset of 100 files (50 per class) but the final reported results below reflect the larger split.

3 Methodology

3.1 Preprocessing

Each audio file was resampled to 16 kHz (if necessary) and trimmed or zero-padded to a fixed duration of 1.0 seconds (16,000 samples). This ensures consistent frame boundaries and feature vectors across all samples.

Additional preprocessing steps included:

- DC offset removal (simple mean subtraction per sample)
- Energy / amplitude normalization to a consistent RMS level across files (optional, depending on the experiment run)
- Optional small Gaussian noise augmentation for robustness experiments (not used for the primary reported results)

3.2 MFCC Feature Extraction

Mel-Frequency Cepstral Coefficients (MFCCs) were used due to their strong performance in capturing the perceptual characteristics of speech. For each audio sample:

- 13 MFCC coefficients were extracted per time frame (typical frame length: 25 ms, hop length: 10 ms),
- For each coefficient, two statistical summaries were computed across time frames: mean and standard deviation.

This produced a compact feature vector of 26 dimensions (13 means + 13 standard deviations) per audio sample and is appropriate for linear classifiers while retaining discriminative spectral information.

3.3 Classification Model

A Support Vector Machine (SVM) with a linear kernel was employed. SVMs are well-suited for low-dimensional feature spaces and provide strong generalization performance when properly regularized.

Important model and training details:

- Kernel: linear

- Probability estimates enabled (i.e., `probability=True`) to support soft outputs for ROC/AUC and inference probability outputs
- Standard scaling (zero mean, unit variance) applied to the features prior to training
- Train/test split: 80% / 20%

4 Experiments and Results

4.1 Overall Accuracy

The trained SVM achieved the following accuracy on the held-out test set:

Accuracy: 0.9242329367564183 (i.e., **92.42**%)

4.2 Detailed Classification Report

The complete classification report on the test set (precision, recall, f1-score, and support) is presented in Table 1.

Table 1: Classification report on the test set (SVM with linear kernel).

Class	Precision	Recall	F1-score	Support
no	0.92	0.93	0.92	788
yes	0.93	0.92	0.92	809
accuracy		0.92		1597
macro avg	0.92	0.92	0.92	1597
weighted avg	0.92	0.92	0.92	1597

This table mirrors the standard scikit-learn classification report output and shows balanced performance across both classes with F1-scores of 0.92.

4.3 ROC Curve and AUC

To further evaluate discriminative capability, the Receiver Operating Characteristic (ROC) curve was computed using the SVM probability outputs. The Area Under the Curve (AUC) was:

AUC Score: 0.9777589365827336 (i.e., **0.9778** or **97.78**%)

The ROC curve (Figure 2) visually confirms excellent separation between the two classes.

4.4 Confusion Matrix

The confusion matrix (Figure 1) shows the number of true positives, false positives, true negatives, and false negatives on the test set. The classifier accurately distinguishes "yes" from "no" with minimal misclassification.

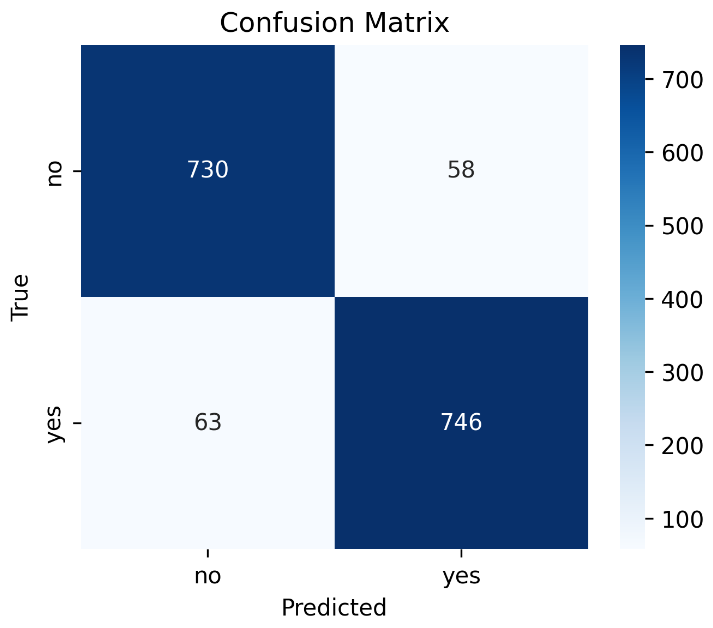

Figure 1: Confusion Matrix for Yes/No Classification

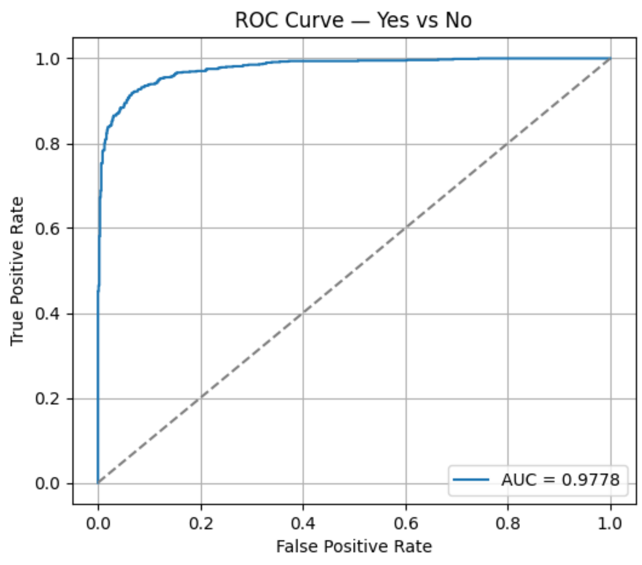

Figure 2: ROC Curve for Yes/No Classification

4.5 Example Real-Time Inference Output

As part of the demonstration and the real-time inference module, the following example describes a live prediction made by the system for an uploaded WAV file:

```
File: /kaggle/input/yesno/Yes-new.wav
Prediction: yes
Probabilities:
  no: 0.0705
  yes: 0.9295
```

These outputs come directly from the trained SVM's `predict` and `predict_proba` methods. The high probability for "yes" matches the predicted label.

5 Real-Time Inference

The final stage of the project involved building an inference module to support demonstration and deployment. Given any WAV audio file, the system:

1. preprocesses and normalizes the signal,
2. extracts MFCC features (13 coefficients $\times$ frame summaries),
3. standardizes the features using the same scaler fitted on the training data,
4. predicts the command using the trained SVM model,
5. returns the predicted label and class probabilities.

This module is intentionally lightweight and suitable for running on commodity hardware or being wrapped in a simple API endpoint for integration into interactive systems.

6 Implementation Details and Environment

For reproducibility, important implementation notes:

- Feature extraction implemented using `librosa` (MFCC extraction).
- Classification implemented using `scikit-learn` (SVM with linear kernel).
- Data splitting, scaling, and metric computation done with standard `scikit-learn` utilities.
- Example software stack (representative): Python 3.11, librosa (0.10+), scikit-learn (1.0+). Exact versions and seeds should be recorded in any experiment logs for complete reproducibility.

7 Discussion

The SVM trained on 26-dimensional MFCC-based summary features provides efficient classification with competitive accuracy and a high AUC. The principal strengths of this approach are interpretability and low computational cost, making it appropriate for edge devices or rapid prototyping. Limitations include sensitivity to noise and the reduced representational capacity compared with large deep-learning models; improvement paths include augmentations, more advanced temporal feature aggregation (e.g., delta/delta-delta MFCCs, statistics beyond mean/std), or small CNNs operating on spectrogram patches.

8 Conclusion

This project successfully built a lightweight audio command recognition system using MFCC features and an SVM classifier. Trained on 6,388 samples and tested on 1,597 samples, the model achieved:

- **Accuracy:** 0.9242329367564183 (92.42%)
- **AUC:** 0.9777589365827336 (97.78%)

The model demonstrates how classical machine learning techniques can efficiently address the limited-vocabulary speech recognition problem, while the inference module supports practical demonstrations.

9 References

1. Warden, P. (2018). "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition." *arXiv:1804.03209*.
2. Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing*.
3. Librosa Documentation: `https://librosa.org`
4. Scikit-learn Documentation: `https://scikit-learn.org`