Signal Classification Project

Machine Learning + Digital Signal Processing + Streamlit Web App

Author: Utkarsh Awasthi

Live App: https://signal-classification-demo-ury6wkngocgfryaee8zzst.streamlit.app/

GitHub: https://github.com/Awasthiutk564/Signal-classification-demo

Project Summary:

This project generates synthetic signals (sine, square, sawtooth, and noisy), extracts time- and frequency-domain features, trains a Random Forest classifier, and visualizes results. The Streamlit app provides live signal generation, FFT visualization, Butterworth low-pass filtering, upload/predict, and download features. The codebase includes scripts for training, evaluation (confusion matrix, feature importance), and DSP processing (FFT, filtering).

Technical Details

- Signal types: sine, square, sawtooth, noisy (synthetic)

- Feature extraction: mean, std, min, max, RMS, FFT energy, band energy ratios

- Model: RandomForestClassifier (scikit-learn)

- DSP: FFT (scipy.fft), Butterworth low-pass filter (scipy.signal)

- Web: Streamlit app with live controls, upload, download, and visualization

How to Run & Deploy

1) Install requirements: pip install -r requirements.txt

2) Run ML script (offline): python signal_classification.py

   - Generates confusion matrix, FFT plots, filtered CSV, and feature importance PNGs

3) Run web app locally: python -m streamlit run app.py

4) Deployment: push to GitHub and use Streamlit Cloud 'New app' → select repo → deploy

Results & Artifacts

- confusion_matrix.png

- fft_before.png, fft_after.png

- feature_importance.png

- filtered_signal.csv

- multi_signals.png

Notes:

All artifacts are generated by running the main script and are also downloadable from the web app.